



TAMPERE UNIVERSITY OF TECHNOLOGY

Henri Linnainmaa

Calibration and Implementation of Stochastic Volatility Models for Pricing Autocallable Structures

MASTER OF SCIENCE THESIS

Examiner: Professor Juho Kanniainen
Examiner and topic approved by
the Council of the Faculty of
Business and Built Environment
on 4 September 2013

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY

Master's Degree Programme in Industrial Engineering and Management

Master of Science Thesis, 54 pages

April 2014

Major: Industrial Management

Examiner: Professor Juho Kannainen

Keywords: Stochastic volatility, autocallable structured products, derivatives pricing, model calibration

The goal of this study was to find out what benefits do stochastic volatility models provide over the standard Black-Scholes model in pricing autocallable structured products. Also the practical difficulties of calibrating the models to option data and pricing autocallables using the calibrated models were studied.

The models chosen for this study were the Black-Scholes model and the stochastic volatility models of Heston and Bates. The models were calibrated to market implied volatility data of vanilla options using a simulated annealing calibration algorithm. The volatility surfaces of the S&P 500 index and Neste Oil Oyj were used to test the model calibration. For the pricing task, an autocallable structure called the Neste Oil Autocall was chosen. A Monte-Carlo simulation was used to price the structure for each of the models and for different pricing scenarios. Finally, the pricing accuracy, volatility surface fit and practical performance of each of the models were compared.

We concluded that the Black-Scholes model is not reliable in pricing autocallable structures consistently with the market. The Bates model provided a slightly better implied volatility surface fit than the Heston model, but the calibration was not as fast and stable as the Heston model calibration. The pricing accuracy of the Heston and Bates models was almost identical. Price differences are noticeable only when the shortest maturities affect the payoff of the autocallable.

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO

Tuotantotalouden koulutusohjelma

Diplomityö, 54 sivua

Huhtikuu 2014

Pääaine: Teollisuustalous

Tarkastaja: Professori Juho Kanniainen

Avainsanat: Stokastinen volatiliteteetti, strukturoitu sijoitustuote, johdannaisten hinnoittelu, mallien kalibrointi

Tämän diplomityön tavoitteena oli selvittää, mitä etuja stokastisen volatilitetin mallien käytöstä on verrattuna perinteiseen Black-Scholes -malliin autocallable-tyypistien strukturoitujen sijoitustuotteiden hinnoittelussa. Lisäksi työssä tutkittiin, mitä käytännön haasteita liittyy mallien kalibrointiin markkinadataan sekä mallien käyttoon autocallable-tuotteiden hinnoittelussa.

Tutkittavaksi malleiksi valittiin Black-Scholes -malli sekä Hestonin ja Batesin stokastisen volatilitetin mallit. Mallit kalibroitiin S&P 500-indeksin sekä Neste Oil Oyj:n optioiden markkinadataan käyttäen globaalialia simulated annealing -kalibointialgoritmia. Hinnoittelutesteissä käytettiin Neste Oil Autocall -nimistä strukturoitua sijoitustuotetta. Hinnoittelut toteutettiin Monte Carlo -simulointimenetelmällä usealle eri skenaariolle. Lopulta malleja arvioitiin perustuen niiden hinnoittelutarkkuuteen, kykyyn mallinta markkinoilla havaittu volatilitettipinta sekä mallien käytännön ominaisuuksiin.

Lopputuloksena todettiin, että Black-Scholes -mallia ei voida käyttää luotettavasti autocallable-tuotteiden hinnoittelussa. Batesin malli pystyi kuvaamaan markkinoilla havaitun volatilitettipinnan hieman paremmin kuin Hestonin malli, mutta sen kalibrointi oli hankalampaa ja hitaampaa kuin Hestonin mallin kalibrointi. Molempien stokastisen volatilitetin mallien antamat hinnat olivat hyvin lähellä toisiaan. Hintaeroja havaittiin ainoastaan silloin, kun autocallable-tuotteen hinnan määräytymiseen vaikuttivat kaikkein lyhyimmät maturiteetit.

PREFACE

This thesis was written for the derivatives trading desk in Pohjola Markets/Pohjola Bank Plc while studying in the Industrial Engineering and Management programme of Tampere University of Technology. The subject was chosen to gain understanding of model selection for pricing autocallable structured products. During the study, I also gained important knowledge that will help me continue my work in the field of derivatives pricing.

First, I would like to thank Professor Juho Kannainen for valuable advice and help in writing the thesis and finding the position. I am also grateful to Senior Manager Petri Kangas for giving me insight in structured products and Senior Manager Lauri Tamminen for sharing his expertise in financial engineering. My thanks also go to the whole derivatives trading team for being extremely supportive and making it possible to focus on writing the thesis with minimal distractions.

Table of Contents

1.	Introduction	1
2.	Autocallable structures	3
2.1	Basic properties	3
2.2	Hedging autocallables	5
3.	Models	7
3.1	Black-Scholes model	7
3.2	Heston model	9
3.3	Bates model	11
3.4	Closed-form European option prices for Heston and Bates models . .	13
3.5	The Fourier integral	15
3.6	Choosing the pricing model	16
4.	Model calibration	18
4.1	Calibration methods	18
4.2	Simulated annealing	21
5.	Pricing	23
5.1	Monte Carlo method	23
5.2	Discretization of stochastic processes	23
5.3	Variance reduction techniques	26
5.3.1	Antithetic variates	26
5.3.2	Control variates	27
5.3.3	Low-discrepancy sequences	28

5.4 QuantLib implementation	30
6. Numerical results	33
6.1 Test data	33
6.2 Model parameter estimates and fit to volatility surfaces	35
6.2.1 Calibration to the S&P 500 index options	37
6.2.2 Calibration to the volatility surface of Neste Oil Oyj	42
6.2.3 Out-of-sample volatility surface fit	46
6.3 Pricing results	47
7. Discussion	49
7.1 Conclusion	49
7.2 Further research	50
Bibliography	52

List of definitions

t_0	Current time
T	Time at maturity
S_t	Spot price of an asset at time t
C, P	Price of European call or put option
K	Strike price
μ	Drift of stock S
W_t^S	Standard Wiener process for stock S
r_t	Risk-free rate at time t
q	Dividend yield
σ	Volatility
$V(t)$	Variance at time t
f	Price of any derivative instrument
κ	Mean-reversion of variance
θ	Long-run variance
ξ	Volatility of variance
ρ	Correlation (between W_t^S and W_t^V)
q	Standard Poisson (jump-diffusion) process
J	Jump size of the jump-diffusion process
λ	Jump intensity
\bar{k}	Average jump size
δ	Standard deviation of the jump size

1. Introduction

A model that describes financial asset price developments well is key in pricing financial derivatives. Essentially, a financial model has to meet two requirements. First, it has to realistically describe the price movements of an asset, and second, the prices obtained for derivative instruments have to be free of arbitrage to prevent any mispricing. Combined with interest rate modelling, this is all that is needed to price derivatives of an asset. It is easy to find a model that prices vanilla derivatives consistently with the market, but for more complex derivatives, the price given by different models can vary greatly.

Autocallable structured bonds are a good example of financial instruments that require a precise pricing model. They are heavily path dependent and sensitive to volatility movements. Usually they contain deep in- or out-of-the-money barriers, requiring a good prediction of the probability of unusually big price movements. Moreover, they cannot be replicated with vanilla instruments and usually have no closed-form solution, so they have to be priced separately using some numerical methods such as the Monte Carlo method. These reasons make interesting the problem of pricing autocallable instruments correctly.

The study of stochastic calculus is relatively new in the field of mathematics, focusing on the 20th century. Its practical possibilities have changed the world of financial asset pricing substantially. Since the introduction of the Black-Scholes model in 1973, the methods have come a long way. Countless different models have been suggested, each with their own strengths and weaknesses. The models studied here are stochastic volatility models, which are based on a standard geometric Brownian motion process, but in which the volatility parameter is also stochastic. We focus on two models: The Heston model, which is a ‘standard’ stochastic volatility model and the Bates model, which is a jump-diffusion extension of the Heston model.

This thesis is based on two fundamental research questions. Firstly, we want to see what and how significant benefits do stochastic volatility models offer over the standard Black-Scholes model in pricing autocallable structures. Secondly, we want to find out what practical limitations do the tasks of model calibration and pricing have

and how we can overcome these limitations without compromising the calibration and pricing accuracy.

The thesis is structured in the following way. We will start by presenting the asset class of autocallable structures in chapter two and shortly consider the hedging of autocallables. Next we introduce the models, namely the Black-Scholes model, the Heston model and the Bates model. Also the closed-form solution of European option prices is presented for each of the models.

Chapter four deals with model calibration, briefly discussing calibration methods in general and presenting the main calibration method used, the simulated annealing algorithm. In chapter five, we consider the practical implementation of the pricing engine, dealing with the Monte Carlo method, the discretization of stochastic processes and different variance reduction techniques. The practical implementation is based on a financial mathematics C++ library Quantlib. Finally, we present the numerical results of the model calibration and pricing using sample data and conclude with a short discussion of the results.

2. Autocallable structures

2.1 Basic properties

In general, *autocallable structures* (autocallables) are structured bonds with a distinctive property of non-pretermitted maturity date. If given autocall barrier conditions are met on a observation date, the product will mature immediately, usually paying back the notional amount and cancelling all future coupon payments. The name *autocallable* derives from the asset class of *callable bonds*, for which the issuer has the right, but not the obligation, to buy back the bond with some pretermitted conditions, after which the bond will cease to exist. For autocallable structures, however, the call event is determined only by the price of the underlying, and is hence automatic. The coupon payments are also dependent on some trigger barrier, which can be different from the autocall barrier. Moreover, autocallables usually incorporate a downside risk if the underlying falls significantly.

Autocallables can be divided into *discrete* and *continuous* structures. Discrete autocallables have pre-determined observation dates on which the barrier conditions are observed, and the value of the underlying between the observation dates does not affect the payoff in any way. For continuous autocallables, the observation is done continuously so that the barrier condition is met instantly if the underlying hits the barrier. In practice, this requires a precise definition of the product barrier conditions. Almost all of the autocallables currently available in the market use discrete call dates, and hence will be focused on.

Aside from the possibility of autocall event, there are no strictly defined limits of what kind of rules autocallable structures can have. There are, however, some common conventions which will be presented here. Usually there is a constant *call barrier* B_{call} which triggers the call of the structure on an observation date t_i , if $S_i \geq B_{call}$.

The coupon payments can be constant until the maturity (for example, $x\%$ of the notional) or they can be dependent on some *coupon barrier* B_{coupon} . The coupon

barrier can be the same as the call barrier or lower. For example, if $B_{coupon} = S_0$, the coupon payments are made only if the underlying is at or above its initial price ($S_t \geq S_0$). Cumulating coupons are also possible, sometimes referred to as the snowball variation. For instance if $S_{1,2} < B_{coupon}$ but $S_3 \geq B_{coupon}$, the coupons that were not paid at times 1 and 2 would be paid at time 3 in addition to the 3rd coupon.

Autocallable structures are usually not capital protected, capital protection meaning that the investor is guaranteed to receive at least the initial investment in any scenario. Usually a *protection barrier* $B_{protection}$ is defined, meaning that if the underlying value is below the protection barrier at maturity ($S_T < B_{protection}$), the investor will lose some or all of the invested capital. This can effectively be achieved, for example, by selling a put option to the underlying with maturity T and strike $B_{protection}$. Especially in the current environment of historically low interest rates, it would be difficult to structure completely capital protected products with attractive coupon payments.

Another important factor is the method of calculation for the underlying. The underlying can be a single asset, a basket or an index. All of these are available in the market. Some different kind of rules for calculation of the underlying value are also possible, and can make interesting variations. One example is a *worst-of*-structure, where the ‘underlying value’ is the lowest value of a basket of equities at a given time. This makes the structure quite cheap, but also quite risky. The price of a worst-of -structure is also heavily dependent on the correlation of underlying assets, so that a higher correlation is better for the investor.

An example of the payoff possibilities of a standard autocallable instruments is shown in figure (2.1). The common convention of indexing $S_0 = 100\%$ is used. For the price of the autocallable, the quoting convention is the same as for other structured bonds, so that a price of 100 equals the notional amount.

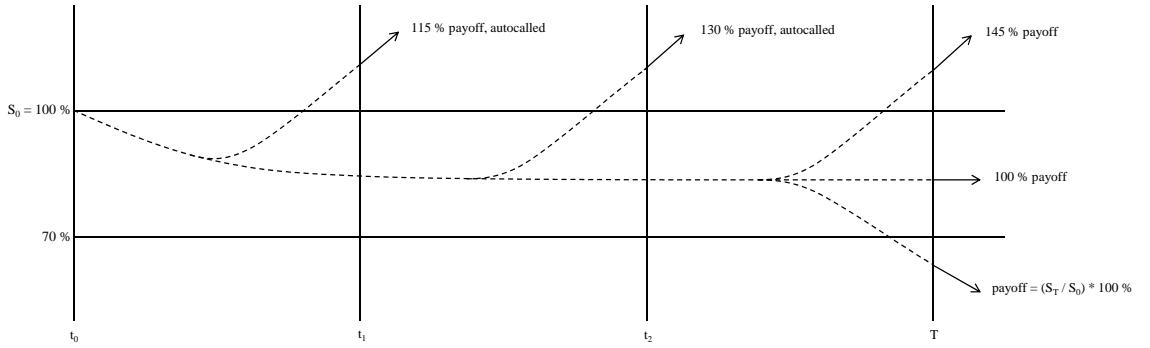


Figure 2.1: Payoffs of an autocallable with $B_{call} = 100$, $B_{coupon} = 100$, $B_{protection} = 70$ and snowball coupons.

2.2 Hedging autocallables

Comparing autocallable structures to ‘traditional’ structured bonds with a predetermined maturity, the hedging of autocallables is a bit more complicated. The traditional way of building a structured bond is buying a (zero-coupon) bond with the same notional as the structured bond, guaranteeing a notional payoff at maturity. The remainder is then used to buy the option. Assuming that the bond is risk-free, this hedge is perfect: the issuer takes no risk and does not need to adjust its hedging portfolio before maturity, unless the structured bond is traded also during its lifetime. It is important to note that the capital protection of the customer is guaranteed only on maturity, not during the contact.

In the case of an autocallable structure, capital protection cannot be achieved with a single bond, because the maturity date is not known beforehand. Let us consider, for simplicity, an autocallable that has two alternative maturity dates, t_1 and t_2 . Intuitively, the hedge would have to be constructed according to the probabilities of maturity on each date. For an autocallable with notional N , one would buy a bond portfolio with $P_{M1}N$ of bond 1 and $P_{M2}N$ of bond 2, where P_{Mi} is the probability of autocallable maturing at time t_i . The portfolio would then have to be adjusted according to the change in probabilities. However, the interest rates are not constant during the period of the contract. In case the interest rates increase and probability of maturity at t_2 grows so that one has to rebalance the position by selling bond 1 and buying bond 2, the rebalancing would result in a net loss. This is because the price of bond 2 would increase more than the price of bond 1.

