Open Mathematics Open Access

Research Article

Milan Mrázek and Jan Pospíšil*

Calibration and simulation of Heston model

DOI 10.1515/math-2017-0058
Received July 29, 2016; accepted April 10, 2017.

Abstract: We calibrate Heston stochastic volatility model to real market data using several optimization techniques. We compare both global and local optimizers for different weights showing remarkable differences even for data (DAX options) from two consecutive days. We provide a novel calibration procedure that incorporates the usage of approximation formula and outperforms significantly other existing calibration methods.

We test and compare several simulation schemes using the parameters obtained by calibration to real market data. Next to the known schemes (log-Euler, Milstein, QE, Exact scheme, IJK) we introduce also a new method combining the Exact approach and Milstein (E+M) scheme. Test is carried out by pricing European call options by Monte Carlo method. Presented comparisons give an empirical evidence and recommendations what methods should and should not be used and why. We further improve the QE scheme by adapting the antithetic variates technique for variance reduction.

Keywords: Heston model, Stochastic volatility, Option pricing, Monte Carlo simulation, Calibration

MSC: 60H10, 60H35, 65K10, 91G20, 91G60

1 Introduction

Since the introduction of the Black Scholes model in [1] a number of complex models have been proposed to reflect the behaviour of markets and the derivatives. Particularly, the Black Scholes model of option valuation relied on constant volatility, which does not represent the dynamics in the financial markets sufficiently anymore. Relaxing this assumption led to the models called stochastic volatility models.

Among the first publications about the class of stochastic volatility models were Hull and White [2], Scott [3], Stein and Stein [4] and Heston [5]. Heston model was one of the first models that allowed a calibration to real market data using thee semi-closed form solution for European call and put option prices. In Heston model, one cas also consider a correlation between the asset price and the volatility process as for example opposed to Stein and Stein [4].

Heston model is one of the most popular models for option pricing. It can be calibrated using the vanilla option prices and then used to price exotic derivatives for which there is no closed form valuation formula. For this purpose a method for simulating the evolution of variable of interest is necessary. Although the Heston model was already introduced in 1993, the research on discretization methods of its dynamics appeared only very recently. Different Euler schemes implementations were examined and various ways how to deal with negative values of the variance process were suggested, for example Higham and Mao [6] or Lord et. al. [7]. An exact scheme was developed by Broadie and Kaya [8], which simulates the Heston processes from its exact distributions. Higher order Milstein scheme is also available as well as QE scheme, which is the most popular one right now, introduced by Andersen [9].

Milan Mrázek: NTIS-New Technologies for the Information Society, Faculty of Applied Sciences, University of West Bohemia, Univerzitní 8, 306 14, Plzeň, Czech Republic
***Corresponding Author: Jan Pospíšil:** NTIS-New Technologies for the Information Society, Faculty of Applied Sciences, University of West Bohemia, Univerzitní 8, 306 14, Plzeň, Czech Republic, E-mail: honik@kma.zcu.cz

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

The calibration of the model is a crucial process and a price to pay with more complex model is the increased complexity of the calibration process. The industry standard approach is to minimize the difference between the observed prices and the model prices. We can follow for example Mikhailov and Nögel [10], who used the method of adaptive simulated annealing and weights of their fine choice. Unfortunately, it is quite common in the literature that only few data and mostly at the money options are used to demonstrate the calibration process, which of course leads to small errors and it makes the calibration problem easy to solve. Also synthetic option prices with simple parameters are often calculated and used for the calibration. See for example Kienitz and Wetterau [11], where synthetic option prices are calculated and the model parameters are recovered from the calibration process. A significant role is played by the optimizer, because local optimizers heavily depend on the initial guess and thus the distance between the model prices and the market prices does not necessarily have to result from the model, but it can be due to the calibration process.

We demonstrate the complexity of the calibration process and employ variety of optimizers. To overcome the problem with the initial guess we use the global optimizers, in particular adaptive simulated annealing and genetic algorithm. We use different optimizers with a specific approach to calibrate the model to data obtained from the real market, namely we used daily data for DAX Options obtained using the Bloomberg Terminal. We present the results of the calibration for two consecutive days. A combination of a global and local optimizer will be compared to the results obtained by running a large amount of local optimizers stating at different initial points of a deterministic grid. The idea of a deterministic grid was thoroughly tested in the Master's thesis [12] and, in terms of relative error, offers the best results. We show that the combination of a global and local optimizers can provide reasonably good results in much shorter computational times.

Calibration of the Heston model together with the recent approximative fractional stochastic volatility model is briefly described and compared also in the author's papers [13, 14], where only calibration results are presented. In this paper we focus on the Heston model only, for which we deeply compare not only calibration techniques, but also several simulation methods. We further improve the calibration process by also incorporating the approximation formula by Alòs, de Santiago and Vives [15] and the Monte Carlo simulation is then improved by considering antithetic variates in the QE scheme.

