

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA APLIKOVANÝCH VĚD
KATEDRA MATEMATIKY

Hestonův model stochastické volatility

Mgr. Milan Mrázek

rigorózní práce
k získání akademického titulu doktor přírodních věd

V souladu s pravidly rigorózního řízení na Fakultě aplikovaných věd byla podána žádost o uznání diplomové práce za rigorózní práci.

Čestně prohlašuji, že text rigorózní práce je shodný s textem diplomové práce, která byla obhájena dne 19. června 2013.

datum

podpis

Plzeň 2013

UNIVERSITY OF WEST BOHEMIA IN PILSEN
FACULTY OF APPLIED SCIENCES
DEPARTMENT OF MATHEMATICS

Heston stochastic volatility model

Mgr. Milan Mrázek

**Doctoral thesis for the award of the degree
of Doctor of Natural Sciences (RNDr.)**

The request for the recognition of my diploma thesis as a doctoral (RNDr.) thesis was submitted to the Rigorous Examination Board, in accordance with the rigorous examination procedure at the Faculty of Applied Sciences, on July 24th, 2013.

I hereby declare that the text of the doctoral (RNDr.) thesis is identical with the text of my diploma thesis, which I defended on June 19th, 2013.

Date

Signature

Pilsen 2013

DECLARATION

I do hereby declare that this diploma thesis is solely my own work and that I have used only the cited sources.

In Plzeň

author's signature

Preface

The subject of the diploma thesis is the Heston model. One part of the thesis illustrates the complexity of the calibration process of the model. This is done using synthetic option prices, where the model implied parameters are known. Calibration of the model to the data obtained from the market is then carried out using approach combining local and global optimizers. Results are presented for two consecutive days.

Another part of the thesis deals with the various Monte Carlo schemes for simulation of the Heston processes. Introduction is given to the known schemes - log-Euler, Milstein, QE, Exact scheme, IJK scheme and also a scheme combining the higher order Milstein scheme with an approximation to the Exact scheme is suggested. Parameters obtained from the market are used to test the schemes while also the range of options available on market is mimicked.

Hereby I would like to thank my supervisor, Ing. Jan Pospíšil Ph.D., for his suggestions, remarks and patient guidance throughout the time of developing this thesis. Also I would like to thank Mgr. Ondřej Pokora, Ph.D., Faculty of Science, Masaryk University in Brno, for his insights and help generously offered as part of his participation in project A-Math-Net. My one week internship at Masaryk University in Brno was supported by the project A-Math-Net - knowledge transfer network in applied mathematics (project no. CZ.1.07/2.4.00/17.0100). This project is co-financed by the European Social Fund and the state budget of the Czech Republic.

Last but not least, I would like to thank my parents for their continuous support.

Plzeň, May, 2013.

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 2 | Preliminaries | 3 |
| 2.1 | Brownian motion | 4 |
| 2.2 | Stochastic integral and SDE | 7 |
| 2.3 | Numerical solution for SDE | 9 |
| 2.3.1 | Euler scheme | 9 |
| 2.3.2 | Milstein scheme | 11 |
| 3 | Options | 13 |
| 3.1 | Option markets | 15 |
| 3.2 | Option prices | 16 |
| 3.3 | Option valuation | 17 |
| 3.3.1 | Volatility smile | 18 |
| 3.3.2 | Historical volatility | 19 |
| 3.3.3 | Other models | 21 |
| 4 | Heston model | 23 |
| 4.1 | Model dynamics | 23 |
| 4.2 | Option valuation | 24 |
| 4.3 | Valuation under risk-neutral measure | 26 |
| 4.4 | The Little Heston Trap | 27 |
| 5 | Simulation | 29 |
| 5.1 | Heston simulation schemes | 29 |
| 5.1.1 | Dynamics of the model | 29 |
| 5.1.2 | Euler scheme | 29 |
| 5.1.3 | Milstein scheme | 31 |
| 5.1.4 | Kahl-Jäckel scheme | 31 |
| 5.1.5 | Haastrecht and Pelsser [34] revised | 32 |
| 5.1.6 | Exact scheme | 33 |
| 5.1.7 | QE scheme | 34 |
| 5.1.8 | E+M scheme | 36 |

| | | |
|----------|---|-----------|
| 5.2 | MC simulations | 36 |
| 5.3 | MC test | 38 |
| 5.4 | MC test results | 39 |
| 6 | Calibration | 43 |
| 6.1 | Calibration in progress | 44 |
| 6.1.1 | Data generated | 44 |
| 6.1.2 | Measured errors | 45 |
| 6.1.3 | Considered weights | 45 |
| 6.2 | Genetic algorithm | 46 |
| 6.3 | modSQP | 49 |
| 6.4 | Excel's Solver | 50 |
| 6.5 | Adaptive Simulated Annealing | 52 |
| 6.6 | MATLAB's <code>lsqnonlin</code> | 54 |
| 6.7 | SA | 56 |
| 6.8 | Conclusion | 56 |
| 7 | Calibration to real market | 59 |
| 7.1 | March 18, 2013 | 59 |
| 7.1.1 | GA | 59 |
| 7.1.2 | ASA | 62 |
| 7.1.3 | Deterministic grid for <code>lsqnonlin</code> | 64 |
| 7.1.4 | Excel's Solver | 65 |
| 7.1.5 | <code>lsqnonlin</code> | 69 |
| 7.1.6 | Conclusion | 72 |
| 7.2 | March 19, 2013 | 74 |
| 7.2.1 | GA | 74 |
| 7.2.2 | ASA | 76 |
| 7.2.3 | Deterministic grid for <code>lsqnonlin</code> | 78 |
| 7.2.4 | MS Excel's Solver | 79 |
| 7.2.5 | <code>lsqnonlin</code> | 80 |
| 7.2.6 | Conclusion | 82 |
| 8 | Conclusion | 85 |
| A | Risk-Neutral Valuation | 87 |
| B | Content of the CD | 93 |
| | Bibliography | 96 |

Chapter 1

Introduction

Financial models are an essential tool in today's world of finance and economics. They are needed to estimate fair values of various securities, to estimate their risks and to show how to control these risks. However, the world changes as well as the market does and in response to that newer models always become necessary. As pointed out on Paul Wilmott's Blog ¹, the most important question about any financial model is how wrong it is likely to be, and how useful it is despite its assumptions.

Since the introduction of the Black Scholes model [5] a number of complex models have been proposed to reflect the behavior of markets and the derivatives. Particularly, the Black Scholes model of option valuation relied on constant volatility, which does not represent the dynamics in the financial markets sufficiently anymore. Relaxing this assumption led to the models called stochastic volatility models. Among the first publications on the class of stochastic volatility models were Hull and White [16] in 1987, Scott [31] in 1987, Stein and Stein [33] in 1991 and Heston [14] in 1993. Heston model was the first model that allowed reasonable calibration to the market option data together with semi-closed² form solution for European call/put option prices. Heston model also allows correlation between the asset price and the volatility process as opposed to Stein and Stein [33]. We discuss the Heston model in Chapter 4, whereas the particular drawbacks of the Black Scholes model and motivation for more complex models is given as a part of Chapter 3.

Heston model is one of the most popular models for option pricing. It can be calibrated using the vanilla option prices and then used to price exotic derivatives for which there is no closed form valuation formula. For this purpose a method for simulating the evolution of variable of interest is necessary. Although the Heston model was already introduced in 1993, the research on discretization methods of its dynamics appeared only very recently. Different Euler schemes implementations were examined and various ways how to deal with negative values of the variance process were suggested, for example Higham and Mao [15] in 2005 or Lord et. al. [24] in 2010. An exact scheme was developed by Broadie and Kaya [6] in 2006, which simulates the He-

¹<http://www.wilmott.com/blogs/paul/>

²It involves numerical integral.

ston processes from its exact distributions. This topic is covered in detail in Chapter 5, where we also discuss higher order Milstein schemes and QE scheme, which is the most popular one right now. It was introduced by Andersen [2] in 2008.

Most of the tests for pricing using Monte Carlo in the papers omit certain schemes and also choose their parameters and discretization rather wisely when comparing the schemes or showing their performance. We aimed to eliminate this phenomenon by calibrating the Heston model to real market data and obtaining a set of parameters which explained the prices of European call options on the real market. We used them afterwards in the Monte Carlo simulations in Chapter 5 and compared the Monte Carlo results with the prices implied by the Heston model.

The calibration of the model is a crucial process and a price to pay with more complex model is the increased complexity of the calibration process. The industry standard approach is to minimize the difference between the observed prices and the model prices. We can follow for example Nögel and Mikhailov [29], they used the method of Adaptive Simulated Annealing and weights of their fine choice. Unfortunately it is quite common in the literature that only few data and mostly at the money options are used to demonstrate the calibration process, which of course leads to small errors, it makes the calibration problem easy to solve. Also synthetic option prices with simple parameters are often calculated and used for the calibration. See for example Kienitz and Wetterau [22], where synthetic option prices are calculated and the model parameters are recovered from the calibration process. A significant role is played by the optimizer, because local optimizers heavily depend on the initial guess and thus the distance between the model prices and the market prices does not necessarily be because of the model but it can be due to the calibration process.

Chapter 6 demonstrates the complexity of the calibration process and employs variety of optimizers. We used the newest available version of Adaptive Simulated Annealing, we also tested the Genetic algorithm from the MATLAB's Optimization Toolbox. In Chapter 7 we use the different optimizers with a specific approach to calibrate the model to data obtained from the real market, namely we used daily data for DAX Options obtained using the Bloomberg Terminal³. We presented the results of the calibration for two consecutive days.

The structure of the thesis is as follows. We start with addressing mathematical preliminaries in Chapter 2. In Chapter 3 we define option contract and give theoretical and graphical motivation for stochastic volatility models. Chapter 4 introduces the Heston model. In Chapter 5 we present various simulation schemes and results of their test. Chapter 6 discusses the calibration problem and introduces the used optimizers. Chapter 7 contains the results of calibration to real market data. The final conclusion is given in Chapter 8.

³Computer system provided by Bloomberg L.P. enabling access to the Bloomberg Professional service for the purpose of monitoring and analyzing real-time financial market data.

Chapter 2

Preliminaries

The purpose of this section is to introduce basic definitions and concepts that will be used throughout the thesis. This is not intended to be a complete comprehensive overview of the stochastic theory with proofs. We will follow Steven E. Shreve [32], Paul Glasserman [12] and Bohdan Maslowski [26].

We will start with defining the Brownian motion¹, that is the most common stochastic building block used for random walks in finance. At first, let us define a probability space and a stochastic process.

Definition 1. Let Ω be a nonempty set, and let \mathcal{F} be a σ -algebra of subsets of Ω . A probability measure \mathbb{P} is a function that, to every set $A \in \mathcal{F}$, assigns a number in $[0, 1]$, called the probability of A and written $\mathbb{P}(A)$. We require

- (i) $\mathbb{P}(\Omega) = 1$,
- (ii) $\mathbb{P}(\emptyset) = 0$, and
- (iii) whenever A_1, A_2, \dots is a sequence of disjoint sets in \mathcal{F} , then

$$\mathbb{P}\left(\bigcup_{n=1}^{\infty} A_n\right) = \sum_{n=1}^{\infty} \mathbb{P}(A_n).$$

The triple $(\Omega, \mathcal{F}, \mathbb{P})$ is called a probability space.

Having the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ we will now proceed to the definition of stochastic process.

Definition 2. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. A stochastic process is a collection of random variables $X = \{X(t, \omega), t \in [0, T], \omega \in \Omega\}$ on $(\Omega, \mathcal{F}, \mathbb{P})$ that takes values in \mathbb{R} . By sample path (or trajectory) of the process we understand the function

$$t \rightarrow X(t, \omega) \text{ for each given } \omega \in \Omega,$$

¹Also called Wiener process.

alternatively the process X can be viewed as a random variable

$$\omega \rightarrow X(t, \omega) \text{ for any fixed parameter } t \in [0, T].$$

In what follows we use the notation $X = (t, \omega) = X(t)$, i.e. for simplicity we omit writing the dependence on $\omega \in \Omega$.

2.1 Brownian motion

Definition 3. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. For each $\omega \in \Omega$, suppose there is a continuous function $W(t)$ of $t \geq 0$ that satisfies $W(0) = 0$ and that depends on ω . Then $W(t)$, $t \geq 0$, is a Brownian motion if for all $0 = t_0 < t_1 < \dots < t_m$ the increments

$$W(t_1) - W(t_0), W(t_2) - W(t_1), \dots, W(t_m) - W(t_{m-1})$$

are independent and each of these increments is normally distributed with

$$\begin{aligned} \mathbb{E}[W(t_{i+1}) - W(t_i)] &= 0, \\ \text{Var}[W(t_{i+1}) - W(t_i)] &= t_{i+1} - t_i, \end{aligned}$$

where \mathbb{E} denotes mathematical expectation (mean value) and Var denotes variance.

Remark 1. We can think of ω as one particular Brownian motion path, as an outcome of a random experiment, which value at time t is $W(t)$. Thus, the sample space Ω is the set of all possible outcomes of this random experiment, \mathcal{F} is the σ -algebra of subsets of Ω and their probabilities are defined by \mathbb{P} , which is a probability measure. For each $A \in \mathcal{F}$, the probability of A is a number in $[0, 1]$.

Simulation of the Brownian motion can be done in the following way. We use the fact that the increments are normally distributed with mean zero and variance equal to $t_{i+1} - t_i$. Setting $t_0 = 0$ and starting with $W(0) = 0$, subsequent values of one Brownian motion path can be obtained as follows

$$W(t_{i+1}) = W(t_i) + \sqrt{t_{i+1} - t_i} Z_{i+1}, \quad i = 0, \dots, n-1, \quad (2.1)$$

where Z_i is a sequence of standard normal variables.

Definition 4. Let X be a random variable defined on a nonempty sample space Ω . Let \mathcal{G} be a σ -algebra of subsets of Ω . If every set in $\sigma(X)$ is also in \mathcal{G} , we say that X is \mathcal{G} -measurable.

A random variable X is \mathcal{G} -measurable if and only if the information in \mathcal{G} is sufficient to determine the value of X . Our next aim is to define the filtration for Brownian motion. The term *filtration* represents the increasing collection of σ -algebras $\mathcal{F}(t)$, $t \geq 0$ generated in our case by $W(t)$, $t \geq 0$ and containing the information available at time t .

Definition 5. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space on which a Brownian motion $W(t)$ is defined, $t \geq 0$. A filtration for the Brownian motion is a collection of σ -algebras $\mathcal{F}(t) \subset \mathcal{F}$, $t \geq 0$, satisfying:

- (i) For $0 \leq s < t$, every set in $\mathcal{F}(s)$ is also in $\mathcal{F}(t)$.
- (ii) For each $t \geq 0$, the Brownian motion $W(t)$ at time t is $\mathcal{F}(t)$ -measurable (the information at time t is sufficient enough to evaluate the the Brownian motion $W(t)$ at time t).
- (iii) For $0 \leq t \leq u$, the increment $W(u) - W(t)$ is independent of $\mathcal{F}(t)$.

In the view of the previous definition we will also define a stochastic process adapted to a filtration.

Definition 6. Let us consider a stochastic process $X(t)$, $t \geq 0$. We say that $X(t)$ is adapted to the filtration $\mathcal{F}(t)$ if for each $t \geq 0$ the random variable $X(t)$ is $\mathcal{F}(t)$ -measurable.

If it is obvious to what filtration $\mathcal{F}(t)$ the process $X(t)$ is adapted, we may simply write that $X(t)$ is adapted. Next we give a definition of martingale and then show that Brownian motion is a martingale.

Definition 7. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, let T be a fixed positive number, and let $\mathcal{F}(t)$, $0 \leq t \leq T$, be a filtration of sub- σ -algebras of \mathcal{F} . Consider an adapted process $M(t)$, $0 \leq t \leq T$. We say $M(t)$ is a martingale if

$$\mathbb{E}[M(t)|\mathcal{F}(s)] = M(s) \text{ for all } 0 \leq s \leq t \leq T,$$

where $\mathbb{E}[M(t)|\mathcal{F}(s)]$ denotes conditional expectation of $M(t)$ given $\mathcal{F}(s)$.

In other words, a martingale has no tendency to rise or fall.

Lemma 1. Brownian motion is a martingale, i.e. for $0 \leq s < t$ we have that

$$\mathbb{E}[W(t)|\mathcal{F}(s)] = W(s).$$

Proof. Let $0 \leq s < t$ be given. Then

$$\begin{aligned} \mathbb{E}[W(t)|\mathcal{F}(s)] &= \mathbb{E}[(W(t) - W(s)) + W(s)|\mathcal{F}(s)] \\ &= \mathbb{E}[W(t) - W(s)|\mathcal{F}(s)] + \mathbb{E}[W(s)|\mathcal{F}(s)] \\ &= \mathbb{E}[W(t) - W(s)] + W(s) \\ &= W(s). \end{aligned}$$

□

When simulating the paths of financial assets later in the thesis, we will need to construct two correlated Brownian motions. We will do this by Cholesky decomposition of the correlation matrix.

Definition 8. The correlation matrix of m random variables X_1, \dots, X_m is the $m \times m$ symmetric matrix whose i, j entry is $\rho(X_i, X_j)$ defined as

$$\rho(X_i, X_j) = \frac{\text{Cov}(X_i, X_j)}{\sigma(X_i)\sigma(X_j)} = \frac{\mathbb{E}[(X_i - \mu_i)(X_j - \mu_j)]}{\sigma(X_i)\sigma(X_j)},$$

where $\sigma(X_i)$ stands for standard deviation, Cov for covariance and

$$\mu_i = \mathbb{E}[X_i].$$

Remark 2. In our case we will deal with standard normal variables X and Y , hence standard deviations will be 1 for both X and Y and $\text{Cov}(X, Y) = \rho(X, Y)$.

We would like to simulate two correlated Brownian motions $W(t)^{(1)}$ and $W(t)^{(2)}$, with correlation ρ . We showed how to simulate one Brownian motion, thus we will now need to generate two correlated random numbers $Z^{(1)}$ and $Z^{(2)}$. A correlation matrix C for two random variables is according to the definition above given by

$$C = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}, \quad (2.2)$$

where ρ is the correlation between $W(t)^{(1)}$ and $W(t)^{(2)}$.

Lemma 2. Let $\rho \in [-1, 1]$ and C be given by 2.2. Then there exists an upper triangular matrix U such that

$$C = U^T U, \quad (2.3)$$

namely

$$U = \begin{bmatrix} 1 & \\ 0 & \sqrt{1 - \rho^2} \end{bmatrix}.$$

Proof.

$$U^T U = \begin{bmatrix} 1 & 0 \\ \rho & \sqrt{1 - \rho^2} \end{bmatrix} \cdot \begin{bmatrix} 1 & \rho \\ 0 & \sqrt{1 - \rho^2} \end{bmatrix} = \begin{bmatrix} 1 & \rho \\ \rho & \rho^2 + (1 - \rho^2) \end{bmatrix} = C.$$

□

Remark 3. The matrix U can be found using the Cholesky decomposition, if C is positive definite. It is satisfied for $\rho \in (-1, 1)$. For $\rho = \pm 1$ the assertion (2.3) is also true.

Two correlated random numbers $Z^{(1)}, Z^{(2)}$ with correlation ρ are then generated by transforming two uncorrelated independent random numbers z_1, z_2 by

$$\begin{bmatrix} Z^{(1)} \\ Z^{(2)} \end{bmatrix} = [z_1 \quad z_2] \begin{bmatrix} 1 & \rho \\ 0 & \sqrt{1-\rho^2} \end{bmatrix},$$

thus we arrive to $Z^{(1)} = z_1$ and $Z^{(2)} = \rho z_1 + \sqrt{1-\rho^2} z_2$.

2.2 Stochastic integral and SDE

We have already defined the Brownian motion and a filtration for the Brownian motion, these are necessary for the forthcoming definition of the stochastic integral, which will be followed by the definition of the stochastic differential equation in this section.

Definition 9. Let $W(t), t \geq 0$, be a Brownian motion and $\phi(t), t \geq 0$, be a stochastic process adapted to the filtration $\mathcal{F}(t), t \geq 0$, such that

$$\mathbb{E} \int_0^T \phi^2(t) dt < \infty.$$

We define the Itô stochastic integral of $\phi(t)$ with the driving process $W(t)$ by the formula

$$\int_0^T \phi(u) dW(u) = \lim_{\|D_n\| \rightarrow 0} \sum_{i=0}^{n-1} \phi(t_i) (W(t_{i+1}) - W(t_i)), \quad (2.4)$$

where $D_n = \{0 = t_0 < t_1 < \dots < t_n = T\}$ is the partition of the interval $[0, T]$ and

$$\|D_n\| = \max_{j \in \{0, \dots, n-1\}} |t_{j+1} - t_j|.$$

Some of the properties of Itô stochastic integral are given in the following theorem.

Theorem 1. Let $I(t) = \int_0^t \phi(u) dW(u)$ be a random variable as defined above. The stochastic integral $I(t)$ has the following properties

- (i) For each t , $I(t)$ is $\mathcal{F}(t)$ -measurable.
- (ii) $\text{Var}[I(t)] = \mathbb{E}[I(t)^2] = \mathbb{E} \int_0^t \phi^2(u) dW(u)$.
- (iii) $\int_0^t (a\phi(u) + b\phi(u)) dW(u) = a \int_0^t \phi(u) dW(u) + b \int_0^t \phi(u) dW(u)$,
where a and b are constants.
- (iv) $I(t)$ is a martingale.

Theorem 1 is provided without a proof, reader is referred to section 3.2.A and 3.2.B of Karatzas and Shreve [21] or 4.2 and 4.3 of Shreve [32]. Itô stochastic integral is first constructed for so called simple processes (also step functions). For such integrands it is not too difficult to verify the above properties of Itô stochastic integral. General integrands ϕ are then approximated using a sequence of simple processes converging to ϕ . The properties of $I(t)$ are then inherited from the properties of Itô stochastic integral for simple integrands as pointed out in Theorem 4.3.1 by Shreve in [32].

Definition 10. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and let $W(t)$ be a Brownian motion on Ω with respect to $\mathcal{F}(t)$. Also let $a : [0, T] \times \mathbb{R} \rightarrow \mathbb{R}$ and $b : [0, T] \times \mathbb{R} \rightarrow \mathbb{R}$ be measurable functions and $X(0)$ be $\mathcal{F}(0)$ -measurable random variable. The stochastic differential equation is a stochastic differential:

$$dX(t) = a(t, X(t))dt + b(t, X(t))dW(t), t \in (0, T) \quad (2.5)$$

with an initial condition

$$X(0) = x_0 \in \mathbb{R}, \quad (2.6)$$

which may be random. A solution to the equation is a continuous stochastic process $X(t)$ with respect to $\mathcal{F}(t)$ satisfying

$$X(t) = x_0 + \int_0^t a(s, X(s))ds + \int_0^t b(s, X(s))dW(s), t \in [0, T],$$

where the first integral is Lebesgue integral and the second one is an Itô stochastic integral. The solution is also called strong solution.

The existence of the solution for the above defined stochastic differential equation and its uniqueness is provided by the following Existence and uniqueness of solutions theorem for stochastic differential equations.

Theorem 2. Let $a : [0, T] \times \mathbb{R} \rightarrow \mathbb{R}$ and $b : [0, T] \times \mathbb{R} \rightarrow \mathbb{R}$ be measurable functions satisfying

$$|a(t, x)| + |b(t, x)| \leq C(1 + |x|), x \in \mathbb{R}; t \in [0, T],$$

for some constant C and

$$|a(t, x) - a(t, y)| + |b(t, x) - b(t, y)| \leq D|x - y|; x, y \in \mathbb{R}, t \in [0, T]$$

for some constant D . Then the equation (2.5) - (2.6), where $X(0)$ is $\mathcal{F}(0)$ -measurable random variable satisfying $\mathbb{E}|X(0)|^2 < \infty$ has a continuous and unique solution $X(t)$ with respect to $\mathcal{F}(t)$.

Proof 1. Proof can be found in Maslowski [26] under Theorem 5.2.

The first *linear growth condition* ensures that the solution process $X(t)$ does not explode, i.e. that $X(t)$ does not tend to go to infinity in finite time. The second, *Lipschitz condition* guarantees that equation has a unique solution. Here uniqueness means that if $X(t)$ and $Y(t)$ are two continuous processes satisfying the differential equation (2.5) - (2.6) then

$$X(t) = Y(t), t \in [0, T], \text{ almost surely.}$$

Such processes as $X(t)$ and $Y(t)$ are called indistinguishable.

Lemma 3. *Itô formula.* Let V be a twice continuously differentiable function and let $X(t)$, $t \geq 0$, be a process defined in Definition 10, where σ and b satisfy the conditions from Theorem 2. Then the process $Y(t) = V(t, X(t))$ has a stochastic differential

$$dY(t) = \left(\frac{\partial V}{\partial t} + a(t, X(t)) \frac{\partial V}{\partial x} + \frac{1}{2} b^2(t, X(t)) \frac{\partial^2 V}{\partial x^2} \right) dt + a(t, X(t)) \frac{\partial V}{\partial X} dW(t). \quad (2.7)$$

Proof 2. The proof of Itô formula can be found for example in Karatzas and Shreve [21] as Theorem 3.3.

2.3 Numerical solution for SDE

2.3.1 Euler scheme

One of the simplest time discrete approximations of a stochastic differential equation is the Euler approximation, also known as Euler-Maruyama approximation. We consider stochastic process $X = X(t)$, $0 \leq t \leq T$ following stochastic differential equation

$$dX(t) = a(t, X(t))dt + b(t, X(t))dW(t) \quad (2.8)$$

on $0 \leq t \leq T$ with initial value

$$X(0) = X_0. \quad (2.9)$$

Taking time discretization $0 = \tau_0 < \tau_1 < \dots < \tau_n < \dots < \tau_N = T$ of the interval $[t_0, T]$, an explicit Euler approximation is a continuous time stochastic process $Y = \{Y(t), t_0 \leq t \leq T\}$ following the iterative scheme

$$Y_{n+1} = Y_n + a(\tau_n, Y_n)(\tau_{n+1} - \tau_n) + b(\tau_n, Y_n)(W_{\tau_{n+1}} - W_{\tau_n}), \quad (2.10)$$

where $n = 0, 1, 2, \dots, N - 1$ with initial value

$$Y_0 = X_0, \quad (2.11)$$

and where Y_n denotes the value of approximation at the time of discretization τ_n

$$Y_n = Y(\tau_n). \quad (2.12)$$

We will denote

$$\Delta_n = \tau_{n+1} - \tau_n \quad (2.13)$$

the n th time increment. Note that for $b \equiv 0$, the equation (2.10) reduces to deterministic Euler scheme.

In order to compute the sequence $Y_n, n = 0, 1, \dots, N$ of values of the Euler approximation we need to generate the random increments

$$\Delta W_n = W_{\tau_{n+1}} - W_{\tau_n}, \quad (2.14)$$

for $n = 0, 1, \dots, N - 1$ of the Brownian motion $W(t), t \geq 0$. As described earlier these increments are independent Gaussian random variables with

$$\mathbb{E}[W_n] = 0 \quad (2.15)$$

and

$$\mathbb{E}[(\Delta W_n)^2] = \Delta_n. \quad (2.16)$$

For this purpose we can use a sequence of independent Gaussian pseudo-random numbers and we can simulate ΔW_n by

$$\Delta W_n = Z_n \sqrt{\Delta_n}, \quad (2.17)$$

where Z_n is a sequence of standard normal random variables, $Z_n \sim \mathcal{N}(0, 1)$ ². We will shorten the scheme and from now on use the following form

$$Y_{n+1} = Y_n + a\Delta_n + bZ\sqrt{\Delta_n}, \quad (2.18)$$

for $n = 0, 1, \dots, N - 1$, where Z is a sample generated from the standard normal distribution.

To measure the convergence of the approximations we shall define the following criterions. Let

$$\delta = \max_n \Delta_n \quad (2.19)$$

and call it the time step. Y^δ denotes the discrete time approximation with time step δ .

Definition 11. We say that the time discrete approximation Y^δ converges weakly with order $\beta > 0$ to X at time T if for any polynomial g there exists a constant $C > 0$ such that

$$|\mathbb{E}(g(X(T))) - \mathbb{E}(g(Y_N))| \leq C\delta^\beta,$$

for each $\delta, 0 < \delta < T$.

²In MATLAB we can use for example function randn.

Definition 12. We say that the time discrete approximation Y^δ converges strongly with order $\gamma > 0$ to X at time T if there exists a constant $C > 0$, such that

$$\mathbb{E}(|X(T) - Y_N^\delta|) \leq C\delta^\gamma,$$

for each $\delta, 0 < \delta < T$.

The Euler approximations (2.18) converge in the strong sense with order $\gamma = 0.5$ and in the weak sense with order $\beta = 1$, the proof can be found in Kloeden and Platen [23].

2.3.2 Milstein scheme

Refining the Euler scheme (2.18) by adding an extra term using Itô lemma, see for example [12], results in

$$Y_{n+1} = Y_n + a\Delta_n + bZ\sqrt{\Delta_n} + \frac{1}{2}b'b(Z^2 - 1)\Delta_n. \quad (2.20)$$

The above scheme was derived by Milstein [28] and thus is referred to as Milstein scheme. Milstein approximations (2.20) converge in strong order $\gamma = 1$ and weak order $\beta = 1.5$, for proof see Kloeden and Platen [23].

Chapter 3

Options

Before we get to the point and define an option, let us describe what is an asset. We will use the asset to describe any financial object whose value is known at the present time but subject to changes in the future. We would mention following examples, which would qualify as an asset

- shares in company,
- commodities such as gold, oil or wheat,
- currencies, for example, the value of \$1 in €.

Definition 13. *A European call option gives its holder the right, but not the obligation to purchase from the writer a prescribed asset for a prescribed price at a prescribed time in the future.*

The right, but not the obligation is what distinguishes options from other derivatives¹ such as forwards and futures, where the holder is obligated to buy or sell the underlying asset. We will further refer to the prescribed purchase price as the strike price and the prescribed time in the future as the expiry date (also maturity).

The opposite of a European call option is a European put option.

Definition 14. *A European put option gives its holder the right, but not the obligation to sell to the writer a prescribed asset for a prescribed price at a prescribed time in the future.*

Options are very popular, sometimes more money is invested in the options and other than in the underlying assets. There are at least two good reasons to do so

- (i) Options are attractive to investors for both speculation and hedging.
- (ii) There is a particular way to determine how much they are worth, and thus they can be bought and sold with some confidence.

¹By derivative we understand a financial instrument whose value depends on the price of an underlying asset.

Hedgers use derivatives to reduce risk they face from future movements on the market, whereas speculators use them to bet on the future direction of the market. For more on types of trading, see for example Hull [17].

Let K denote the strike price and $S(T)$ denote the asset price at the expiry date T . $S(t)$ is then the asset price at a general time t . We say that the value of the European call option at the expiry date, denoted by c , is

$$c = (S(T) - K)^+, \quad (3.1)$$

which means, that at expiry date, if $S(T) > K$, then the holder can buy the asset for K and sell it on the market for $S(T)$, making profit $S(T) - K$. On the other hand, if $S(T) \leq K$, then the holder gains nothing. We can see this illustrated on the payoff diagrams in Figure 3.1, the piecewise linear payoff curves are sometimes called (ice) hockey sticks. The value of the European put option, denoted as p is then

$$p = (K - S(T))^+. \quad (3.2)$$

We will consider data on index options, where settlement is always in cash, rather than by delivering the portfolio underlying the index. Also one contract is usually to buy or sell 100 times the index at specified strike price. These are called contract multipliers. Consider for example one call contract with a strike price of K . If it is exercised when the value of the index is S , the writer of the contract pays the holder $(S - K) * 100$ in cash.