Also because of the uncertainty of the maturity date, autocallable structures cannot be replicated with vanilla options. The first intuition for replicating an autocallable

would be with digital call options on the underlying for each observation date. A digital call option $C_D(S, N, K, T)$ on underlying S for strike K and expiry T pays the notional amount N if $S \geq K$ at time T , and 0 otherwise. For the most basic type of autocallable that pays $n*C$ and matures at observation date t_n if $S_n \geq B$, this would mean buying a digital call of notional $n*C$ for each observation date. The problem is, however, that if the autocallable is called prematurely (a digital option matures in the money), the options on later maturities should not be in the replicating portfolio anymore. To achieve this, the later options should have a knock-out barrier condition on all of the previous observation dates. Unfortunately, these type of options are not available in the market. We do not know the conditional prices $\{C_D(S, nC, B, t_n) | S_1 < B\}$ of later options at time t_1 either, because we do not know how much below the barrier the underlying will be. Otherwise the conditional payoffs of the autocallable at each observation date would be exactly known.

The pricing of the hedging cost can in practice be done separately and is often easier to implement. After pricing the autocallable using standard risk-neutral procedures, the hedging cost can be independently evaluated using simulated sample underlying paths. For instance if delta hedging is used, the delta of the autocallable is calculated for each hedging interval and then the hedging cost is accumulated over the lifetime of the option. The calculation of the hedging cost involves several assumptions concerning at least the future interest rate levels and the bid-ask spreads for hedging instruments, and would hence require an in-depth study of its own. Therefore the calculation of the hedging cost is left outside this study, but we will instead focus on pricing the option assuming zero hedging cost.

3. Models

3.1 Black-Scholes model

The Black-Scholes model, introduced by Black and Scholes in 1973, is without doubt the most used model in pricing financial derivatives. It is widely known and even though quite simple, it is still applicable for certain types of derivatives. A more thorough explanation of the model is found in almost any book about fundamentals of derivative pricing, so only a brief introduction to the model will be presented here, following Hull (2009).

The Black-Scholes model includes an assumption of constant volatility, in contrast to stochastic volatility models where the volatility of the asset changes through time. The model is used here as a proxy for stochastic volatility models to see in which scenarios the price differences are significant.

A process called the *geometric Brownian motion*, or GBM process, is assumed to be the price process of the underlying in the Black-Scholes model. For a stock that does not pay dividends, the GBM process can be defined as

$$dS_t = \mu S_t dt + \sigma S_t dW_t^S, \quad (3.1)$$

where μ is the drift of the underlying asset σ is the volatility of the asset and dW_t^S is a standard Wiener process. For the Wiener process, $\mathbb{E}(dW_t^S) = 0$ and $\Delta W_t^S = N(0, 1)\sqrt{\Delta t}$. This is the real-world (\mathbb{P} -measure) process that describes the true price evolution of the asset. Similarly, in the risk-neutral probability measure measure \mathbb{Q} , the process is

$$dS_t = r S_t dt + \sigma S_t dW_t^{*S}, \quad (3.2)$$

where r is the risk-free interest rate and dW_t^{*S} is the Wiener process in the risk-neutral probability measure \mathbb{Q} . This process is equivalent to the real-world process (3.1) in pricing options, when the payoffs are discounted with the risk-free rate r .

The relationship between dW_t^S and dW_t^{*S} is

$$dW_t^{*S} = dW_t^S + \frac{\mu - r}{\sigma} t, \quad (3.3)$$

where $\frac{\mu - r}{\sigma}$ is the *market price of risk*.

The most important result of Black and Scholes was, at the time, the Black-Scholes differential equation, which could be used to obtain a closed-form solution for the price of vanilla options. The Black-Scholes differential equation can be obtained using *Itô's lemma* and a riskless portfolio consisting of a derivative instrument and the underlying. Itô's lemma states that if a random variable x follows a GBM process $dx = a(x, t)dt + b(x, t)dZ$, the variable $G(x, t)$ follows the process (Hull 2009):

$$dG = \left(\frac{\partial G}{\partial x} a + \frac{\partial G}{\partial t} + \frac{1}{2} \frac{\partial^2 G}{\partial S^2} b^2 \right) dt + \frac{\partial G}{\partial x} b dZ. \quad (3.4)$$

For the Black-Scholes process, Itô's lemma gives

$$df = \left(\frac{\partial f}{\partial S} \mu S + \frac{\partial f}{\partial t} + \frac{1}{2} \frac{\partial^2 f}{\partial S^2} \sigma^2 S^2 \right) dt + \frac{\partial f}{\partial S} \sigma S dW_t^S, \quad (3.5)$$

where f is the price of any derivative instrument whose value depends on S and t . The riskless portfolio Π consisting of derivative instrument f and underlying S is

$$\Pi = -f + \frac{\partial f}{\partial S} S, \quad (3.6)$$

for which the change of value for a small time interval Δt is given by

$$\Delta\Pi = -\Delta f + \frac{\partial f}{\partial S} \Delta S. \quad (3.7)$$

Discretizing equations (3.1) and (3.5) and inserting them into equation (3.7) gives

$$\Delta\Pi = \left(-\frac{\partial f}{\partial t} - \frac{1}{2} \frac{\partial^2 f}{\partial S^2} \sigma^2 S^2 \right) \Delta t. \quad (3.8)$$

As we can see, the Wiener process dW_t^S has dropped out, so the portfolio is riskless. Assuming absence of arbitrage, the yield of the portfolio must equal the risk-free market interest rate r ,

$$\Delta\Pi = r \Pi \Delta t. \quad (3.9)$$

Finally substituting here equations (3.8) and (3.6) and rearranging, we arrive to the

Black-Scholes differential equation:

$$\frac{\partial f}{\partial t} + r S \frac{\partial f}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} = rf. \quad (3.10)$$

Solving this differential equation gives the price of any single-underlying derivative instrument f , given the assumption that the hedging portfolio can be balanced continuously so that it yields the risk-free interest rate r . For an European call, for example, the equation is solved by setting a boundary condition $f = \max(S - K, 0)$ when $t = T$. The prices of european call and put options according to the Black-Scholes model are (Hull 2009):

$$C = S_0 N(d_1) - K e^{-rT} N(d_2) \quad (3.11)$$

and

$$P = K e^{-rT} N(-d_2) - S_0 N(-d_1), \quad (3.12)$$

where

$$d_1 = \frac{\ln(s_0/K) + (r - \sigma^2/2) T}{\sigma \sqrt{T}} \quad (3.13)$$

and

$$d_2 = d_1 - \sigma \sqrt{T}. \quad (3.14)$$

3.2 Heston model

The Heston model was suggested by Heston (1993) and is one of the best known stochastic volatility models and also widely used. The reason for this is that even though it is quite simple and arguably not as realistic as some other models, it provides a relatively good fit to most volatility surfaces while being computationally efficient due to its closed form solution for European options (Gatheral 2006).

The definition of the model as well as the derivation of the closed-form solution of European options and the Heston characteristic function is presented in several different literary sources. Here we will follow closely to Gatheral (2006), where the definitions and derivations are presented in a clear form.

The Heston process (in the real probability measure \mathbb{P}) is defined as a set of differ-

ential equations

$$dS_t = \mu S_t dt + \sqrt{V_t} S_t dW_t^S \quad (3.15)$$

$$dV_t = \kappa (\theta - V_t) dt + \xi \sqrt{V_t} dW_t^V \quad (3.16)$$

$$dW_t^S dW_t^V = \rho dt, \quad (3.17)$$

where κ is the mean reversion rate of variance, θ the long-run variance and ξ the volatility of volatility.

As it can be seen, the first equation (3.15) is the standard geometric brownian motion, and is the same process that is assumed by Black & Scholes (1973). The difference is that instead of a constant volatility, the volatility is modelled also as a stochastic process (3.16). This process is most commonly called the *CIR process* according to Cox, Ingersoll & Ross (1985). The Wiener processes dW_t^S and dW_t^V are correlated with correlation ρ .

For the risk-neutral process, some of the parameters have to be modified to separate the *market price of volatility risk*. It is the risk premium caused by the volatility of volatility ξ in a similar fashion as the market price of risk in the GBM process, which is caused by σ . The derivation of the change of measure from the real-world \mathbb{P} -measure to risk-neutral \mathbb{Q} -measure is left out, but can be found in detail, for example, in Gatheral (2006). Define $\Phi(S, V, t)$ as the market price of volatility risk. In the risk-neutral measure, κ and θ will be replaced with risk-adjusted versions

$$\kappa^* = \kappa + \Phi \quad \text{and} \quad \theta^* = \frac{\kappa \theta}{\kappa + \Phi}.$$

Now, the Heston process in the risk-neutral probability measure is:

$$dS_t = r S_t dt + \sqrt{V_t} S_t dW_t^{*S} \quad (3.18)$$

$$dV_t = \kappa^* (\theta^* - V_t) dt + \xi \sqrt{V_t} dW_t^{*V} \quad (3.19)$$

$$dW_t^{*S} dW_t^{*V} = \rho dt. \quad (3.20)$$

The Heston parameters have the constraint $2\kappa\theta > \xi^2$, known as the *Feller constraint*, which ensures that the variance process cannot reach zero. This makes the derivation of the closed-form European option prices easier. In practical applications, however, the Feller constraint is often relaxed, as for example in Albrecher et al. (2007) and Andersen (2008).

The Heston model can be effectively converted into a local volatility model by fixing

the correlation parameter ρ to ± 1 . For this parameter choice, the computational evaluation of the model becomes simpler since it has only one dimension of uncertainty ($dW_t^V = \pm dW_t^S$). Usually the price and volatility processes have negative correlation, so this simplification may work well in most cases. It is not, however, as general as the full Heston model, which provides more flexible and thus general dynamics. A related model worth mentioning is the Heston-Nandi model (Heston & Nandi 2000). It is a GARCH (Generalized Autoregressive Conditional Heteroskedasticity) model with discrete time steps. In the continuous-time limit, however, the Heston-Nandi model converges into the Heston model with $\rho = \pm 1$.

3.3 Bates model

The Bates model combines the stochastic volatility of the Heston model with a jump-diffusion process first suggested by Merton (1976). In the literature, it is also sometimes referred to as the Heston-Merton process or just a ‘stochastic volatility with jumps’ model. The process was suggested by Bates (1993). It was originally formulated as a foreign exchange rate process, but can be used for describing stock price movements as well. Probably the most important reason for using jump-diffusions in addition to the stochastic volatility is that the Heston model cannot always be fitted to the near-maturity option volatility smiles well enough. The jump term allows for wider tails of the underlying probability distribution in the short end.

While it is true that the Bates process better describes the real development of underlying prices, this is not yet enough to make the model generally better. The underlying price paths look better to the eye than in the Heston model, but this is practically irrelevant as only the transition probability densities between discrete observation dates matter. Also, the additional flexibility of jumps comes with the price of making the calibration more difficult due to added parameters. Some may claim that the jump process is problematic as the noncontinuous price paths make hedging impossible. Even though theoretically the Heston price path can be continuously delta hedged and Bates path cannot, in practice hedging can never be done continuously. The comparison of models is based on how they fit to the probability distributions of the underlying asset implied by the market, or equivalently, how accurately they can produce the market prices of liquid options.

The presented equations follow Bates (1993) with parameters changed correspondingly with the Heston model. The real measure Bates process is defined as a system

of differential equations

$$\frac{dS_t}{S_t} = (\mu - \lambda \bar{k}) dt + \sqrt{V_t} dW_t^S + k dq \quad (3.21)$$

$$dV_t = \kappa(\theta - V_t) dt + \xi \sqrt{V_t} dW_t^V \quad (3.22)$$

$$dW_t^S dW_t^V = \rho dt \quad (3.23)$$

$$P(dq = 1) = \lambda dt \quad (3.24)$$

$$\ln(1 + k) \sim N\left(\ln(1 + \bar{k}) - \frac{1}{2}\delta^2, \delta^2\right). \quad (3.25)$$

It is similar to the Heston model, with the difference of the additional jump diffusion term in equation (3.21). For the jump diffusion, dq in (3.21) is a Poisson counting process with intensity λ and jump size $J = \ln(1 + k)$. J is normally distributed, the distribution of k being log-normal so that jumps to negative values are not possible.

Transformed to the risk-neutral \mathbb{Q} -measure, the Bates process is

$$\frac{dS_t}{S_t} = (r - \lambda^* \bar{k}^*) dt + \sqrt{V_t} dW_t^{*S} + k^* dq^* \quad (3.26)$$

$$dV_t = \kappa^*(\theta^* - V_t) dt + \xi \sqrt{V_t} dW_t^{*V} \quad (3.27)$$

$$dW_t^{*S} dW_t^{*V} = \rho dt \quad (3.28)$$

$$P(dq^* = 1) = \lambda^* dt \quad (3.29)$$

$$\ln(1 + k^*) \sim N\left(\ln(1 + \bar{k}^*) - \frac{1}{2}\delta^2, \delta^2\right) \quad (3.30)$$

with risk-adjusted κ^* and θ^* as in the risk-neutral Heston process. Similarly as the Heston process was adjusted for the market price of volatility risk, the \mathbb{Q} -measure Bates process needs an adjustment for the *market price of jump risk*, which concerns the parameters λ and \bar{k} . The risk-adjusted λ^* and \bar{k}^* are

$$\lambda^* = \lambda(1 + \bar{k}) \quad \text{and} \quad \bar{k}^* = \bar{k} + \frac{\delta^2}{1 + \bar{k}}.$$

The Bates model does not include a connection between jumps and instantaneous volatility. This is not empirically accurate, as the implied volatility tends to rise after sudden movements in stock price. Gatheral (2006) compares the Bates model to another jump diffusion model (SVJJ) that has simultaneous jumps in the stock price and volatility processes. It would be intuitive to think that a model that better describes the observed stock market behaviour would also provide a better fit to the implied volatility surface. However, Gatheral (2006) notes that the Bates model is much easier to fit to the volatility surface than the SVJJ model. The numerical tests showed better fit to the Bates model, even though it is a special case of the SVJJ

model. This has the implication that the increased amount of parameters makes the calibration of the SVJJ model too difficult.

3.4 Closed-form European option prices for Heston and Bates models

For the Heston and Bates stochastic volatility models, the solution of European option prices is obtained in a similar fashion as in the Black-Scholes model. It is the solution of the *Garman's partial differential equation*, which is obtained by applying the Itô's lemma and standard arbitrage arguments to the stochastic differential equation process in a similar manner as with the Black-Scholes process (Mikhailov & Nögel 2004). The derivation of the Garman's PDE is left out, but its solution will be presented.

The price of a European call option C for underlying S that follows a Heston process can be calculated from

$$C(S_0, K, V_0, t, T) = S_0 P_1 - K e^{-(r-q)(T-t)} P_2, \quad (3.31)$$

where P_1 is the delta of the call option and P_2 the conditional, risk-neutral probability that the call option will mature in the money ($P(S(T) > K)$). This the form used by Mikhailov & Nögel (2004), and a similar formula is used also by Heston (1993) and Bates (1996). Sepp (2003) specifically notes that it is applicable for both of the presented models.

The probabilities P_1 and P_2 can be calculated from

$$P_j = \frac{1}{2} + \frac{1}{\pi} \int_0^\infty \text{Re} \left[\frac{e^{-i u \ln K} \varphi_j(S_0, V_0, t, T, u)}{i u} \right] du, \quad j = 1, 2, \quad (3.32)$$

where φ_1 , φ_2 are the corresponding characteristic functions:

$$\varphi_j(u) = \mathbb{E}[e^{i u \ln S_t}]. \quad (3.33)$$

The integral in (3.32) has to be calculated numerically. The method of numerical integration is described in chapter (3.5).

The difference in the analytic call prices for Heston and Bates models comes from

the difference in the characteristic functions. The characteristic function φ_j for the Heston model, as presented by Mikailov & Nögel (2004), has the form:

$$\varphi_j(S_0, V_0, \tau; u) = \exp(C_j(\tau; u) + D_j(\tau; u) V_0 + i u S_0), \quad \tau = T - t, \quad (3.34)$$

where

$$C_j(\tau, u) = (r - q) u i \tau + \frac{\kappa \theta}{\sigma^2} \left[(b_j - \rho \sigma u i - d) \tau - 2 \ln \left(\frac{1 - g e^{-d \tau}}{1 - g} \right) \right] \quad (3.35)$$

and

$$D_j(\tau, u) = \frac{b_j - \rho \sigma u i - d}{\sigma^2} \frac{1 - e^{-d \tau}}{1 - g e^{-d \tau}}, \quad (3.36)$$

where

$$\begin{aligned} g &= \frac{b_j - \rho \sigma u i - d}{b_j - \rho \sigma u i + d}, \\ d &= \sqrt{(\rho \sigma u i - b_j)^2 - \sigma^2 (2 u_j u i - u^2)}, \\ u_1 &= \frac{1}{2}, \quad u_2 = -\frac{1}{2}, \\ a &= \kappa \theta, \\ b_1 &= \kappa + \lambda - \rho \sigma \quad \text{and} \quad b_2 = \kappa + \lambda. \end{aligned}$$

In fact, the form of Mikailov & Nögel (2004) had d instead of $-d$ in functions C_j (3.35) and D_j (3.36). These are in fact equivalent, because d is a complex square root, which is a multi-valued function. Using $-d$ instead of d is justified by Albrecher et al. (2007) who note that the other form can result in some instabilities in the pricing for certain set of parameters and maturities.

For the Bates model, the characteristic function φ_j has the form (Bates 1996):

$$\varphi_j(S_0, V_0, \tau; u) = \exp(C_j(\tau; u) + D_j(\tau; u) V_0 + i u S_0 + \lambda \tau E(u)). \quad (3.37)$$

It is otherwise identical to the Heston characteristic function (3.34), except for the additional term $\lambda \tau E(u)$. Following Sepp (2003), it can be written as

$$\begin{aligned} E(u) &= \exp((\bar{k} + I_j \delta^2 / 2) i u - \delta^2 u^2 / 2 + I_j (\bar{k} + \delta^2 / 2)) \\ &\quad - 1 - (it + I_j)(e^{\bar{k} + \delta^2 / 2} - 1) \end{aligned} \quad (3.38)$$

with

$$I_j = u_j + \frac{1}{2}.$$

3.5 The Fourier integral

The Fourier integral in the inverse Fourier transform formula (3.32),

$$\int_0^\infty \operatorname{Re} \left[\frac{e^{-i \ln K} \varphi_j(u)}{i u} \right] du,$$

is calculated using the ten-point Gaussian quadrature approximation as done by Sepp (2003). Alternatively a Fast Fourier Transform method could be used, but the direct integration is recommended by Kilin (2006) who notes that its numerical evaluation is faster. The integral is rewritten for the approximation as

$$\int_0^\infty f(k) dk = \lim_{N \rightarrow \infty} \sum_{j=1}^N \int_{(j-1)h}^{jh} f(k) dk. \quad (3.39)$$

The sub-integrals are calculated using the Gauss-Legendre quadrature. The integration is continued until the contribution of the last strip $f(k_j)$ becomes smaller than 10^{-12} . This integration procedure is already implemented in QuantLib. The Gauss-Legendre quadrature approximates the integral over a closed interval as

$$\int_a^b f(k) dk \approx \frac{b-a}{2} \sum_{i=1}^n w_i f\left(\frac{b-a}{2} x_i + \frac{a+b}{2}\right) \quad (3.40)$$

where x_i are the roots of $P_n(x)$, $P_n(x)$ being the n :th Legendre polynomial

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} [(x^2 - 1)^n] \quad (3.41)$$

and w_i are the weights,

$$w_i = \frac{2}{(1 - x_i^2) [P'_n(x_i)]^2}. \quad (3.42)$$

The 10th Legendre polynomial used in the quadrature is

$$P_{10}(x) = \frac{46\,189\,x^{10} - 109\,395\,x^8 + 90\,090\,x^6 - 30\,030\,x^4 + 3\,465\,x^2 - 63}{256}. \quad (3.43)$$

3.6 Choosing the pricing model

Choosing the pricing model for a financial product is always a compromise between its accuracy and practicality. An accurate model describes the price development of the asset well and does not make unnecessary assumptions. In reality, however, it is not possible to take into account all of the factors that influence the asset price. A model must be chosen that takes into account the most important parameters that affect the price of the product.

Autocallables are heavily path-dependent products, meaning that their net present value (NPV) is dependent on assumptions of expected underlying values on several different observation dates. The expected payoffs cannot be valued independently, because the future payoffs are dependent on realization of earlier barrier events. Therefore a small inaccuracy in estimating the transition probability distribution at the first observation date can result in a great error in estimating the future payoffs.

When pricing vanilla options, especially at-the-money (ATM) options with a long maturity, the shape of the probability distribution does not make a big difference. However, for barrier options or out-of-the-money options, it is important for the model to precisely capture the Black-Scholes implied volatility smile. As autocallable products include several barriers which strongly affect the total payoff of the product, the assumption is that the model's capability to reproduce the B-S implied volatility smile is necessary for it to give correct prices. This is the primary requirement for the model.

Assuming absence of hedging costs, the model selection can be based only on the model's ability to produce observed option prices and its computational efficiency. In practice, however, the hedging costs can form a significant part of an asset's price. What else than transition densities affects the hedging costs is the prevailing interest rate and bid-ask spread of hedging instruments. Some assumption of the future interest rates could be made, or the interest rate could even be made stochastic in the model to simulate the hedging cost over time. Also the spread has to be taken into account in some way; it could be assumed constant or dependent on, for example, the instantaneous volatility.

As explained in section (2.2), the effect of interest rates on the hedging cost of an autocallable over its lifetime can be significant. However, the calculation of the hedging cost is problematic compared to pricing the option. The hedging cost is not directly dependent on the price path of the underlying, but rather on the path of the derivative instrument that is hedged, or more specifically the delta of the instrument.

When using a Monte Carlo simulation approach, the delta should be calculated for each discrete hedging date. This would increase the amount of simulation steps that should be calculated and also add the time complexity of the simulation to order N^2 .

Because of the complexity of hedging calculation, we will choose the pricing model regardless of the hedging cost and price the options as if the hedging cost would be zero. The hedging cost can then be separately valued based on some sample realizations of the underlying price path. The specific analysis of pricing the hedging cost is left outside this thesis, as it would require more in-depth study of how the delta should be calculated and what assumptions should be made regarding the interest rates and bid-ask spreads.

What also has to be taken into account in pricing the hedging cost is the correlation between the underlying and interest rates. The underlying value affects the probabilities of maturity on different dates, which requires a rebalancing of the hedging portfolio. This hedging cost is dependent on the interest rates.

For a basic autocallable, a negative correlation between the underlying and interest rates would be best for the issuer, as the longer maturity bonds with greater duration are more sensitive to changes in interest rates. This note was made by Giese (2006). The benefits of a negative correlation can be seen in the following scenario: the price of the underlying falls, which increases the expected duration of the autocallable, as the premature expiration becomes less probable. Now the duration of the bond portfolio would also have to be increased. This would result in a net profit, as the longer maturity bonds that would be bought would be relatively cheaper than the short bonds that would be sold. A rise in the underlying would similarly result in a net hedging profit. On the other hand, a positive correlation between the underlying and interest rates would result in a hedging loss. (Giese 2006)

4. Model calibration

4.1 Calibration methods

The calibration of a model to a given set of data means simply finding a set of parameters for the model that replicate the data as well as possible. A model that simulates the price of a financial asset can be calibrated either to historical data of the asset or current market data of its derivatives.

The advantage of calibration to historical data is that the data is (almost) always available, and can be said to be ‘exact’ in the sense that it describes the true path of the asset. However, the longer back we go in history, the less relevant the data is in representing the current properties of the asset. Using historical data is based on the assumption that the fundamental properties of the asset have not changed significantly during the period. Moreover, it expects that the distribution of historical up- and down-movements describe the expected probability distribution of the asset returns. The sample data has a lot of small movements, but for big movements such as ones that happen once a year, there are only few data points. Therefore, the estimate of the tails of the probability distribution is not necessarily accurate.

On the other hand, current market data on liquid options is believed to represent the current state of the asset reasonably well. The idea is to use a set of financial derivatives of the underlying asset, whose payoff depends on the price of the asset at some time in the future. The market price of the derivatives then imply a probability distribution for the underlying asset. In theory, almost any kind of derivatives could be used, but in practice the calibration is done to vanilla options, as they have enough liquidity for different maturities and strike levels. Option data is available for most stocks, but not necessarily for smaller companies, and in the scale of Finland, possibly not even for some bigger companies.

For multi-asset options, the estimation of each asset’s marginal probability distribution for each maturity is not enough, as the correlation between the assets also affects their joint probability distributions. Strictly speaking, the knowledge of cor-

relation and marginal probabilities is not enough to form a joint distribution, but specific correlation analysis is not considered in this study. It is, however, important to note that for some types of multi-asset options, for example worst-of-type or credit default linked structures, the correlation between assets affects the valuation of the structure greatly. In fact, sometimes the correlation analysis can become the most important factor affecting the valuation. This is why we will focus only on calibration and pricing of single-asset structures.

The calibration to option data is chosen as a primary method for calibration due to its better representation of the asset. This is also the method that the numerical tests will be based on. In case of a multi-asset option, the correlations would be estimated from the historical data. In this study, however, the main focus of the numerical results will be on single-asset autocallables because in multi-asset structures, the correlation plays such an important role that an unbiased comparison of different models would be difficult.

When calibrating prices to an implied volatility surface, the idea is to calculate prices of each option in the surface with initial parameters of the model, check how they correspond to the real option prices, adjust the parameters and repeat until the model replicates the real option values with sufficient accuracy. To measure the error between real and model values, an error function is defined. The problem then becomes a minimization problem of the error function. When using numerical methods in the minimization, the error function can be chosen quite freely. Usually a mean-squared error function between the market and model prices is chosen, as done by [Mikhailov & Nögel \(2004\)](#):

$$\begin{aligned} Err(\Theta) = & \sum_{i=1}^N \sum_{j=1}^M w_{ij} \left[C_{market}(K_i, \tau_j) \right. \\ & - C_{model}(K_i, \tau_j, S(t), fwd_j, df_j, \Theta) \left. \right]^\alpha \\ & + Penalty(\Theta, \Theta_0). \end{aligned} \tag{4.1}$$

Here K_i are strikes, τ_j are maturities, fwd_j and df_j are maturity specific forwards and discount factors, Θ is the vector of model parameters, w_{ij} is a option specific weight factor and α is an even integer. The weighted errors are summed over N strikes and M maturities. Another choice is to define error as the difference between implied volatilities instead of the option prices:

$$Err(\Theta) = \sum_{i=1}^N \sum_{j=1}^M w_{ij} [\sigma_{market} - \sigma_{model}(\Theta)]^\alpha + Penalty(\Theta, \Theta_0). \tag{4.2}$$

Mikhailov & Nögel (2004) also proposed the penalty term $\text{Penalty}(\Theta, \Theta_0)$ to be used in the error function (4.1). The penalty is measured as a difference of the model values and some pre-terminated parameter values, for example, the values obtained in the previous calibration of the same underlying. It is desirable that the parameters would stay relatively stable between consecutive calibrations, instead of finding values that are different but produce a similar volatility surface. The penalty term helps the calibration to stay in reasonable limits and makes the calibration more efficient. However, the penalty term cannot cause any bias in the calibration, so intuitively its derivative with regard to all of the parameters must be 0 in the vicinity of the initial values. Mikhailov & Nögel (2004) suggested a penalty term of the form $|\Theta - \Theta_0|^2$.

Another note they make is that the choice of the weight factors w_{ij} plays a big role in the calibration. They do not, however, make any suggestions regarding the weighting. Chen (2007) gives some thought on the selection of the weighting method, especially the BS-Vega weighting suggested by Cont (2005), which is applied to the calibration to option prices (4.1) rather than volatilities. Nevertheless, Chen finally chooses an uniform weighting where w_{ij} is a constant. Another possibility would be to choose the weighting according to the liquidity or spread of specific options, if available, to give more weight to data points that are more accurate and valid.

The market does not, of course, quote a single ‘right’ price for any option, but instead there is always a spread between bid and ask prices. It is not obvious that the mid-price would best describe the true volatility of the asset. The calibration can either be done to mid-prices or some other price calculated from the bid and ask prices, or alternatively both the bid and ask prices can be included in the error function. Especially for underlyings with relatively low trading volumes, the liquidity and spread of options should be taken into account in the calibration, as their bid-ask spread is greater. The volatility surfaces used in this study will be described in the Numerical results chapter.

To minimize the error function, any suitable minimization algorithm can be used. As the shape of the error function is not known, a local minimization algorithm is not sufficient. Even if the final values of the last calibration would be used as the initial values, the minima can make unexpected shifts in the parameter space. Therefore an efficient global optimization algorithm that suits the calibration of stochastic volatility models must be found. In this study, we chose a calibration algorithm called the *simulated annealing*. Other commonly used optimization algorithms are, for example, the Simplex algorithm (Nelder & Mead 1965), some evolutionary algorithm (e.g. Jin & Branke 2005; Hamida & Cont 2005) or the Levenberg-Marquardt

algorithm (Moré 1978), which is local, but can be combined with some stochastic global algorithm.

It could also be considered to use a local minimization algorithm in daily calibration with previous calibration values as starting parameters. Assuming that there are no big fundamental changes in the underlying, the changes in the error function should be quite small. The full calibration could then be done, for example, weekly.

4.2 Simulated annealing

The global optimization task is not straightforward to execute efficiently but still accurately, especially when the error function is computationally expensive. Global optimization methods have been studied quite extensively, but since different optimization algorithms work better for different kind of minimization tasks, the choice of the optimization algorithm is always, to some extent, a guess. The simulated annealing algorithm chosen for this study is used for calibrating stochastic volatility models for example by Mikhailov & Nögel (2004) and Chen (2007). It was originally presented by Kirkpatrick et al. in 1983. The use of simulated annealing in calibrating stochastic volatility models is usually not very well described in literature. However, for example, Chen (2007) explains the algorithm quite specifically. Chen uses an analogy in a slowly cooling thermodynamic system that can be roughly described with the concepts of *state*, *temperature* and *energy*.