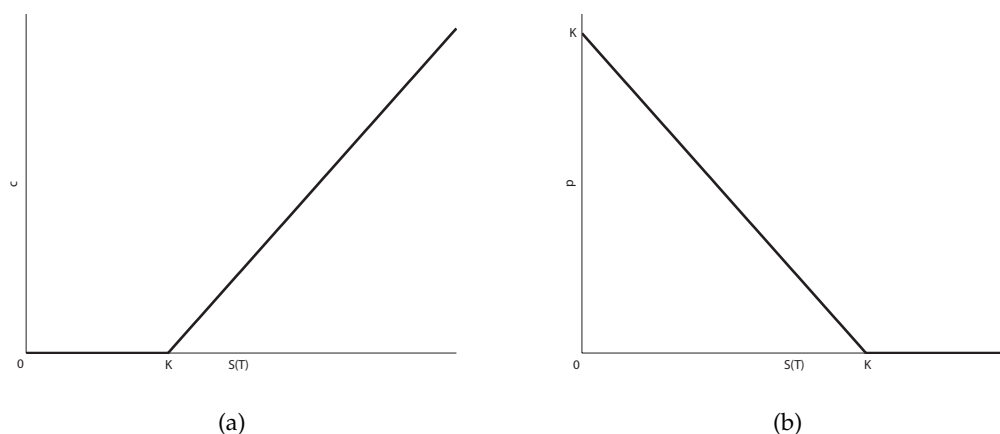


Figure 3.1: Payoff diagram for a European call option (a) according to formula (3.1) and for a European put (b) following formula (3.2).

3.1 Option markets

Options are traded on a number of official exchanges. First of them, Chicago Board Options Exchange (CBOE), was established by Chicago Board of Trade in 1973. Most exchanges work through the use of market makers, individuals, who are obliged to buy or sell options at any time they are asked to do so. Market makers will quote two prices, the bid and the ask. They will buy the option for the bid price and they will sell it for the ask price. Bid-ask spread is the difference between the ask and bid price and that is how the market makers make their profit, they do not speculate on the market.

Options are also traded directly between financial institutions, fund managers or corporations, these are called over-the-counter (OTC) deals. The trades in the over-the-counter market are typically much larger than trades in the exchange-trade market.

Another derivatives exchange is Eurex, established in 1998. One of the products it offers are the options on DAX (Deutscher Aktien Index). DAX is a stock market index consisting of the 30 major German companies trading on the Frankfurt Stock Exchange. The following Figure 3.2 is a snapshot of Bloomberg Option Monitor (OMON)² listing the bid and ask prices for the European Call options on DAX with expire date on September 20, 2013. Picture was taken on March 19, 2013 (185 days before expiry date).

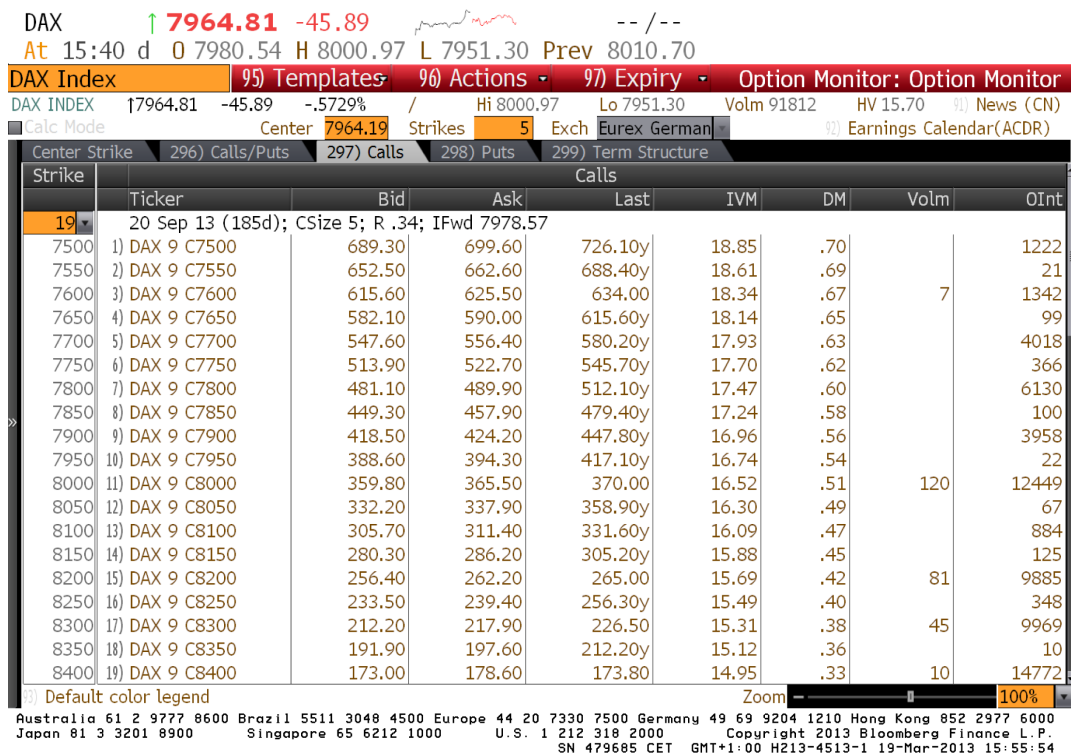


Figure 3.2: DAX index Call Option Price Quotations. Source: Bloomberg Finance L.P.

²Option Monitor is a screen providing all current and future Calls and Puts for the particular equity.

3.2 Option prices

Option prices are available for a range of strike prices and expiry dates. Figure 3.2 shows prices only for a particular expiry date. The next Figure 3.3 shows market values for S&P500 index call and put options, that were calculated as

$$\text{Mid} = \frac{(\text{Bid} + \text{Ask})}{2} \quad (3.3)$$

and for various strike prices and expiry dates³. We can see that for a given expiry date, the call option decreases as the strike price increases. Increasing the strike price has a negative effect on the payoff and hence it reduces the call option's worthiness. Similarly with the price of the put option, where the put price increases with increasing strike price.

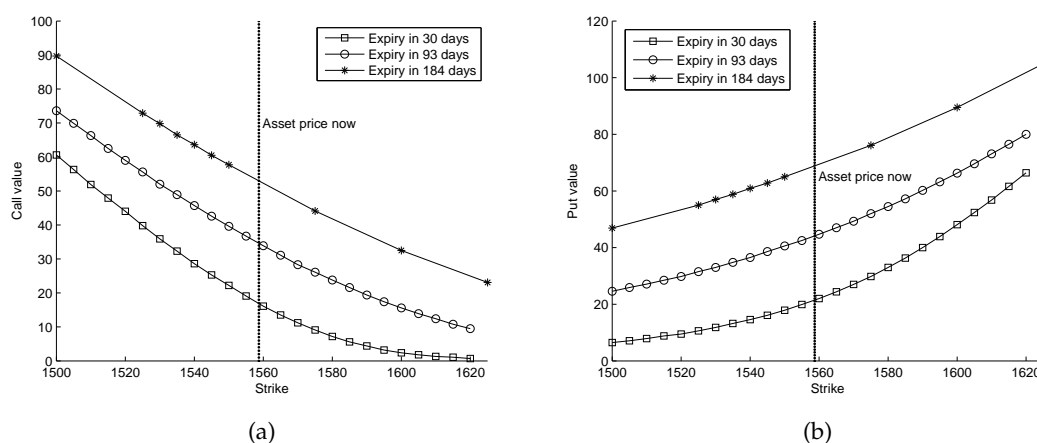


Figure 3.3: Market values for S&P500 Index call (a) and put (b) options for a range of strike prices and expiry dates from March 21, 2013.

Another observation can be made from the figures above. For a given strike price, both the call and the put option prices increase with time increasing to expiry date.

We will now introduce more financial jargon. At time t , a European call option is said to be

- In the money (ITM) if $S(t) > K$
- out of the money (OTM) if $S(t) < K$, and
- at the money (ATM) if $S(t) = K$.

Together, this concept is called "Moneyness of Options". In general, in the money means that there will be a positive payoff if the asset price stays as it is. Out of the

³Source for bid and ask prices: Bloomberg Option Monitor.

money means the asset must change by significant amount in order for a positive pay-off. The interval between in and out of the money is then defined as at the money.

3.3 Option valuation

In the early 1970s, Merton tackled a problem that had been partially solved by two other economists, Fischer Black and Myron S. Scholes: deriving a formula for the 'correct' price of a stock option. Grasping the intimate relation between an option and the underlying stock, Merton completed the puzzle with an elegantly mathematical flourish. Then he graciously waited to publish until after his peers did; thus the formula would ever be known as the Black–Scholes model. Few people would have cared given that no active market for options existed. But coincidentally, a month before the formula appeared, the Chicago Board Options Exchange had begun to list stock options for trading. Soon, Texas Instruments was advertising in The Wall Street Journal, 'Now you can find the Black–Scholes value using our . . . calculator.' This was the true beginning of the derivatives revolution. Never before had professors made such an impact on Wall Street.

- Roger Lowenstein [25].

In this section we will briefly discuss the drawbacks and broken assumptions of the Black-Scholes-Merton model. But first we will give a very brief introduction to the Black Scholes model. In 1973 Fischer Black and Myron Scholes published their paper called "The Pricing of Options and Corporate Liabilities" [5]. Price of a call option $c(S, t)$ follows the Black-Scholes-Merton⁴ partial differential equation

$$\frac{\partial c}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 c}{\partial S^2} + rS \frac{\partial c}{\partial S} - rc = 0, \quad (3.4)$$

where r is the risk free interest rate, σ is volatility, both constant parameters, t is time and S is the stock price, satisfying terminal condition

$$c(S, T) = (S - K)^+ \quad (3.5)$$

and boundary conditions

$$c(0, t) = 0, \text{ for } 0 \leq t \leq T \quad (3.6)$$

and

$$c(S, t) \rightarrow S, \text{ as } S \rightarrow \infty. \quad (3.7)$$

⁴It is well documented that the three men cooperated on development of the option-pricing formula and in recognition of this Scholes and Merton were awarded by the 1997 Nobel Prize in Economics and it is widely accepted that Fischer Black, who died in 1995, would have also shared the prize if he had been still alive.

Analytical solution to the value of a European call option at time t is

$$C(S, t) = SN(d_+) - Ke^{-r(T-t)}N(d_-), \quad (3.8)$$

where

$$d_{\pm} = \frac{1}{\sigma\sqrt{T-t}} \left[\ln \frac{S}{K} + \left(r \pm \frac{\sigma^2}{2} \right) (T-t) \right], \quad (3.9)$$

and N is the cumulative distribution function for the standard normal variable $\mathcal{N}(0;1)$. $(T-t)$ is the time to expiry date, K is the strike price as previously defined. S is the stock price at time t and the formula assumes it follows a geometric Brownian motion

$$dS(t) = rS(t)dt + \sigma S(t)dW(t), \quad (3.10)$$

where r and σ are constant defined as above. $W(t)$ is Brownian motion. We will be particularly interested in the constant volatility assumption, which is listed in Willmot [35] as first assumption or as ideal condition (b) in Black and Scholes [5], since the volatility parameter does not need to be constant as stated in Willmot [35] and still we can find solution. We will give some arguments to break this assumption in the next parts of this chapter. Paul Willmot discusses some more assumptions in [35] as well as the consequences of dropping them.

Despite the forthcoming arguments, the formula (3.8) is still very popular among option traders because it is simple and easy to use, with very few parameters. Also its drawbacks are quite well understood.

3.3.1 Volatility smile

The Black–Scholes call and put values depend on $S, K, r, T-t$ and σ . Of all these parameters, only the asset volatility σ can not be observed directly. One approach is to extract the volatility from the observed market data. Knowing a quoted option call value c , and based on observed S, t, K, r and T , we can find such σ that leads to this value. A σ computed this way is known as an implied volatility (σ is implied by data on the market). To find such σ we can use for example MATLAB function

```
sigma = blsimpv(S, K, r, tau, c) %tau=T-t
```

Let us use the following quote as a bridge to the definition of the phenomenon mentioned in the name of the section.

"A smiley implied volatility is the wrong number to put in the wrong formula to obtain the right price."

- Riccardo Rebonato [30].

Volatility smile is the pattern we can observe on a graph of implied volatility vs. strike price for a given expiry date. It will form an upturned curve similar to the shape of a smile, because in the money and out of the money options are observed to have higher implied volatilities than at the money options. However, as mentioned in Hull [17] equity options traded in American markets did not show a volatility smile before the Crash of 1987 but began showing one afterwards. Despite the disparities, the Black–Scholes theory continues to be highly regarded by both academics and market traders. It is common among traders for option values to be quoted in terms of volatility rather than price, because the implied volatility tends to be less variable than the option price.

A related concept is that of term structure of volatility. It describes how (implied) volatility differs for related options with different maturities. The next Figure 3.4 shows an implied volatility surface 3D plot that plots volatility smile and term structure of volatility in a three-dimensional surface for DAX call options.

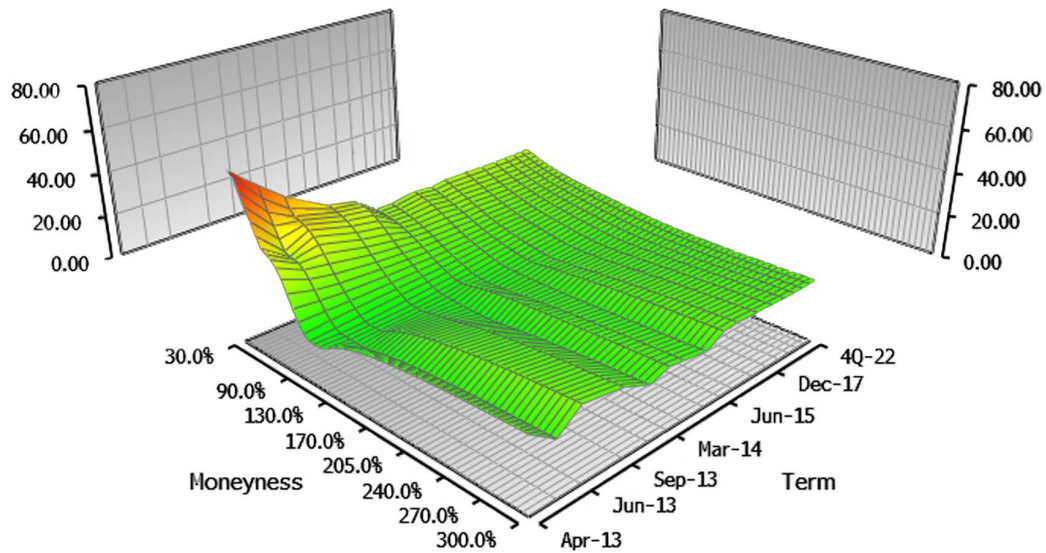


Figure 3.4: Graph of the DAX-implied volatility surface on March 21, 2013. Source: Bloomberg Finance L.P.

3.3.2 Historical volatility

Another approach to obtain the volatility parameter σ for the Black-Scholes model is to use the historical data. But as we can see in Figure 3.5 the volatility in real life is not constant, but changes over time.

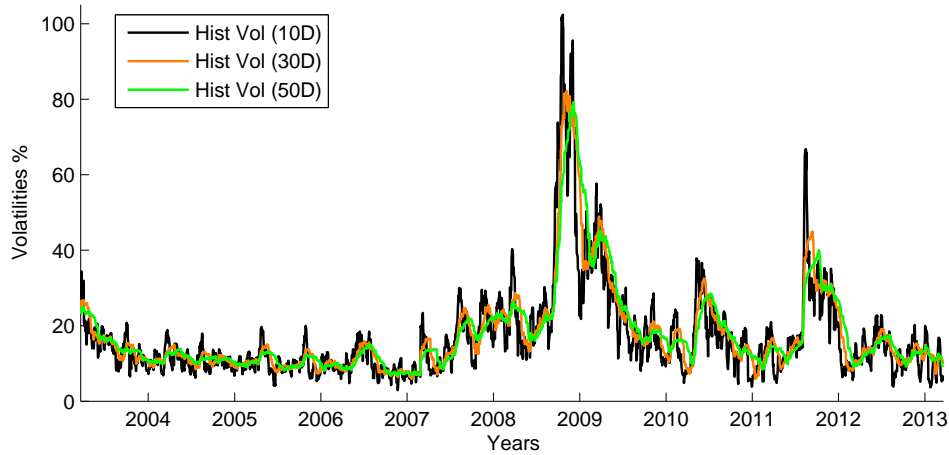


Figure 3.5: 10/30/50-day historical volatilities for S&P 500 Index.

Whatsmore, fall in the stock price is often accompanied by a rise in volatility. It is easy to observe this in the second half of 2008 in Figure 3.6 and especially in the weeks following the bankruptcy of Lehman Brothers (September 15, 2008) and problems of other financial companies.

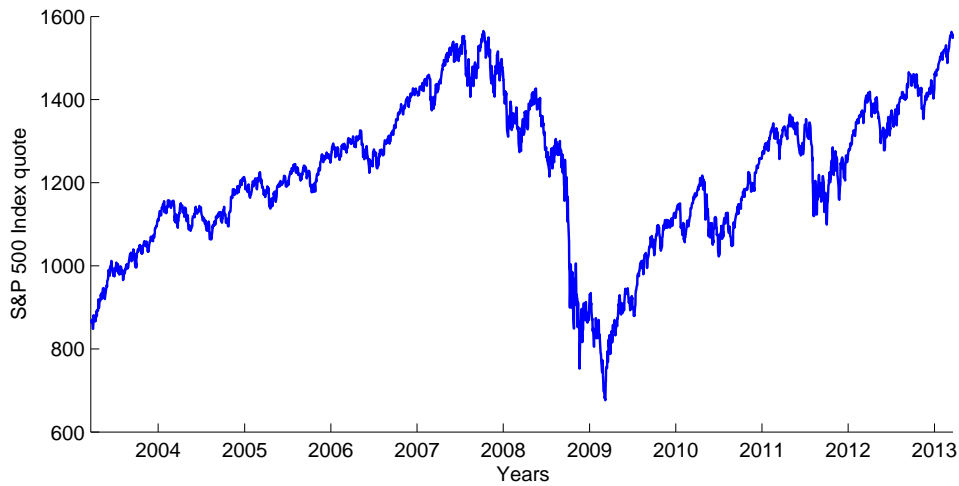


Figure 3.6: Historical quotes for S&P 500 Index.

Another valid point against using historical volatility is that we should be estimating volatility for the time period that is about to come, but measuring volatility from a

stock price series only yields an estimate for the volatility over the past time period.

3.3.3 Other models

Last but not least the Black-Scholes stochastic differential equation results in a lognormal distribution of the random variable $S(t)$ (its log is normally distributed). This is not the case for the real market data, fat tails and the high central peaks can be observed and are characteristics of mixtures of distributions with different variances [11]. We can see the fat tails and highly skewed central peak in the following Figure 3.7, extreme events indeed occur more frequently than a model based on normal random variables would predict.

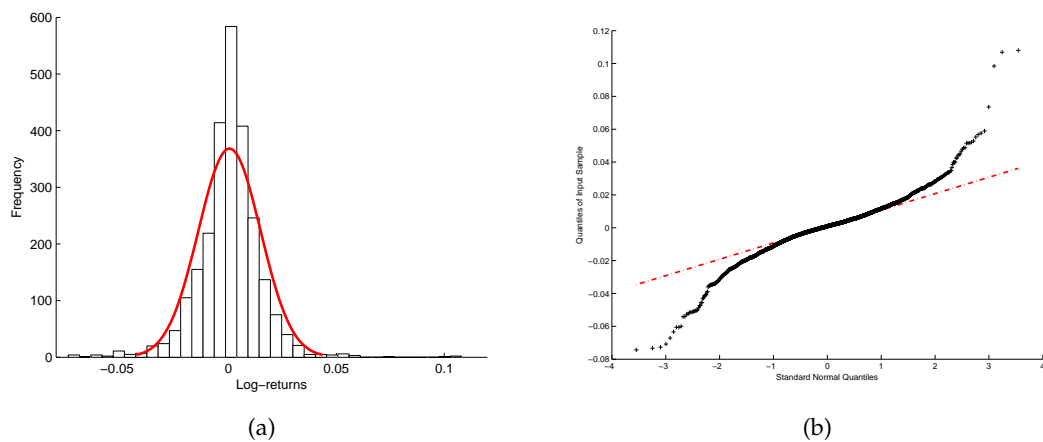


Figure 3.7: DAX daily log-returns vs. the normal distribution in red. Frequency distribution of 10 years in picture (a) and Q-Q plot in (b).

There have been numerous attempts to develop generalizations or alternatives to the lognormal asset price model. Many of these were motivated by the observations of the real market data. One known approach is to allow the volatility to be stochastic, another is to allow the asset to undergo ‘jumps’. In the next section we will introduce the Heston model, which allows the volatility to be stochastic.

Chapter 4

Heston model

4.1 Model dynamics

In this chapter we focus on stochastic volatility model introduced by Steven Heston in [14]. The spot asset S at time t follows the stochastic differential equation

$$dS(t) = rS(t)dt + \sqrt{v(t)}S(t)dW^S(t), \quad (4.1)$$

where $W^S(t)$ is a Wiener process. The parameter r stands for the interest rate and for simplicity it is assumed to be a constant. The square root of $v(t)$ is assumed to be an Ornstein-Uhlenbeck process satisfying

$$d\sqrt{v(t)} = -\beta\sqrt{v(t)}dt + \delta dW^v(t), \quad (4.2)$$

where $\beta > 0$ and δ are real constants and $W^v(t)$ is another Wiener process, that has a correlation ρ with $W^S(t)$, i.e.

$$dW^S(t)dW^v(t) = \rho dt. \quad (4.3)$$

We can show that $v(t)$ is the CIR process named after its creators John C. Cox, Jonathan E. Ingersoll, and Stephen A. Ross [8]. Supposing $X(t) = \sqrt{v(t)}$ and using Itô's lemma we can calculate the differential of $v(t) = X^2(t)$

$$\begin{aligned} dv(t) &= 2X(t)dX(t) + (dX(t))^2 \\ &= 2\sqrt{v(t)}[-\beta\sqrt{v(t)}dt + \delta dW^v(t)] + \delta^2 dt, \\ &= [\delta^2 - 2\beta v(t)]dt + 2\delta\sqrt{v(t)}dW^v(t), \end{aligned}$$

this can be rewritten as

$$dv(t) = \kappa[\theta - v(t)]dt + \sigma\sqrt{v(t)}dW^v(t), \quad (4.4)$$

which is the square-root mean reverting process with the long run variance θ , mean reversion rate κ and with parameter σ , which is referred to as the volatility of the variance. It was described by Cox, Ingersoll and Ross in [8]. The variance process is always positive and can not reach zero if $2\kappa\theta > \sigma^2$, this is often referred to as the Feller condition [10].

4.2 Option valuation

Using the standard arbitrage arguments from Merton [27] and Black-Scholes [5], the value of any asset $c(S, v, t)$ must satisfy the partial differential equation

$$\frac{1}{2}S^2v\frac{\partial^2c}{\partial S^2} + \rho\sigma Sv\frac{\partial^2c}{\partial Sv} + \frac{1}{2}\sigma^2v\frac{\partial^2c}{\partial v^2} + rS\frac{\partial c}{\partial S} + \{\kappa[\theta - v(t)] - \lambda(S, v, t)\}\frac{\partial c}{\partial v} - rc + \frac{\partial c}{\partial t} = 0, \quad (4.5)$$

where ρ , σ , κ are as above and the term $\lambda(S, v, t)$ represents the price of volatility risk and must be independent of the particular asset. Heston [14] assumed the risk is proportional to v

$$\lambda(S, v, t) = \lambda v. \quad (4.6)$$

A European call option maturing at time T with strike price K satisfies the PDE (4.5) subject to the boundary conditions:

$$c(S, v, T) = (S - K)^+ \quad (4.7)$$

$$c(0, v, t) = 0 \quad (4.8)$$

$$c_S(\infty, v, t) = 1 \quad (4.9)$$

$$rS\frac{\partial c}{\partial S}(S, 0, t) + \kappa\theta\frac{\partial c}{\partial v}(S, 0, t) - rc(S, 0, t) + \frac{\partial c}{\partial t}(S, 0, t) = 0 \quad (4.10)$$

$$c(S, \infty, t) = S \quad (4.11)$$

Heston was looking for the solution of (4.5) in the form corresponding to Black-Scholes formula

$$c(S, v, t) = SP_1 - KP(t, T)P_2, \quad (4.12)$$

where the first term represents the present value of the spot asset upon optimal exercise and the second term is the present value of the strike price payment

$$P(t, T) = e^{-r(T-t)}. \quad (4.13)$$

" P_1 is the delta of the European call option and P_2 is the conditional risk neutral probability that the asset price will be greater than K at the maturity." Both of these terms must satisfy the original PDE. Heston wrote them in terms of the logarithm of the spot price

$$x = \ln S. \quad (4.14)$$

When we substitute the solution (4.12) to the original equation (4.5), it shows that P_1 and P_2 must satisfy

$$\frac{1}{2}v \frac{\partial^2 P_j}{\partial x^2} + \rho\sigma v \frac{\partial^2 P_j}{\partial x \partial v} + \frac{1}{2}\sigma^2 v \frac{\partial^2 P_j}{\partial v^2} + (r + u_j v) \frac{\partial P_j}{\partial x} + (a - b_j v) \frac{\partial P_j}{\partial v} + \frac{\partial P_j}{\partial t} = 0, \quad (4.15)$$

for both $j = 1, 2$, where the parameters u_j , a_j and b_j are defined as follows

$$u_1 = \frac{1}{2}, \quad u_2 = -\frac{1}{2}, \quad a = \kappa\theta, \quad b_1 = \kappa + \lambda - \rho\sigma, \quad b_2 = \kappa + \lambda. \quad (4.16)$$

For the option price to satisfy the boundary conditions (4.7) - (4.11), the partial differential equations (4.15) are subject to the terminal condition

$$P_j(x, v, T; \ln[K]) = \mathbb{I}_{\{x \geq \ln[K]\}}, \quad (4.17)$$

where \mathbb{I} is an indicator function. As explained in the Appendix of Heston [14], when x is the stochastic process satisfying

$$\begin{aligned} dx(t) &= [r + u_j v(t)]dt + \sqrt{v(t)}dW^S(t), \\ dv(t) &= [a - b_j v(t)]dt + \sigma\sqrt{v(t)}dW^v(t), \end{aligned}$$

where the parameters u_j , a and b_j remain as defined in (4.16), then P_j is the conditional probability that the option expires in-the-money

$$P_j(x, v, T; \ln[K]) = \mathbb{P}[x(T) \geq \ln[K] \mid x(t) = x, v(t) = v]. \quad (4.18)$$

Appendix of Heston [14] shows that the characteristic functions of these probabilities satisfy the same partial differential equations (4.15), subject to the terminal condition

$$f_j(x, v, T; \phi) = e^{i\phi x}. \quad (4.19)$$

The characteristic function solution takes the form

$$f_j(x, v, t; \phi) = \exp\{C(T - t; \phi) + D(T - t; \phi)v + i\phi x\}, \quad (4.20)$$

where

$$C(\tau; \phi) = r\phi i\tau + \frac{a}{\sigma^2} \left\{ (b_j - \rho\sigma\phi i + d)\tau - 2\ln \left[\frac{1 - ge^{dr}}{1 - g} \right] \right\}, \quad (4.21)$$

$$D(\tau; \phi) = \frac{b_j - \rho\sigma\phi i + d}{\sigma^2} \left[\frac{1 - e^{dr}}{1 - ge^{dr}} \right], \quad (4.22)$$

and

$$g = \frac{b_j - \rho\sigma\phi i + d}{b_j - \rho\sigma\phi i - d'}, \quad (4.23)$$

$$d = \sqrt{(\rho\sigma\phi i - b_j)^2 - \sigma^2(2u_j\phi i - \phi^2)}, \quad (4.24)$$

for $j = 1, 2$.

The terms P_1, P_2 are defined via the inverse Fourier transformation, for $j = 1, 2$

$$P_j(x, v, T; \ln[K]) = \frac{1}{2} + \frac{1}{\pi} \int_0^\infty \operatorname{Re} \left[\frac{e^{-i\phi \ln[K]} f_j(x, v, T; \phi)}{i\phi} \right] d\phi. \quad (4.25)$$

The equations (4.12), (4.25) and (4.20) give the solution for European call options as first introduced by Heston in [14]. The integrand in equation (4.25) is a smooth function that decays rapidly and presents no difficulties as was pointed out also in [14].

4.3 Valuation under risk-neutral measure

In order to eliminate the parameter λ we define the risk-adjusted parameters

$$\kappa^* = \kappa + \lambda$$

and

$$\theta^* = \frac{\kappa\theta}{\kappa + \lambda},$$

thus substituting into equation (4.5) gives under the risk-neutral measure $\tilde{\mathbb{P}}$, also called equivalent¹ martingale measure (more on this topic can be found for example in Chapter 5 of Shreve [32])

$$\frac{1}{2}S^2v \frac{\partial^2 c}{\partial S^2} + \rho\sigma Sv \frac{\partial^2 c}{\partial Sv} + \frac{1}{2}\sigma^2v \frac{\partial^2 c}{\partial v^2} + rS \frac{\partial c}{\partial S} + \{\kappa^*[\theta^* - v(t)]\} \frac{\partial c}{\partial v} - rc + \frac{\partial c}{\partial t} = 0. \quad (4.26)$$

¹Equivalent measures agree about which sets have probability zero, see Definition 1.6.3 in Shreve [32].

Since we are interested in pricing, we will assume that the Heston process with parameters fitted to option prices generates the risk-neutral measure so that the market price of volatility risk λ in the general valuation equation (4.5) is set to zero in equation (4.26). Derivation of (4.26) can be found in Appendix. Under the risk neutral measure we have now

$$a = \kappa^* \theta^*, \quad b_1 = \kappa^* - \rho\sigma, \quad b_2 = \kappa^*.$$

In what follows, we will assume that the SDEs for S and v are in risk-neutral terms because we are interested in fitting the Heston model to option prices. Effectively, we assume that we are imputing the risk-neutral measure directly by fitting the parameters of the process that we are imposing, see Gatheral [11].

4.4 The Little Heston Trap

The Characteristic function solution is (4.20) as presented by Heston in [14], however this formulation of the Heston characteristic function was later the subject of the paper called "The Little Heston Trap" [1] by Albrecher et al., who investigated in detail the properties and relations between the original formulation given by Heston and other formulation presented for example in Gatheral [11], which is identical to the Heston formulation, but causes fewer numerical problems in the implementation of the model. Change of term (4.23) is desired and the new form is

$$g' = \frac{b_j - \rho\sigma\phi i - d}{b_j - \rho\sigma\phi i + d}, \quad (4.27)$$

which is in fact

$$g' = g^{-1}, \quad (4.28)$$

as well as change of formulation of terms $C(\tau; \phi)$ given by (4.21) and $D(\tau; \phi)$ given by (4.22). The new formulations read

$$C(\tau; \phi) = r\phi i \tau + \frac{a}{\sigma^2} \left\{ (b_j - \rho\sigma\phi i - d)\tau - 2 \ln \left[\frac{1 - g e^{d\tau}}{1 - g} \right] \right\}, \quad (4.29)$$

$$D(\tau; \phi) = \frac{b_j - \rho\sigma\phi i - d}{\sigma^2} \left[\frac{1 - e^{d\tau}}{1 - g e^{d\tau}} \right]. \quad (4.30)$$

These changes produce an equivalent formulation to the original Heston formulation, but regarding properties of both formulations Albrecher et al. in [1] provided proofs that the Heston formulation is unstable under certain conditions and the alternative formulation is stable under the full parameter space. They also established a threshold maturity from which the Heston formulation suffers from instability. When the Feller condition is exactly satisfied, they encountered no problems in any of both versions. The paper of Kahl and Jäckel [19] also concerns itself with this problem and

use the rotation count algorithm to overcome the discontinuities in the original Heston formulation.

Chapter 5

Simulation

5.1 Heston simulation schemes

Monte Carlo approach is a popular and flexible pricing alternative. As computational power still increases it may result in even wider application of Monte Carlo techniques in the near future. Especially for certain derivatives which can not be priced in a closed form.