A n-dimensional bounded state space Ω includes all the possible parameter variations, *states*, of n parameters, and a single set of parameters corresponds to one state of the system. The energy (ϵ) of the system is the value of the function that is to be minimized with given parameters. In other words, the goal is to find the state of the system with the lowest possible energy. In simulated annealing, the system can be at only one state at a time and has no memory. The system moves to a randomly chosen *neighbour state* with some probability which is calculated depending on the current temperature and energy difference of current and neighbour states. If the energy of the neighbour state is lower, then the probability of transition is 1. The temperature (T) is the distance (or maximum distance) between the current and neighbouring states. The simulation is started with a large temperature value and then the temperature is reduced gradually, until the temperature goes to 0 and the final, low-energy state is found.

There are several parameters in the simulated annealing process itself that can be

changed. These include starting temperature, cooling speed, the neighbour choosing function and the transition probability function. There is not a general rule to what kind of settings should be used, and the simulated annealing process has to be adjusted according to each object function. For a stable function, for example, a fast cooling schedule can be used, whereas an unpredictably fluctuating function may require a long time to settle to a single low-energy state.

The simulated annealing algorithm used here is based on the described basic principles, but does not follow any specific guides given in the literature or use any external code. It also includes some custom extensions that were proven to increase the stability and convergence of the algorithm. Required starting parameters include the initial values X_0 , parameter limits X_{min} and X_{max} , starting temperature T_0 , up-move probability parameter and amount of iterations. The algorithm is roughly as follows:

1. Calculate current temperature based on the cooling function
2. Randomly draw a new parameter transition vector from the uniform sphere with maximum length = current temperature
3. Calculate neighbour state parameters; if outside the limits, modify them
4. Calculate the energy of the neighbouring state
5. Set $e = e_{neighbour}$ if the new energy is smaller, or if bigger, with some probability depending on temperature and energy difference
6. If the energy is the smallest found so far, save it to a separate variable
7. If the algorithm has not been in the minimum energy state for a defined time (for example 5 % of the remaining time), set $e = e_{min}$
8. Loop until maximum number of iterations is achieved (temperature = 0).

5. Pricing

5.1 Monte Carlo method

The autocallable structures will be priced using Monte Carlo simulation. After calibrating the model, the NPV of the structure is obtained by simulating n paths of the stochastic process and taking the expected value of the discounted payoffs. The hedging cost is not taken into account in the simulation, but will be independently evaluated afterwards.

Even though the Monte Carlo method is relatively simple to implement compared to, for example, the calibration process, there are some choices to be made. First we will present some methods for reducing the variance of the simulation. Variance reduction techniques are mostly independent of the product priced or model used. Second, we will talk about different methods of discretizing the stochastic processes. The performance of discretization methods are model-specific, as they are a part of the stochastic process itself. The better the discretization method, the less amount of time steps have to be simulated between each observation dates. The choice of how many simulations are needed for the NPV to converge with a sufficient accuracy is made based on an analysis of convergence for different variance reduction techniques. The discretization method does not affect the variance of the simulation significantly, but rather causes a bias in the results.

5.2 Discretization of stochastic processes

As an introduction to the discretization of stochastic processes, we will shortly present the *Euler discretization* method. It is the simplest way of estimating the integral of a stochastic process. Following Rouah (2011), the Euler scheme discretizes the Q-measure process $dS_t = r S_t dt + \sigma S_t dW_t$ as

$$S_{t+dt} = S_t + r S_t dt + \sigma S_t \sqrt{dt} Z, \quad (5.1)$$

where Z is a standard normal variable. For the log-price $x_t = \ln S_t$ the corresponding discretization is

$$x_{t+dt} = x_t + (r - \frac{1}{2}\sigma^2) dt + \sigma \sqrt{dt} Z. \quad (5.2)$$

Discretizing the log-price process has the advantage that there is no discretization error when discretizing the constant volatility Black-Scholes process. This is also the discretization that is used in discretizing the Heston stock price process. For the process of stochastic volatility in the Heston model, discretization is done in a similar manner:

$$V_{t+dt} = V_t + \kappa (\theta - V_t) dt + \xi \sqrt{V_t} \sqrt{dt} Z_V. \quad (5.3)$$

The problem with this discretization of stochastic volatility is that the volatility cannot be allowed to become negative. To prevent this, a truncation must be performed. According to Gatheral (2006), the most common truncation methods of Euler discretization are absorption ($V_{truncated} = \max(0, V)$) and reflection ($V_{truncated} = |V|$). However, these truncation methods require a large number of simulations before giving correct results. Gatheral proposes another scheme, the Milstein scheme, to deal with the problem of negative variance. The Milstein discretization scheme uses a higher order Itô-Taylor expansion of V_{t+dt} , where the variance process is discretized as

$$V_{t+dt} = V_t + \kappa (\theta - V_t) dt + \xi \sqrt{V_t} \sqrt{dt} Z + \frac{\xi^2}{4} dt (Z^2 - 1). \quad (5.4)$$

With this discretization, the probability of hitting a negative variance is reduced substantially.

A recent study by Lord et al. (2008) compares different truncations of the Euler scheme, namely absorption, reflection, Higham and Mao truncation, partial truncation and Full Truncation. The Full Truncation scheme is proposed in their paper as a new method. These Euler truncations are also compared to a quasi-second order scheme by Kahl and Jäckel (2006) and the exact scheme of Broadie and Kaya (2006). Lord et al. note that the difference between different Euler truncation schemes is important even for small time steps, even though one might intuitively think that the probability of hitting a negative variance would then become insignificantly small. The results of the paper of Lord et al. state that the Full Truncation scheme is superior to the other compared schemes, as the error compared to an exact simulation in pricing vanilla options is clearly the smallest. It is also simple and computationally very efficient. The Full Truncation scheme uses the log-stock price discretization of equation (5.2) and truncates the variance process in the following way:

$$\tilde{V}_{t+\Delta t} = \tilde{V}_t + \kappa (\theta - \tilde{V}_t^+) \Delta t + \xi \sqrt{\tilde{V}_t^+} \Delta W_t \quad (5.5)$$

and

$$V_t = \tilde{V}_t^+, \quad (5.6)$$

using the notation

$$(x)^+ = \max(x, 0). \quad (5.7)$$

In other words, the variance process in the Full Truncation scheme is a separate process \tilde{V}_t (that can also become negative), and the process V_t used in the stock price process is obtained with \tilde{V}_t^+ for each step. The numerical tests of Lord et al. (2008) strongly support the Full Truncation scheme compared to the other alternatives. They did tests for pricing an at-the-money call option with Heston and Bates models using 20, 40, 80 and 160 time steps per year. For the Heston model, the Full Truncation scheme with Euler discretization of the stock log-price process produced a bias of 0.01 percentage points in volatility for 160 timesteps per year, or a bias of 0.07 percentage points for only 20 timesteps per year. For the Bates model, the bias observed with the Full Truncation scheme was even smaller than in the Heston model. Compared to the bias observed in other truncation schemes, the Full Truncation scheme was clearly superior with a wide margin in all of the test scenarios.

Another recent study by Van Haastrecht & Pelsser (2010) adds to the comparison some truncation methods that are based on the exact method of Broadie and Kaya (2006) but use some approximation of either the Fourier inversions or the distribution of the variance process. The exact method itself, even though theoretically appealing, is not useful in practice due to its computational inefficiency. The schemes compared by Van Haastrecht & Pelsser are the Euler Full Truncation scheme, the Quadratic Exponential (QE-M) scheme of Andersen (2008), the IM-IJK scheme of Kahl & Jäckel (2006), the new Non-central Chi-squared Inversion (NCI-M) scheme presented in their paper, the combined NCI-QE-M scheme and the Drift Interpolation (BK-DI-M) scheme of Broadie and Kaya. These methods are more complicated than the Euler discretizations and will not be presented here. The article of Van Haastrecht & Pelsser (2010) provides a good introduction to these schemes.

The results of Van Haastrecht & Pelsser (2010) concluded that the QE-M scheme was most efficient, followed closely by the other schemes based on the exact method of Broadie and Kaya (2006). They noted that even though the Full Truncation method is appealing in its simplicity, it requires at least 32 time steps per year for the bias to be statistically insignificant. For the Quadratic Exponential scheme, they noted that 8 time steps per year seems to be enough for the bias to be insignificant.

Both the Full Truncation and Quadratic Exponential schemes are supported by our Monte Carlo option pricer. Based on the numerical results of Van Haastrecht & Pelsser (2010), the Quadratic Exponential method will be used in the discretization of Heston and Bates models.

5.3 Variance reduction techniques

Variance reduction techniques are used to improve the computational efficiency of the Monte Carlo simulation. They manipulate the random process in some way to reduce the variance of the process without changing the expected outcome. Increasing the amount of simulations could also be considered a variance reduction technique, but the goal of variance reduction is to reduce the variance without increasing the computational time significantly. The most used variance reductions are compared by Boyle et al. (1997). The variance reduction methods presented here are *antithetic variates*, *control variates* and *low discrepancy sequences*, and will follow the derivations of Boyle et al (1997).

5.3.1 Antithetic variates

Antithetic variates is the most simple variance reduction technique in terms of computation. For a symmetrical random variable Z , it is statistically identical to draw random values from the distribution of Z as it is from the distribution $-Z$. For example, consider a standard Black-Scholes process for which a single price realization can be replicated from

$$S_T^i = S_0 e^{(r - \frac{1}{2}\sigma^2)T + \sigma\sqrt{T}Z_i}. \quad (5.8)$$

For a single price realization $S_T^i = S_T(Z_i)$, the corresponding antithetic realization is $\tilde{S}_T^i = S_T(-Z_i)$. For a option P with a payoff depending only on S_T , the unbiased estimator of the option price without an antithetic simulation is

$$\hat{P} = \frac{1}{n} \sum_{i=1}^n P_i. \quad (5.9)$$

Similarly, for an antithetic simulation, the unbiased estimator is

$$\hat{P}_A = \frac{1}{n} \sum_{i=1}^n \frac{P_i + \tilde{P}_i}{2}. \quad (5.10)$$

For the same n , the antithetic simulation is of course slower. But, for most cases, the antithetic Monte Carlo simulation with n pairs of $(Z_i, -Z_i)$ gives more accurate results than the regular Monte Carlo simulation with $2n$ realizations. This is because the distribution of antithetic pairs $(Z_i, -Z_i)$ is more regular than the distribution of $2n$ independent samples. However, assuming that generating \hat{P}_A takes twice the computational resources than generating \hat{P} , the antithetic simulation surely increases computational efficiency only if $Cov(P_i, \tilde{P}_i) \leq 0$. This is clearly true for options with monotonic price as a function of Z . If this is not the case, then the efficiency of the antithetic simulation must be tested independently for each pricing task.

5.3.2 Control variates

The control variates method relies on using some known value in reducing the variance of an unknown value. In the example of Boyle et al., the price of a geometric Asian option P_G is used as a control variate in pricing a arithmetic Asian option P_A , because the price of the geometric Asian option can be analytically calculated. Consider unbiased estimators \hat{P}_A and \hat{P}_G that are the discounted payoffs for the same simulated path of the underlying asset. In the control variates method, instead of using $P_A = \mathbb{E}(\hat{P}_A)$, the price P_A is expressed as

$$P_A = P_G + \mathbb{E}[\hat{P}_A - \hat{P}_G]. \quad (5.11)$$

Now, the unbiased estimator of P_A is

$$\hat{P}_A^{CV} = \hat{P}_A + (P_G - \hat{P}_G). \quad (5.12)$$

This is not yet optimal. A better result can be achieved by multiplying the term $(P_G - \hat{P}_G)$ with a constant. This is allowed, because its expected value is 0. Considering a family of unbiased control variate estimators

$$\hat{P}_A^\beta = \hat{P}_A + \beta(P_G - \hat{P}_G), \quad (5.13)$$

the variance of \hat{P}_A^β can be expressed as

$$Var[\hat{P}_A^\beta] = Var[\hat{P}_A] + \beta^2 Var[\hat{P}_G] - 2\beta Cov[\hat{P}_A, \hat{P}_G], \quad (5.14)$$

so the variance-minimizing β is

$$\beta^* = \frac{Cov[\hat{P}_A, \hat{P}_G]}{Var[\hat{P}_G]}. \quad (5.15)$$

The estimator in equation X1 might even increase the variance, if the correlation between \hat{P}_G and \hat{P}_G is negative. However, the estimator of equation X2 using optimal β^* is guaranteed not to increase variance, and decreases variance always if \hat{P}_G and \hat{P}_G are correlated. The correlation is not usually known beforehand, and has to be estimated by means of regression analysis. A small amount of simulations is first used to estimate β^* , and then a larger number of simulations is used to estimate P_A with the estimator $\hat{P}_A^{\beta^*}$.

Also, several estimators can be used simultaneously. For example, under risk-neutral probabilities, the current price of the underlying asset, S_0 , must equal $e^{-rT}\mathbb{E}[S_T]$. Therefore, we can use the estimator

$$\hat{P}_A^{\beta_1, \beta_2} = \hat{P}_A + \beta_1(P_G - \hat{P}_G) + \beta_2(S_0 - e^{-rT}S_T). \quad (5.16)$$

The variance-minimizing β_1 and β_2 are similarly determined by regression analysis.

5.3.3 Low-discrepancy sequences

Low-discrepancy sequences are quasi-random sequences that try to produce a sequence of numbers that follow a given distribution as precisely as possible. The discrepancy of a sequence is a measure of how evenly the points are dispersed, or how well the sequence describes a random distribution. Low-discrepancy sequences are not, in fact, random in any way, but deterministic. However, they can be used in a similar way to a ‘true’ random sequence. For example, for a one-dimensional uniform distribution $U(0, 1)$, we could take n values that are evenly distributed in the interval $[0, 1]$. For a Monte Carlo simulation that uses the uniform distribution $U(0, 1)$, the evenly distributed values would probably give a more accurate result than the independent random realizations of $U(0, 1)$. This is quite intuitive, as the result of the Monte Carlo simulation is not affected by the order of realized simulations. However, the problem arises when trying to generate uncorrelated variables in multiple dimensions, which is not trivial at all. A one-dimensional low-discrepancy sequence cannot be used as-is in a Monte Carlo simulation where independent random variables are needed, because the successive low-discrepancy numbers are not independent. A multi-dimensional low-discrepancy sequence must be used. The

measure of discrepancy is not simple, but the idea is that for a n-dimensional low-discrepancy sequence, each of the sub-spaces should be as evenly distributed as possible.

There are several different low-discrepancy sequences proposed by different authors. The sequence that will be considered here is the Sobol sequence, first presented by Sobol in 1967. The generation of the sequence itself is quite technical and will not be presented here. However, some notes about the performance will be made. Several open-source implementations of the Sobol sequence are available, and for example, QuantLib includes a Sobol sequence generator.