5.1.1 Dynamics of the model

Let us consider the system

$$dS(t) = rS(t)dt + \sqrt{v(t)}S(t)d\tilde{W}^S(t), \quad S(0) := S_0 \geq 0, \quad (5.1)$$

$$dv(t) = \kappa(\theta - v(t))dt + \sigma\sqrt{v(t)}d\tilde{W}^v(t) \quad v(0) := v_0 \geq 0, \quad (5.2)$$

where r , κ , θ and σ are as in previous chapter and $(\tilde{W}^S, \tilde{W}^v)$ is a two-dimensional Brownian motion under the risk-neutral measure $\tilde{\mathbb{P}}$ with instantaneous correlation ρ , i.e.

$$d\tilde{W}^S(t)d\tilde{W}^v(t) = \rho dt. \quad (5.3)$$

5.1.2 Euler scheme

Let us now apply the Euler scheme (2.18) to the variance process (5.2), a naive Euler discretization of the variance process reads

$$v_{n+1} = v_n + \kappa(\theta - v_n)\Delta_n + \sigma\sqrt{v_n}Z_v\sqrt{\Delta_n} \quad (5.4)$$

where Z_v is standard normal random variable. The main problem when using the above scheme is the fact, that the scheme does not preserve the positivity of the process. The discrete process v_n can become negative with non-zero probability,

$$P(v_{n+1} < 0) = P(Z_v < \frac{-v_n - \kappa(\theta - v_n)\Delta_n}{\sigma\sqrt{v_n}\sqrt{\Delta_n}}) = \Phi(\frac{-v_n - \kappa(\theta - v_n)\Delta_n}{\sigma\sqrt{v_n}\sqrt{\Delta_n}}),$$

as Z_v is standard normal variable, Φ denotes the standard normal cumulative distribution function. This makes the computation of $\sqrt{v_n}$ impossible and the time-stepping scheme fails. To avoid this, the practitioners adopted two ways of fixing it. First approach is referred to as the absorption. This approach follows the rule that process equals to zero whenever it attains a negative value. The other approach is called reflection. The process is being reflected by its origin. i.e. : if $v < 0$ then $v = -v$. [11]

The various schemes can be unified in the following framework

$$v_{n+1} = f_1(v_n) + \kappa(\theta - f_2(v_n))\Delta_n + \sigma\sqrt{f_3(v_n)}Z_v\sqrt{\Delta_n}, \quad (5.5)$$

where $f_i(x) = x$ for $x \leq 0$ and $i = 1, 2, 3$ and $f_3(x) \leq 0$ for all $x \in R$.

The particular choices for the "fixing" functions $f_i(x)$ in the literature are the identity function ($f(x) = x$), for the absorption ($f(x) = x^+$), where $x^+ = \max(x, 0)$ and ($f(x) = |x|$) for reflection. In the paper by Higham and Mao [15] the scheme uses the following configuration $f_1(x) = f_2(x) := x$ and $f_3(x) := |x|$, whereas for example in paper by Deelstra and Delbaen [9] $f_3(x) := x^+$ is used. All schemes coincide and are to be bias-free as $\Delta_n \rightarrow 0$ and so the choice of the fix functions may seem almost indifferent, but the contrary is true. While some schemes are extremely biased for practical sizes of the time step, others turn out to be almost bias-free for not "too extreme parameter configurations" as pointed out in [34] by Van Haastrecht and Pelsler.

According to the paper by Lord et al. [24] scheme called the full truncation scheme produces the smallest bias, closely followed by the moment-matching, Euler scheme using log-normal approximation by Andersen and Brotherton-Ratcliffe [3] and the partial truncation scheme by Deelstra and Delbaen in [9].

The full truncation scheme uses $f_1(x) := x$ and $f_2(x) = f_3(x) := x^+$. The resulting Euler scheme reads

$$v_{n+1} = v_n + \kappa(\theta - v_n^+)\Delta_n + \sigma\sqrt{v_n^+}Z_v\sqrt{\Delta_n}, \quad (5.6)$$

where $x^+ = \max(x; 0)$.

The simulation schemes of the corresponding asset price process (5.1) is then

$$S_{n+1} = S_n + rS_n\Delta_n + S_n\sqrt{v_n^+}Z_S\sqrt{\Delta_n}, \quad (5.7)$$

where Z_S is standard normal random variable with correlation ρ to Z_v . This pair of correlated standard normal random variables Z_v and Z_S with a correlation of ρ between the driving Brownian motions using Cholesky decomposition can be simulated by setting

$$Z_v := Z_1 \quad (5.8)$$

and

$$Z_S := \rho Z_v + \sqrt{1 - \rho^2} Z_2, \quad (5.9)$$

where Z_1 and Z_2 are two independent draws from the standard normal distribution.

Alternatively, the process for $\log S(t)$ can be simulated, and then exponentiated. If we discretize log-stock price S rather than the stock price S , there are no higher order corrections to the Euler discretization for details, see [11]. The log-Euler scheme

$$\log(S_{n+1}) = \log(S_n) + [r - \frac{1}{2}v_n^+] \Delta_n + \sqrt{v_n^+} Z_S \sqrt{\Delta_n}. \quad (5.10)$$

as noted in [34] does not entail any discretization error in the stock price direction, on the other hand the scheme usually shows biases in the Euler discretization of the variance process (and thus in resulting stock prices).

5.1.3 Milstein scheme

Now we consider the scheme (2.20), which is a refinement to the Euler scheme produced by adding an extra term using Itô lemma. As pointed out by Gatheral in [11], applying this scheme to the variance process yields reduced frequency with which the process goes negative relative to the Euler scheme. Nevertheless, negative values still appear and we have to adopt similar fixes. As $a = \kappa(\theta - v_n)$ and $b = \sigma\sqrt{v_n}$ the scheme reads

$$v_{n+1} = v_n + \kappa(\theta - v_n^+) \Delta_n + \sigma\sqrt{v_n^+} Z_v \sqrt{\Delta_n} + \frac{1}{4}\sigma^2(Z_v^2 - 1)\Delta_n, \quad (5.11)$$

where Z_v stands for standard normal random variable. Similarly we can apply the scheme (2.20) to the the process $S(t)$, as this time $a = rS(t)$ and $b = S(t)\sqrt{v_t}$ produces

$$S_{n+1} = S_n + rS_n \Delta_n + S_n \sqrt{v_n^+} Z_S \sqrt{\Delta_n} + \frac{1}{2}S_n v_n^+ (Z_S^2 - 1)\Delta_n, \quad (5.12)$$

where Z_S is standard normal random variable with correlation ρ to Z_v . This however is not a refinement to the log-Euler scheme for the log-stock price S introduced earlier as (5.10).

5.1.4 Kahl-Jäckel scheme

Introduced in Kahl and Jäckel [20] as IJK scheme, it follows

$$v_{n+1} = \frac{v_n + \kappa\theta\Delta t + \sigma\sqrt{v_n} Z_v \sqrt{\Delta t} + \frac{1}{4}\sigma^2\Delta t(Z_v^2 - 1)}{1 + \kappa\Delta t},$$

$$\begin{aligned}\log(S_{n+1}) = & \log S_n + r\Delta_n - \frac{1}{4}(V_{n+1} + V_n)\Delta_n + \rho\sqrt{V_n}\sqrt{\Delta_n}Z_v + \frac{1}{2}(\sqrt{v_n} + \\ & + \sqrt{v_{n+1}})(Z_S - \rho Z_v)\sqrt{\Delta_n} + \frac{1}{4}\sigma\rho(Z_v^2 - 1)\Delta_n.\end{aligned}$$

This scheme was developed as a *drop-in* replacement for the standard log-Euler procedure since it requires no additional random numbers per step or any adaptive refinement. Authors Kahl and Jäckel [20] claim the IJK scheme is not superior in convergence order to the log-Euler scheme, but they praise the fact that it excels over other equally simple methods by showing a superior convergence behaviour that is faster by a multiplicative factor.

5.1.5 Haastreht and Pelsser [34] revised

Haaserecht and Pelsler commented on Kahl-Jäckel scheme implemented in the following way. Conditional on time s a discretization of the log Stock price process S for $t > s$ (with $\delta t := t - s$) reads exactly

$$\begin{aligned}\log(S(t)) = & \log(S(s)) + \left[r - \frac{v(s) + v(t)}{4}\right]\Delta_t + \rho\sqrt{v(s)}Z_V\sqrt{\Delta_t} + \\ & + \frac{1}{2}(\sqrt{v(s)} + \sqrt{v(t)})(Z_S + \rho Z_V)\sqrt{\Delta_t} + \frac{\rho\tilde{\zeta}\Delta t}{2}(Z_V^2 - 1)\end{aligned}$$

We left the original notation as well. This however does not match our implementation of the scheme. The Heston log price S comes from the formula by Kahl and Jäckel [20], which is exactly

$$\begin{aligned}\ln S_{t_{n+1}} = & \ln S_{t_n} + \mu\Delta t_n - \frac{1}{4}(V_{t_n}^{2p} + V_{t_{n+1}}^{2p})\Delta t_n + \rho V_{t_n}^p \Delta Z_n + \\ & + \frac{1}{2}(V_{t_n}^p + V_{t_{n+1}}^p)(\Delta W_n - \rho\Delta Z_n) + \frac{1}{2}\rho p V_{t_n}^{p-1} b_n \cdot (\Delta Z_n^2 - \Delta t_n).\end{aligned}$$

For Heston model we have parameter $p = \frac{1}{2}$. Also note that $\Delta W = Z_S\sqrt{\Delta t}$ and ΔZ is $Z_V\sqrt{\Delta t}$ to match the notation a bit as well.

Clearly the sign in the term $(Z_S + \rho Z_V)$ is wrong and should be $(Z_S - \rho Z_V)$ instead. What appears not to be a typo is the the mistake in the last term, where there is $\frac{1}{2}$ instead of $\frac{1}{4}$ compared to our equation. b_n is ζv_n^p for Heston in the original article by Kahl and Jäckel [20]. This implies $b(s) = \zeta v(s)^{1/2}$, which cancels out with $v(s)^{-1/2}$ and leaves ζ in the notation of Haastreht and Pelsser. This is probably due to the fact that the IJK scheme in Kahl-Jäckel [20] was not stated explicitly for Heston model. The wrongly interpreted scheme published in [34] can be found in some other papers as well.

5.1.6 Exact scheme

In Broadie and Kaya [6] an exact simulation algorithm has been devised for the Heston model. In numerical comparisons of their algorithm to an Euler discretisation with the absorption fix, they find that for the pricing of European options in the Heston model and variations thereof, the exact algorithm compares favourably in terms of root-mean-squared (RMS) error. According to Broadie and Kaya [6] the volatility at time t , given the values of v_u and for $u < t$, can be written as

$$v_t = v_u + \kappa\theta(t - u) - \kappa \int_u^t v_s ds + \sigma \int_u^t \sqrt{v_s} dW_s^{(1)} \quad (5.13)$$

and similarly the stock price at time t , given the values of S_u and v_u for $u < t$, can be written as

$$\log(S_t) = \log(S_u) + r(t - u) - \frac{1}{2} \int_u^t v_s ds + \rho \int_u^t \sqrt{v_s} dW_s^{(1)} + \sqrt{1 - \rho^2} \int_u^t \sqrt{v_s} dW_s^{(2)}, \quad (5.14)$$

where $W^{(1)}$ and $W^{(2)}$ are uncorrelated Brownian motions. Substituting $\int_u^t \sqrt{v_s} dW_s^{(1)}$ from (5.13) into (5.14) yields

$$\begin{aligned} \log(S_t) = & \log(S_u) + r(t - u) + \left(\frac{\kappa\rho}{\sigma} - \frac{1}{2}\right) \int_u^t v_s ds + \frac{\rho}{\sigma}(v_t - v_u - \kappa\theta(t - u)) \\ & + \sqrt{1 - \rho^2} \int_u^t \sqrt{v_s} dW_s^{(2)}. \end{aligned} \quad (5.15)$$

The exact simulation method uses sampling exactly from the distribution of v_t . This can be done given v_u for some $u < t$ by using sampling from the noncentral chi-squared distribution. The next step having v_t and sample from v_u is generating a sample from the distribution of the integral $\int_u^t v_s ds$.

This is done by inverting the characteristic function. Broadie and Kaya uses this approach to obtain the probability function of the random variable $V(u, t)$, which has the same distribution as the integral $\int_u^t v_s ds$ conditional on v_u and v_t . They use the trapezoidal rule to compute the probability distribution. The value of the integral is then simulated by the inverse transform method. By first generating a uniform standard normal variable U and then looking for the value of x for which the $\Pr\{V(u, t) \leq x\}$ is equal to U .

Knowing all the terms in (5.13) except for $\sigma \int_u^t \sqrt{v_s} dW_s^{(1)}$, the value of $\sigma \int_u^t \sqrt{v_s} dW_s^{(1)}$ can be easily obtained from (5.13). Furthermore they use the fact that v_t is independent of the Brownian motion $W(t)^{(2)}$ and therefore the distribution of $\int_u^t \sqrt{v_s} dW_s^{(2)}$ given the path generated by v_t is normal with mean 0 and variance $\int_u^t \sqrt{v_s}$. This scheme gives an unbiased estimator of the derivative price.

The algorithm is however very time-consuming, because of the application of the numerical inversion of a cumulative distribution using the characteristic function involving modified Bessel function of the first kind. Also we found out that with finer discretization the modified Bessel function of the first kind takes too large values (in particular - unrepresentable values). Therefore it is not recommended for the pricing of options when requiring the value of the asset price on a larger number of time instants. Nonetheless, (5.15) will be used in the next sections.

5.1.7 QE scheme

The Quadratic Exponential scheme, also known as QE was introduced by Andersen [2]. It exploits the fact that the value v_{n+1} conditional on value v_n follows a non-central chi-square distribution. It uses two different approximations to the non-central chi-square distribution, depending on the values of the variance.

The non-centrality parameter of the distribution for v_{n+1} is proportional to v_n and for sufficiently large values of v_n the scheme uses a power function applied to a standard normal variable Z_v to approximate v_{n+1} . Thus a critical level $\psi_C \in [1, 2]$ is introduced and compared to ψ , which is given by

$$\psi = s^2/m^2,$$

where given v_n

$$s^2 = \frac{v_n \sigma^2 e^{-\kappa \Delta_n}}{\kappa} (1 - e^{-\kappa \Delta}) + \frac{\theta \sigma^2}{2\kappa} (1 - e^{-\kappa \Delta})^2$$

and

$$m = \theta + (v_n - \theta)e^{-\kappa \Delta_n}.$$

The algorithm uses U_v , which is a uniform random number.

1. If $\psi \leq \psi_C$ we set

$$v_{n+1} = a(b + Z_v)^2,$$

where

$$a = \frac{m}{1 + b^2},$$

$$b^2 = 2\psi^{-1} - 1 + \sqrt{2\psi^{-1} - 1} \sqrt{2\psi^{-1} - 1} \geq 0$$

and

$$Z_v = \Phi^{-1}(U_v).$$

2. If $\psi > \psi_C$ we set

$$v_{n+1} = \Psi^{-1}(U_v; p, \beta),$$

where

$$\Psi^{-1}(u) = \Psi^{-1}(u; p, \beta) = \begin{cases} 0 & \text{for } 0 \leq u \leq p \\ \beta^{-1} \log\left(\frac{1-p}{1-u}\right) & \text{for } p < u \leq 1 \end{cases},$$

$$p = \frac{\psi - 1}{\psi + 1},$$

and

$$\beta = \frac{1-p}{m}.$$

As stated above the value of ψ_C indicates which approximation to use. The first approximation can be only used for $\psi \leq 2$. For higher values of ψ , that correspond to low values of v_n , the scheme will fail. For $\psi \geq 1$ the second one can be used. Therefore $\psi_C \in [1, 2]$. Andersen in [2] states that the exact choice for ψ_C appears to have relatively small effects on the quality of the overall simulation scheme and proposes $\psi_C = 1.5$.

This scheme can not be used with the Euler discretization of $S(t)$, since it would result in too feeble effective correlation and, consequently, paths of S with poor distribution tails. Instead Andersen used the exact scheme from Broadie and Kaya (5.15) and tried to preserve the key driver of the correlation between S_{n+1} and v_{n+1} , which is $\frac{\rho}{\sigma}v_{n+1}$. Instead of sampling the integral $\int_u^t v_s ds$ he used an approximation

$$\int_u^t v_s ds \approx \Delta_n[\gamma_1 v_n + \gamma_2 v_{n+1}],$$

where setting $\gamma_1 = 1, \gamma_2 = 0$ yields Euler like setting. For the last integral same approach as in (5.14) is applied, thus

$$\int_u^t \sqrt{v_s} dW_s^{(2)} \approx \sqrt{\gamma_1 v_n + \gamma_2 v_{n+1}} Z_S \sqrt{\Delta_n},$$

where Z_S is standard normal random variable. This leads to the following discretization scheme for $S(t)$

$$\log(S_{n+1}) = \log(S_n) + K_0 + K_1 v_n + K_2 v_{n+1} + \sqrt{K_3 v_n + K_4 v_{n+1}} \cdot Z_S, \quad (5.16)$$

where K_0, \dots, K_4 are given by:

$$K_0 = -\frac{\rho\kappa\theta}{\sigma}\Delta_n, \quad K_1 = \gamma_1 \Delta_n \left(\frac{\kappa\rho}{\sigma} - \frac{1}{2} \right) - \frac{\rho}{\sigma}, \quad K_2 = \gamma_2 \Delta_n \left(\frac{\kappa\rho}{\sigma} - \frac{1}{2} \right) + \frac{\rho}{\sigma},$$

$$K_3 = \gamma_1 \Delta_n (1 - \rho^2), \quad K_4 = \gamma_2 \Delta_n (1 - \rho^2).$$

and $\gamma_1 = \gamma_2 = 0.5$ for a central discretization of the time integral of v in the exact scheme.

Equation (5.16) is the exact proposal as can be found in Andersen [2]. However, we will also include the drift, thus the resulting scheme we will use for $S(t)$ reads

$$\log(S_{n+1}) = \log(S_n) + r\Delta_n + K_0 + K_1v_n + K_2v_{n+1} + \sqrt{K_3v_n + K_4v_{n+1}} \cdot Z_S. \quad (5.17)$$

5.1.8 E+M scheme

We will also consider a combination of the exact scheme for the underlying asset process with Milstein scheme for the volatility process. The motivation for this is the fact that the frequency of negative values of the variance process decreases for the Milstein scheme (5.11) compared to Euler as pointed out for example in Gatheral [11]. The motivation behind the use of the exact scheme by Broadie and Kaya (5.15) is that it can be used for discretizations on a larger number of time instants when not trying to sample the integral, but using an approximation instead.

Thus, we expect this scheme to perform better than Milstein and log-Euler scheme. The question is whether it can outperform the QE scheme, because QE scheme approximates the non-central chi-square distribution and therefore does not suffer from negative values for the variance process. However, the proposed scheme does not include switching between different approximations, which makes it easier to implement and quite possibly faster compared to the QE scheme.

5.2 MC simulations

Most of the tests done in previous literature are quite easy to pass, they typically involve at-the-money options and model parameters are not similar to what we are about to obtain in the Chapter 7. Thus we aim to test the schemes using parameters obtained from real market and for range of Strikes, that is also similar to market range.

Table 5.1: Set of used parameters from March 19, 2013.

| Parameter | Value |
|-----------|----------|
| v0 | 0.02497 |
| kappa | 1.22136 |
| theta | 0.06442 |
| sigma | 0.55993 |
| rho | -0.66255 |
| S0 | 7962.31 |
| r | 0.00207 |

We use the set of parameters listed in Table 5.1. The model implied parameters were obtained by Genetic algorithm and then refined using nonlinear least-squares method. Using the model with this set of parameters we were able to obtain prices with mean absolute relative error of 0.65% and maximum absolute relative error of 2.22%. For details, see Chapter 7.

However, as we stated above the choice of the parameters can cause problems. The following Figure 5.1 shows the mean path of the underlying asset for the examined schemes. We used $\Delta = 2^{-6}$ and $T = 1$. Number of simulations was nPaths=100 000.

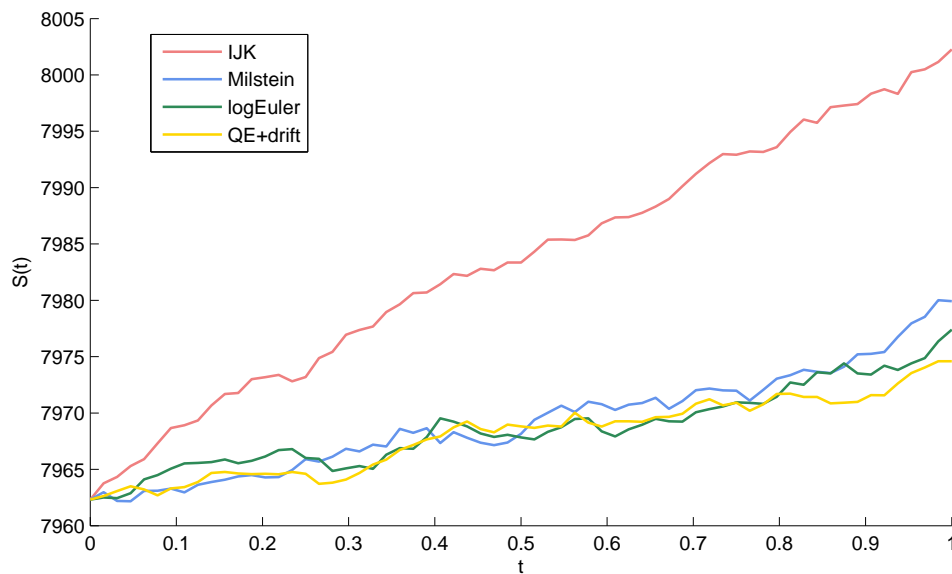


Figure 5.1: Mean of 100000 paths using parameters from Table 5.1, $\Delta = 2^{-6}$ and $T = 1$.

It is clear in Figure 5.1 that the paths simulated by IJK scheme differ significantly. This behaviour is invariable over changing number of simulation and the step size Δ and results in great errors when trying to price the options using Monte-Carlo method. IJK scheme clearly does not work for this set of parameters and thus we will consider it as a competing scheme. Nonetheless the results for IJK scheme will be presented for illustration.

It is arguable whether the IJK scheme is of any practical use, since even with so to speak friendlier set of parameters the IJK scheme seems to be outperformed by the other schemes presented here.

5.3 MC test

We designed the following test for measuring performance of pricing the options using the previous schemes.

We have four maturities at times $T=0.25, 0.5, 0.75$ and 1 . We will be pricing 28 options for each maturity with strikes

$$[7100, 7200, \dots, 7500, 7550, 7600, \dots, 8350, 8400, 8500, \dots, 9000]$$

We can compute the price of the options at time 0 denoted as $C(0)$ with the parameters from Table 5.1 using the pricing formula from the previous chapter for all values of K and for each maturity. Resulting in $4 \times 28 = 112$ option prices.

Similarly we can use the Monte-Carlo method and compute the approximate value $\hat{C}(0)$ by computing the expectation

$$\hat{C}(0) = e^{-rT} \mathbb{E}[(\hat{S}(T) - K)^+], \quad (5.18)$$

where we will be using the introduced schemes to approximate the value of $S(T)$ by $\hat{S}(T)$. In particular, for each scheme for $\hat{S}(T)$ we will draw $nPaths$ independent samples of $\hat{S}^{(1)}(T), \hat{S}^{(2)}(T), \dots, \hat{S}^{(nPaths)}(T)$, where $nPaths = 100\,000, 250\,000$ and $500\,000$ using time grid with step $\Delta = 2^{-6}, \dots, 2^{-11}$. Thus $\hat{C}(0)$ will be estimated as

$$\hat{C}(0) \approx e^{-rT} \frac{1}{nPaths} \sum_{i=1}^{nPaths} (\hat{C}^{(i)}(T) - K)^+. \quad (5.19)$$

This will lead to error

$$e = C(0) - \hat{C}(0). \quad (5.20)$$

We will be measuring errors for different steps Δ and also number of simulations used in terms of mean absolute relative error, thus

$$\hat{\varepsilon}_j = \frac{1}{N} \sum_{n=1}^N \frac{|C(0) - \hat{C}(0)|}{C(0)}, \quad (5.21)$$

where N is the number of options, in our case $N = 112$. However $\hat{\varepsilon}_j$ will be a random variable taking different value in different batch $j = 1, \dots, M$. We will estimate the variance $\sigma_{\hat{\varepsilon}}^2$ of $\hat{\varepsilon}$ by arranging the $nPaths$ simulations into M batches and use it to construct a confidence interval for the absolute error ε . In particular we will estimate the mean of the batch averages

$$\hat{\varepsilon} = \frac{1}{M} \sum_{j=1}^M \varepsilon_j \quad (5.22)$$

and then use formula

$$\hat{\sigma}_\varepsilon^2 = \frac{1}{M-1} \sum_{j=1}^M (\hat{\varepsilon}_j - \hat{\varepsilon})^2 \quad (5.23)$$

to estimate the variance σ_ε^2 . For the Student t-distribution with $M - 1$ degrees of freedom an $100(1 - \alpha)\%$ confidence interval for ε has the form

$$(\hat{\varepsilon} - \Delta\hat{\varepsilon}, \hat{\varepsilon} + \Delta\hat{\varepsilon}), \quad (5.24)$$

where

$$\Delta\hat{\varepsilon} = t_{1-\alpha, M-1} \sqrt{\frac{\hat{\sigma}_\varepsilon^2}{M}} \quad (5.25)$$

with $t_{1-\alpha, M-1}$ is determined from the Student t-distribution with $M - 1$ degrees of freedom.

5.4 MC test results

This section presents the results. Figures 5.2 - 5.4 correspond to Tables 5.2 - 5.4. They differ in the number of simulated paths. As we can see the mean absolute relative errors decrease with higher number of simulated paths. The behaviour of the particular schemes is consistent through all the three values. Results show that the QE scheme is the best choice for cases where we do not need small step size Δ . We did not include the results for Δ smaller than 2^{-8} since the scheme was not improving and it also took too long to simulate the paths this way.

This is a factor which we mentioned earlier and indeed the E+M scheme was better in terms of speed, but as we can see from the pictures it needed four times finer discretization to reach the level of mean absolute relative error of the QE scheme.

Table 5.2: Results for $\Delta = 2^{-4} - 2^{-11}$, for 100 000 simulated paths, number of batches $M = 100$. IJK scheme is included.

| Δ | logEuler | | Milstein | | QE+drift | | E+M | | IJK | |
|-----------|---------------------|---------------------------|---------------------|---------------------------|---------------------|---------------------------|---------------------|---------------------------|---------------------|---------------------------|
| | $\hat{\varepsilon}$ | $\hat{\varepsilon}\Delta$ | $\hat{\varepsilon}$ | $\hat{\varepsilon}\Delta$ | $\hat{\varepsilon}$ | $\hat{\varepsilon}\Delta$ | $\hat{\varepsilon}$ | $\hat{\varepsilon}\Delta$ | $\hat{\varepsilon}$ | $\hat{\varepsilon}\Delta$ |
| 2^{-4} | 3.730% | 0.069% | 2.466% | 0.024% | 0.440% | 0.041% | 0.728% | 0.050% | 27.981% | 0.112% |
| 2^{-5} | 1.477% | 0.070% | 1.308% | 0.027% | 0.395% | 0.033% | 0.566% | 0.055% | 28.209% | 0.103% |
| 2^{-6} | 0.682% | 0.055% | 0.770% | 0.031% | 0.412% | 0.042% | 0.417% | 0.035% | 28.259% | 0.109% |
| 2^{-7} | 0.472% | 0.051% | 0.528% | 0.038% | 0.394% | 0.033% | 0.407% | 0.039% | 28.191% | 0.113% |
| 2^{-8} | 0.448% | 0.044% | 0.450% | 0.038% | 0.423% | 0.038% | 0.400% | 0.035% | 28.297% | 0.117% |
| 2^{-9} | 0.390% | 0.035% | 0.417% | 0.031% | - | - | 0.398% | 0.035% | 28.412% | 0.096% |
| 2^{-10} | 0.414% | 0.037% | 0.424% | 0.039% | - | - | 0.421% | 0.042% | 28.368% | 0.109% |
| 2^{-11} | 0.403% | 0.034% | 0.422% | 0.033% | - | - | 0.420% | 0.041% | 28.289% | 0.109% |

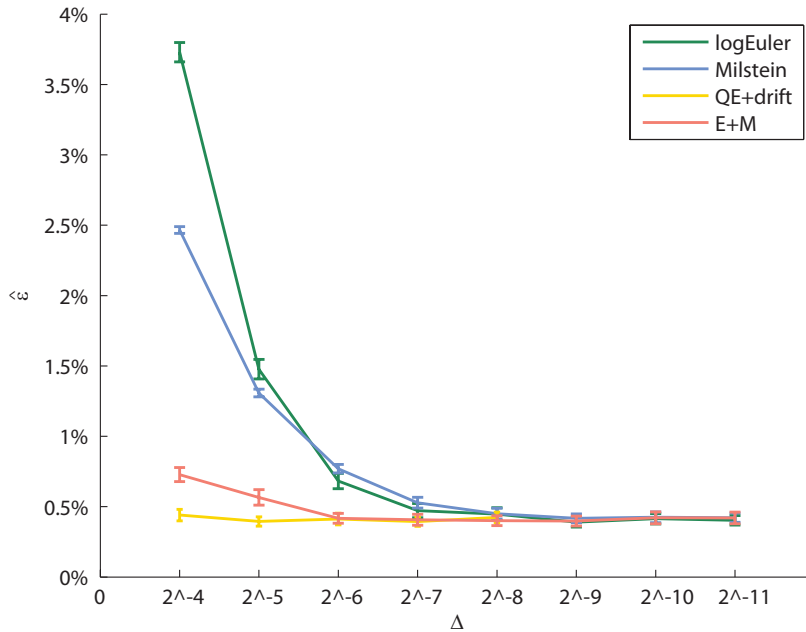


Figure 5.2: Results for $\Delta = 2^{-4} - 2^{-11}$, for 100 000 simulated paths, number of batches $M = 100$.

Nevertheless the combination of the exact approach provided with the variance process discretized using Milstein scheme performed significantly better than the log-Euler scheme and plain Milstein scheme without much of an additional code needed. And this scheme can be used when greater precision is required for larger number of times steps.

We also included the results of IJK scheme although it proved to be not useful for this set of parameters. We can see this also in Table 5.2. The errors are high and do not depend on the time step Δ . The following Figure 5.3 corresponds to Table 5.3. This time 250 000 paths were simulated.

The errors are dropping with rising number of simulations and the average absolute relative error for the QE scheme across the range of 28 Strikes for four maturities is 0.203% with $\pm 0.015\%$ 95% confidence interval when simulating 500 000 paths and taking $\Delta = 2^{-4}$, which can be observed in the last Figure 5.4. For the full results see Table 5.4 on the next page.