Boyle et al. (1997) tested the performance of the Monte Carlo simulation of the Black-Scholes process with Sobol low-discrepancy sequence compared to a simulation with standard, pseudo-random number generator. They got a lower root-mean-squared relative error with a Sobol sequence simulation of 192 points than that they got with a standard simulation of 10,000 Monte Carlo points. Also, it was noted that the generation of Sobol sequence numbers is even faster than most pseudo-random number generators. However, the higher the dimension, the less advantage there is in using a low-discrepancy sequence.

An example of the Monte Carlo price convergence using pseudo- and low-discrepancy sequence generators is presented in figure (5.1). The asset used is a standard auto-callable structure.

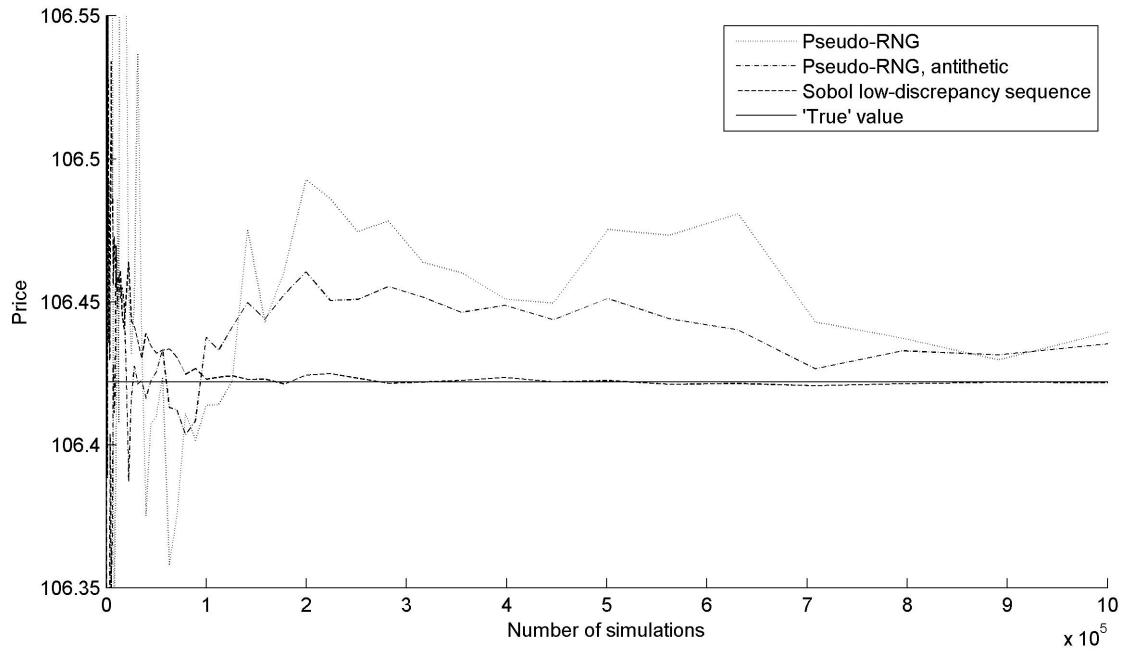


Figure 5.1: Convergence comparison of pseudo- and low-discrepancy random sequence generators used in Monte Carlo pricing of an option.

5.4 QuantLib implementation

QuantLib is an open-source C++ library that provides several kinds tools of quantitative finance. The main focus is in pricing and modelling risk of financial instruments and derivatives. Quantlib code is very high quality and its class hierarchy is designed so that it's easy to expand, even though it might take some time to get familiar with all of the key classes and design patterns used. The first release of QuantLib was in 2000, and since then it has gained popularity inside the quant community. (*QuantLib* website; Firth 2004)

Quantlib offers a solid base to build on, rather than ready-to-use pricing tools for derivatives. The pricing tools for most of the vanilla instruments are quite well refined, but for most exotic derivatives, a custom pricer must be built using the provided base classes. This is the case also with autocallable instruments. For Monte Carlo simulation, QuantLib provides a cleverly built framework that natively supports, for instance, different kind of variance reduction techniques.

The most important classes created for this project and their class hierarchy is presented in the class diagram of figure (5.2).

The **Autocallable** class describes the instrument itself and includes the properties of

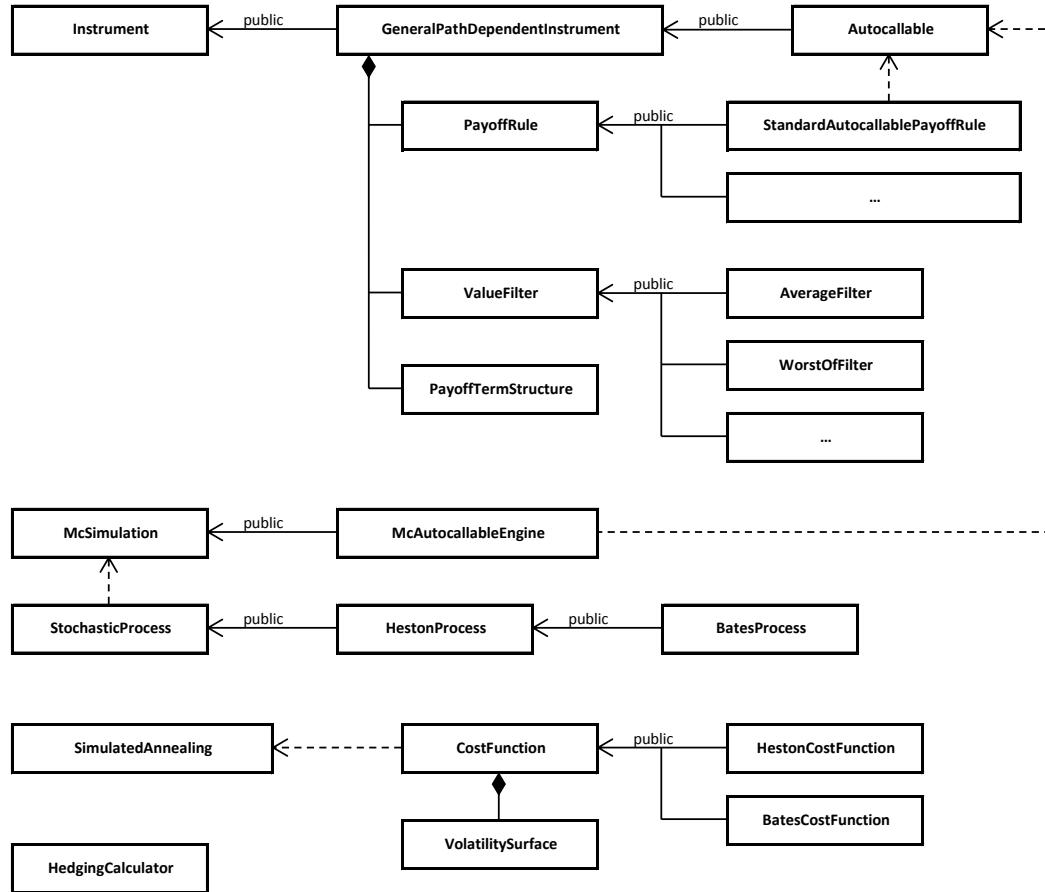


Figure 5.2: Class diagram of the calibration & pricing engine C++ implementation.

the instrument that do not change over time. It is derived from the more general class `GeneralPathDependentInstrument`, which was created to describe any instrument whose payoff depends purely on the price path of the underlying. From this class, different kinds of path dependent instruments can be derived in the future.

The Autocallable instrument contains a pointer to an instance of `PayoffRule`, that defines the payoff structure of the instrument for any price path of the underlying. Several payoff rules can be connected to a single autocallable instrument, if necessary. The `PayoffRule` class is an abstract class, and for the standard autocallable, the payoff rule is implemented in the `StandardAutocallablePayoffRule` class. It defines the most common properties of an autocallable structure, such as different barrier levels and possible snowball property. The implementation of the `StandardAutocallablePayoffRule` class is relatively simple, considering that the payoff is defined only based on the the price path and is not affected by, for example, decisions of the autocall holder.

The abstract class `ValueFilter` defines rules that the underlying values are filtered by, for example in the case of an average or worst-of -type structure. It is connected to a specific payoff rule and a set of underlyings. Specific filter rules, like the classes `AverageFilter` and `WorstOfFilter`, are derived from the `ValueFilter` class.

Finally, the Autocallable class includes a `PayoffTermStructure`, which describes the observation dates and corresponding settlement dates. In practice, the observation dates can be defined in different ways, for instance sometimes the observed value is defined as the average of daily close prices in one week. The `PayoffTermStructure` is also connected to a specific payoff rule.

The calibration of the pricing model is done by the class `SimulatedAnnealing`. It calibrates the model according to given parameters, model specific `CostFunction` and given `VolatilitySurface`. The `CostFunction` class defines the function to be minimized by the simulated annealing algorithm. The `VolatilitySurface` class includes the implied volatility surface to which the model is calibrated. Also some test cost function classes were used to verify the results of the simulated annealing algorithm.

For the pricing, a class `MCAutocallableEngine` is used. It is derived from the abstract base class `McSimulation`, which is included in QuantLib. The autocallable engine is given the underlying process (in this case, a GBM, Heston or Bates process) with specific model parameters, as well as information about simulation parameters such as the random generator, variation reduction techniques used and the amount of samples to be simulated.

6. Numerical results

6.1 Test data

As a test product for the calibration and pricing problems, an autocallable structure called the *Neste Autocall* will be used. It is an autocallable structure whose payoff depends on the stock price of a publicly listed Finnish fuel company Neste Oil Oyj, and is hence a single-asset structure. Even though the product is based on a existing autocallable structure, the issue date is not the same as in the real product. The spot price, volatilities and interest rate curves are of 10 January 2014.

The structure has $B_{autocall} = B_{coupon}$ and has a snowball property (cumulating coupons), so that on the first observation date that $S_t \geq B_{autocall}$, the structure will be autocalled and a redemption of $N + nC$ will be paid, where n is the amount of years from the issue date. As is usual, the structure does not have full capital protection. If it has not autocalled before maturity and $S_T < B_{protection}$, the payoff is S_T (indexed $S_0 = 100$). This is roughly equivalent to selling a down-and-in put option, that is to say that the barrier is observed only at the maturity. The specific properties of the product are presented in table (6.1). The algorithm that defines the payoff depending on the asset price on different observation dates is specified in table (6.2).

The models are first calibrated to European option implied volatility data of the S&P 500 index. This is to test the general ability of the models to fit market data as well as the performance of the calibration algorithm. Then, the models are fitted to the Neste Oil implied volatility surface. The data is provided by Bloomberg and the options in the surface are filtered according to the method described in Cui & Frank (2011). The option prices used are essentially mid-prices of the most liquid options available. Additionally, the Bloomberg option data is filtered so that only the options closest to the barrier dates are used. Calibration is done to the implied volatility rather than option prices. The implied option data is visualised in the corresponding sections.

Table 6.1: The properties of the Neste Autocall used for pricing.

Adjusted S_0	100
Autocall barrier	100
Coupon barrier	100
Coupon type	Cumulating (snowball)
Protection barrier	50
Notional (N)	100
Coupon (C)	7 (semi-annual)
Observation dates (t)	19.8.2014; 19.2.2015; 19.8.2015; 19.2.2016; 19.8.2016; 17.2.2017
Settlement dates (t_S)	26.8.2014; 26.2.2015; 26.8.2015; 26.2.2016; 26.8.2016; 24.2.2017

Table 6.2: The payoff rules of the Neste Autocall.

if $S_1 \geq B_{autocall}$	\rightarrow	payoff = $N + C = 107$	(paid at time t_{S1})
else if $S_2 \geq B_{autocall}$	\rightarrow	payoff = $N + 2C = 114$	(paid at time t_{S2})
else if $S_3 \geq B_{autocall}$	\rightarrow	payoff = $N + 3C = 121$	(paid at time t_{S3})
else if $S_4 \geq B_{autocall}$	\rightarrow	payoff = $N + 4C = 128$	(paid at time t_{S4})
else if $S_5 \geq B_{autocall}$	\rightarrow	payoff = $N + 5C = 135$	(paid at time t_{S5})
else if $S_6 \geq B_{autocall}$	\rightarrow	payoff = $N + 6C = 142$	(paid at time t_{S6})
else if $S_6 \geq B_{protection}$	\rightarrow	payoff = $N = 100$	(paid at time t_{S6})
else ($S_6 < B_{protection}$)	\rightarrow	payoff = S_6	(paid at time t_{S6})

The pricing tests will be done for several different scenarios. The Heston and Bates prices are compared to the Black-Scholes price with two different constant volatilities. The pricing is done at 4 different dates and two different spot prices, so a total of 8 scenarios are priced with each of the models. The reason for this is that the pricing ability during the lifetime of the contract is important if the bond is traded during its lifetime. Also for the Heston and Bates models, the price differences might be especially interesting when the bond is valued close to an observation date.

The discounting yield curve used in the calculations is presented in figure (6.1). It is formed by cubic spline interpolation using EURIBOR, FRA and swap market rates. The product-specific dividend yield curve is constructed using forward prices implied by the at-the-money put-call parity. If forward prices are not available for later maturities, a dividend yield based on Bloomberg analyst estimates will be used.

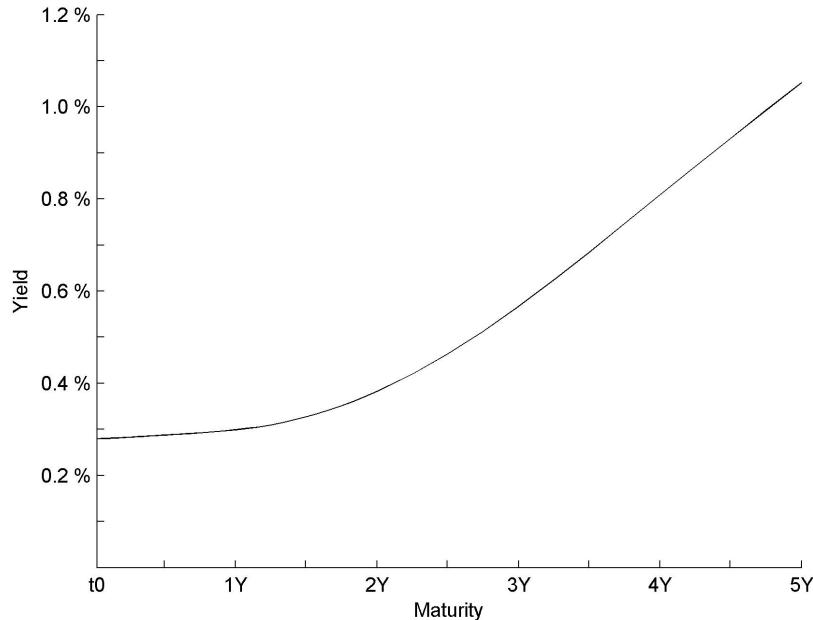


Figure 6.1: The discounting yield curve.