Table 5.3: Results for $\Delta = 2^{-4} - 2^{-11}$, for 250 000 simulated paths, number of batches $M = 100$. IJK scheme is included.

| Δ | logEuler | | Milstein | | QE+drift | | E+M | | IJK | |
|-----------|---------------------|---------------------------|---------------------|---------------------------|---------------------|---------------------------|---------------------|---------------------------|---------------------|---------------------------|
| | $\hat{\varepsilon}$ | $\hat{\varepsilon}\Delta$ | $\hat{\varepsilon}$ | $\hat{\varepsilon}\Delta$ | $\hat{\varepsilon}$ | $\hat{\varepsilon}\Delta$ | $\hat{\varepsilon}$ | $\hat{\varepsilon}\Delta$ | $\hat{\varepsilon}$ | $\hat{\varepsilon}\Delta$ |
| 2^{-4} | 3.777% | 0.051% | 2.471% | 0.016% | 0.245% | 0.020% | 0.733% | 0.041% | 27.992% | 0.071% |
| 2^{-5} | 1.388% | 0.044% | 1.262% | 0.017% | 0.261% | 0.025% | 0.419% | 0.033% | 28.140% | 0.071% |
| 2^{-6} | 0.579% | 0.034% | 0.702% | 0.017% | 0.261% | 0.022% | 0.277% | 0.026% | 28.246% | 0.060% |
| 2^{-7} | 0.307% | 0.024% | 0.429% | 0.021% | 0.273% | 0.024% | 0.273% | 0.022% | 28.261% | 0.072% |
| 2^{-8} | 0.282% | 0.024% | 0.307% | 0.022% | 0.270% | 0.028% | 0.256% | 0.020% | 28.249% | 0.072% |
| 2^{-9} | 0.269% | 0.025% | 0.267% | 0.021% | - | - | 0.261% | 0.026% | 28.357% | 0.062% |
| 2^{-10} | 0.237% | 0.021% | 0.258% | 0.022% | - | - | 0.263% | 0.023% | 28.296% | 0.060% |
| 2^{-11} | 0.248% | 0.021% | 0.242% | 0.021% | - | - | 0.271% | 0.028% | 28.386% | 0.074% |

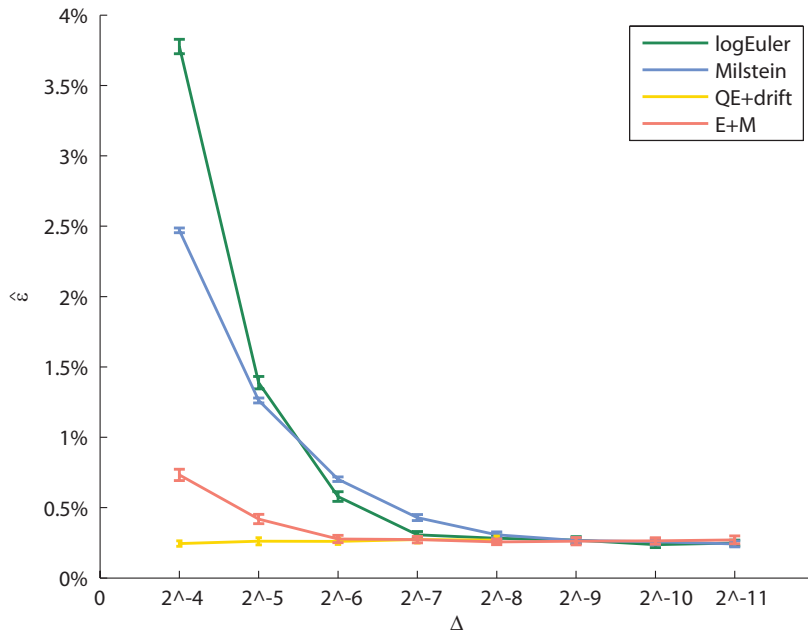


Figure 5.3: Results for $\Delta = 2^{-4} - 2^{-11}$, for 250 000 simulated paths, number of batches $M = 100$.

Table 5.4: Results for $\Delta = 2^{-4} - 2^{-11}$, for 500 000 simulated paths, number of batches $M = 100$. IJK scheme is included.

| Δ | logEuler | | Milstein | | QE+drift | | E+M | | IJK | |
|-----------|---------------------|---------------------------|---------------------|---------------------------|---------------------|---------------------------|---------------------|---------------------------|---------------------|---------------------------|
| | $\hat{\varepsilon}$ | $\hat{\varepsilon}\Delta$ | $\hat{\varepsilon}$ | $\hat{\varepsilon}\Delta$ | $\hat{\varepsilon}$ | $\hat{\varepsilon}\Delta$ | $\hat{\varepsilon}$ | $\hat{\varepsilon}\Delta$ | $\hat{\varepsilon}$ | $\hat{\varepsilon}\Delta$ |
| 2^{-4} | 3.746% | 0.036% | 2.456% | 0.010% | 0.203% | 0.015% | 0.638% | 0.026% | 27.941% | 0.051% |
| 2^{-5} | 1.396% | 0.032% | 1.255% | 0.012% | 0.166% | 0.014% | 0.361% | 0.025% | 28.103% | 0.053% |
| 2^{-6} | 0.559% | 0.025% | 0.655% | 0.011% | 0.184% | 0.014% | 0.243% | 0.022% | 28.236% | 0.051% |
| 2^{-7} | 0.286% | 0.020% | 0.376% | 0.015% | 0.188% | 0.017% | 0.195% | 0.019% | 28.291% | 0.048% |
| 2^{-8} | 0.219% | 0.017% | 0.260% | 0.018% | 0.179% | 0.020% | 0.183% | 0.014% | 28.312% | 0.046% |
| 2^{-9} | 0.187% | 0.020% | 0.212% | 0.016% | - | - | 0.188% | 0.017% | 28.354% | 0.044% |
| 2^{-10} | 0.202% | 0.019% | 0.210% | 0.018% | - | - | 0.180% | 0.018% | 28.374% | 0.043% |
| 2^{-11} | 0.181% | 0.015% | 0.193% | 0.015% | - | - | 0.191% | 0.018% | 28.370% | 0.047% |

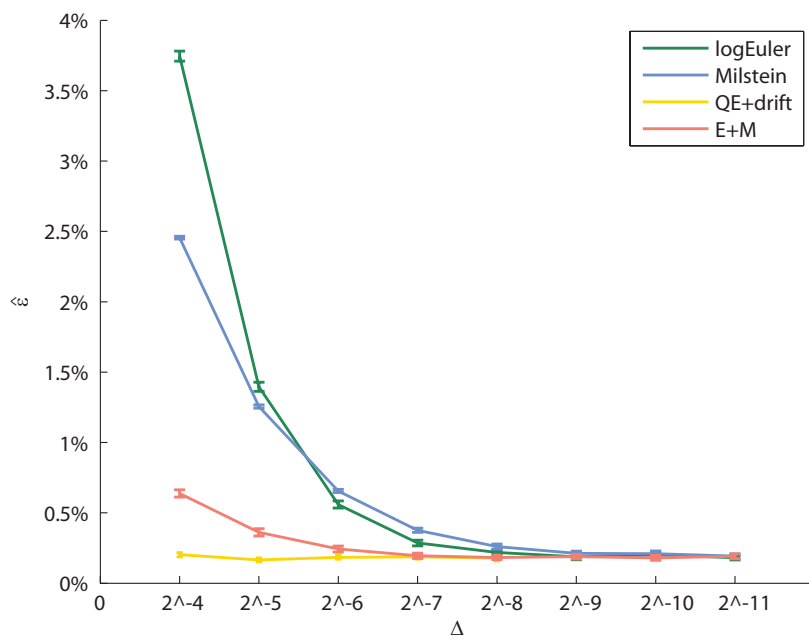


Figure 5.4: Results for $\Delta = 2^{-4} - 2^{-11}$, for 500 000 simulated paths, number of batches $M = 100$.

Chapter 6

Calibration

Market prices are used to calibrate an option pricing model, which can then be used to compute prices of more complex (exotic) options or compute hedge ratios.

Determining the model parameters to match the market prices of a set of options is what is commonly referred to as "model calibration" problem, it is in fact the inverse problem to the option pricing problem.

The price to pay for more realistic model such as the Heston model is the increased complexity of model calibration. As pointed out by Jacquier and Jarrow [18], the estimation method becomes as crucial as the model itself.

In order to compute option prices using the Heston model, we need input parameters that are not observable from the market data, therefore we can not use empirical estimates. Bakshi et al. [4] documented that the implied structural parameters differ significantly from their time-series estimated counterparts. For instance, the magnitudes of time-series correlation coefficient of the asset return and its volatility estimated from the daily prices were much lower than their option-implied counterparts.

One of the difficulties in solving the calibration problem is that the information observed from market data is insufficient to exactly identify the parameters. Several sets of parameters may perform well and produce model prices that are consistent with the prices observed on the market. This causes the ill-posedness of the problem.

In practice it is not possible nor meaningful to match exactly the observed prices. Therefore the problem of calibrating the model is formulated as an optimization problem. Our aim is to minimize the pricing error between the model prices and the market prices for a set of traded options. A common approach to measure this error is to use the squared difference between market and model prices, this approach leads to the nonlinear least square method

$$\inf_{\Theta} G(\Theta), \quad G(\Theta) = \sum_{i=1}^N w_i |C_i^{\Theta}(t, S_t, T_i, K_i) - C_i^*(T_i, K_i)|^2, \quad (6.1)$$

where w_i is a weight, $C_i^*(T_i, K_i)$ is the market price of the call option observed at

time t . C^Θ denotes the model price computed using vector of model parameters

$$\Theta = (\kappa, \theta, \sigma, v_0, \rho).$$

Thinking of G as an objective function of Θ , the optimization problem (6.1) is not an easy one. Function G is not convex and is not of any particular structure. Moreover it may have more than one global minimum and even if it had not, it is not clear whether the unique minimum can be reached by gradient based algorithm.

Local deterministic algorithms can be used, but there is always the possibility to end up in a local minimum, one has to choose an initial guess, which is "good enough". Due to high sensitivity to initial parameters various stochastic algorithms are suggested. Nevertheless, for example Mikhailov and Nögel [29] use built in Excel solver, which is a local optimizer using Generalized Reduced Gradient method.

Another method used to calibrate the model is regularisation. This method adds to the objective function (6.1) penalization function, e.g. $f(\Theta)$ such that

$$\inf G(\Theta) + \alpha f(\Theta), \tag{6.2}$$

is convex, hence gradient based optimizing procedure can be used. This yields another parameter to be estimated α , which is called regularisation parameter. For further details, see Cont and Hamida [7].

6.1 Calibration in progress

We will first test various optimizers on a set of generated prices. That means we will actually know the optimal set of parameters that fully explain the synthetic market prices. Then we proceed to use these optimizers on a real data observed on market.

6.1.1 Data generated

We generated 97 option prices using the following set of parameters.

Table 6.1: Set of used parameters

| | |
|-------|--------|
| v0 | 0.034 |
| kappa | 2.210 |
| theta | 0.073 |
| sigma | 0.568 |
| rho | -0.952 |

The choice of parameters was not random, but it was chosen to be close to parameters, that explained prices of ODAX, options on DAX¹. The initial stock price was set to 7260.33 and interest rate was 0.0022. We computed prices for 6 Maturities T with strike prices K varying from 13 to 19 for each maturity. By doing so we strived for imitation of the situation when obtaining the real market data available.

6.1.2 Measured errors

We measured the following errors:

1.MAAE (Maximum Absolute Relative Error)

$$\text{MARE}(\Theta) = \max_i \frac{|C_i^\Theta - C_i^*|}{C_i^*}; \quad i = 1, \dots, N.$$

2.AARE (Average Absolute Relative Error)

$$\text{AARE}(\Theta) = \frac{1}{N} \sum_{i=1}^N \frac{|C_i^\Theta - C_i^*|}{C_i^*} .$$

3.RMSE (Root-mean-square Error)

$$\text{RMSE}(\Theta) = \frac{1}{\sqrt{N}} \sqrt{\sum_{i=1}^N (C_i^\Theta - C_i^*)^2} .$$

In our case $N = 97$.

6.1.3 Considered weights

As defined in the beginning of this chapter w_i stands for particular weight. In practice the weights are put where the most liquid quotes are on the market. This is usually around ATM. With our market data we will use the bid ask spreads. Hence

$$\text{Spread} = \text{Ask} - \text{Bid}.$$

We aim to have the model prices as close to the Mid price defined in (3.3) as possible. Thus we will put weight on the small spreads. According to U. Nögel everybody does his own fine tuning on that, depending on the market and the products. We decided not

¹DAX (Deutscher Aktien Index = German stock index), 30 major German companies trading on the Frankfurt Stock Exchange

to pick up just one choice for the weight function, but rather test more of these choices and see if there is a significant influence on the results caused by the particular choice of the weight function.

For one particular option we denoted the weights by capital letters A,B,C, where

$$\text{weight A denotes } \frac{1}{|\text{Spread}|},$$

$$\text{weight B denotes } \frac{1}{(\text{Spread})^2}$$

$$\text{and weight C denotes } \frac{1}{\sqrt{\text{Spread}}}.$$

The bigger the spread the less weight is put on the particular difference between the model price and the market price during the calibration process.

6.2 Genetic algorithm

Genetic algorithm is part of MATLAB's Global Optimization Toolbox, hence it is ideal tool for looking for global minima of our nonlinear problem:

A genetic algorithm is a method for solving both constrained and unconstrained optimization problems based on a natural selection process that mimics biological evolution. The algorithm repeatedly modifies a population of individual solutions. At each step, the genetic algorithm randomly selects individuals from the current population and uses them as parents to produce the children for the next generation. Over successive generations, the population "evolves" toward an optimal solution.²

Genetic algorithm is easy to implement, see the following code, and also improves chances of finding a global solution for functions that are not differentiable nor continuous.

```
options = gaoptimset('Generations',500,'PopulationSize',60,'TolFun',1e-12);
[calib,fval] = ga(@error_f,5,[],[],[],[],l_bounds,u_bounds,[],options)
```

Variables `l_bounds` and `u_bounds` store the bounds for the set of optimal parameters we are looking for, thus they remain unchanged for all the used optimizers.

²Sourcing MATLAB Documentation.

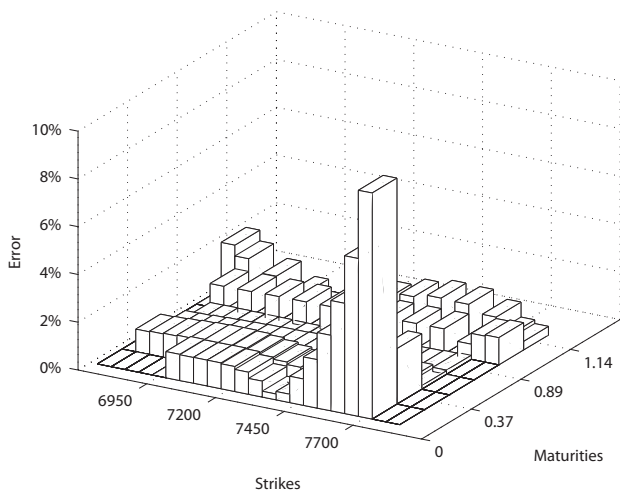
Table 6.2 below shows the parameters we were able to retrieve using Genetic algorithm with different weights w_i .

Table 6.2: Set of retrieved parameters using different weights.

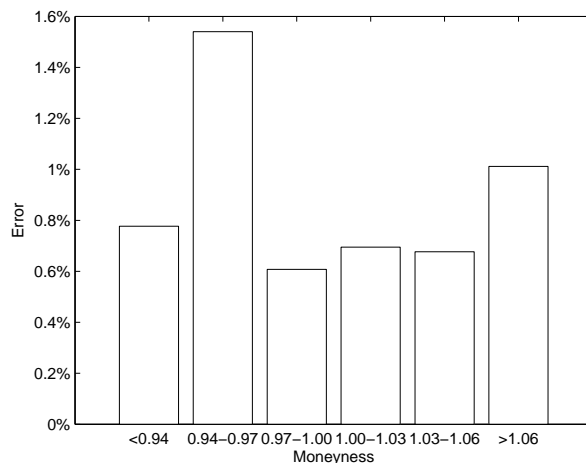
| | Used Weights | | |
|-------|--------------|----------|----------|
| | A | B | C |
| v0 | 0.03399 | 0.03537 | 0.03610 |
| kappa | 0.79087 | 0.39258 | 0.17567 |
| theta | 0.11581 | 0.18695 | 0.28482 |
| sigma | 0.47311 | 0.49517 | 0.36132 |
| rho | -0.84858 | -0.82075 | -0.91083 |

Next page contains graphical depiction of the absolute relative errors for each strike and maturity with different weights. This can be seen in the left column of Figure 6.1. The column on the right shows mean absolute relative errors in terms of moneyness for the particular choice of weights.

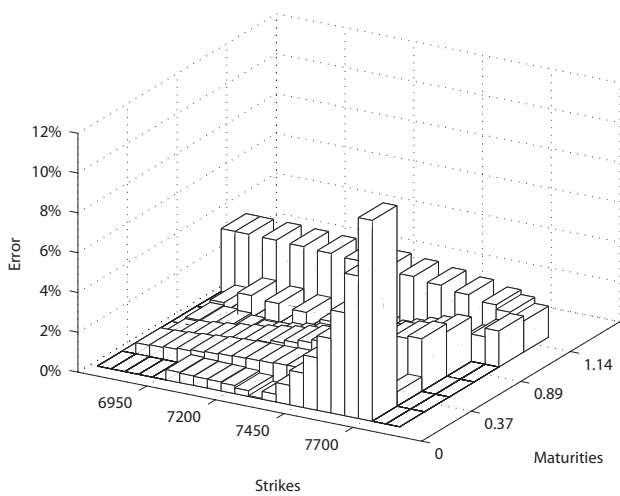
We know the parameters that generated this set of prices. These were presented in Table 6.1. Genetic algorithm did not provide us with the set that would be very close to them. But looking at the right column we can see that the absolute relative error for options at the money is still under 1%. The greatest errors are for the out of the money options with shorter maturities. This fact can be observed in pictures (a), (c) and (d) in Figure 6.1.



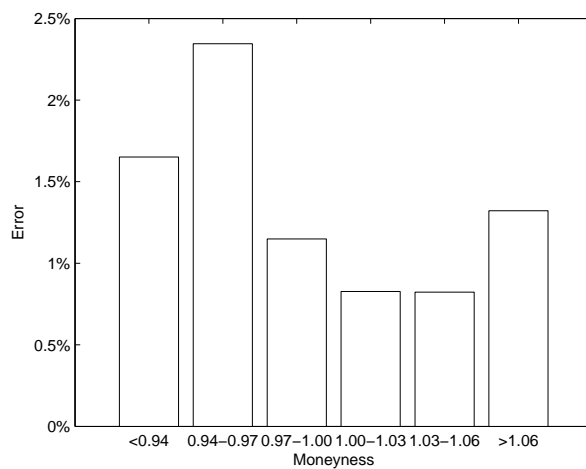
(a)



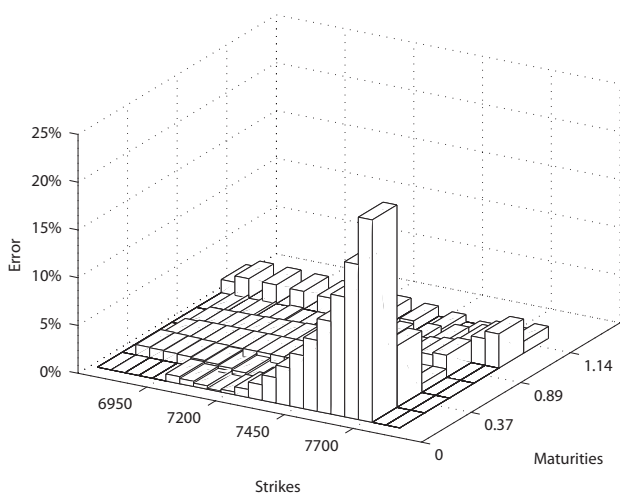
(b)



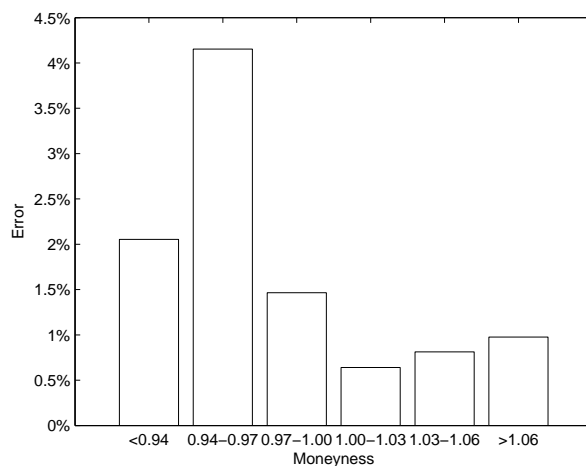
(c)



(d)



(e)



(f)

Figure 6.1: Results for Genetic algorithm. Left column portrays the absolute relative errors, average absolute relative errors in terms of moneyness are shown on the right. Each row corresponds to a particular choice of weights. First row shows errors when using weights A, the rows are then sorted in the alphabetical order.

6.3 modSQP

The most popular Newton method based algorithm for non-linear constrained optimization problems is the sequential quadratic programming (SQP) method as suggested by Kienitz and Wetterau in [22]. The algorithm was tested on a simple calibration example in the book [22]. We implemented it to test it on the data closer to the real market ones.

```
x0 = [0.2 0.5 0.5 1 -0.5]'; % initial guess
maxeval = 20*length(x0);
tol = 1e-12;
[xMin,fMin,perfSQP] = modSQP(@error_f,x0,[],l_bounds,u_bounds,[],maxeval,tol)
```

Below we can see the parameters we were able to obtain starting from initial point $x_0=[0.2 \ 0.5 \ 0.5 \ 1 \ -0.5]$. However only the result for weights A are included, since the algorithm did not converge to solution for weights B and C.

Table 6.3: Set of retrieved parameters using weights A.

| | |
|-------|-----------|
| v0 | 0.035187 |
| kappa | 0.061271 |
| theta | 0.726061 |
| sigma | 0.328934 |
| rho | -0.996691 |

The results given in Figure 6.2 are comparable to the ones obtained by the Genetic algorithm, although modSQP is a local optimizer. We can also observe greater absolute relative errors for out of the money options with short maturities.

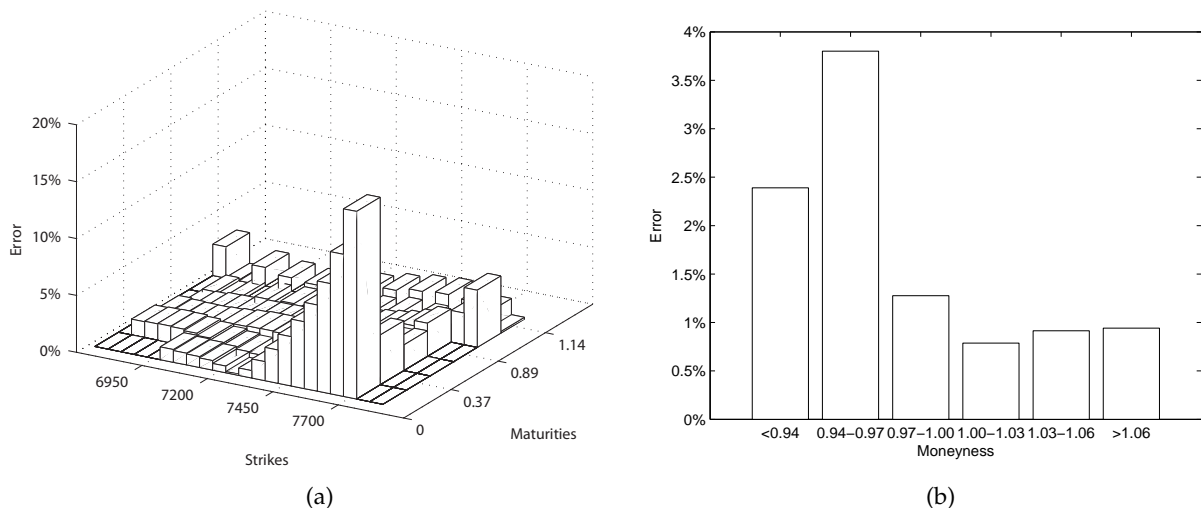


Figure 6.2: Results for SQP. (a) the absolute relative errors, (b) the average absolute relative errors in terms of moneyness.

6.4 Excel's Solver

We used the Solver built in Microsoft Excel 2003, which according to the official site is using Generalized Reduced Gradient Algorithm. This can be done by using the Spreadsheet Link™ EX. It allows us to execute any MATLAB function directly from MS Excel and thus optimize our cost function with the parameters from MS Excel sheet using the Solver. We did try to use newer versions of MS Excel and their built in solvers, but they did not give a better performance.

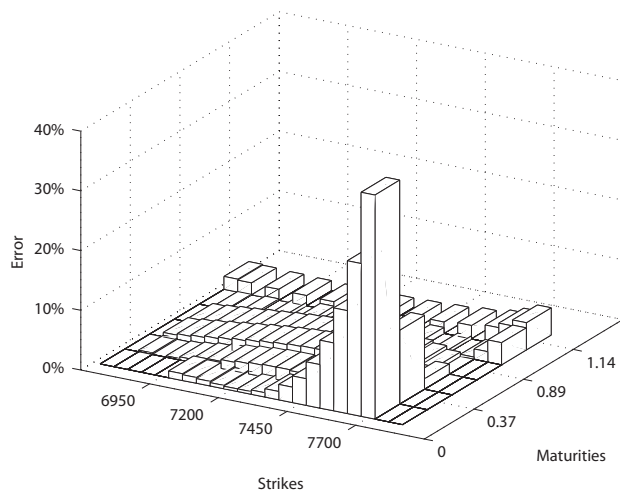
The following Table 6.4 shows the retrieved set of parameters. Again they are not close to the set which generated our prices, but we were quite impressed by the performance of the Solver. We strived for same conditions as this is another local optimizer so we started with initial guess same as for `modSQP`, hence

$$x_0 = [0.2 \ 0.5 \ 0.5 \ 1 \ -0.5].$$

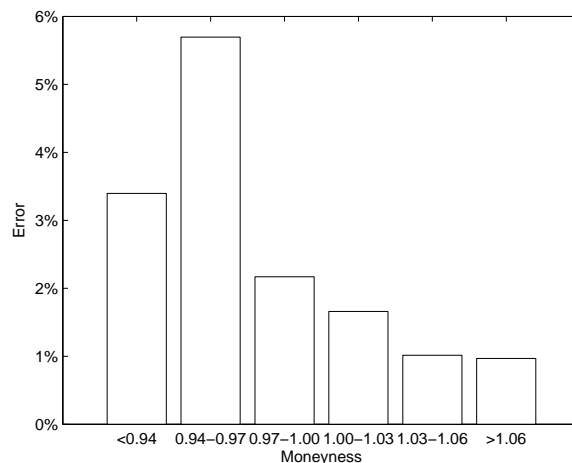
Table 6.4: Set of retrieved parameters using different weights.

| | Used Weights | | |
|-------|--------------|----------|----------|
| | A | B | C |
| v0 | 0.04416 | 0.03530 | 0.03654 |
| kappa | 0.37454 | 0.61008 | 0.88055 |
| theta | 0.25995 | 0.13617 | 0.10582 |
| sigma | 1.06295 | 0.50483 | 0.54082 |
| rho | -0.48373 | -0.84125 | -0.76837 |

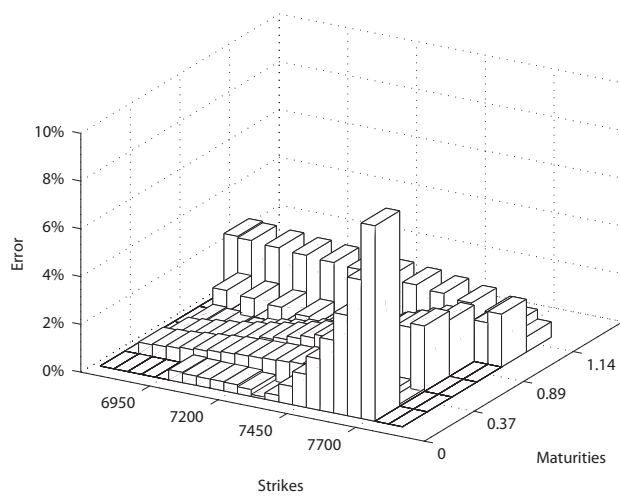
The result using weights A was not great, see first row of Figure 6.3. Also notice significantly higher value of v0 in Table 6.4. However, using weights B and C we were able to obtain comparable results to Genetic algorithm, see the pictures (c) and (e) in Figure 6.3 for the absolute relative errors using different weights presented in the graphical form. The absolute relative errors in terms of moneyness are portrayed in pictures (d) and (f).



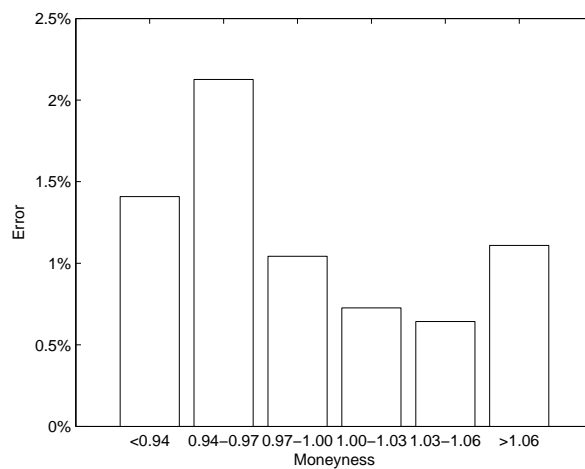
(a)



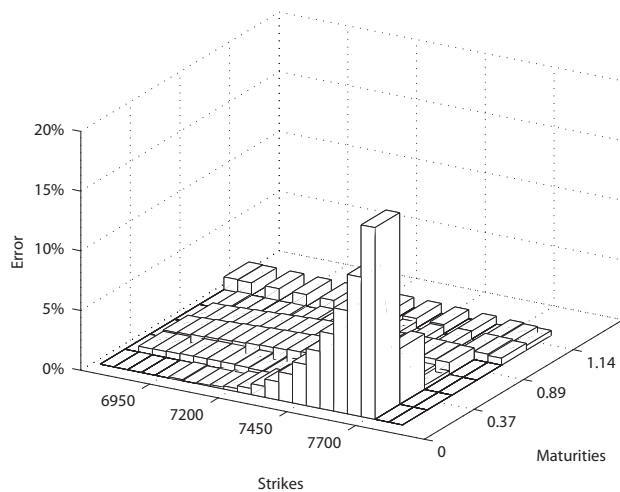
(b)



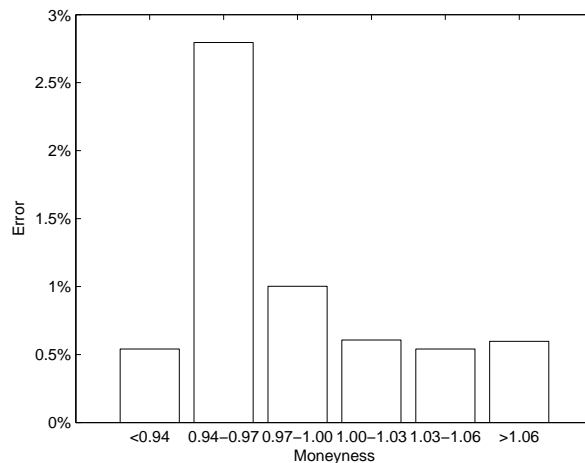
(c)



(d)



(e)



(f)

Figure 6.3: Results for Excel's Solver. Left column portrays the absolute relative errors, average absolute relative errors in terms of moneyness are shown on the right. Each row corresponds to a particular choice of weights. First row shows errors when using weights A, the rows are then sorted in the alphabetical order.

6.5 Adaptive Simulated Annealing

We used ASA 28.14 package³. The ASA code was first developed in 1987 as Very Fast Simulated Reannealing (VFSR) to deal with the necessity of performing adaptive global optimization on multivariate nonlinear stochastic systems.

Beginning in January 1993, many adaptive features were developed, largely in response to users' requests, leading to this ASA code. After installing we can use ASA program in order to optimize a cost function coded in MATLAB language.

```
x0 = [0.2 0.5 0.5 1 -0.5]
xt=-1 * ones(5,1);
[fval,calib]=asamin('minimize','error_f',x0,l_bounds,u_bounds,xt)
```

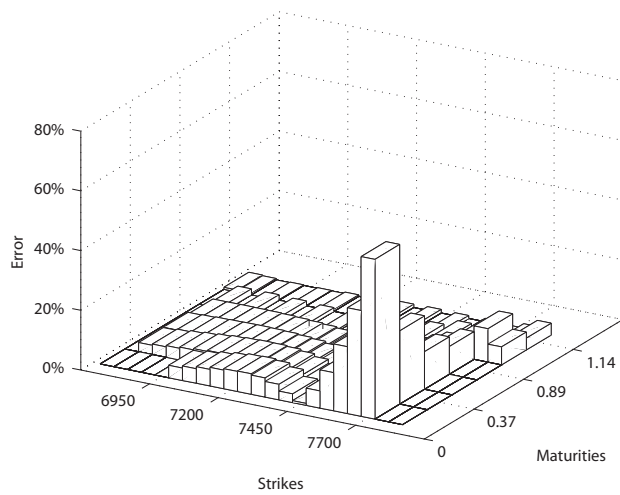
The vector `xt` indicates the desired form of the output argument. If `xt` is equal to -1, it means the result is a real. The following Table 6.5 shows the retrieved parameters using ASA algorithm.

Table 6.5: Set of retrieved parameters using different weights.

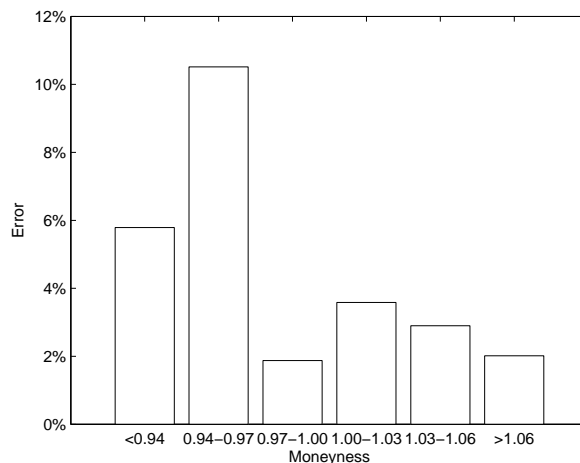
| | Used Weights | | |
|-------|--------------|----------|----------|
| | A | B | C |
| v0 | 0.04218 | 0.04510 | 0.053049 |
| kappa | 3.45525 | 0.85780 | 1.00718 |
| theta | 0.08428 | 0.15793 | 0.16211 |
| sigma | 2.16748 | 1.15227 | 1.67937 |
| rho | -0.20591 | -0.65788 | -0.52136 |

Surprisingly, the results for parameters obtained using ASA algorithm were quite poor. We immediately recognize in Table 6.5 rather high parameters sigma accompanied by the lowest values of rho so far. The absolute relative errors for individual strikes and maturities as well as the absolute relative errors in terms of moneyness are depicted in Figure 6.4 on the next page.

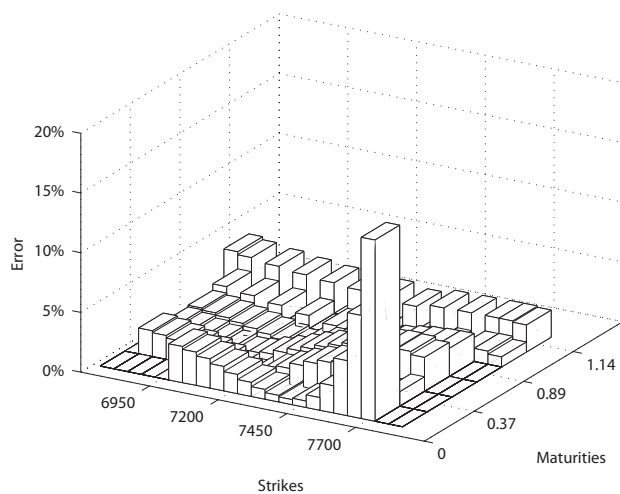
³Package is available at <http://ingber.com>.



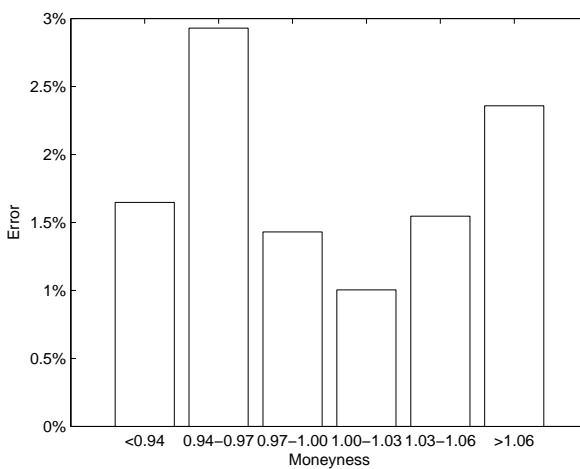
(a)



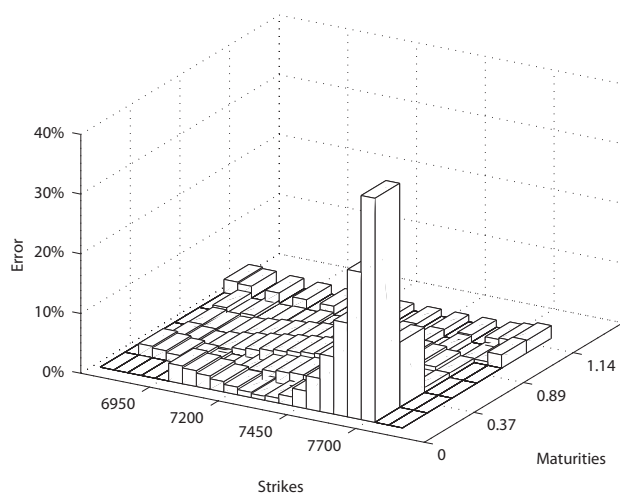
(b)



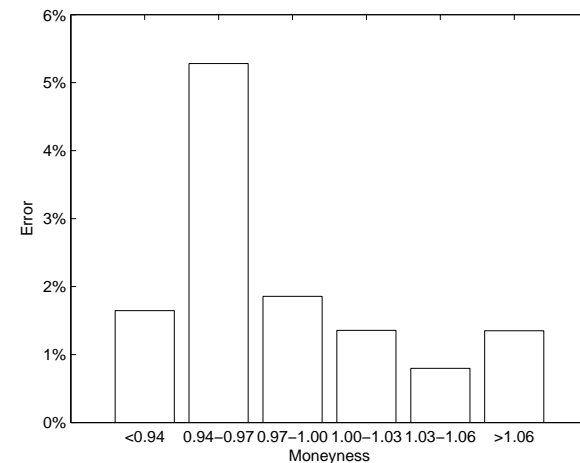
(c)



(d)



(e)



(f)

Figure 6.4: Results for ASA. Left column portrays the absolute relative errors, average absolute relative errors in terms of moneyness are shown on the right. Each row corresponds to a particular choice of weights. First row shows errors when using weights A, the rows are then sorted in the alphabetical order.

6.6 MATLAB's `lsqnonlin`

MATLAB's function `lsqnonlin` is an intuitive choice for our problem since it solves nonlinear least-squares problems, including nonlinear data-fitting problems. Nevertheless, as we are about to see it is extremely sensitive to initial guess.

To overcome this we used a deterministic grid to divide the state space of parameters in order to generate a set of initial points for this function. As a result, only around 6.5% of initial points were close enough to converge to set of parameters, which produced mean absolute relative error of less than 1%. This shows how important is the choice of the initial point when using `lsqnonlin`. Nonetheless, we were able to retrieve the original set of parameter using this deterministic grid method. The parameters are presented in the following Table 6.6.

Table 6.6: Closest set of retrieved parameters

| | |
|-------|----------|
| v0 | 0.03400 |
| kappa | 2.21004 |
| theta | 0.07300 |
| sigma | 0.56800 |
| rho | -0.95199 |

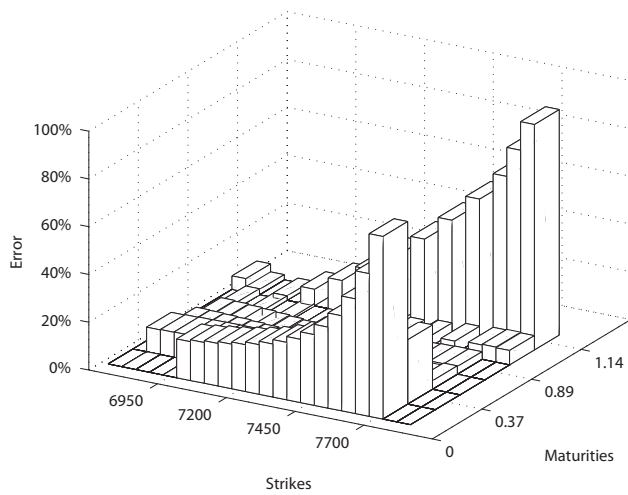
To illustrate the extreme sensitivity some more, we will also include the results when using the initial guess $x_0 = [0.2 \ 0.5 \ 0.5 \ 1 \ -0.5]$. We obtained the parameters presented in Table 6.7.

```
x0=[0.2 0.5 0.5 1 -0.5];
options=optimset('MaxFunEvals',20000,'TolFun',1e-12,'TolX',1e-12);
[calib]=lsqnonlin(@error_f,x0,l_bounds,u_bounds,options);
```

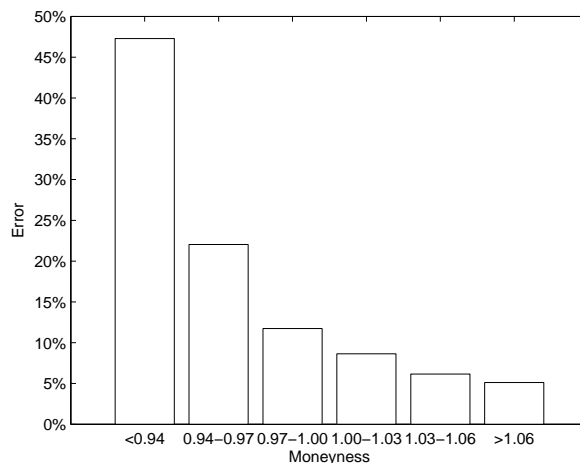
Table 6.7: Set of retrieved parameters using different weights.

| | Used Weights | | |
|-------|--------------|----------|----------|
| | A | B | C |
| v0 | 0.08957 | 0.10251 | 0.09199 |
| kappa | 0.35932 | 0.35858 | 0.35601 |
| theta | 0.35569 | 0.35493 | 0.35197 |
| sigma | 1.96603 | 2.18412 | 1.93802 |
| rho | -0.64682 | -0.69506 | -0.63865 |

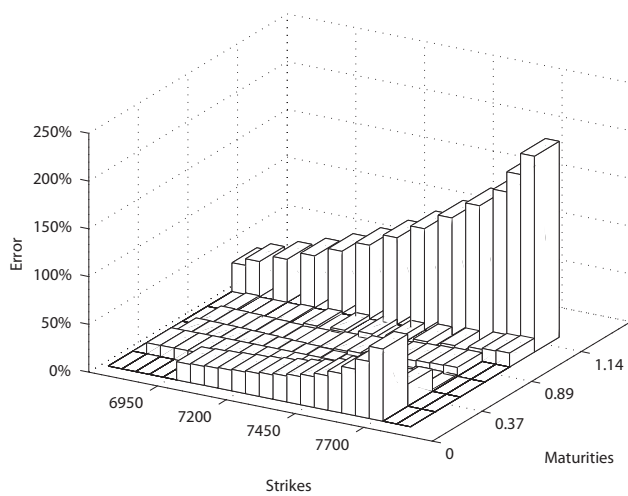
The next page portrays the absolute relative errors and mean absolute relative errors in terms of moneyness. Figure 6.5 gives the big picture of what we mentioned earlier, thus high sensitivity to the initial guess.



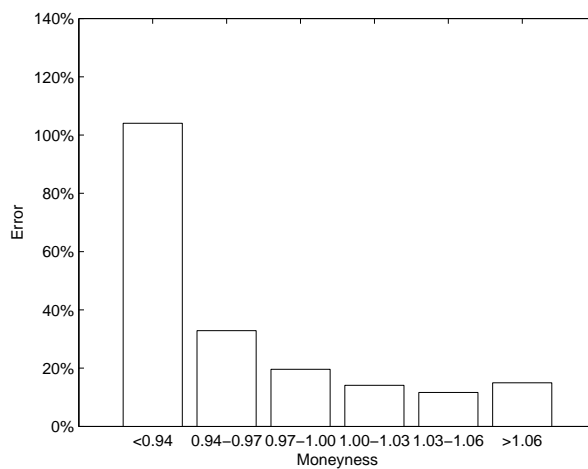
(a)



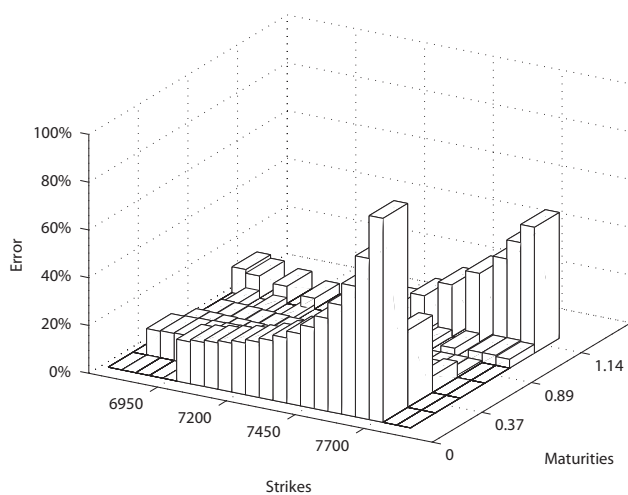
(b)



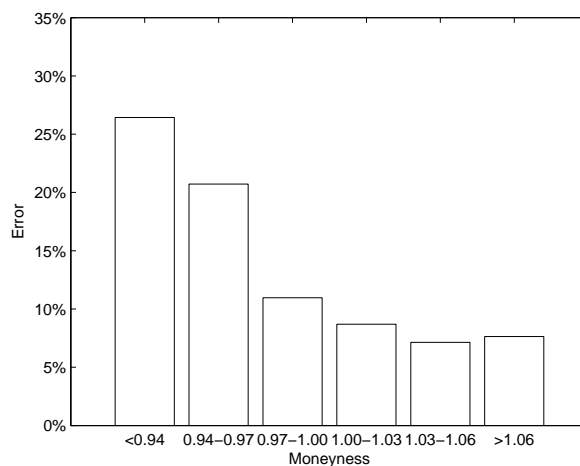
(c)



(d)



(e)



(f)

Figure 6.5: Results for `lsqnonlin`. Left column portrays the absolute relative errors, average absolute relative errors in terms of moneyness are shown on the right. Each row corresponds to a particular choice of weights. First row shows errors when using weights A, the rows are then sorted in the alphabetical order.

6.7 SA

We also considered MATLAB's Simulated annealing (SA) method for solving unconstrained and bound-constrained optimization problems, which is part of Global Optimization Toolbox. However, we decided not to include the results, since they were also very sensitive to the initial guess and in comparison with `lsqnonlin` the SA method was too time consuming.

6.8 Conclusion

The results are presented in the following Table 6.8. We generated this set of prices, therefore we not only knew that there was an optimal set of parameters but we also knew the actual values of the parameters. We used this artificial data to test the selected optimizers. We compared two global optimizers - MATLAB's Genetic algorithm (GA) and Adaptive Simulated Annealing (ASA).

Looking at the results, it's safe to say that GA provided better results on this set of data, although chosen weights played an important role. Excel's Solver performed very well bearing in mind it is a local optimizer and it is very sensitive to the initial guess. `modSQP` was rather unstable and was able to converge to a possible local minimum using only weights A. Although it worked on a school example in Kienitz and Wetterau [22], when applied to our set of data it failed to deliver the results for weights B and C.

MATLAB's `lsqnonlin` also suffers from sensitivity to initial point, but generating number of initial points within the state space of possible parameters we were able to get close enough and retrieve the original parameters. Briefly mentioned SA did not tackle the problem in a way which could be of any use to us.

Table 6.8: Calibration results

| Algorithm | Weights | AARE | RMSE | SSE | MARE | Crit A ^a | Crit B | Crit C | v0 | kappa | theta | sigma | rho |
|-----------|---------|--------|--------|------------|---------|---------------------|----------|------------|---------|---------|---------|---------|----------|
| Genetic | A | 0.92% | 5.24 | 2659.98 | 9.46% | 201.51 | 27.49 | 693.03 | 0.03399 | 0.79087 | 0.11581 | 0.47311 | -0.84858 |
| Genetic | B | 1.36% | 10.02 | 9740.60 | 10.08% | 629.04 | 59.26 | 2402.48 | 0.03537 | 0.39258 | 0.18695 | 0.49517 | -0.82075 |
| Genetic | C | 1.86% | 7.34 | 5222.08 | 21.18% | 638.83 | 142.12 | 1678.59 | 0.03610 | 0.17567 | 0.28482 | 0.36132 | -0.91083 |
| modSQP | A | 1.79% | 7.98 | 6174.93 | 16.49% | 584.04 | 99.79 | 1778.04 | 0.03519 | 0.06127 | 0.72606 | 0.32893 | -0.99669 |
| Excel | A | 2.76% | 11.06 | 11875.07 | 37.52% | 1490.03 | 318.77 | 3910.63 | 0.04416 | 0.37454 | 0.25995 | 1.06295 | -0.48373 |
| Excel | B | 1.20% | 7.78 | 5873.67 | 8.18% | 413.30 | 42.64 | 1505.39 | 0.03530 | 0.61008 | 0.13617 | 0.50483 | -0.84125 |
| Excel | C | 1.26% | 4.82 | 2256.26 | 16.05% | 301.14 | 68.59 | 763.41 | 0.03654 | 0.88055 | 0.10582 | 0.54082 | -0.76837 |
| ASA | A | 4.96% | 18.75 | 34107.55 | 53.49% | 4326.86 | 836.78 | 11490.41 | 0.04218 | 3.45525 | 0.08428 | 2.16748 | -0.20591 |
| ASA | B | 1.81% | 11.29 | 12357.17 | 15.17% | 856.58 | 89.24 | 3137.03 | 0.04510 | 0.85780 | 0.15793 | 1.15227 | -0.65788 |
| ASA | C | 2.43% | 9.24 | 8286.40 | 37.53% | 1131.77 | 262.63 | 2834.11 | 0.05305 | 1.00718 | 0.16211 | 1.67937 | -0.52136 |
| lsqnonlin | A | 13.82% | 113.44 | 1248309.93 | 94.40% | 74434.03 | 6526.21 | 296348.09 | 0.08957 | 0.35932 | 0.35569 | 1.96603 | -0.64682 |
| lsqnonlin | B | 23.60% | 288.49 | 8072819.05 | 204.92% | 439401.05 | 26774.39 | 1867922.02 | 0.10251 | 0.35858 | 0.35493 | 2.18412 | -0.69506 |
| lsqnonlin | C | 12.78% | 68.52 | 455380.19 | 85.40% | 36437.46 | 5315.30 | 120687.25 | 0.09199 | 0.35601 | 0.35197 | 1.93802 | -0.63865 |

^aThe value of (6.1) using weights A. Analogously applied for Crit B and Crit C.

Chapter 7

Calibration to real market

In this chapter we will calibrate Heston model to data obtained from real market. We used Bloomberg's Option Monitor to obtain data for two consecutive days for ODAX call options. The following Table 7.1 sums up the data we will use to calibrate the Heston model.

Table 7.1: Real market data

| Day | Maturities | Strikes | S_0 | r |
|----------------|------------|---------|---------|--------|
| March 18, 2013 | 5 | 97 | 7953.70 | 0.206% |
| March 19, 2013 | 6 | 107 | 7962.31 | 0.207% |

As a risk free interest rate r we used the corresponding 3 month EURIBOR rate¹, these are widely available. S_0 is the current DAX value at the time of obtaining the data.

7.1 March 18, 2013

In this section we present the results of calibration of the Heston model to the first set of data from March 18, 2013. We were using GA and ASA as global optimizers and MS Excel's Solver and MATLAB's `lsqnonlin` as local optimizers.

7.1.1 GA

The following Table 7.2 shows the result of the calibration using Genetic algorithm (GA) with different weights. We were able to obtain a negative correlation ρ between the underlying asset and volatility. Using weights A and B resulted in similar set of

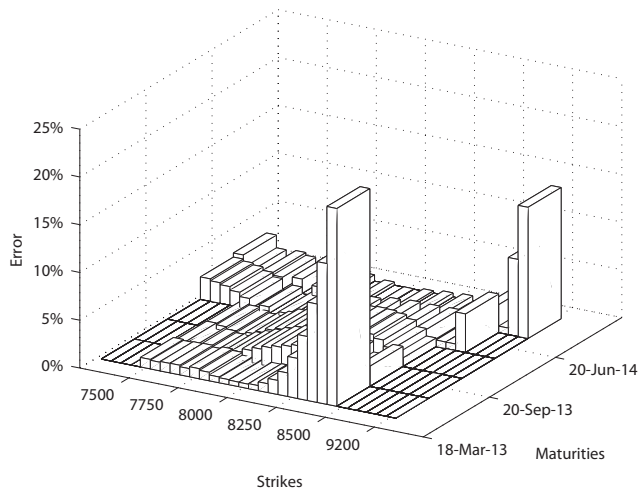
¹The Euribor rates are based on the average interest rates at which European banks borrow funds from one another.

parameters which naturally implies similar absolute relative errors, see pictures (a) - (d) in Figure 7.1.

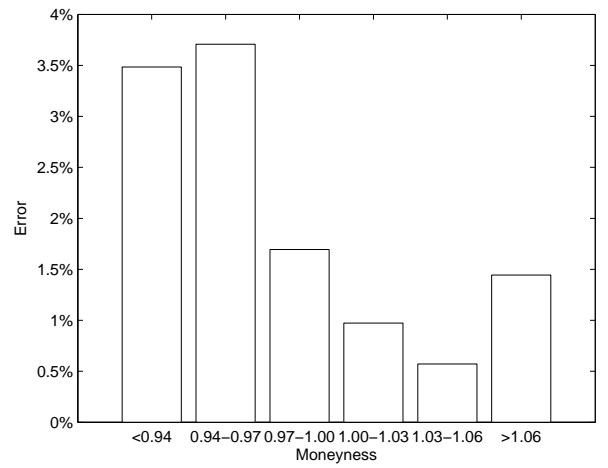
Table 7.2: Set of retrieved parameters using different weights.

| | Used Weights | | |
|-------|--------------|----------|----------|
| | A | B | C |
| v0 | 0.03226 | 0.03193 | 0.03035 |
| kappa | 0.07065 | 0.07747 | 0.55662 |
| theta | 0.73827 | 0.73826 | 0.11191 |
| sigma | 0.81988 | 0.85729 | 0.71420 |
| rho | -0.52083 | -0.55003 | -0.55050 |

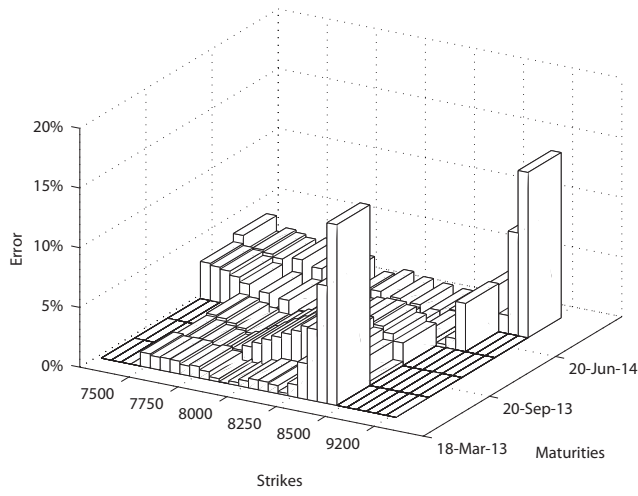
The best result in terms of absolute relative errors was obtained by using weights C. We can still observe higher absolute relative errors in Figure 7.1, picture (e) for out of the money options, especially those maturing on June 21, 2013. Represented by the first row, not to be confused with June 20, 2014. But higher parameter kappa and lower long mean average resulted in lowest average absolute relative error, this can be partly observed in terms of moneyness in picture (f) and fully in Table 7.9 which provides the comprehensive results at the end of this section.



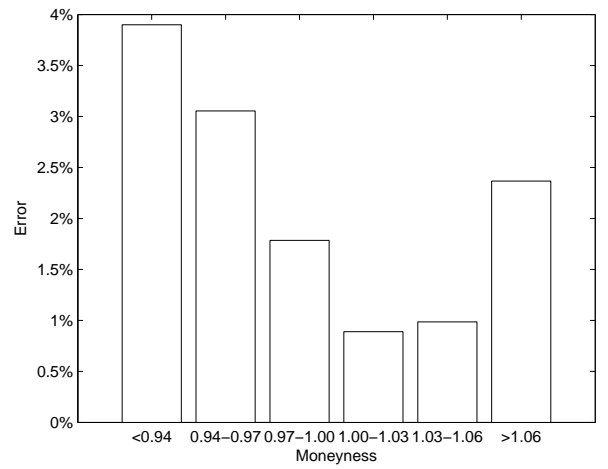
(a)



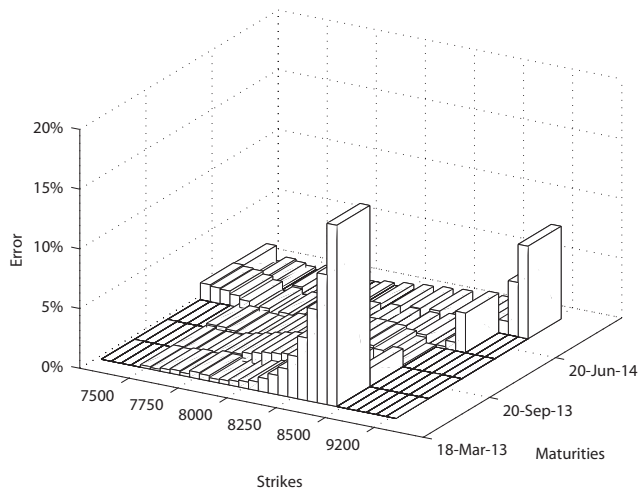
(b)



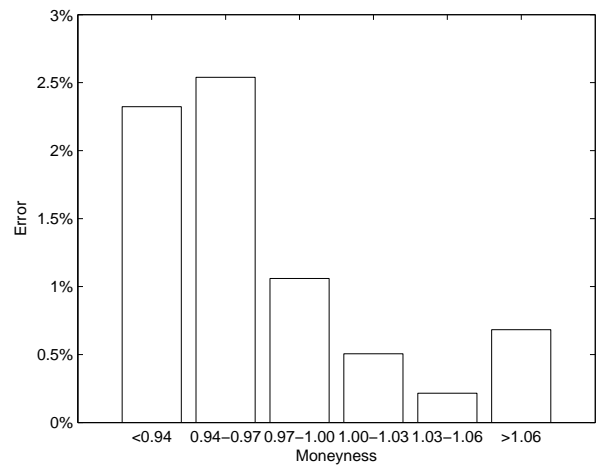
(c)



(d)



(e)



(f)

Figure 7.1: Results for Genetic algorithm. Left column portrays the absolute relative errors, average absolute relative errors in terms of moneyness are shown on the right. Each row corresponds to a particular choice of weights. First row shows errors when using weights A, the rows are then sorted in the alphabetical order.

7.1.2 ASA

Table 7.3 shows the result of the calibration using Adaptive Simulated Annealing (ASA) with different weights.

We can observe three quite different sets of parameters. All three contain negative correlation parameter ρ for the underlying asset and volatility, but other than that the parameters quite vary.

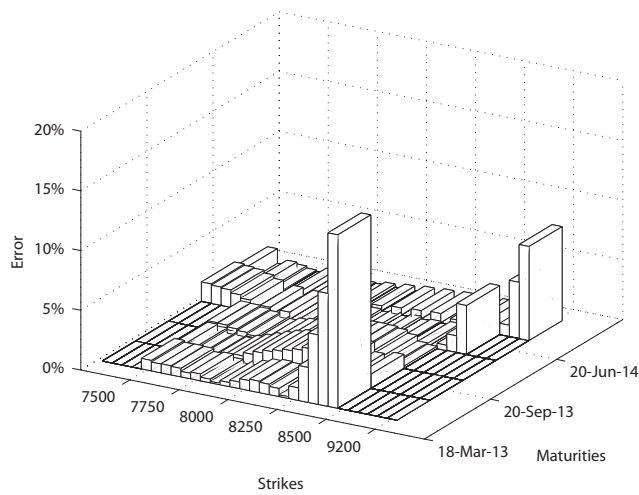
This produces three different portraits of absolute relative errors. Using weights A we were able to calibrate the model with average absolute relative error similar to the one using GA with weights C, see Table 7.9. This error is also accompanied by higher absolute relative errors for out of the money options with short maturity, this can be seen in figure 7.2, picture (a).

Using weights B produced very low absolute relative errors. Only deep out of money options suffered higher than 0.8% average absolute relative error. This is the best result obtained so far and absolute relative errors can be seen in Figure 7.2 pictures (c) and (d).

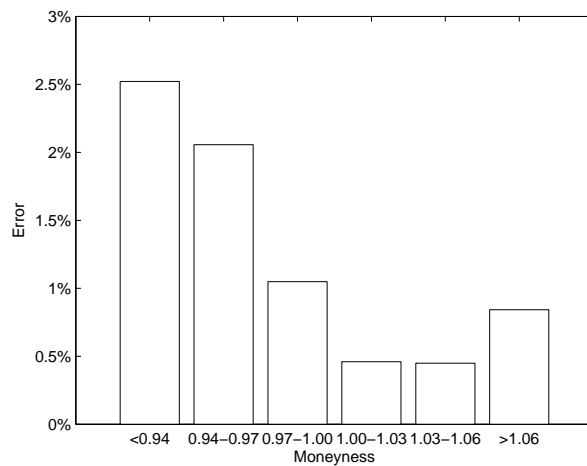
When using weights C we obtained rather high initial volatility v_0 together with high volatility σ . As can be seen in pictures (e) and (f) in Figure 7.2 this is not the set of parameters that explains the market values of options very well. The maximum absolute relative error reaches 33.54%.

Table 7.3: Set of retrieved parameters using different weights.

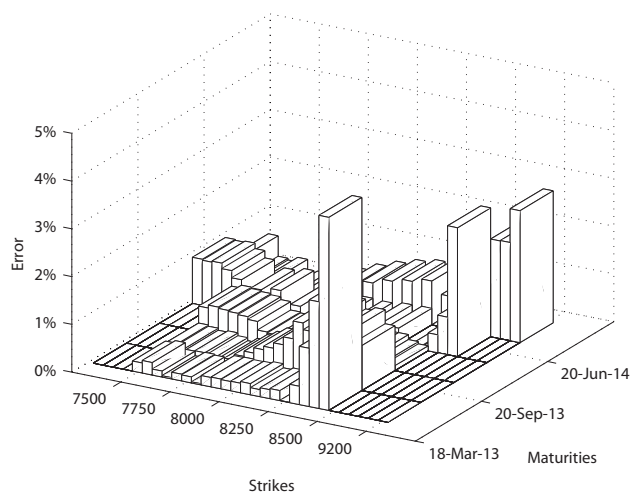
| | Used Weights | | |
|----------|--------------|----------|----------|
| | A | B | C |
| v_0 | 0.03219 | 0.02845 | 0.04111 |
| κ | 1.12162 | 1.26339 | 0.80249 |
| θ | 0.08278 | 0.06718 | 0.13210 |
| σ | 0.96401 | 0.67255 | 1.55269 |
| ρ | -0.54227 | -0.62816 | -0.47895 |



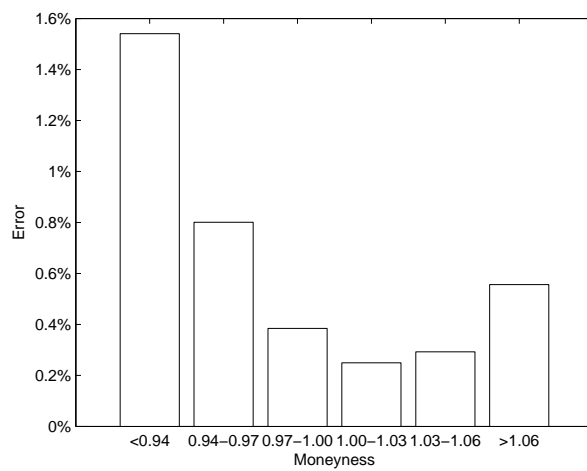
(a)



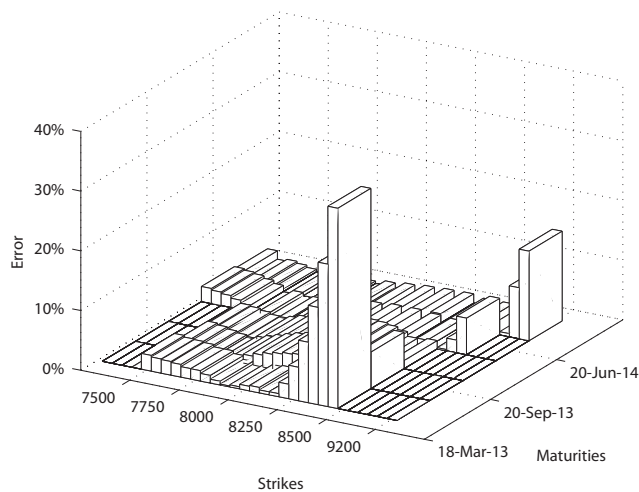
(b)



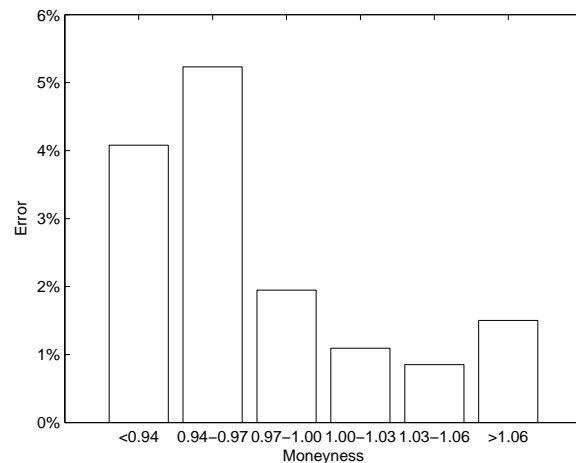
(c)



(d)



(e)



(f)

Figure 7.2: Results for ASA. Left column portrays the absolute relative errors, average absolute relative errors in terms of moneyness are shown on the right. Each row corresponds to a particular choice of weights. First row shows errors when using weights A, the rows are then sorted in the alphabetical order.