6.2 Model parameter estimates and fit to volatility surfaces

The calibration of Heston and Bates models with the Simulated annealing algorithm is done with the set of calibration parameters shown in table (6.3). The parameter limits and initial values will be the same, except that for the Heston model the

Table 6.3: The parameters used in simulated annealing.

Number of iterations (n)	10 000
Starting temperature (T_0)	0.5
Cooling function	$T_i = T_0 \left(1 - \frac{i}{n}\right)^2$
Transition probability	$\min \left(1.0, \exp \left(-\frac{\Delta\epsilon}{T} \frac{\ln 2}{c}\right)\right)$ with $c = 0.1$
Parameter limits	$V_0 \in [0, 0.5]$, $\kappa \in [0, 6]$, $\theta \in [0, 1.0]$, $\xi \in [0, 3]$, $\rho \in [-1, 1]$, $\lambda \in [0, 1]$, $\bar{k} \in [-1, 1]$, $\delta \in [0, 1]$
Initial values	$V_0 = 0.3$, $\kappa = 2.0$, $\theta = 0.3$, $\xi = 0.5$, $\rho = -0.5$, $\lambda = 0.1$, $\bar{k} = 0.0$, $\delta = 0.1$
Boundary type	Reflecting

jump-diffusion parameters λ , \bar{k} and δ will not be used. The parameter limits are chosen so that they are as tight as possible without excluding the optimal parameter values. This is easy, for instance, for ρ which is always in the range $[-1, 1]$, but for parameters that are unbounded by the model, some guesswork has to be done. If at any time the optimal set of parameters seems to be close to the edges of the optimization space, the limits can be expanded. The transition probability is adjusted so that for $\Delta\epsilon/T = c$, the probability of transition is 0.5. Here $c = 0.1$ was chosen. The lower the temperature, the less probable transitions to higher energy states become. The reflecting boundary means that if the neighbouring state happens to be outside the limits, it is reflected back in for each dimension, for example $\kappa = 5.3 \rightarrow 4.7$.

The error function used in the calibration is

$$Err(\Theta) = \left(\sum_{i=1}^N \left[\frac{1}{N} \left(\sigma_{i,market} - \sigma_{i,model}(\Theta) \right)^2 \right] \right)^{\frac{1}{2}} \quad (6.1)$$

with N being the amount of options in the surface. It is the standard root mean squared error of the difference between market implied and model volatilities with equal weighting. Since option volatility data is available, the choice of calibration to volatilities instead of option prices works better because of discrete dividends.

The volatility surface is smooth, whereas the price data has a discontinuity at the expected dividend date.

To test the calibration algorithm, the calibration is first done to a highly liquid volatility surface, in this case the S&P 500 index, for both Heston and Bates models. This also allows us to compare the general ability of Heston and Bates models to describe the market. After we have confirmed that the calibration algorithm works as required, we fit the models to the volatility surface of Neste Oil Oyj.

6.2.1 Calibration to the S&P 500 index options

The option volatility data of the S&P 500 index was obtained from Bloomberg. The sample contains volatility data of 176 options from 16 strikes and 11 maturities. The method used to construct the volatility surface is described in detail in the Bloomberg documentation (Cui & Frank 2011). It mostly involves filtering out bad data, preserving only strikes and maturities that are liquid enough and have proper bid and ask prices. Also, only out-of-the-money options are used to form the surface. The data is intraday data and thus is not perfectly smooth. The spot value of the index was 1839.48. The volatility smiles of S&P 500 index for chosen maturities at 10 January 2014 is presented in figure (6.2).

The volatility smile of the S&P 500 options for the first maturity date, 17 Jan 2014, is quite steep. The first calibration attempts showed that the Heston model cannot describe the short maturity smile at all. This affects the whole calibration, as the error function is dependent on the square of data point errors, which become quite high for the short maturities. The high volatilities of the short end dominate the calibration and result in a worse fit also for other maturities. The RMSE of the full option data calibration was almost 2 percentage points. For these reasons, the shortest options were left out of the calibration for the Heston model, using options of 22 Feb 2014 as the first maturity.

The calibration algorithm was run 50 times to test its reliability. The calibration algorithm seemed to consistently converge close to the same parameters for the Heston model, implying that the algorithm is reliable.

The best values found by the simulated annealing algorithm for the Heston model were close to the average parameter values over the 50 algorithm runs (within 0.5 standard errors). The best found parameters and the standard errors (SE) of individual parameters are presented in table (6.4).

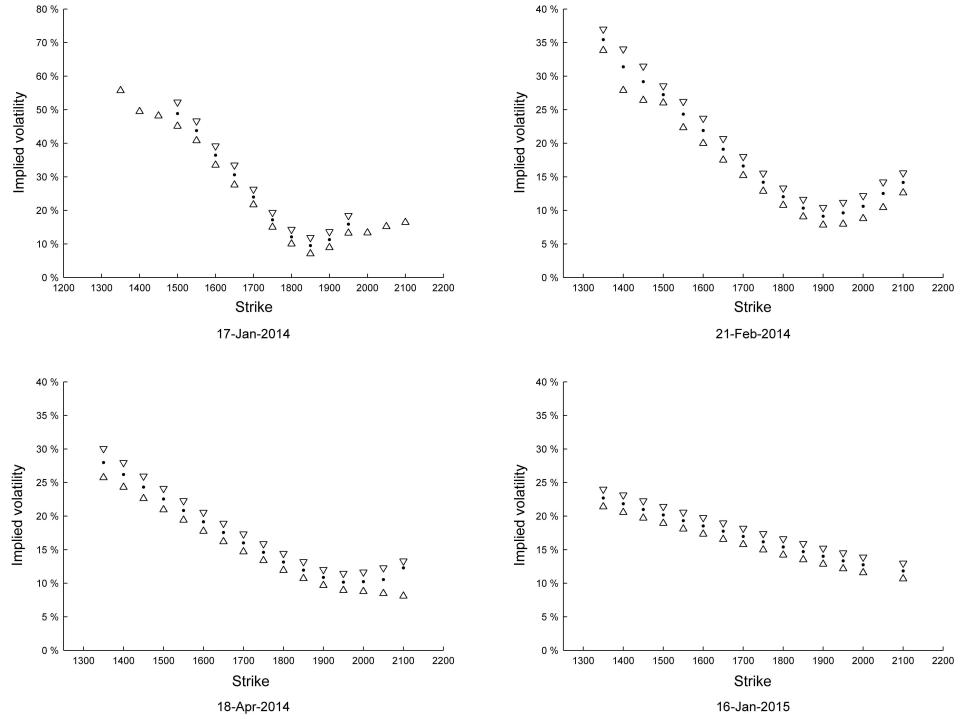


Figure 6.2: Volatilities of the S&P 500 index options at 10 Jan 2014. Note the different scaling of the first maturity. Maturity dates are shown under the graphs.

Table 6.4: Parameters of the S&P 500 Heston calibration.

	V_0	κ	θ	ξ	ρ
Value	0.0080	4.2389	0.0424	1.3214	-0.6270
SE	0.00015	0.88	0.0039	0.12	0.0081

The biggest standard error is seen in the mean reversion of variance and volatility of volatility parameters. This is understandable as they both have quite similar effects on the volatility fluctuation. The root mean squared error of the difference between model and market volatilities was 0.00851 ± 0.00014 , or 85.1 ± 1.4 volatility basis points. It should be noted that the parameters do not satisfy the Feller constraint $2\kappa\theta > \xi^2$. The volatility surface implied by the model, compared to the market volatility surface, is presented in figure (6.3). The errors between model and market implied volatilities for each option in the data is presented in figure (6.4).

For the Bates model, the calibration algorithm did not converge as well as for the

Heston model due to the added complexity of three extra dimensions. With the ten thousand iterations used, the algorithm did not always find to the lowest energy state. With 100 000 iterations the convergence was good, but the calibration algorithm became already quite slow, running close to one minute. The best found parameters of the Bates model calibration, with parameter standard errors, are shown in table (6.5).

Table 6.5: Parameters of the S&P 500 Bates calibration.

	V_0	κ	θ	ξ	ρ	λ	\bar{k}	δ
Value	0.0088	2.3336	0.0399	0.6005	-0.7423	0.0388	-0.1752	0.3018
SE	0.00043	0.22	0.0092	0.029	0.0064	0.024	0.13	0.12

The RMSE of the Bates model calibration was 0.00324 ± 0.00025 , or 32.4 ± 2.5 volatility basis points. This is for calibration to the full volatility surface, including the shortest maturity. In this case, leaving out the first maturity of the option data did not improve the model fit significantly, as it did for the Heston model. It can thus be said that the Bates model fit is good even for very short maturities. The Bates model volatility surface, also compared to the market volatilities, is presented in figure (6.5). A corresponding bar chart of the errors is presented in figure (6.6).

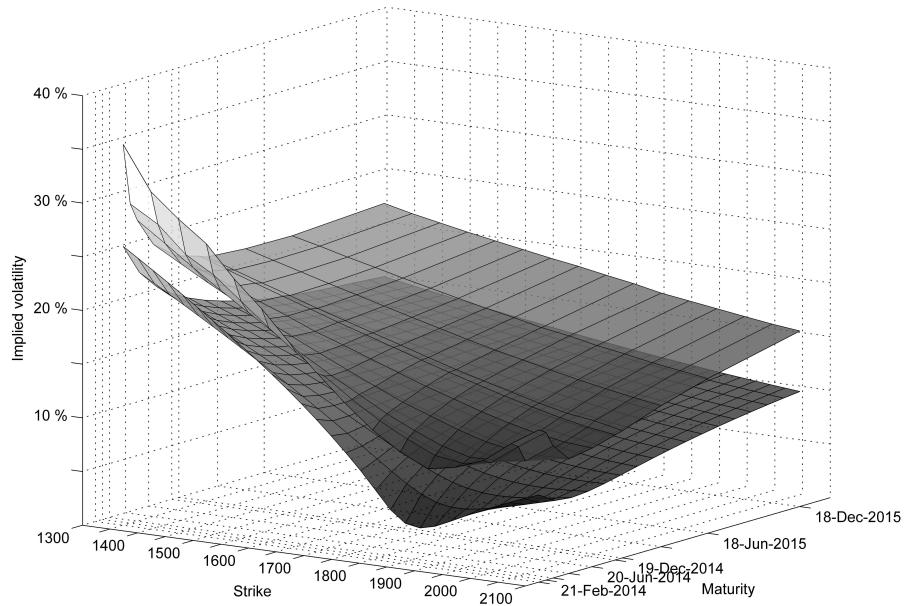


Figure 6.3: Volatility surface of the Heston model calibrated to the S&P 500 index. The Heston surface is shifted 5 percentage points down for the sake of clarity.

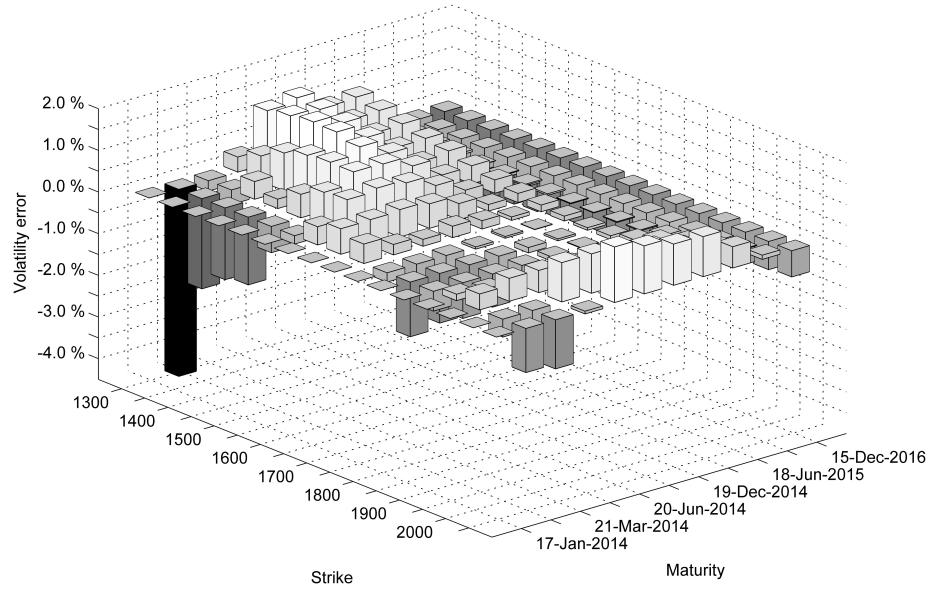


Figure 6.4: The errors between Heston model and market implied volatilities of the S&P 500 index options. The first maturity is not included in the calibration.

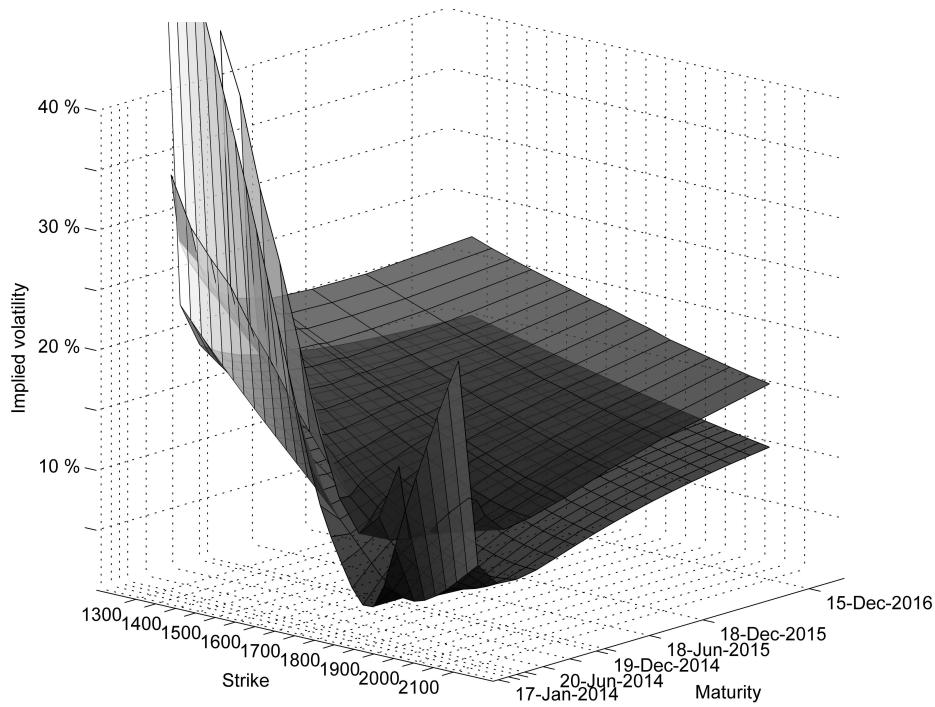


Figure 6.5: Volatility surface of the Bates model calibrated to the S&P 500 index. The Bates surface is shifted 5 percentage points down. Unlike with the Heston model, also the first maturity is calibrated.