7.1.3 Deterministic grid for `lsqnonlin`

We showed in the previous chapter, that `lsqnonlin` is extremely sensitive to the provided initial value. To overcome this we divided the state space of possible parameters' values again in order to obtain a set of initial values. We used these as initial values and this method gave us the best results in terms of absolute relative errors, also the maximum relative error was in line with the rest of the errors, we do not observe high values for the out of the money options with short maturities in Figure 7.3, picture (a).

We used weights B, which according to further results appears to be the best choice for `lsqnonlin`.

Table 7.4 shows the set of parameters and we can see the lowest value of initial volatility v_0 and volatility of volatility so far. Also the negative correlation is the highest one so far. This results in maximum absolute relative error under 2.5%.

However this method is not quite elegant. We will try another approach in the next section and see if we can match the magnitude of the errors for the parameters obtained in this section.

| | |
|----------|----------|
| v_0 | 0.02741 |
| κ | 1.18184 |
| θ | 0.06586 |
| σ | 0.57479 |
| ρ | -0.66686 |

Table 7.4: Set of retrieved parameters using weights B.

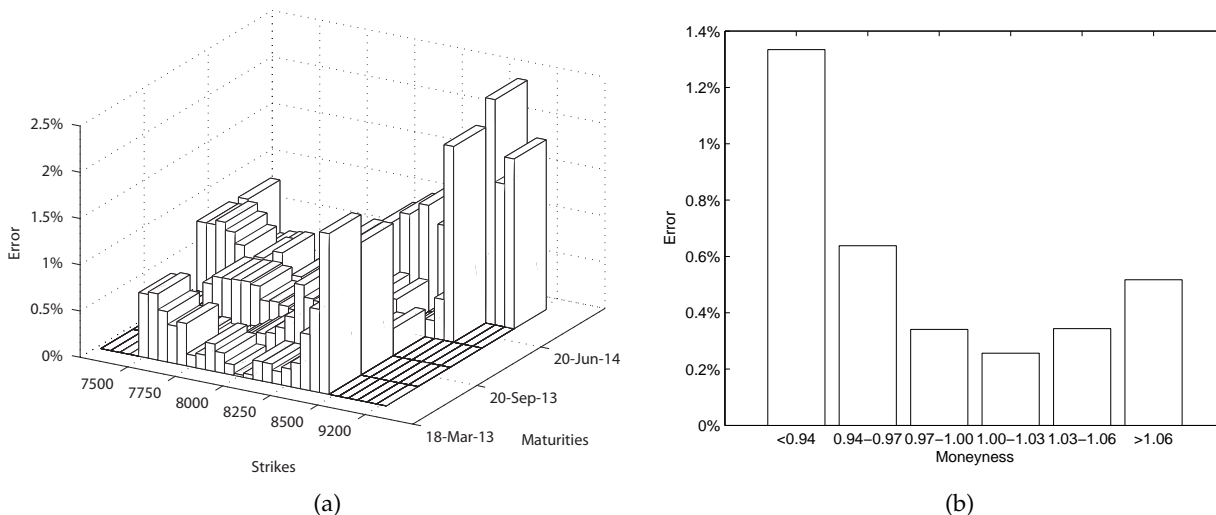


Figure 7.3: Results for `lsqnonlin`. (a) absolute relative errors, (b) the average absolute relative errors in terms of moneyness.

7.1.4 Excel's Solver

We did observe quite high absolute relative errors for out of the money options with short maturities even for the best results when using global optimizers ASA and GA. In order to eliminate these errors, we tried a different approach and the results will be presented in this section.

At first we employed MS Excel's Solver and provided it with the initial guesses from the global optimizers (GA, ASA) and the results were as follows. We denoted the initial point obtained by GA by x^*

$$x^* = [0.03035 \quad 0.55662 \quad 0.11191 \quad 0.71420 \quad -0.55050]$$

and x^{**}

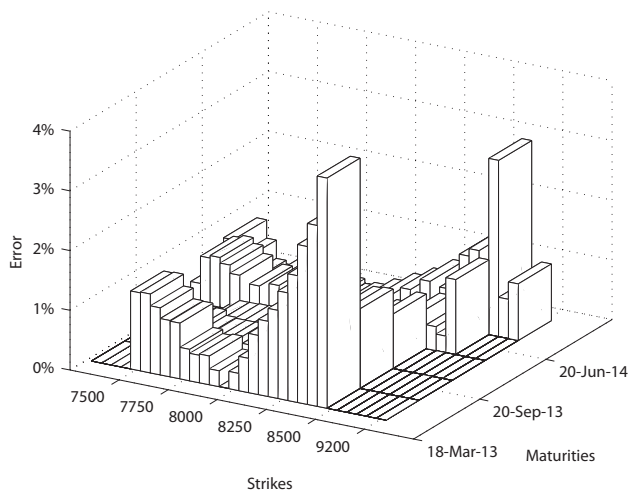
$$x^{**} = [0.02845 \quad 1.26339 \quad 0.06718 \quad 0.67255 \quad -0.62816],$$

which is the initial point obtained by ASA. Table 7.5 shows obtained parameters for the model using initial point x^* . We can see that except for when using weights C for x^* Excel was able to refine the values of parameters.

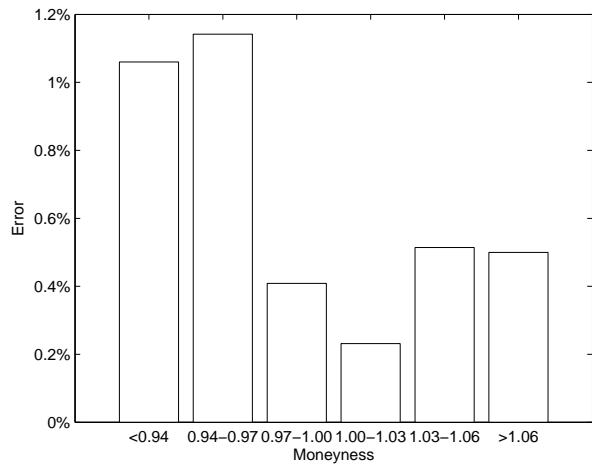
Table 7.5: Set of retrieved parameters using different weights starting from initial point x^* . Initial point x^* is included for comparison.

| | x^* | Used Weights | | |
|-------|----------|--------------|----------|----------|
| | | A | B | C |
| v0 | 0.03035 | 0.02746 | 0.02683 | 0.03035 |
| kappa | 0.55662 | 1.12422 | 0.66747 | 0.55663 |
| theta | 0.11191 | 0.06762 | 0.08426 | 0.11192 |
| sigma | 0.71420 | 0.57479 | 0.46984 | 0.71417 |
| rho | -0.55050 | -0.66342 | -0.67899 | -0.55050 |

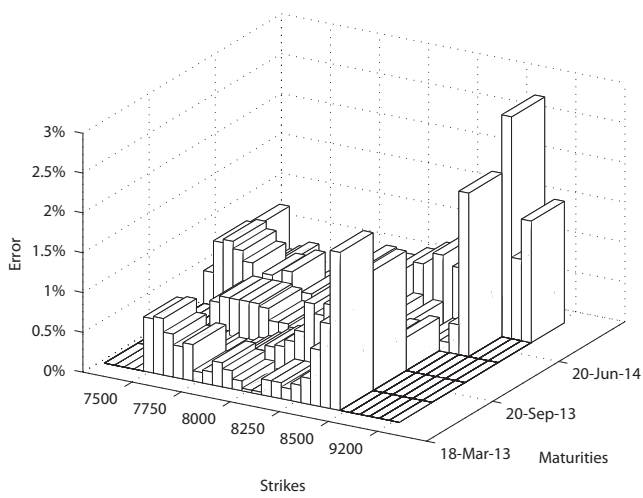
For the choice of weights B we were able to match the best result in terms of average absolute relative error, this can be partly observed in Figure 7.4 (c) and (d), also we were able to obtain maximum absolute relative error under 3%.



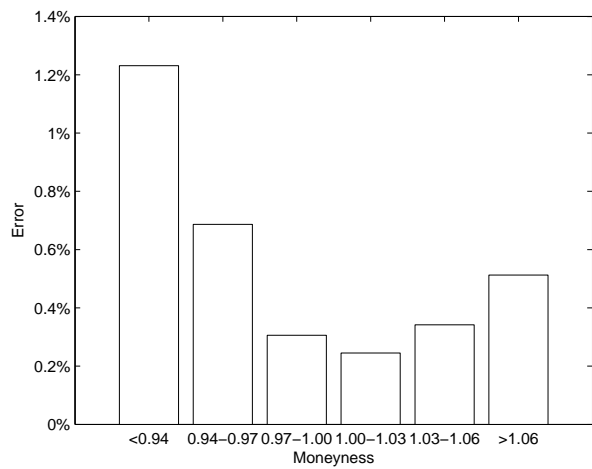
(a)



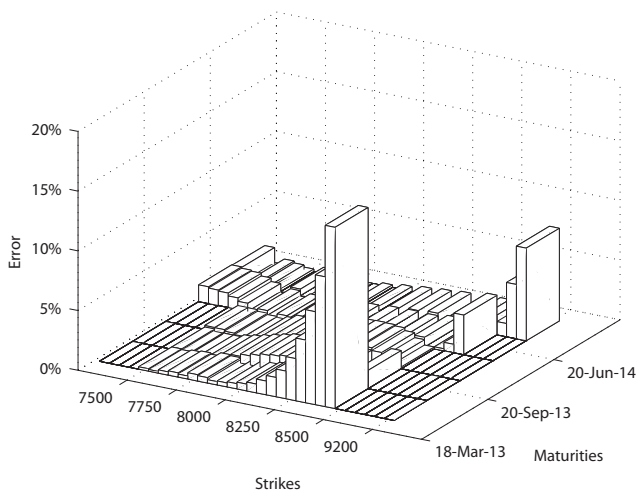
(b)



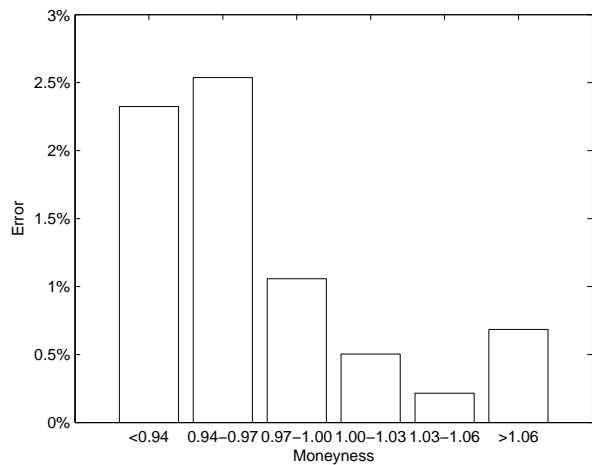
(c)



(d)



(e)



(f)

Figure 7.4: Results for Excel's Solver. Left column portrays the absolute relative errors, average absolute relative errors in terms of moneyness are shown on the right. Each row corresponds to a particular choice of weights. First row shows errors when using weights A, the rows are then sorted in the alphabetical order. The initial point was x^* .

In the following Table 7.6 we can see the parameters obtained when starting MS Excel's Solver from x^{**} . Although the average absolute error was already at the low value of 0.58%, we were able to refine the parameters using MS Excel's Solver.

The lowest average absolute error was obtained using weights A, this can be seen partly in Figure 7.5. Also the maximum absolute relative error is lowered to 3.53%. Similar results were obtained using weights C, see figure 7.6 (c) and (d). Using weights B, Excel's Solver did not refine the parameters significantly.

Table 7.6: Set of retrieved parameters using different weights starting from initial point x^{**} . Initial point x^{**} is included for comparison.

| | x^{**} | Used Weights | | |
|-------|----------|--------------|----------|----------|
| | | A | B | C |
| v0 | 0.02845 | 0.02745 | 0.02843 | 0.02729 |
| kappa | 1.26339 | 1.09385 | 1.26363 | 1.06117 |
| theta | 0.06718 | 0.06818 | 0.06716 | 0.06852 |
| sigma | 0.67255 | 0.57187 | 0.67246 | 0.55391 |
| rho | -0.62816 | -0.64966 | -0.62834 | -0.65495 |

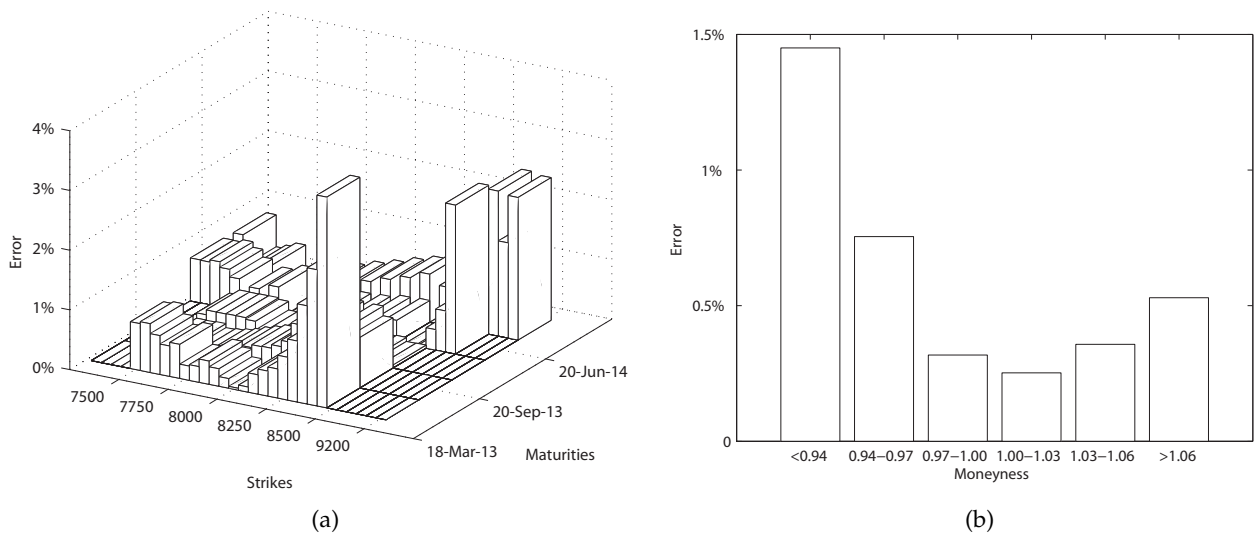
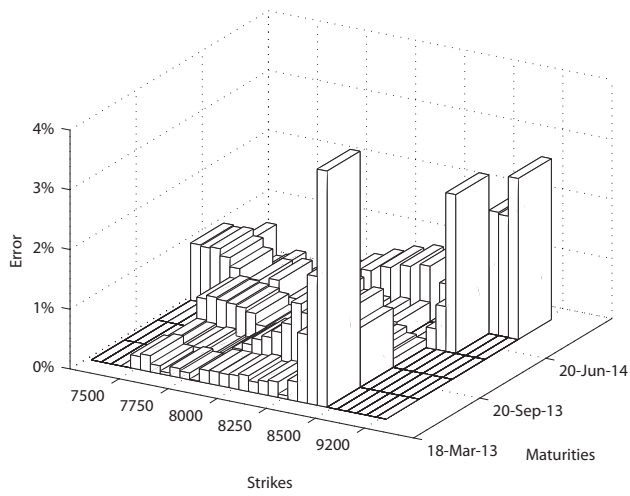
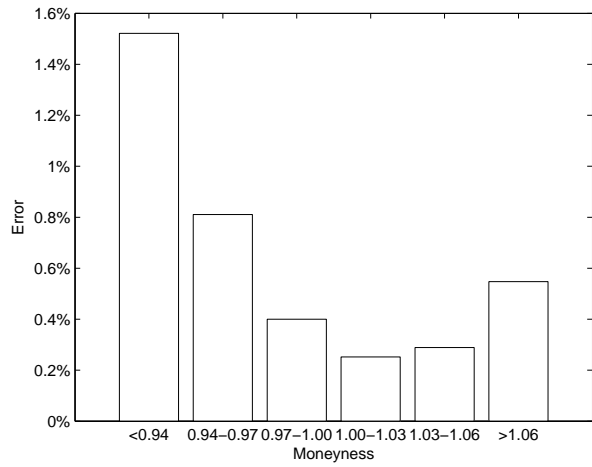


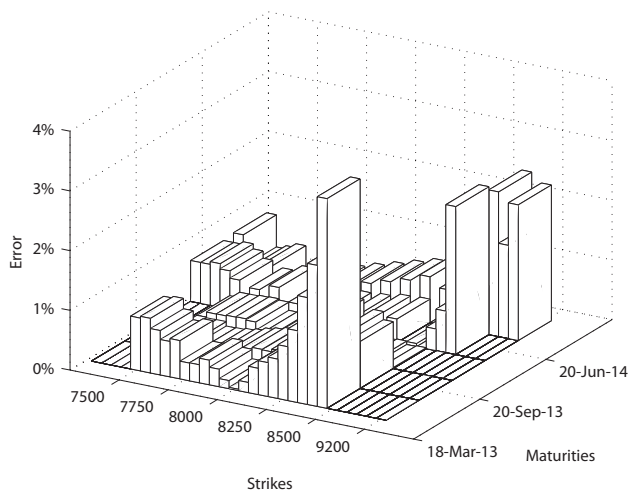
Figure 7.5: Results for Excel's Solver. The absolute relative errors in (a), average absolute relative errors in terms of moneyness are shown on the right in (b). Errors were obtained using weights A and the initial point was x^{**} .



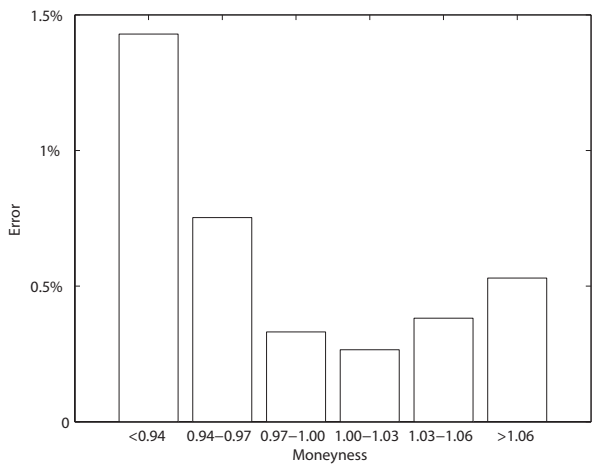
(a)



(b)



(c)



(d)

Figure 7.6: Results for Excel's Solver. Left column portrays the absolute relative errors, average absolute relative errors in terms of moneyness are shown on the right. Each row corresponds to a particular choice of weights. First row shows errors when using weights B, the second row is for weights C. The initial point was x^{**} .

7.1.5 lsqnonlin

We applied the same approach for `lsqnonlin` as previously for MS Excel's Solver, thus starting from initial points x^* and x^{**} . `lsqnonlin` was able to refine the parameters using all the weights and starting from both x^* and x^{**} .

Using weights B and starting from the initial point provided by GA we were able to calibrate the model with average absolute relative error of 0.52%, also we obtained the lowest maximum absolute error of 2.33%, see Figure 7.7 (c) and (d).

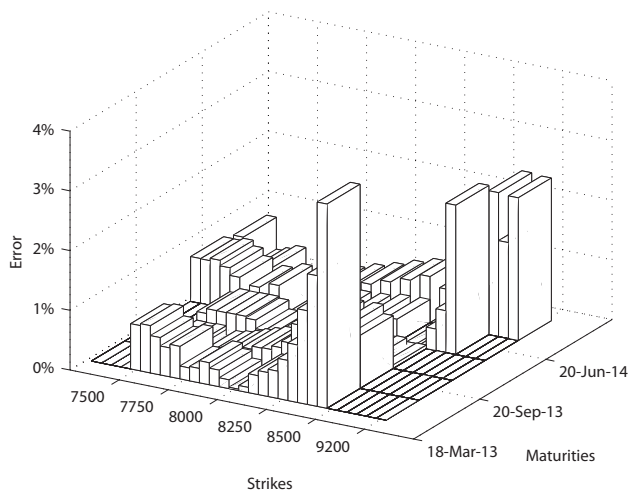
Table 7.7: Set of retrieved parameters using different weights starting from initial point x^* . Initial point x^* is included for comparison.

| | x^* | Used Weights | | |
|-------|----------|--------------|----------|----------|
| | | A | B | C |
| v0 | 0.03035 | 0.02747 | 0.02760 | 0.02732 |
| kappa | 0.55662 | 1.09567 | 1.20011 | 0.97657 |
| theta | 0.11191 | 0.06829 | 0.06601 | 0.07120 |
| sigma | 0.71420 | 0.57399 | 0.59282 | 0.54564 |
| rho | -0.55050 | -0.65043 | -0.65886 | -0.65127 |

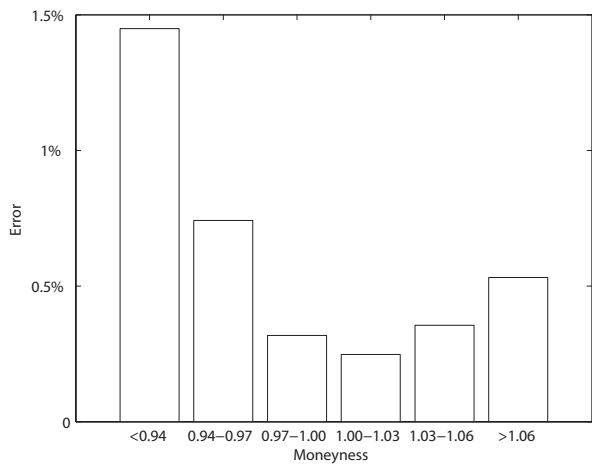
The best result when starting from x^{**} was also obtained when using weights B and that was in terms of both maximum absolute relative error and average absolute relative error. This can be seen in Figure 7.8 (c) and (d). Nevertheless it is not better than what we were able to obtain when using initial guess from GA.

Table 7.8: Set of retrieved parameters using different weights starting from initial point x^{**} . Initial point x^{**} is included for comparison.

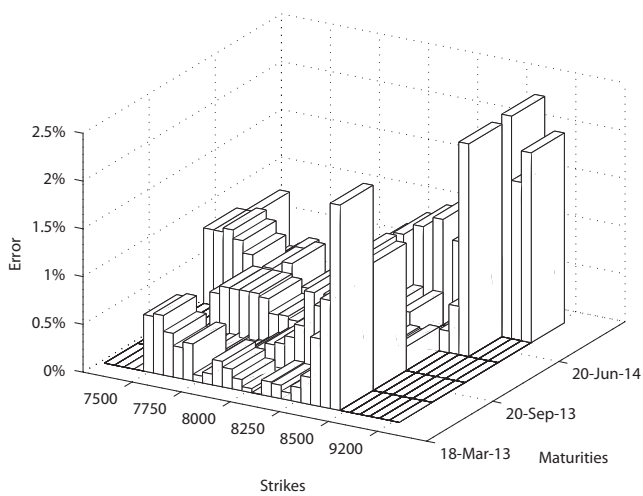
| | x^{**} | Used Weights | | |
|-------|----------|--------------|----------|----------|
| | | A | B | C |
| v0 | 0.02845 | 0.02750 | 0.02786 | 0.02730 |
| kappa | 1.26339 | 1.11668 | 1.24433 | 0.97637 |
| theta | 0.06718 | 0.06781 | 0.06596 | 0.07113 |
| sigma | 0.67255 | 0.57870 | 0.62264 | 0.54339 |
| rho | -0.62816 | -0.64958 | -0.64732 | -0.65279 |



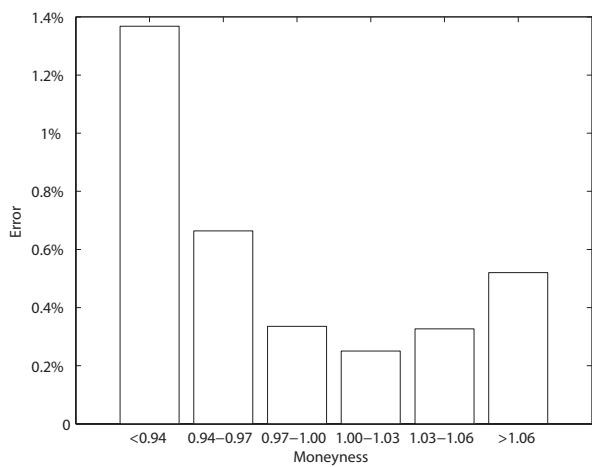
(a)



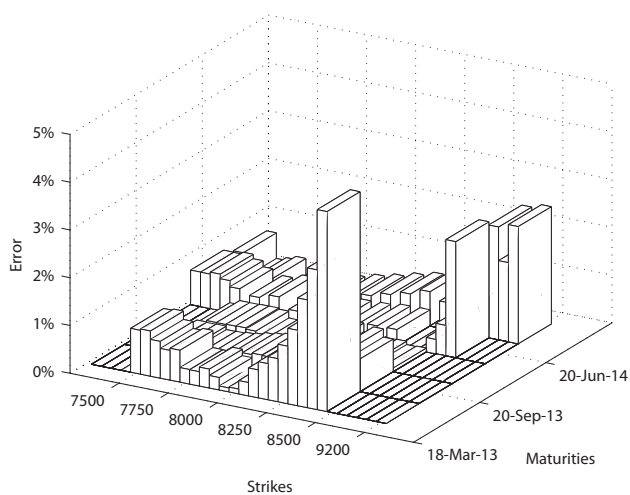
(b)



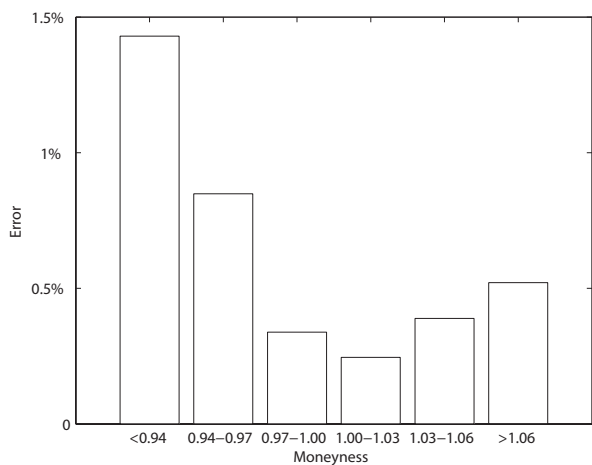
(c)



(d)

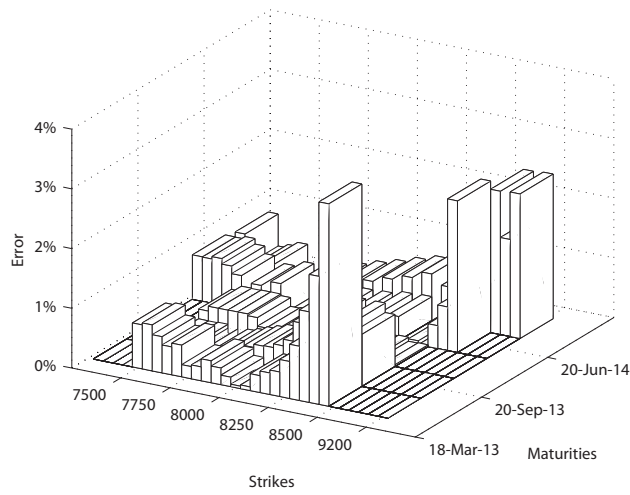


(e)

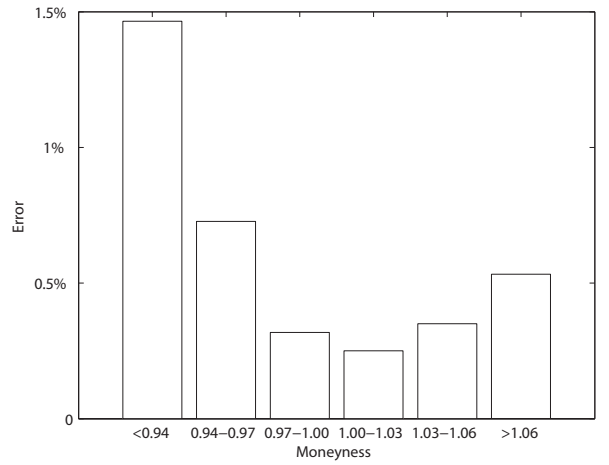


(f)

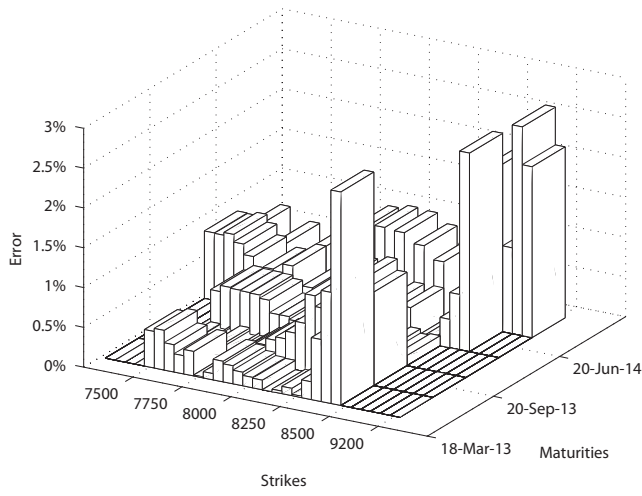
Figure 7.7: Results for `lsqnonlin`. Left column portrays the absolute relative errors, average absolute relative errors in terms of moneyness are shown on the right. Each row corresponds to a particular choice of weights. First row shows errors when using weights A, the rows are then sorted in the alphabetical order. The initial point was x^* .



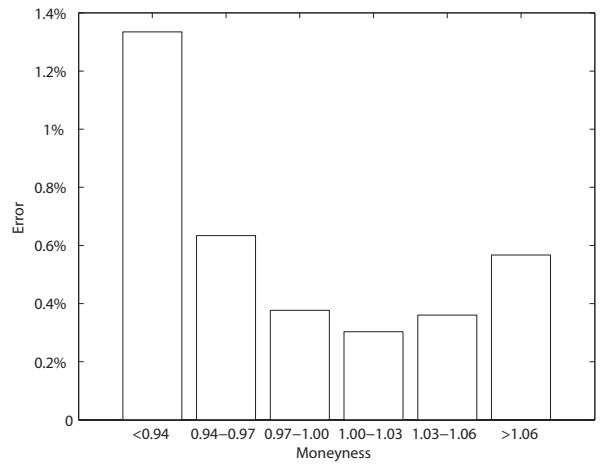
(a)



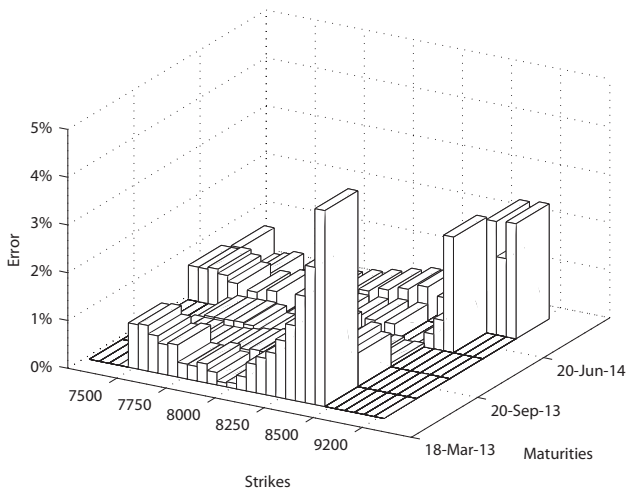
(b)



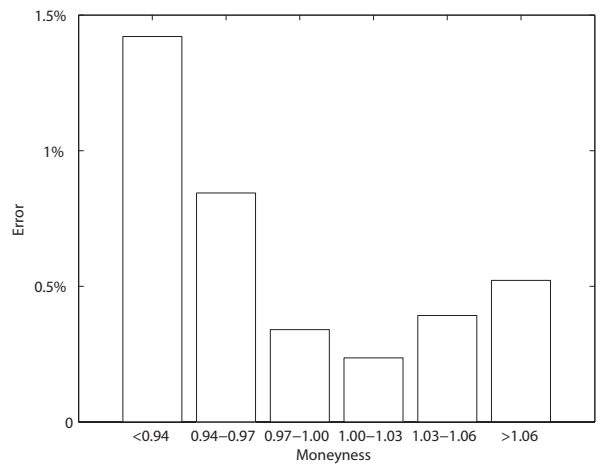
(c)



(d)



(e)



(f)

Figure 7.8: Results for `lsqnonlin`. Left column portrays the absolute relative errors, average absolute relative errors in terms of moneyness are shown on the right. Each row corresponds to a particular choice of weights. First row shows errors when using weights A, the rows are then sorted in the alphabetical order. The initial point was x^{**} .

7.1.6 Conclusion

The following Table 7.9 sums up the results. This time we compared the global optimizers ASA and GA on the real set of data, thus there probably is not a set of parameters which would fully explain the market prices. However, using the `lsqnonlin` and starting from predetermined number of initial points we were able to calibrate the model with average absolute relative error (AARE) 0.51% and maximum absolute relative error (MARE) of 2.44%. The illustrations are given in Figure 7.3.

As mentioned before this is not quite a way to calibrate the model since it is very time consuming. But we can consider it as a some sort of benchmark value.

Using the global optimizers and then employing MS Excel's Solver we were able to obtain set of parameters resulting in a similar magnitude of errors. For example in terms of average absolute relative error, we managed to calibrate the model using an initial point provided by GA with error of 0.51%, whereas the maximum relative error was 2.79%. The mean average relative error was of the same value as for `lsqnonlin` using deterministic grid for the initial points.

The lowest maximum absolute relative error was obtained when using `lsqnonlin` with initial point provided again by GA and it was 2.33% with average absolute relative error of 0.52%.

We can say that best initial point for local optimization was provided by GA algorithm, which is part of MATLAB's Global Optimization Toolbox and requires no additional installation as opposed to ASA.

The choice of weights plays an important role though. We are not able to distinguish, which weight is the most suitable one at this time. Since for example using weights C with Excel did not work for an initial point provided by GA. But considering an initial point provided by ASA, the solver provided us with the best result as far as the maximum absolute relative error is concerned.

It is very likely based on the results in Table 7.9 that weights B suit `lsqnonlin` the best in terms of both average absolute relative error and maximum absolute relative error.

Table 7.9: Calibration results.

| Algorithm | Weights | AARE | RMSE | SSE | MARE | Crit A | Crit B | Crit C | v0 | kappa | theta | sigma | rho |
|---------------------|---------|--------------|-------|----------|--------------|--------|--------|---------|---------|---------|---------|---------|----------|
| Genetic | A | 2.00% | 10.40 | 10496.41 | 20.70% | 623.58 | 86.56 | 2296.51 | 0.03226 | 0.07065 | 0.73827 | 0.81988 | -0.52083 |
| Genetic | B | 2.07% | 14.04 | 19117.07 | 15.13% | 867.68 | 71.98 | 3820.04 | 0.03193 | 0.07747 | 0.73826 | 0.85729 | -0.55003 |
| Genetic | C | 1.24% | 5.76 | 3215.33 | 15.17% | 225.43 | 39.56 | 750.11 | 0.03035 | 0.55662 | 0.11191 | 0.71420 | -0.55050 |
| ASA | A | 1.19% | 6.12 | 3633.19 | 14.52% | 227.37 | 32.59 | 823.12 | 0.03219 | 1.12162 | 0.08278 | 0.96401 | -0.54227 |
| ASA | B | 0.58% | 3.83 | 1425.71 | 4.04% | 70.48 | 5.29 | 304.60 | 0.02845 | 1.26339 | 0.06718 | 0.67255 | -0.62816 |
| ASA | C | 2.55% | 11.19 | 12152.02 | 33.54% | 984.00 | 178.25 | 3075.41 | 0.04111 | 0.80249 | 0.13210 | 1.55269 | -0.47895 |
| LSQdet ^q | B | 0.51% | 3.67 | 1308.63 | 2.44% | 64.88 | 4.30 | 281.65 | 0.02741 | 1.18184 | 0.06586 | 0.57479 | -0.66686 |
| Excel * | A | 0.65% | 3.49 | 1178.53 | 3.86% | 80.08 | 9.26 | 287.21 | 0.02683 | 0.66747 | 0.08426 | 0.46984 | -0.67899 |
| Excel * | B | 0.51% | 3.48 | 1174.29 | 2.79% | 60.22 | 4.24 | 256.14 | 0.02746 | 1.12422 | 0.06762 | 0.57479 | -0.66342 |
| Excel * | C | 1.24% | 5.76 | 3217.37 | 15.17% | 225.33 | 39.55 | 750.09 | 0.03035 | 0.55663 | 0.11192 | 0.71417 | -0.55050 |
| Excel ** | A | 0.55% | 3.46 | 1162.85 | 3.53% | 59.75 | 5.03 | 250.79 | 0.02745 | 1.09385 | 0.06818 | 0.57187 | -0.64966 |
| Excel ** | B | 0.58% | 3.82 | 1415.59 | 3.95% | 70.26 | 5.27 | 303.07 | 0.02843 | 1.26363 | 0.06716 | 0.67246 | -0.62834 |
| Excel ** | C | 0.56% | 3.43 | 1143.00 | 3.51% | 60.80 | 5.39 | 250.01 | 0.02729 | 1.06117 | 0.06852 | 0.55391 | -0.65495 |
| lsqnonlin* | A | 0.55% | 3.46 | 1163.12 | 3.42% | 59.55 | 4.90 | 250.85 | 0.02747 | 1.09567 | 0.06829 | 0.57399 | -0.65043 |
| lsqnonlin* | B | 0.52% | 3.68 | 1315.56 | 2.33% | 64.71 | 4.30 | 282.08 | 0.02760 | 1.20011 | 0.06601 | 0.59282 | -0.65886 |
| lsqnonlin* | C | 0.58% | 3.38 | 1108.65 | 4.19% | 61.89 | 6.25 | 246.18 | 0.02732 | 0.97657 | 0.07120 | 0.54564 | -0.65127 |
| lsqnonlin** | A | 0.55% | 3.48 | 1175.03 | 3.39% | 59.59 | 4.82 | 252.47 | 0.02750 | 1.11668 | 0.06781 | 0.57870 | -0.64958 |
| lsqnonlin** | B | 0.54% | 3.96 | 1518.40 | 2.68% | 72.52 | 4.77 | 320.34 | 0.02786 | 1.24433 | 0.06596 | 0.62264 | -0.64732 |
| lsqnonlin** | C | 0.58% | 3.37 | 1103.20 | 4.10% | 61.68 | 6.20 | 245.23 | 0.02730 | 0.97637 | 0.07113 | 0.54339 | -0.65279 |

^qDenotes the results for the use of the deterministic grid for lsqnonlin.

7.2 March 19, 2013

We did find out in the previous section that the global optimizers GA and ASA are comparable, although it seemed that GA provided us with a better starting point for local optimizers. MS Excel's Solver was competing very well against MATLAB's `lsqnonlin`. We will repeat what we did in the previous section on the data obtained on March 19, 2013. This time we will have 10 more data available with one more maturity.

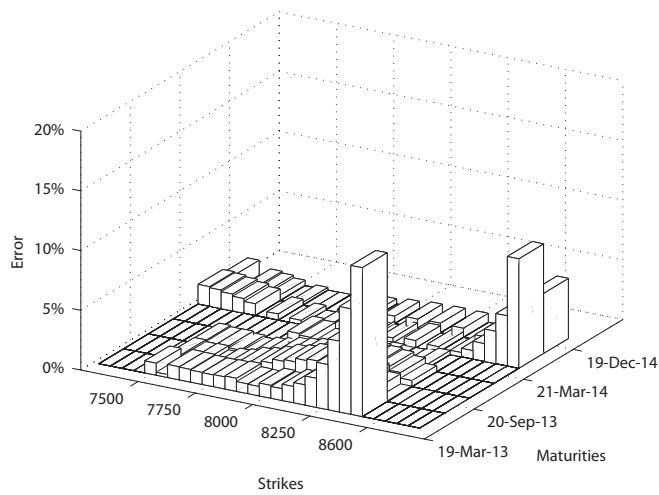
7.2.1 GA

Table 7.10 shows the results using different weights and GA optimizer. This time we did not obtain similar set of parameters as we did on the previous day. The best result appears to be obtained using weights A. This can be observed in Figure 7.9 in pictures (a) and (b) in comparison to (c) - (f).

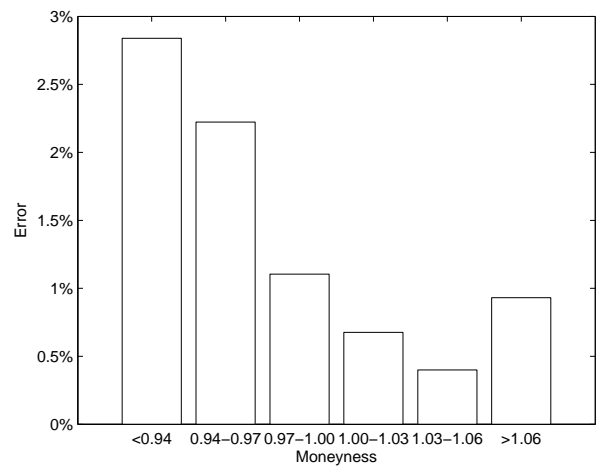
Table 7.10: Set of retrieved parameters using different weights.

| | Used Weights | | |
|-------|--------------|----------|----------|
| | A | B | C |
| v0 | 0.02897 | 0.03073 | 0.03300 |
| kappa | 0.68921 | 0.06405 | 0.83930 |
| theta | 0.10313 | 0.94533 | 0.10826 |
| sigma | 0.79492 | 0.91248 | 1.14674 |
| rho | -0.53769 | -0.53915 | -0.49923 |

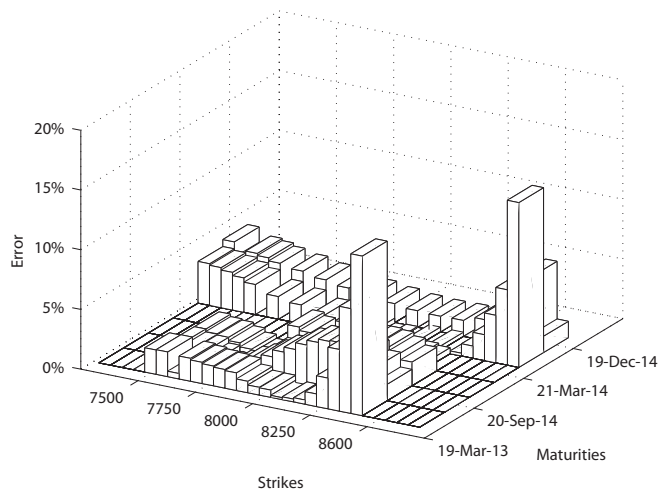
The pictures in Figure 7.9 are similar to what we obtained on the previous day, thus we can observe higher absolute relative errors for out of the money options.



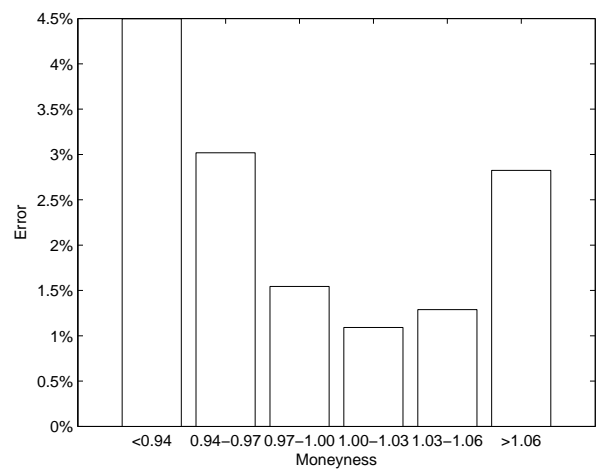
(a)



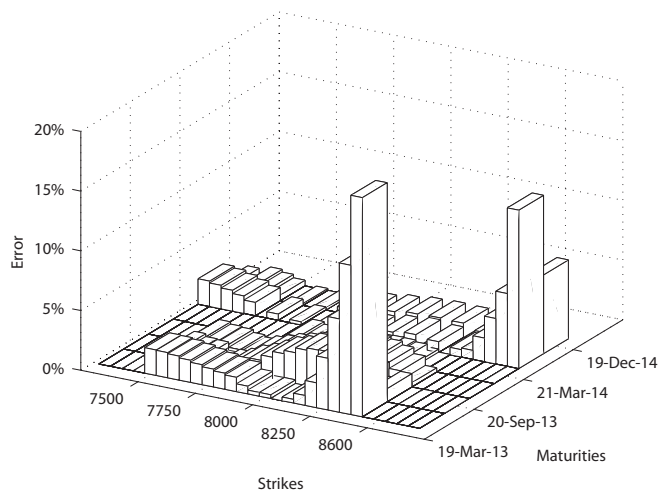
(b)



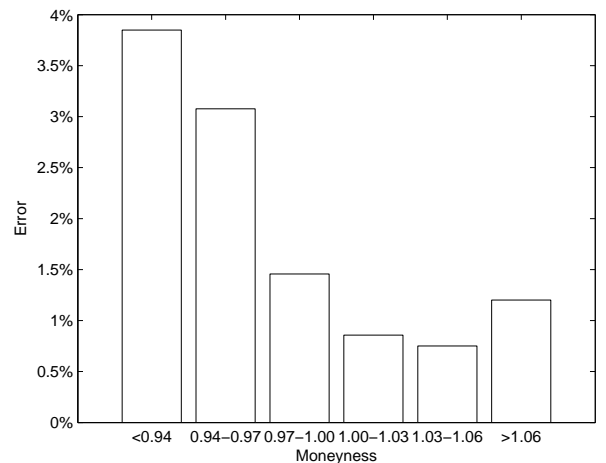
(c)



(d)



(e)



(f)

Figure 7.9: Results for Genetic algorithm. Left column portrays the absolute relative errors, average absolute relative errors in terms of moneyness are shown on the right. Each row corresponds to a particular choice of weights. First row shows errors when using weights A, the rows are then sorted in the alphabetical order.

7.2.2 ASA

This section gives the results for the other global optimizer ASA. Table 7.11 provides us with the set of retrieved parameters using different. We can see less negative values of correlation parameter ρ compared to previous section. Doing some more comparison to the previous section, using ASA we also obtained three rather different sets of parameters.

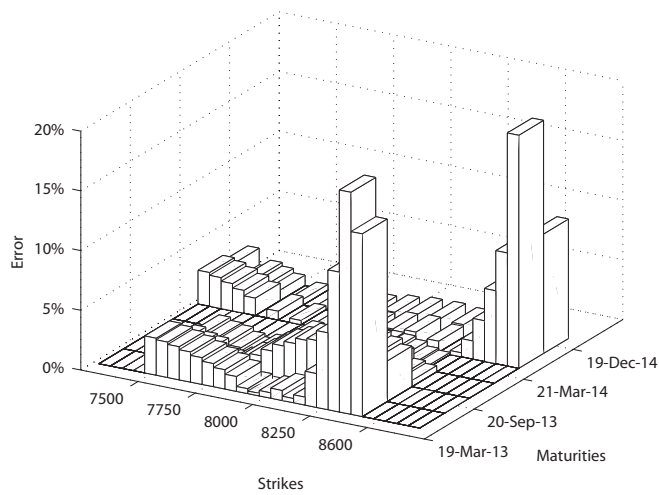
Best results were obtained using weights C. The worst with B, which is in contrast to the results from previous day, on which weights B gave the best results and weights C performed the worst.

The graphs portraying the absolute relative errors using weights B can be seen in Figure 7.10, pictures (c) and (d). The errors are rather high for deep out of the money options indeed.

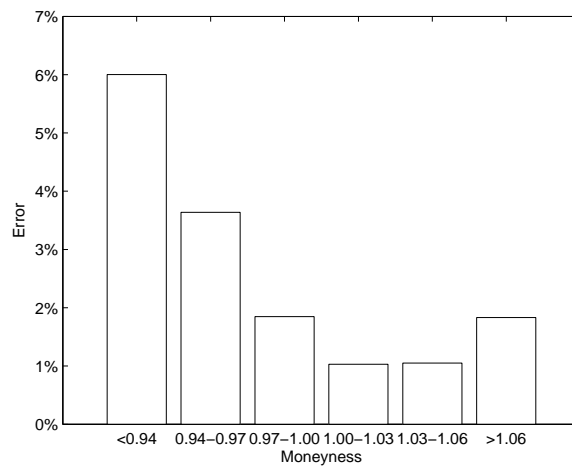
The best performing in terms of average absolute relative errors and maximum absolute relative errors were weights C and the errors are portrayed in pictures (e) and (f) of Figure 7.10.

Table 7.11: Set of retrieved parameters using different weights.

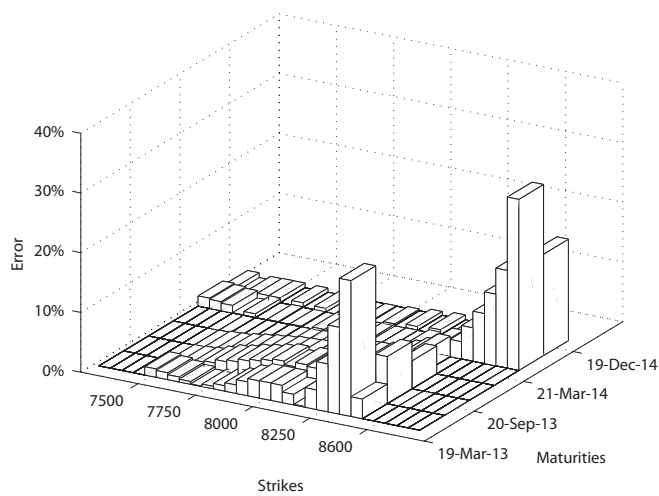
| | Used Weights | | |
|-------|--------------|----------|----------|
| | A | B | C |
| v0 | 0.03876 | 0.03721 | 0.03550 |
| kappa | 0.80811 | 1.45765 | 1.22482 |
| theta | 0.13781 | 0.09663 | 0.09508 |
| sigma | 1.63697 | 1.86941 | 1.44249 |
| rho | -0.46680 | -0.37053 | -0.49063 |



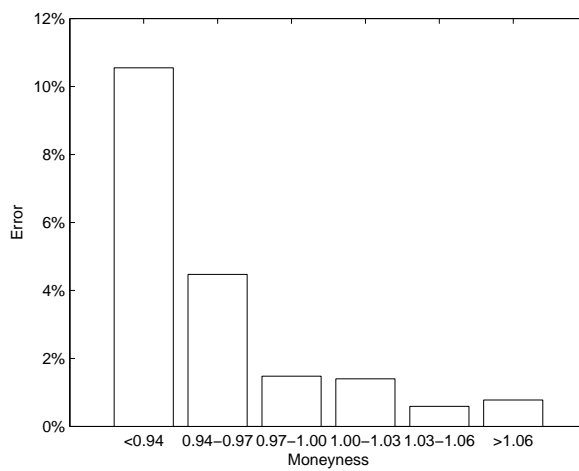
(a)



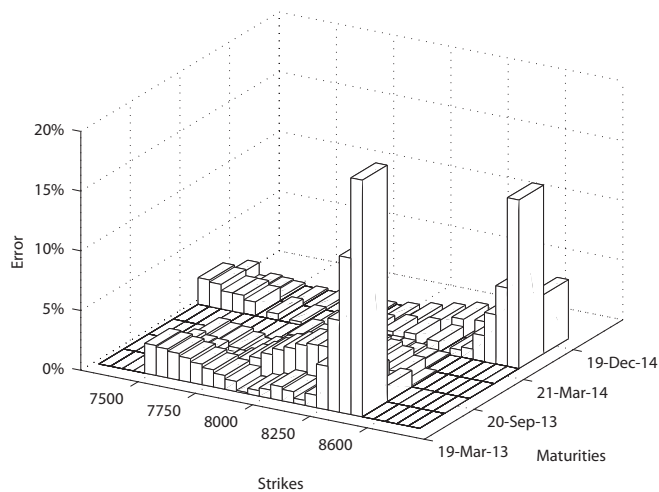
(b)



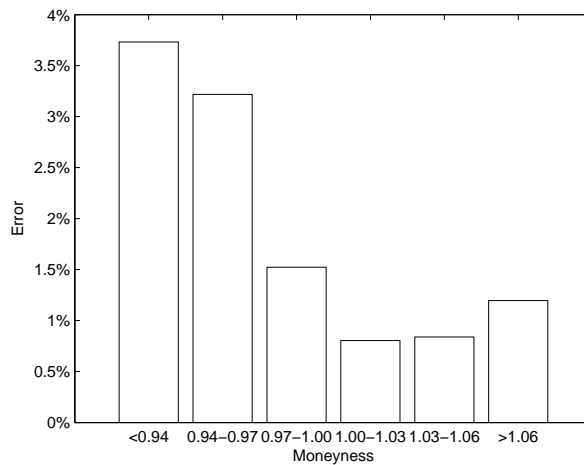
(c)



(d)



(e)



(f)

Figure 7.10: Results for ASA. Left column portrays the absolute relative errors, average absolute relative errors in terms of moneyness are shown on the right. Each row corresponds to a particular choice of weights. First row shows errors when using weights A, the rows are then sorted in the alphabetical order.

7.2.3 Deterministic grid for `lsqnonlin`

For this set of data we also generated a grid of initial points and used these as starting values for MATLAB's `lsqnonlin`. We obtained the following set of parameters in table 7.12. The results are again the best so far in terms of absolute relative errors. The maximum absolute relative error is less than 3.5%, as can be seen in Figure 7.11.

As far as the parameters are concerned, we can observe the lowest values v_0 , θ and σ . On the other hand κ is of the highest value so far. Also we can see the strongest negative correlation ρ .

We will see in the next section if the local optimizers can refine the values provided by the global optimizers ASA and GA and get closer in the sense of errors to these parameters.

Table 7.12: Set of retrieved parameters using weights B.

| | |
|----------|----------|
| v_0 | 0.02382 |
| κ | 1.75680 |
| θ | 0.04953 |
| σ | 0.42134 |
| ρ | -0.84493 |

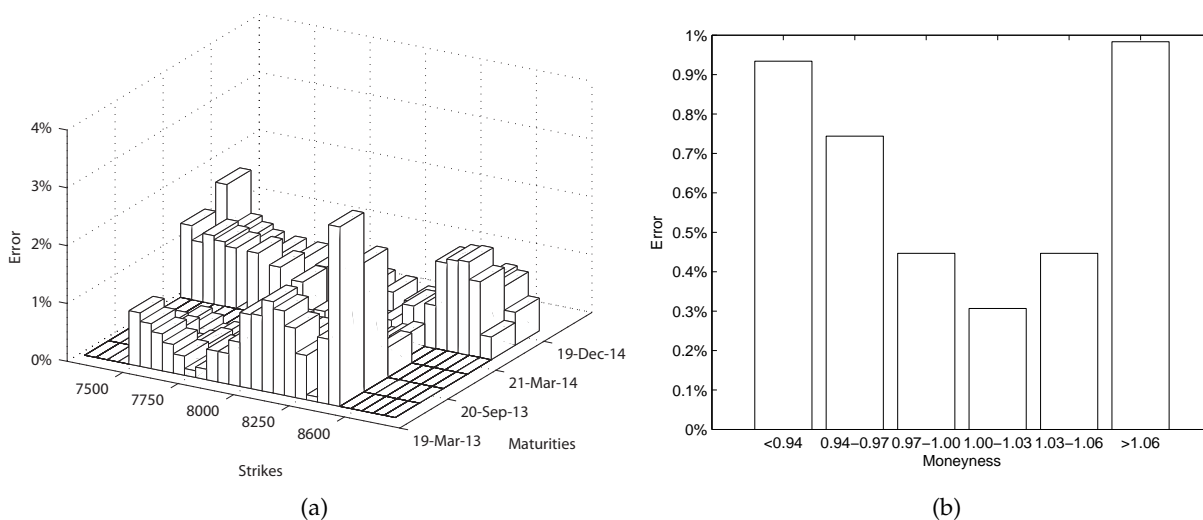


Figure 7.11: (a) the absolute relative errors, (b) the average absolute relative errors in terms of moneyness.

7.2.4 MS Excel's Solver

We will use the same approach as for the data from previous day. We will first employ MS Excel's Solver and provide it with the initial guesses from the global optimizers (GA, ASA). This time denoting the initial point obtained by GA by x^*

$$x^* = [0.02897 \quad 0.68921 \quad 0.10313 \quad 0.79492 \quad -0.53769]$$

and the initial point obtained by ASA by x^{**}

$$x^{**} = [0.03550 \quad 1.22482 \quad 0.09508 \quad 1.44249, -0.49063]$$

Looking at the Tables 7.13 and 7.14 and the initial points above, we can see that MS Excel's Solver did not converge to any better value provided both initial guesses. The parameters are very much the same and there was no significant refinement. See Table 7.17 for comprehensive information.

Table 7.13: Set of retrieved parameters using different weights and initial point x^* .

| | Used Weights | | |
|-------|--------------|----------|----------|
| | A | B | C |
| v0 | 0.02897 | 0.02896 | 0.02903 |
| kappa | 0.68922 | 0.68921 | 0.68932 |
| theta | 0.10314 | 0.10314 | 0.10294 |
| sigma | 0.79490 | 0.79492 | 0.79464 |
| rho | -0.53769 | -0.53769 | -0.53763 |

Table 7.14: Set of retrieved parameters using different weights and initial point x^{**} .

| | Used Weights | | |
|-------|--------------|----------|----------|
| | A | B | C |
| v0 | 0.03550 | 0.03439 | 0.03550 |
| kappa | 1.22482 | 1.22399 | 1.22482 |
| theta | 0.09509 | 0.09740 | 0.09509 |
| sigma | 1.44248 | 1.43711 | 1.44248 |
| rho | -0.49062 | -0.49115 | -0.49062 |

7.2.5 lsqnonlin

MATLAB's `lsqnonlin` also did not refine the parameters when starting from the initial guess provided by ASA. But it was able to refine the set of parameters when starting from the initial guess provided by GA.

We included both tables, we can see the results for when starting from x^* in Table 7.15 and as we can observe, these are different from the initial point. The improved errors can be seen in Figure 7.12.

| | x^* | Used Weights | | |
|-------|----------|--------------|----------|----------|
| | | A | B | C |
| v0 | 0.02897 | 0.02491 | 0.02497 | 0.02486 |
| kappa | 0.68921 | 0.82270 | 1.22136 | 0.75195 |
| theta | 0.10313 | 0.07597 | 0.06442 | 0.07886 |
| sigma | 0.79492 | 0.48665 | 0.55993 | 0.46936 |
| rho | -0.53769 | -0.67099 | -0.66255 | -0.67266 |

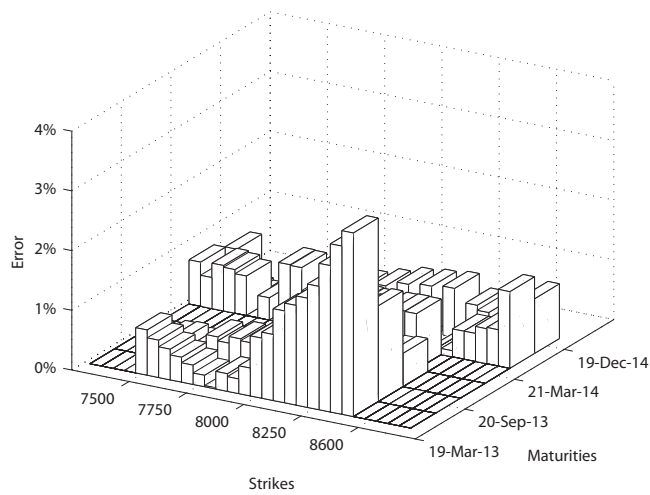
Table 7.15: Set of retrieved parameters using different weights starting from initial point x^* . Initial point x^* is included for comparison.

Table 7.16 shows the parameters obtained when starting from the initial point provided by ASA. These are almost identical to the initial point and did not lead to any significant improvement. This fact can be seen in the final Table 7.17.

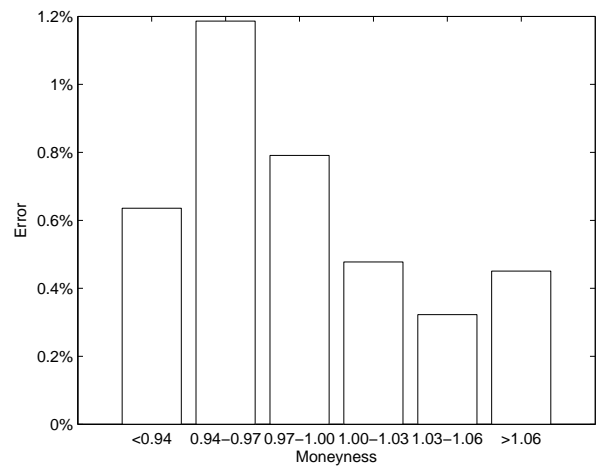
| | x^{**} | Used Weights | | |
|-------|----------|--------------|----------|----------|
| | | A | B | C |
| v0 | 0.03550 | 0.03550 | 0.03511 | 0.03550 |
| kappa | 1.22482 | 1.22482 | 1.22672 | 1.22482 |
| theta | 0.09508 | 0.09508 | 0.09636 | 0.09508 |
| sigma | 1.44249 | 1.44249 | 1.44194 | 1.44249 |
| rho | -0.49063 | -0.49063 | -0.49089 | -0.49063 |

Table 7.16: Set of retrieved parameters using different weights starting from initial point x^{**} . Initial point x^{**} is included for comparison.

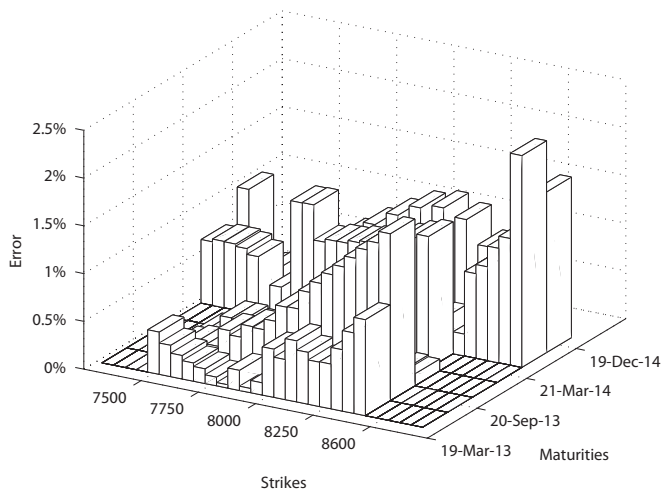
The best result was again when using weights B and we were able to obtain maximum absolute relative error under 2.5%, see Figure 7.12, pictures (c) and (d).