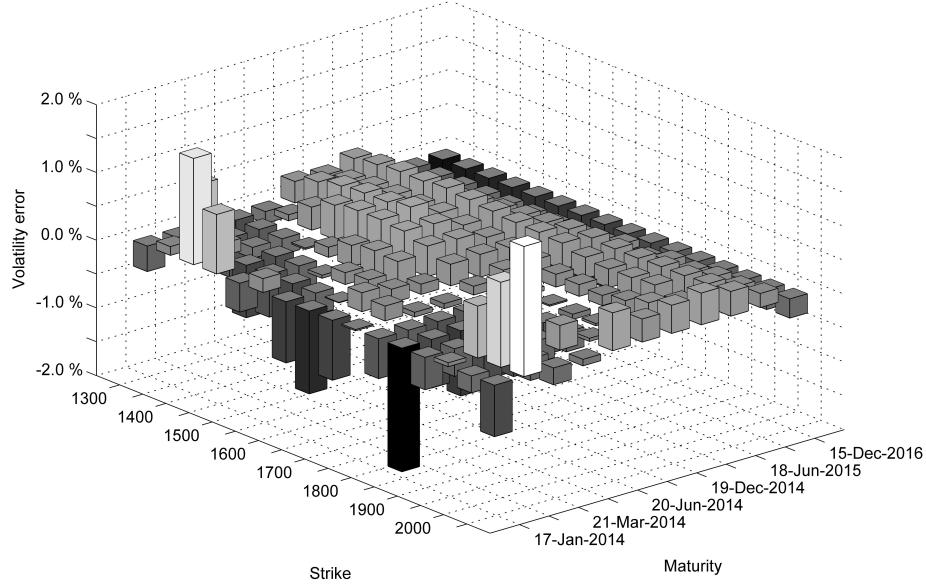


Figure 6.6: The errors between Bates model and market implied volatilities of the S&P 500 index options.

6.2.2 Calibration to the volatility surface of Neste Oil Oyj

The option data for Neste Oil Oyj was obtained from Bloomberg. Similarly as the S&P 500 data, it includes only out-of-the money implied volatilities. There are a total of 49 options from 15 strikes and 5 maturities. The volatility smiles of Neste Oil Oyj options for chosen maturities at 10 January 2014 is presented in figure (6.7).

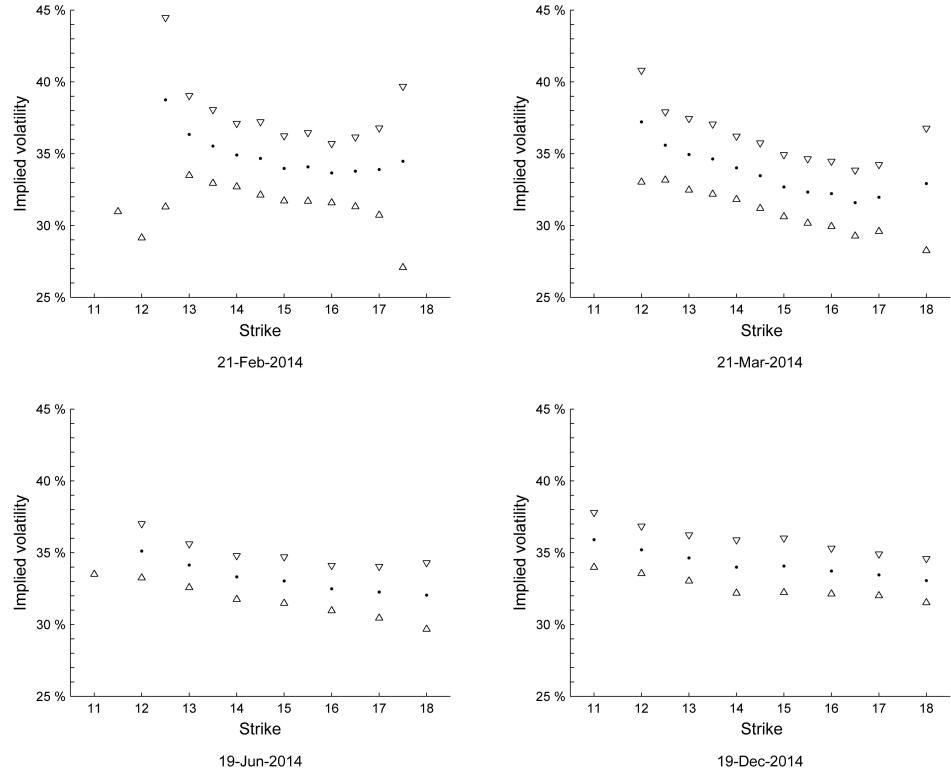


Figure 6.7: Volatilities of Neste Oil options at 10 Jan 2014.

As it can be seen, the spreads are much wider compared to the S&P 500 volatility surface. The short maturity quotes that are far from the at-the-money level have practically no liquidity at all, and thus a similar deep smile to the liquid S&P 500 option data cannot be observed. For this calibration task, the best parameters obtained with parameter standard errors are presented in table (6.6).

Table 6.6: Parameters of the Neste Oil Heston calibration.

	V_0	κ	θ	ξ	ρ
Value	0.1231	4.6340	0.1296	1.3703	-0.2099
SE	0.0016	1.3	0.0043	0.14	0.014

Also in the Neste Oil case, the calibration of the Heston model converged nicely to the same parameters for different calibration runs. The differences in V_0 , θ and ρ between different simulation rounds was very small; the variation was in the 3rd or 4th decimal. κ and ξ showed a little bit more variation, so that their different combinations produced similar values of the error function. They ranged from 4.6 to 4.65 for κ and from 1.35 to 1.40 for ξ .

The RMSE of the Heston calibration to Neste Oil options was 0.01042 ± 0.00047 , or 104.2 ± 4.7 volatility basis points. Considering the noise in the market data, this is an acceptable error. The volatility surface compared to the market volatilities is presented in figure (6.8) and a bar chart of the errors in figure (6.9).

For the Bates model, the calibration was again more problematic than the Heston model calibration. Even though a similar value of the error function was obtained for each calibration round, the difference in parameters was quite large. A correlation was noticeable between volatility process and jump-diffusion parameters: the algorithm could not decide whether to produce the smile using jumps or volatile volatility. When the jump-diffusion parameters λ and δ were larger, the vol-of-vol parameter ξ was smaller and vice versa. After adding the number of calibration steps to 100 000 (ten times the original), the calibration convergence was better, but still not as good as with the Heston model. On the other hand, the importance of stability of the parameters can be questioned - if the model implied volatility surface looks the same, do the exact parameters matter? The best parameters found in the Bates calibration to Neste Oil options are presented in table (6.7).

Table 6.7: Parameters of the Neste Oil Bates calibration.

	V_0	κ	θ	ξ	ρ	λ	\bar{k}	δ
Value	0.1033	4.4024	0.0919	0.6705	0.0149	0.1895	-0.3933	0.0136
SE	0.0090	1.9	0.024	0.37	0.25	0.23	0.27	0.28

The RMSE of the Bates model calibration to Neste Oil options was 0.00906 ± 0.00054 , or 90.6 ± 5.4 volatility basis points. This is a little better compared to the Heston error of 104.2 points. The Bates model volatility surface, also compared to the market volatilities, is presented in figure (6.10), with a error bar chart in figure (6.11).

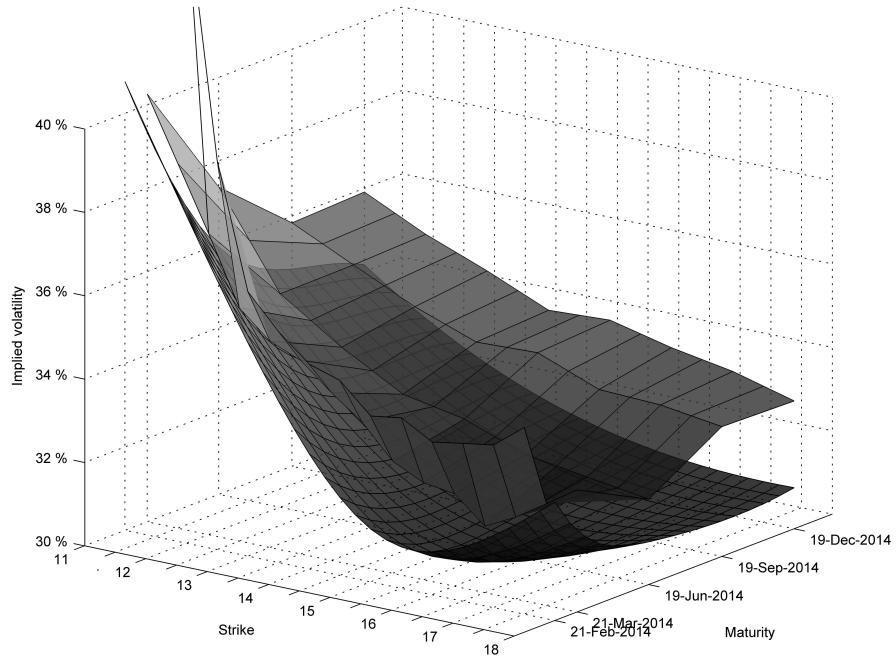


Figure 6.8: Volatility surface of the Heston model calibrated to Neste Oil options. The Heston surface is shifted 5 percentage points down.

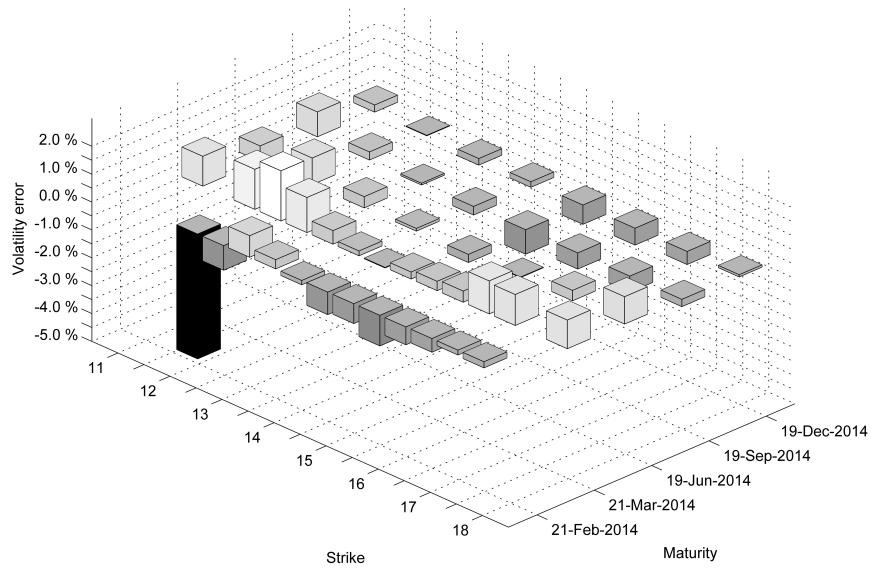


Figure 6.9: The errors between Heston model and market implied volatilities of Neste Oil options.

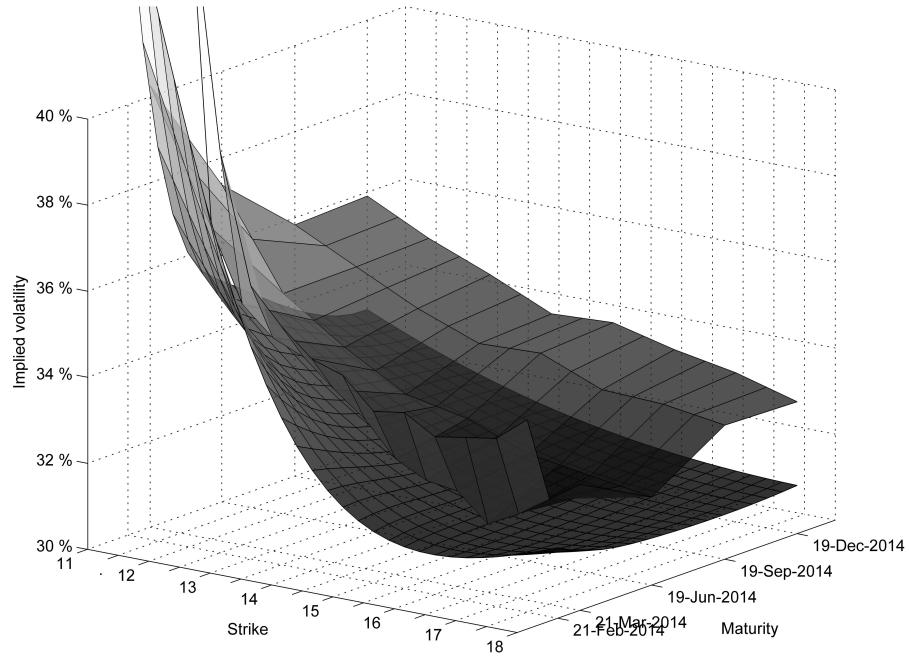


Figure 6.10: Volatility surface of the Bates model calibrated to Neste Oil options. The Bates surface is shifted 5 percentage points down.

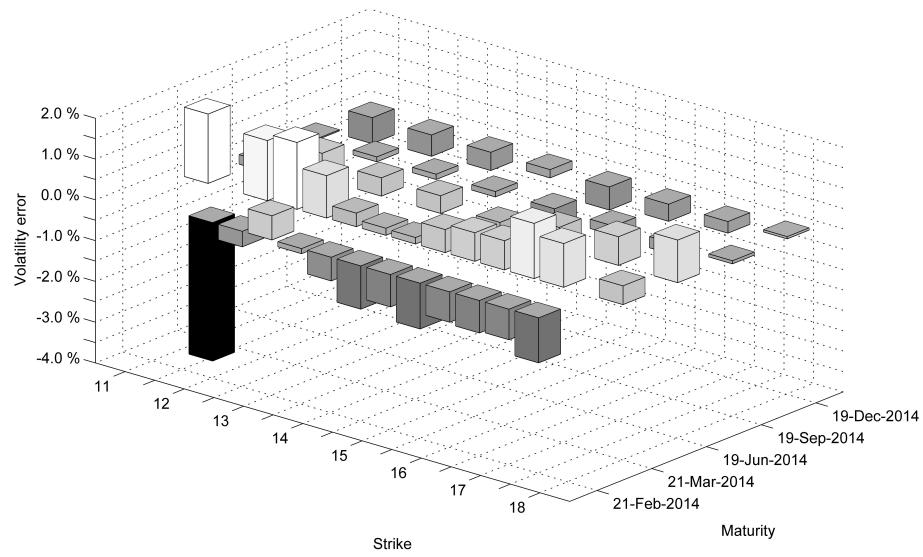


Figure 6.11: The errors between Bates model and market implied volatilities of Neste Oil options.

6.2.3 Out-of-sample volatility surface fit

An out-of-sample test was also performed to test how valid the model parameters are over time. The out-of-sample testing method is used and described, for example, by Bakshi et al. (1997). The idea is that even though a model might fit the data extremely well, it does not yet mean that the model describes the true random process well. For example for a volatility surface of 100 options, one can find a model that can always be fitted to the surface perfectly if the model has at least 100 parameters. This would, however, mean that instead of describing the properties of the asset, the model would be calibrated to the errors of the volatility surface. The idea of the out-of-sample method is to test the performance of the model with a data sample that is not used in the calibration. A model that performs well in the out-of-sample test does not need to be calibrated as often and the parameters are usually more stable over time.