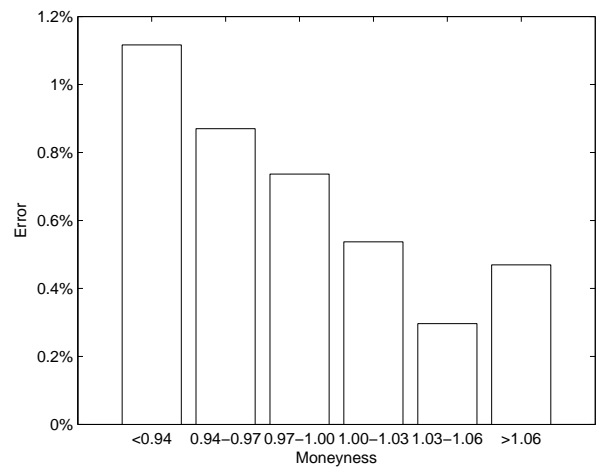
(a)



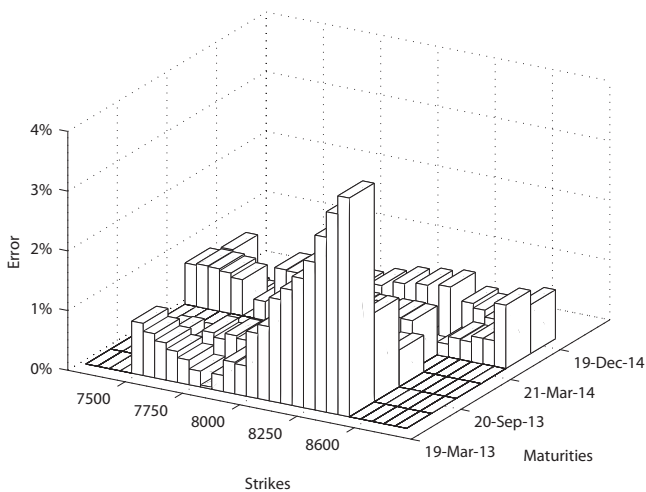
(b)



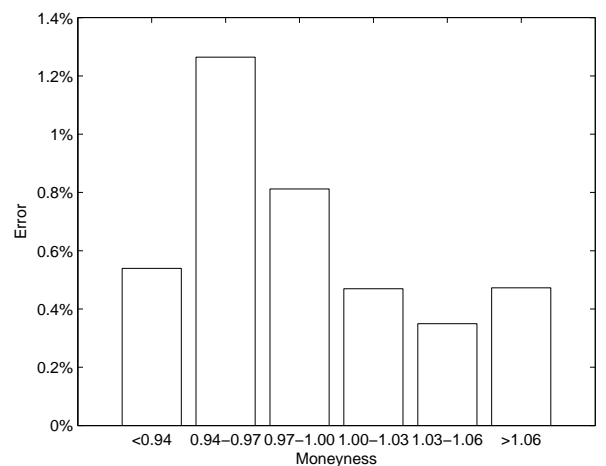
(c)



(d)



(e)



(f)

Figure 7.12: Results for `lsqnonlin`. Left column portrays the absolute relative errors, average absolute relative errors in terms of moneyness are shown on the right. Each row corresponds to a particular choice of weights. First row shows errors when using weights A, the rows are then sorted in the alphabetical order. The initial point x^* .

7.2.6 Conclusion

Although performing well for the previous set of data, this time Excel failed to significantly refine the initial values provided by the global optimizers. On the other hand using MATLAB's `lsqnonlin` we were able to refine the initial set of parameters provided by GA and obtain average absolute relative error comparable to average absolute relative error obtained by the method using deterministic grid as the initial value for `lsqnonlin`.

When comparing the global optimizers GA provided better results than ASA, which again is quite pleasant since it is part of MATLAB's Toolbox and it is easier to work with. Also the initial guess from GA was the one which was later refined by `lsqnonlin` producing lowest maximum absolute relative error (MARE) of 2.22%.

Interesting fact is that as far as the weights are concerned, we can not say which work the best for GA and ASA. Different weights yielded best results for both GA and ASA when compared to the previous set of data. On the contrary according to the results `lsqnonlin` seems to be favoring weights B.

All the discussed above can be seen in Table 7.17 on the following page.

Table 7.17: Calibration results.

| Algorithm | Weights | AARE | RMSE | SSE | MARE | Crit A | Crit B | Crit C | v0 | kappa | theta | sigma | rho |
|-------------|---------|-------|-------|----------|--------|---------|--------|---------|---------|---------|---------|---------|----------|
| Genetic | A | 1.25% | 6.26 | 4195.22 | 12.46% | 299.83 | 40.23 | 1032.16 | 0.02897 | 0.68921 | 0.10313 | 0.79492 | -0.53769 |
| Genetic | B | 2.10% | 14.94 | 23889.28 | 13.80% | 1214.02 | 101.35 | 5092.94 | 0.03073 | 0.06405 | 0.94533 | 0.91248 | -0.53915 |
| Genetic | C | 1.70% | 8.30 | 7375.52 | 18.35% | 591.91 | 80.82 | 1951.42 | 0.03300 | 0.83930 | 0.10826 | 1.14674 | -0.49923 |
| ASA | A | 2.26% | 11.79 | 14867.67 | 19.51% | 1042.47 | 128.15 | 3657.32 | 0.03876 | 0.80811 | 0.13781 | 1.63697 | -0.46680 |
| ASA | B | 2.62% | 13.80 | 20379.47 | 28.65% | 1303.43 | 146.26 | 4814.26 | 0.03721 | 1.45765 | 0.09663 | 1.86941 | -0.37053 |
| ASA | C | 1.73% | 8.42 | 7579.78 | 19.82% | 645.94 | 89.62 | 2077.70 | 0.03550 | 1.22482 | 0.09508 | 1.44249 | -0.49063 |
| LSQdet | B | 0.58% | 4.95 | 2622.96 | 3.10% | 122.48 | 8.77 | 542.09 | 0.02382 | 1.75680 | 0.04953 | 0.42134 | -0.84493 |
| Excel* | A | 1.25% | 6.27 | 4202.86 | 12.46% | 299.82 | 40.19 | 1033.04 | 0.02897 | 0.68922 | 0.10314 | 0.79490 | -0.53769 |
| Excel* | B | 1.25% | 6.27 | 4200.81 | 12.46% | 299.81 | 40.19 | 1032.80 | 0.02896 | 0.68921 | 0.10314 | 0.79492 | -0.53769 |
| Excel* | C | 1.25% | 6.23 | 4146.95 | 12.66% | 300.89 | 41.04 | 1027.36 | 0.02903 | 0.68932 | 0.10294 | 0.79464 | -0.53763 |
| Excel** | A | 1.73% | 8.42 | 7579.87 | 19.82% | 645.89 | 89.61 | 2077.60 | 0.03550 | 1.22482 | 0.09509 | 1.44248 | -0.49062 |
| Excel** | B | 1.78% | 9.79 | 10255.34 | 18.18% | 676.99 | 78.33 | 2465.23 | 0.03439 | 1.22399 | 0.09740 | 1.43711 | -0.49115 |
| Excel** | C | 1.73% | 8.42 | 7579.84 | 19.82% | 645.89 | 89.61 | 2077.59 | 0.03550 | 1.22482 | 0.09509 | 1.44248 | -0.49062 |
| lsqnonlin* | A | 0.67% | 3.49 | 1301.44 | 3.07% | 84.16 | 8.75 | 313.79 | 0.02491 | 0.82270 | 0.07597 | 0.48665 | -0.67099 |
| lsqnonlin* | B | 0.65% | 4.27 | 1955.17 | 2.22% | 99.78 | 6.54 | 430.33 | 0.02497 | 1.22136 | 0.06442 | 0.55993 | -0.66255 |
| lsqnonlin* | C | 0.68% | 3.45 | 1277.06 | 3.66% | 89.02 | 10.27 | 317.55 | 0.02486 | 0.75195 | 0.07886 | 0.46936 | -0.67266 |
| lsqnonlin** | A | 1.73% | 8.42 | 7579.90 | 19.82% | 645.93 | 89.62 | 2077.69 | 0.03550 | 1.22482 | 0.09508 | 1.44249 | -0.49063 |
| lsqnonlin** | B | 1.71% | 9.10 | 8859.31 | 19.48% | 659.70 | 83.48 | 2271.47 | 0.03511 | 1.22672 | 0.09636 | 1.44194 | -0.49089 |
| lsqnonlin** | C | 1.73% | 8.42 | 7579.76 | 19.82% | 645.96 | 89.63 | 2077.72 | 0.03550 | 1.22482 | 0.09508 | 1.44249 | -0.49063 |

Chapter 8

Conclusion

The aim of this thesis was to study the current state of the art of both calibration process and the methods of simulating the paths of the Heston model. We also proposed possible improvements for both.

We studied the available articles and literature to find the most popular and industry standard methods to simulate the Heston model variables. We introduced the most popular ones in the thesis and we ruled out the Exact scheme by Broadie and Kaya [6] from the testing not only because of the computationally expensive numerical inversion but also because of the problems of dividing two modified Bessel functions of the first kind, these two were causing difficulties because of the representation of their rather high values for finer discretizations. Also IJK scheme [20], which is supposed to be a *drop-in* method for simple schemes is tested with parameters obtained from the market and it appears not to be useful for pricing options using Monte Carlo simulations. We also did experiments and proposed a scheme compromising speed and accuracy based on the observed performances of the so far known schemes and we seem to have out-performed the other simple schemes when using the Monte Carlo simulations for pricing options. The proposed scheme uses Milstein scheme for the variance process exploiting the lower frequency of negative values and combines it with the Exact scheme approach by Broadie and Kaya for the asset process and could be a *drop-in* replacement for log-Euler or Milstein scheme if higher precision is required. The best in terms of precision appears to be the QE scheme [2], which is the most popular one among the practitioners these days. All the conclusions were made based on the test that involved mimicking real market situation and thus we avoided parameters or range of strikes and maturities that could favor some of the schemes purposely.

We did experience rather turbulent behaviour with the classical numerical schemes probably due to the possible negative values and the consequent fixes of the variance process and thus we did not provide a proper results in terms of convergence criterion. This appears to be an issue, which could use some further exploration and proper analysis. Also further study of the Exact scheme and possible simplifications seems attractive since practitioners prefer these schemes to the higher order numerical schemes, e.g. Milstein.

Calibration of the Heston model is not covered in the literature in such extent that could be comparable to the coverage of the simulation schemes. We take the real data from the market and try to calibrate the model to them. This is not an easy task, especially when one uses rather high number of options. We approached it by combining the local optimizers, which tend to be sensitive to an initial guess and thus we think one should avoid using only those, with the global optimizers. We suggested using the Genetic algorithm from the MATLAB's Global Optimization Toolbox and we developed what we believe is a stable procedure combining the Genetic algorithm providing an initial guess, which is later refined by non-linear least squares method (`lsqnonlin`). Having the opportunity to run long computations on the departmental server¹ we were able to find benchmark values by creating a grid of the state space for the parameters and find out what the model is capable of. We managed to match the magnitude of the errors by our procedure, but of course in shorter computational time. We believe this is an approach that could be further used to compare for example SABR model to the Heston model. SABR model by Hagan, Kumar, Lesniewski and Woodward [13] is a newer model of the stochastic volatility and is very popular these days. Our procedure could be used for comparison of these two while avoiding inadequate results because of the calibration procedure. We also used Adaptive Simulated Annealing and Excel's Solver and compared the results for two consecutive days. Another conclusion is that slight adjustments to weights used during calibration can play significant role and one can improve the results when experimenting with them.

¹damadama.fav.zcu.cz

Appendix A

Risk-Neutral Valuation

Suppose that under a risk-neutral measure $\tilde{\mathbb{P}}$ a stock price is governed by

$$dS(t) = rS(t)dt + \sqrt{v(t)}S(t)d\tilde{W}^S(t), \quad (\text{A.1})$$

where the interest rate r is constant and the volatility $\sqrt{v(t)}$ is itself a stochastic process governed by the equation

$$dv(t) = (a - bv(t))dt + \sigma\sqrt{v(t)}d\tilde{W}^v(t). \quad (\text{A.2})$$

The parameters a, b and σ are positive constants, and $\tilde{W}^S(t)$ and $\tilde{W}^v(t)$ are correlated Brownian motions under $\tilde{\mathbb{P}}$ with

$$d\tilde{W}^S(t)d\tilde{W}^v(t) = \rho dt$$

for some $\rho \in (-1, 1)$. Since the two dimensional process $(S(t), v(t))$ is governed by the pair of stochastic differential equations (A.1) and (A.2), it is a two dimensional Markov process. So long as trading takes place only in the stock and money market account, this model is incomplete¹. One can create a one-parameter family of risk-neutral measures by changing the dt term in (A.2) without affecting (A.1).

According to risk-neutral pricing formula, for example in Shreve [32], at time t , the risk neutral price of a call expiring at time $T \geq t$ in this stochastic volatility model is $\tilde{\mathbb{E}}[e^{-r(T-t)}(S(T) - K)^+ | \mathcal{F}(t)]$, where $\tilde{\mathbb{E}}$ denotes \mathbb{E} w.r.t. $\tilde{\mathbb{P}}$. Thank to the Markov property, there is a function $c(t, s, v)$ such that

$$c(t, S(t), v(t)) = \tilde{\mathbb{E}}[e^{-r(T-t)}(S(T) - K)^+ | \mathcal{F}(t)], \quad 0 \leq t \leq T. \quad (\text{A.3})$$

This appendix shows that the function $c(t, S, v)$ satisfies the partial differential equation

$$c_t + rSc_S + (a - bv)c_v + \frac{1}{2}S^2vc_{SS} + \rho\sigma Svc_{Sv} + \frac{1}{2}\sigma^2vc_{vv} = rc \quad (\text{A.4})$$

¹There are two sources of risk and only one traded asset (volatility is not a traded asset).

in the region $0 \leq t < T$, $S \geq 0$, and $v \geq 0$. The function $c(t, S, v)$ also satisfies the boundary conditions

$$c(T, S, v) = (S - K)^+ \text{ for all } S \geq 0, v \geq 0, \quad (\text{A.5})$$

$$c(t, 0, v) = 0 \text{ for all } 0 \leq t \leq T, v \geq 0, \quad (\text{A.6})$$

$$c(t, S, 0) = (S - e^{-r(T-t)}K)^+ \text{ for all } 0 \leq t \leq T, S \geq 0, \quad (\text{A.7})$$

$$\lim_{S \rightarrow \infty} \frac{c(t, S, v)}{S - K} = 1 \text{ for all } 0 \leq t \leq T, v \geq 0, \quad (\text{A.8})$$

$$\lim_{v \rightarrow \infty} c(t, S, v) = s \text{ for all } 0 \leq t \leq T, s \geq 0. \quad (\text{A.9})$$

We will be concerned with (A.5).

A.1 Part (i)

We will show that $e^{-rt}c(t, S(t), v(t))$ is a martingale under $\tilde{\mathbb{P}}$, and use this fact to obtain (A.4).

From (A.3) and using the iterated conditioning we get for $0 \leq s \leq t \leq T$

$$\begin{aligned} \tilde{\mathbb{E}}[e^{-rt}c(t, S(t), v(t)) \mid \mathcal{F}(s)] &= \tilde{\mathbb{E}}[\tilde{\mathbb{E}}[e^{-rT}(S(T) - K)^+ \mid \mathcal{F}(t)] \mid \mathcal{F}(s)] \\ &= \tilde{\mathbb{E}}[e^{-rT}(S(T) - K)^+ \mid \mathcal{F}(s)] \\ &= e^{-rs}c(s, S(s), V(s)). \end{aligned}$$

Using the stochastic differential equations (A.1) and (A.2) and noting that $d\tilde{W}^S(t)d\tilde{W}^v(t) = \rho dt$, we take the differential of the $\tilde{\mathbb{P}}$ -martingale $e^{-rt}c(t, S(t), v(t))$. This differential according to Itô product rule² is

$$d(e^{-rt}c(t, S(t), v(t))) = -re^{-rt}c(t, S(t), v(t))dt + e^{-rt}dc(t, S(t), v(t))$$

Following the Two-dimensional Itô-Doeblin formula³ and denoting $c(t, S(t), v(t))$ as c we obtain

²Itô product rule is given as Corollary 4.6.3 in Shreve [32].

³For example Theorem 4.6.2 in Shreve [32].

$$\begin{aligned}
d(e^{-rt}c) &= e^{-rt} \left[-rcdt + c_t dt + c_S dS + c_v dv + \frac{1}{2} c_{SS} dS dS + \frac{1}{2} c_{Sv} dS dv + \frac{1}{2} c_{vv} dv dv \right] \\
&= e^{-rt} \left[-rc + c_t + c_S r S(t) + c_v (a - bv(t)) + \frac{1}{2} c_{SS} v(t) S^2(t) \right. \\
&\quad \left. + \frac{1}{2} c_{Sv} \sigma \rho v(t) S(t) + \frac{1}{2} c_{vv} \sigma^2 v(t) \right] dt + e^{-rt} \left[c_S \sqrt{v(t)} S(t) d\tilde{W}^S(t) \right. \\
&\quad \left. + c_v \sigma \sqrt{v(t)} d\tilde{W}^v(t) \right].
\end{aligned}$$

In order to be a martingale, the dt term must be zero, which implies equation (A.4).

A.2 Part (ii)

We suppose there are functions $f(t, x, v)$ and $g(t, x, v)$ satisfying

$$f_t + \left(r + \frac{1}{2}v \right) f_x + (a - bv + \rho\sigma v) f_v + \frac{1}{2}v f_{xx} + \rho\sigma v f_{xv} + \frac{1}{2}\sigma^2 v f_{vv} = 0, \quad (\text{A.10})$$

$$g_t + \left(r - \frac{1}{2}v \right) g_x + (a - bv) g_v + \frac{1}{2}v g_{xx} + \rho\sigma v g_{xv} + \frac{1}{2}\sigma^2 v g_{vv} = 0, \quad (\text{A.11})$$

in the region $0 \leq t < T$, $-\infty < x < \infty$, and $v \geq 0$. We will show that if we define

$$c(t, S, v) = S f(t, \ln S, v) - e^{-r(T-t)} K g(t, \ln S, v), \quad (\text{A.12})$$

then $c(t, S, v)$ satisfies the partial differential equation (A.4).

Taking (A.4) and (A.12) and denoting $f(t, x, v)$ as f satisfying (A.10) and $g(t, x, v)$ as g satisfying (A.11) we get

$$\begin{aligned}
rc &= \underbrace{Sf_t - re^{-r(T-t)}Kg - e^{-r(T-t)}Kg_t}_{c_t} + \underbrace{rSf + rSf_s - rKe^{-r(T-t)}g_s}_{rSc_s} \\
&+ \underbrace{(a-bv)(Sf_v - e^{-r(T-t)}Kg_v)}_{(a-bv)c_v} + \underbrace{\frac{1}{2}S^2v \left(\frac{1}{S}f_s + \frac{1}{S}f_{SS} - e^{-r(T-t)}K\frac{1}{S^2}g_{SS} + e^{-r(T-t)}K\frac{1}{S^2}g_s \right)}_{\frac{1}{2}S^2vc_{SS}} \\
&+ \underbrace{\rho\sigma Sv \left(f_v + f_{Sv} - e^{-r(T-t)}K\frac{1}{S}g_{Sv} \right)}_{\rho\sigma Sv c_{Sv}} + \underbrace{\frac{1}{2}\sigma^2v \left(Sf_{vv} - e^{-r(T-t)}Kg_{vv} \right)}_{\frac{1}{2}\sigma^2v c_{vv}} \\
&= \left[f_t + \left(r + \frac{1}{2}v \right) f_s + (a-bv + \rho\sigma v) f_v + \frac{1}{2}v f_{SS} + \rho\sigma v f_{Sv} + \frac{1}{2}\sigma^2v f_{vv} \right] S \\
&- \left[g_t + \left(r - \frac{1}{2}v \right) g_s + (a-bv) g_v + \frac{1}{2}v g_{SS} + \rho\sigma v g_{Sv} + \frac{1}{2}\sigma^2v g_{vv} \right] Ke^{-r(T-t)} \\
&+ rSf - re^{-r(T-t)}Kg \\
&= r(Sf - e^{-r(T-t)}Kg) = rc.
\end{aligned}$$

And so (A.12) indeed satisfies the partial differential equation (A.4).

A.3 Part (iii)

Suppose a pair of processes $(X(t), v(t))$ is governed by the stochastic differential equations

$$dX(t) = \left(r + \frac{1}{2}v(t) \right) dt + \sqrt{v(t)} d\tilde{W}^S(t), \quad (\text{A.13})$$

$$dv(t) = (a - bv(t) + \rho\sigma v(t)) dt + \sigma \sqrt{v(t)} d\tilde{W}^v(t), \quad (\text{A.14})$$

where $\tilde{W}^S(t)$ and $\tilde{W}^v(t)$ are Brownian motions under some probability measure \mathbb{P} with $d\tilde{W}^S(t)d\tilde{W}^v(t) = \rho dt$. Define

$$f(t, x, v) = \mathbb{E}^{t,x,v} \mathbb{I}_{\{X(T) \geq \ln K\}}, \quad (\text{A.15})$$

where the notation $\mathbb{E}^{t,x,v}$ emphasizes the dependence on t , x and v .

We will show that $f(t, x, v)$ satisfies the partial differential equation (A.10) and the boundary condition

$$f(T, x, v) = \mathbb{I}_{\{x \geq \ln K\}} \text{ for all } x \in \mathbb{R}, v \geq 0. \quad (\text{A.16})$$

Using Markov property we obtain

$$\mathbb{E}[\mathbb{I}_{\{X(T) \geq \ln K\}} \mid \mathcal{F}(t)] = f(t, X(t), v(t)), \quad (\text{A.17})$$

and so

$$f(T, X(t), v(t)) = \mathbb{I}_{\{X(T) \geq \ln K\}},$$

which implies the boundary condition (A.16) for all $x \in \mathbb{R}, v \geq 0$. Markov property (A.17) and use of iterated conditioning shows that for $0 \leq s \leq t \leq T$

$$\begin{aligned} \mathbb{E}[f(t, X(t), v(t)) \mid \mathcal{F}(s)] &= \mathbb{E}[\mathbb{E}[\mathbb{I}_{\{X(T) \geq \ln K\}} \mid \mathcal{F}(t) \mid \mathcal{F}(s)]] \\ &= \mathbb{E}[\mathbb{I}_{\{X(T) \geq \ln K\}} \mid \mathcal{F}(s)] \\ &= f(s, X(s), v(s)). \end{aligned}$$

We denote $f(t, x, v)$ as f and take its differential using Two-dimensional Itô-Doebelin formula

$$\begin{aligned} df &= f_t dt + f_x dX(t) + f_v dv(t) + \frac{1}{2} f_{xx} v(t) dt + f_{xv} \sigma \rho v(t) dt + \frac{1}{2} f_{vv} \sigma^2 v(t) dt \\ &= \left[f_t + f_x \left(r + \frac{1}{2} v(t) \right) + f_v (a - bv(t) + \rho \sigma v(t)) + \frac{1}{2} f_{xx} v(t) + f_{xv} \sigma \rho v(t) \right. \\ &\quad \left. + \frac{1}{2} f_{vv} \sigma^2 v(t) \right] dt + f_x \sqrt{v(t)} d\tilde{W}^S(t) + f_v \sigma \sqrt{v(t)} d\tilde{W}^v(t). \end{aligned}$$

Setting the drift term to zero in order for f to be a martingale we get the partial differential equation (A.10).

A.4 Part (iv)

Suppose a pair of processes $(X(t), v(t))$ is governed by the stochastic differential equations

$$dX(t) = \left(r - \frac{1}{2} v(t) \right) dt + \sqrt{v(t)} d\tilde{W}^S(t), \quad (\text{A.18})$$

$$dv(t) = (a - bv(t)) dt + \sigma \sqrt{v(t)} d\tilde{W}^v(t), \quad (\text{A.19})$$

where $\tilde{W}^S(t)$ and $\tilde{W}^v(t)$ are Brownian motions under some probability measure \mathbb{P} with $d\tilde{W}^S(t) d\tilde{W}^v(t) = \rho dt$. Define

$$g(t, x, v) = \mathbb{E}^{t, x, v} \mathbb{I}_{\{X(T) \geq \ln K\}}. \quad (\text{A.20})$$

We will show that $g(t, x, v)$ satisfies the partial differential equation (A.11) and the boundary condition

$$g(T, x, v) = \mathbb{I}_{\{x \geq \ln K\}} \text{ for all } x \in \mathbb{R}, v \geq 0. \quad (\text{A.21})$$

Similarly as in previous part we denote $g(t, x, v)$ as g and compute its differential using Two-dimensional Itô-Doebelin formula

$$\begin{aligned} dg &= g_t dt + g_x dX(t) + g_v dv(t) + \frac{1}{2} g_{xx} v(t) dt + g_{xv} \sigma \rho v(t) dt + \frac{1}{2} g_{vv} \sigma^2 v(t) dt \\ &= \left[g_t + g_x \left(r - \frac{1}{2} v(t) \right) + g_v (a - bv(t)) + \frac{1}{2} g_{xx} v(t) + g_{xv} \sigma \rho v(t) \right. \\ &\quad \left. + \frac{1}{2} g_{vv} \sigma^2 v(t) \right] dt + g_x \sqrt{v(t)} d\tilde{W}^S(t) + g_v \sigma \sqrt{v(t)} d\tilde{W}^v(t). \end{aligned}$$

Setting the term dt to zero we get the partial differential equation (A.11). Markov property implies the boundary condition (A.21) as in Part (iii).

A.5 Part (v)

We will show that with $f(t, x, v)$ and $g(t, x, v)$ as in (iii) and (iv), the function $c(t, x, v)$ of (A.12) satisfies the boundary condition (A.5).

The following lines shows the boundary condition (A.5) is satisfied

$$\begin{aligned} c(T, S, v) &= S \mathbb{I}_{\{\ln S \geq \ln K\}} - e^{-r(T-T)} K \mathbb{I}_{\{\ln S \geq \ln K\}} \\ &= (S - K) \mathbb{I}_{\{S \geq K\}} \\ &= (S - K)^+. \end{aligned}$$

Appendix B

Content of the CD

CD contains file with the thesis in folder THESIS, obtained data from the Bloomberg Option Monitor can be found in DATA. Folder CALIBRATION contains codes used to calibrate the Heston model. Folder MC_TESTS provides codes used to test various schemes. Directory structure of the CD:

- CALIBRATION
- DATA
- MC_TESTS
- THESIS

Bibliography

- [1] H. Albrecher, P. Mayer, W. Schoutens, and J. Tistaert, *The little heston trap*, 2006.
- [2] L. Andersen, *Simple and efficient simulation of the heston stochastic volatility model*, *Journal of Computational Finance* **11** (2008), no. 3, 1–42.
- [3] L. B. Andersen and R. Brotherton-Ratcliffe, *Extended LIBOR market models with stochastic volatility*, *Journal of Computational Finance* **9** (2005), no. 1, 1–40.
- [4] G. Bakshi, C. Cao, and Z. Chen, *Empirical performance of alternative option pricing models*, *The Journal of Finance* **52** (1997), no. 5, 2003–2049.
- [5] F. Black and M. S. Scholes, *The pricing of options and corporate liabilities*, *Journal of Political Economy* **81** (1973), no. 3, 637–54.
- [6] M. Broadie and Ö. Kaya, *Exact simulation of stochastic volatility and other affine jump diffusion processes*, *Operations Research* **54** (2006), no. 2, 217–231.
- [7] R. Cont and S. B. Hamida, *Recovering volatility from option prices by evolutionary optimization*, *Journal of Computational Finance* **8** (2005), no. 4, 43–76.
- [8] J. C. Cox, J. E. Ingersoll, and S. A. Ross, *A theory of the term structure of interest rates*, *Econometrica* **53** (1985), no. 2, 385–407.
- [9] G. Deelstra and F. Delbaen, *Convergence of discretized stochastic (interest rate) processes with stochastic drift term*, *Applied Stochastic Models and Data Analysis* **14** (1998), no. 1, 77–84.
- [10] W. Feller, *Two singular diffusion problems*, *The Annals of Mathematics* **54** (1951), no. 1, 173–182.
- [11] J. Gatheral and N.N. Taleb, *The volatility surface: A practitioner’s guide*, Wiley Finance, Wiley, 2006.
- [12] P. Glasserman, *Monte Carlo Methods in Financial Engineering (Stochastic Modelling and Applied Probability)* (v. 53), 1 ed., Springer, 2003.
- [13] Kumar D. Lesniewski A. Hagan, P. S. and D. E Woodward, *Managing smile risk*, *WILMOTT Magazine* (2002), 84–108.

- [14] S. L. Heston, *A closed-form solution for options with stochastic volatility with applications to bond and currency options*, *Review of Financial Studies* **6** (1993), no. 2, 327–343.
- [15] D. J. Higham and X. Mao, *Convergence of monte carlo simulations involving the mean-reverting square root process*, *Journal of Computational Finance* **8** (2005), no. 3, 35–61.
- [16] J. Hull and A. White, *The pricing of options on assets with stochastic volatilities*, *The Journal of Finance* **42** (1987), no. 2, 281–300.
- [17] J. C. Hull, *Options, futures, and other derivatives*, Pearson Prentice Hall, 2006.
- [18] E. Jacquier and R. Jarrow, *Bayesian analysis of contingent claim model error*, *Journal of Econometrics* **94** (2000), no. 1–2, 145 – 180.
- [19] C. Kahl and P. Jäckel, *Not-so-complex logarithms in the heston model*, *Wilmott Magazine* (2005), 94–103.
- [20] C. Kahl and P. Jäckel, *Fast strong approximation monte carlo schemes for stochastic volatility models*, *Quantitative Finance* **6** (2006), no. 6, 513–536.
- [21] I.A. Karatzas and S.E. Shreve, *Brownian motion and stochastic calculus*, Graduate Texts in Mathematics Series, Springer London, Limited, 1991.
- [22] J. Kienitz and D. Wetterau, *Financial modelling: Theory, implementation and practice with matlab source*, The Wiley Finance Series, Wiley, 2012.
- [23] P.E. Kloeden and E. Platen, *Numerical solution of stochastic differential equations*, Applications of Mathematics Series, Springer, 1992.
- [24] R. Lord, R. Koekkoek, and D. van Dijk, *A comparison of biased simulation schemes for stochastic volatility models*, *Quantitative Finance* **10** (2010), no. 2, 177–194.
- [25] R. Lowenstein, *When genius failed: The rise and fall of long-term capital management*, Random House Publishing Group, 2001.
- [26] B. Maslowski, *Stochastic equations and stochastic methods in partial differential equations, (lecture notes)*, *Proceedings of Seminar in Diff. Eq.* (2007), 8–62.
- [27] R. C. Merton, *Theory of rational option pricing*, *Bell Journal of Economics* **4** (1973), no. 1, 141–183.
- [28] G.N. Milstein, *Approximate integration of stochastic differential equations*, *Theory of Probability and its Application* **19** (1974), no. 1, 557–562.
- [29] U. Nögel and S. Mikhailov, *Heston’s stochastic volatility model. implementation, calibration and some extensions*, *WILMOTT Magazine* (2003), 74–79.
- [30] R. Rebonato, *Volatility and correlation: In the pricing of equity, fx and interest-rate options*, Wiley Series in Financial Engineering, Wiley, 1999.

- [31] L. O. Scott, *Option pricing when the variance changes randomly: Theory, estimation, and an application*, *Journal of Financial and Quantitative Analysis* **22** (1987), no. 4, 419–438.
- [32] S. E. Shreve, *Stochastic calculus for finance ii: Continuous-time models*, Springer Finance, Springer Science+Business Media, Inc, 2004.
- [33] J. Stein and E. Stein, *Stock price distributions with stochastic volatility: An analytic approach*, *Review of Financial Studies* **4** (1991), no. 4, 727–752.
- [34] A. van Haastrecht and A. Pelsser, *Efficient, almost exact simulation of the heston stochastic volatility model*, *International Journal of Theoretical and Applied Finance (IJTAF)* **13** (2010), no. 01, 1–43.
- [35] P. Wilmott, *Paul wilmott introduces quantitative finance*, 2 ed., Wiley-Interscience, 2007.