The model given implied volatilities were compared to the volatility surfaces of the next trading day, that is 13 January 2014. All other data was kept unchanged, except for the start date of the implied volatility calculation. The root mean squared error (RMSE) was chosen to describe how well the model fits to the volatility surface. Bakshi et al. (1997) used the relative option price error for this purpose, but here the RMSE was chosen because it was also used in the calibration. For the Heston and Bates models and for the volatility surfaces of Neste Oil Oyj and S&P 500 (SPX), the root mean squared errors for both of the days are shown in table (6.8).

Table 6.8: Root mean squared errors of the in-sample and out-of-sample volatility surface fit (as volatility basis points).

	Heston/SPX	Bates/SPX	Heston/Neste	Bates/Neste
10. Jan 2014	85.1	32.4	104.2	90.6
13. Jan 2014	85.5	35.2	109.0	106.1

Looking at the changes in the RMSEs it can be seen that the Heston model performed better in the out-of-sample test. Especially for the S&P 500 volatility surface, the Heston fit stayed almost as good for the second day as it was for the first day. Still, the Bates fit was better than the Heston model fit. For the Neste Oil surface, the Bates fit worsened even more significantly.

6.3 Pricing results

Finally, the Neste Autocall was priced with the specifications described in section 6.1. For comprehensive comparison, several pricing scenarios were created. The structure was priced with all of the four calibrated models: Heston, Bates and Black-Scholes with two different constant volatilities. The pricing was done at four different dates, 10 January 2014 (issue date), 12 February 2015 (one week before the second observation date), 26 February 2015 (one week after the second observation date) and 20 September 2016 (right after the last observation date before the maturity). The same model parameters, discounting curve and dividend yields were used for each date. Also, the pricing was done with spot prices 100 and 90 for each scenario. This leads to a total of 32 different combinations with four models, four dates and two spot prices.

For the Black-Scholes model, the pricing was done with two different volatilities: the 1-year historical volatility of 40.20 % and the median implied volatility of the surface, 34.09 %. The choice of the median volatility does not have any mathematical basis, but is rather a guess to describe the implied volatilities with one number. It is notable that these volatilities differ significantly. The selection of historical period length does not explain this as the historical volatility is close to 40 % for any historical time range up to at least 10 years. In the implied volatility surface, the liquid volatilities range from 30 % to 39 %. We can thus note that the market does not price the volatility according to past realized volatility. It can also be noted that in the last 6 months, there have been two big daily up-moves unexplained by the normal distribution, other lying 5 and the other 8 standard deviations from the mean. Looking at the history, the option price implied ATM volatility should in fact be even higher than the historical volatility, as the wider tails of the distribution should make the option more expensive. Clearly the market implied volatilities derive from some other source of information.

The prices were obtained by Monte Carlo simulation with 10 million price paths for each model using antithetic variates. This procudes an error estimate of less than 0.01 for each test scenario and model combination. The Heston and Bates models were discretized with the Quadratic Exponential method with 24 time steps per year, based on reasons mentioned in section 5.2. The prices obtained for each of the test scenarios with different models, dates and spot prices are presented in table (6.9). The Heston model prices are absolute prices and other model prices are shown as relative differences compared to the Heston price.

Table 6.9: The pricing results for different models and scenarios.

	Heston	Bates	BS, $\sigma=40.20\%$	BS, $\sigma=34.09\%$
10/1/14, $S_t = 100$	91.17	-0.04 %	-0.86 %	+2.81 %
12/2/15, $S_t = 100$	98.34	+0.11 %	-0.29 %	+1.80 %
26/2/15, $S_t = 100$	88.42	+0.00 %	-0.54 %	+3.47 %
20/9/16, $S_t = 100$	98.67	+0.04 %	-0.60 %	+1.37 %
10/1/14, $S_t = 90$	84.88	-0.02 %	-0.04 %	+3.82 %
12/2/15, $S_t = 90$	85.69	-0.09 %	+0.82 %	+4.52 %
26/2/15, $S_t = 90$	82.54	-0.01 %	+0.28 %	+4.37 %
20/9/16, $S_t = 90$	94.22	-0.07 %	-1.16 %	+1.21 %

The prices given by the calibrated Heston and Bates models are very close to each other. The most significant price difference can be seen at 12 February 2015, one week before the second observation date. Here, the Bates model gives a slightly higher price than the Heston model with $S_t = 100$ but a lower price with $S_t = 90$. After the observation date has passed, the prices for both models are again equal.

The prices obtained from the Black-Scholes model are closer to the Heston and Bates models when using historical volatility of 40.20 %. In this case, the price difference is mostly under 1.0 and it fluctuates above and under the Heston and Bates prices depending on the case. On the other hand, using the median implied volatility of 34.09 % gives prices that are consistently too high. For this volatility, the average difference compared to the Heston and Bates models is 2.5.

The pricing scenarios were chosen to give a comprehensive picture of prices at different parts of the life cycle of the autocallable. As a general note, the prices given by Heston and Bates model do not differ significantly from the perspective of practical asset pricing. Much bigger differences are caused by marginals and hedging cost estimates, for instance. On the other hand, the prices produced by the Black-Scholes model are far from the stochastic volatility model prices. Even if a good guess is made of the Black-Scholes constant volatility, the model either under- or overprices the product depending on the scenario.

7. Discussion

7.1 Conclusion

Based on the numerical results, it can be noted that the prices given by Heston and Bates models are close to each other. In most scenarios the prices are practically the same. Some differences can be seen when close to observation dates but even then the price differences are quite small, being in the magnitude of ten basis points of the price. It can thus be said that the model's ability to describe the short volatility smile is not crucial for obtaining good prices for autocallables. On the other hand, the accurate calibration of the Bates model is quite difficult compared to the Heston model, and from this point of view the Heston model would be a better choice. In addition, according to the out-of-sample test the volatility surface fit of the Heston model seemed to stay better over time than the Bates model fit.

The stochastic volatility models were also compared to the Black & Scholes model. In this case, the price differences were significant. The one-year historical volatility of 40.20 % was found out to describe the prices well on average over different scenarios, but the errors were 50 basis points on average and a maximum of 100 basis points for some scenarios, which is already a significant error even for practical applications. Using the median volatility of the volatility surface as the Black & Scholes constant volatility produced consistently too high prices (250 bp on average) and thus cannot be used. Even though in the Neste Oil case the historical volatility happened to be a good one, it could also be quite far from the truth as it does not react to any fundamental changes in the asset. Therefore the Black & Scholes model cannot be recommended for pricing autocallables.

Looking at the implied volatility surface fit of the Heston and Bates models, it can be noticed that more significant errors are observed in time dimension rather than the strike dimension. Usually the fit is good for a single volatility smile, but fitting all the smiles at the same time seems more difficult. For example, the model given volatilities can be consistently too low for short maturities, too high for middle maturities and again too low for long maturities. This is in fact the case with both

S&P 500 and Neste Oil volatility surfaces and with both models. This problem cannot be solved with a model whose parameters have no time dependency.

7.2 Further research

Even though the volatility fit of the studied models was good, there is still need to explore some extensions to these models. First of all, the volatility fit of a model with time-dependent, piecewise constant parameters should be studied. They can be calibrated to the volatility surface using a bootstrapping algorithm, starting from the shortest maturities and proceeding towards the longer ones. The time-dependent parameters can be incorporated in, for example, the Heston or Bates models. For piecewise constant time-dependent Heston parameters, the vanilla option prices are still known in closed form. The subject is considered in recent studies by, for instance, Elices (2008) and Benhamou et al. (2010).

Also some model that incorporates a stochastic interest rate, such as the Heston model with stochastic interest rates considered by Grzelak & Oosterlee (2011), should be studied. This is because the hedging cost is dependent on the correlation between interest rates and the underlying, or specifically, the delta of the autocallable. Therefore the interest rate can possibly have a significant effect on the price that includes the hedging costs.

A necessary future extension to the pricer is model calibration to historical price data of the underlying. Even though option data calibration is preferred, it is not always available for smaller companies. The calibration of stochastic volatility models to historical data is not straightforward, but fortunately there is some literature available. Especially the maximum likelihood estimation method of Aït-Sahalia & Kimmel (2007) is interesting due to its generality. They have also published a Matlab code that implements their maximum likelihood estimation method. Also a hybrid calibration algorithm that uses both option data and historical prices could be studied. In this case, the objective function is some combination of the option data cost function and the historical data likelihood function.

Finally, another important field of study which is outside the scope of this thesis is the estimation of underlying correlations for multi-asset stochastic volatility models. The model parameters can be calibrated independently for each asset, after which the cross-asset correlations would be estimated. This can be done only by using historical price data, as there is not a lot of publicly traded basket options available.

The size of the correlation matrix depends on the sources of uncertainty in the model. For the Heston model, for example, the full correlation matrix of a 5-asset basket model would be of size 10, with 10 correlation parameters for each model and 90 different cross-correlations. There are some recent studies on this subject available, for instance Szimayer et al. (2009) who studied a multiasset Heston model assuming that only asset-asset cross-correlations are nonzero.

Bibliography

- [1] Aït-Sahalia, Y. & Kimmel, R. 2007. Maximum Likelihood Estimation of Stochastic Volatility Models. *Journal of Financial Economics* 83, pp. 413-452.
- [2] Albrecher, H., Mayer, P., Schoutens, W. & Tistaert, J. 2007. The Little Heston Trap. *Wilmott Magazine*, January, pp. 83-92.
- [3] Andersen, L. 2008. Simple and Efficient Simulation of the Heston Stochastic Volatility Model. *Journal of Computational Finance* 11(3), pp. 1-42.
- [4] Bakshi, G., Cao, C. & Chen, Z. 1997. Empirical Performance of Alternative Option Pricing Models. *The Journal of Finance*, 52(5), pp. 2003-2049.
- [5] Bates, D. S. 1996. Jumps and Stochastic Volatility: Exchange Rate Processes Implicit in Deutsche Mark Options. *Review of financial studies*, 9(1), 69-107.
- [6] Benhamou, E., Gobet, E. & Miri, M. 2010. Time Dependent Heston Model. *SIAM Journal on Financial Mathematics*, 1(1), pp. 289-325.
- [7] Black, F. & Scholes, M. 1973. The Pricing of Options and Corporate Liabilities. *The Journal of Political Economy* 81(3), pp. 637-654.
- [8] Boyle, P., Broadie, M. & Glasserman, P. 1997. Monte Carlo Methods for Security Pricing. *Journal of Economic Dynamics and Control* 21, pp. 1267-1321.
- [9] Broadie, M. & Kaya, Ö. 2006. Exact Simulation of Stochastic Volatility and Other Affine Jump Diffusion Processes. *Operations Research* 54(2), pp. 217-231.
- [10] Chen, B. 2007. Calibration of the heston model with application in derivative pricing and hedging. Master's thesis, Department of Mathematics, Technical University of Delft.
- [11] Cox, J., Ingersoll, J. & Ross, S. 1985. A Theory of the Term Structure of Interest Rates. *Econometrica* 53(2), pp. 358-408.
- [12] Cui, C. & Frank, D. 2011. Equity Implied Volatility Surface. Equities Team, Quantitative Research and Development Bloomberg. Version 3.6.
- [13] Elices, A. 2008. Models with Time-Dependent Parameters Using Transform Methods: Application to Heston's Model. Preprint, ArXiv:0708.2020v2.

- [14] Firth, N. P. 2004. Why Use QuantLib?. Paper available at: <http://www.quantlib.co.uk/publications/quantlib.pdf>
- [15] Gatheral, J. 2006. The Volatility Surface: A Practitioner's Guide. Wiley.
- [16] Giese, A. 2006. On the Pricing of Auto-Callable Equity Structures in the Presence of Stochastic Volatility and Stochastic Interest Rates. MathFinance workshop slides, available at: <http://www.mathfinance.de/workshop/2006/papers/giese/slides.pdf>
- [17] Grzelak, L. & Oosterlee, K. 2011. On the Heston Model with Stochastic Interest Rates. SIAM Journal on Financial Mathematics 2(1), pp. 255-286.
- [18] Hamida, S. B. & Cont, R. 2005. Recovering Volatility from Option Prices by Evolutionary Optimization. Journal of Computational Finance 8(4), pp. 43-76.
- [19] Heston, S. L. 1993. A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options. The Review of Financial Studies 6(2), pp.327-343.
- [20] Heston, S. L. & Nandi, S. 2000. A closed-form GARCH option valuation model. Review of Financial Studies, 13(3), pp. 585-625.
- [21] Hull, J. 2009. Options, Futures, and Other Derivatives, 7th edition. Pearson Education. // check correctness
- [22] Jin, Y. & Branke, J. 2005. Evolutionary Optimization in Uncertain Environments - a Survey. IEEE Transactions on Evolutionary Computation, 9(3), pp. 303-317.
- [23] Kahl, C. & Jäckel, P. 2006. Fast Strong Approximation Monte-Carlo Schemes for Stochastic Volatility Models. Quantitative Finance 6(6), pp. 513-536.
- [24] Kilin, F. 2006. Accelerating the calibration of stochastic volatility models. MPRA paper 2975, University Library of Munich, Germany.
- [25] Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. 1983. Optimization by Simulated Annealing. Science, 220(4598), pp. 671-680.
- [26] Lord, R., Koekkoek, R. & Dijk, D. 2008. A Comparison of Biased Simulation Schemes for Stochastic Volatility Models. Discussion Paper, Tinbergen Institute, Amsterdam.
- [27] Merton, R. C. 1976. Option Pricing when Underlying Stock Returns are Discontinuous. Journal of Financial Economics 3(1), pp. 125-144.

- [28] Moré, J. J. 1978. The Levenberg-Marquardt Algorithm: Implementation and Theory. Numerical analysis, Springer Berlin Heidelberg, pp. 105-116.
- [29] Mikhaliov, S. & Nögel, U. 2004. Heston Stochastic Volatility Model Implementation, Calibration and Some Extensions. Fraunhofer Institute for Industrial Mathematics, Kaiserslautern, Germany.
- [30] Nelder, J. A. & Mead, R. 1965. A Simplex Method for Function Minimization. The Computer Journal, 7(4), pp. 308-313.
- [31] Rouah, F. D. 2011. Euler and Milstein Discretization. Online paper, available at: <http://www.frouah.com>
- [32] Sepp, A. 2003. Fourier Transform for Option Pricing under Affine Jump-Diffusions: An Overview. Online paper, available at: <http://math.ut.ee/~spartak/>
- [33] Szimayer, A., Dimitroff, G. & Lorenz, S. 2009. A Parsimonious Multi-Asset Heston Model: Calibration and Derivative Pricing. Fraunhofer ITWM, Kaiserslautern, Germany.
- [34] Van Haastrecht, A. & Pelsser, A. 2010. Efficient, Almost Exact Simulation of the Heston Stochastic Volatility Model. International Journal of Theoretical and Applied Finance, 13(01), 1-43.