It is worth to mention, that although there have been many papers and even books, e.g. recent book by Rouah [16], published about Heston model since the original Heston's paper [5], one can hardly find practical rules of thumb for calibration of the model to real market data and the performance of the simulation techniques for the parameters obtained by the calibration process. The list of techniques for parameter estimation and schemes for simulation mentioned in Chapters 6 and 7 of the book [16] are far from being satisfactory. This paper aims to fill in this gap by presenting several comparisons, giving some recommendations for the optimization techniques and also by showing what simulation methods should and should not be used and why.

The paper is organized as follows. In section 2, we outline the notations used in this paper together with the properties of the Heston model. Calibration of the model to real market data is presented and optimization techniques are compared in section 3. A special attention is given to the combination of global and local optimizers that use the approximation of Heston formula to get the initial guess of the price really quickly. In section 4 we present and compare the simulation schemes and introduce also a new method combining the exact and Milstein (E+M) scheme. We also show how antithetic variates can further improve the QE scheme. We conclude in section 5.

As for the notation used in this paper, it is worth to mention that many formulas are presented in the notation used in the original cited papers. Although this may seem as a little inconsistency in the notation in this paper, we believe that it can help further investigation of studied formulas and schemes within the cited materials.

2 Heston model

Following the original paper by Heston [5] and the recent book by Rouah [16], we consider the following risk-neutral stock price model

$$dS(t) = rS(t)dt + \sqrt{v(t)}S(t)d\widetilde{W}^S(t), \tag{1}$$

$$dv(t) = \kappa[\theta - v(t)]dt + \sigma\sqrt{v(t)}d\widetilde{W}^v(t), \tag{2}$$

$$d\widetilde{W}^S(t)d\widetilde{W}^v(t) = \rho\, dt, \tag{3}$$

with initial conditions

$$S(0) = S_0 \geq 0, \tag{4}$$

$$v(0) = v_0 \geq 0, \tag{5}$$

where $S(t)$ represents the price of the asset at time t, $v(t)$ is the instantaneous variance at time t, r is the risk-neutral rate of return, θ is the long run average variance (as t tends to infinity, the expected value of $v(t)$ tends to θ), κ is the rate at which $v(t)$ reverts to θ and σ is the volatility of the variance. Processes $\widetilde{W}^S$ and $\widetilde{W}^v$ are two Wiener processes under the risk-neutral measure $\widetilde{\mathbb{P}}$ that are correlated with instantaneous correlation ρ.

Stochastic process $v(t)$ is referred to as the variance process (it is also called a volatility process and the parameter σ then volatility of volatility) and it is the CIR process, the square-root mean reverting process introduced by Cox, Ingersoll and Ross [17]. It is always positive and cannot be zero or negative if the Feller [18] condition $2\kappa\theta > \sigma^2$ is satisfied.

The main parameters of interest in the Heston Model are $v_0, \kappa, \theta, \sigma$ and ρ. High values of κ essentially mean a higher speed of the reversion of the process toward its long-run mean θ. Obviously, an increase in θ increases the prices of options. The parameter ρ affects the skewness of the returns distribution and hence the skewness in the implied volatility smile [5]. Negative values of ρ induce a negative skewness in the returns distribution since lower returns will be accompanied by higher volatility which will stretch the left tail of the distribution. The reverse is true for positive correlation. The parameter σ affects the kurtosis of the returns distribution and hence the steepness of the implied volatility smile. If κ is not too large, large values of σ cause more fluctuation in the volatility process and hence stretch the tails of the returns distribution in both directions.

A European call option maturing at time T with strike price K can be expressed as

$$c(S, v, t) = e^{-r\tau}E[(S(T) - K)^+] = e^x P_1(x, v, \tau) - e^{-r\tau}K\,P_2(x, v, \tau), \tag{6}$$

where $x = \ln S$ and $\tau = T - t$ is the time to expiration. The first term represents the present value of the underlying given an optimal exercise and the second term is the present value of the strike price payment. Moreover, P_1 is the delta of the European call option and P_2 is the conditional risk neutral probability that the asset price will be greater then K at the maturity. Both P_j, $j = 1, 2$, represent the conditional probability of the call option expiring in-the-money given the value $x = x(t)$ of the stock and the value $v = v(t)$ of the volatility at time t. Heston [5] considers the characteristic functions of these probabilities in the form

$$f_j(x, v, t; \phi) = \exp\{C(\tau; \phi) + D(\tau; \phi)v + i\phi x\}, \tag{7}$$

and shows that

$$C(\tau;\phi) = r\phi i\tau + \frac{a}{\sigma^2}\left\{(b_j - \rho\sigma\phi i + d)\tau - 2\ln\left[\frac{1 - ge^{d\tau}}{1 - g}\right]\right\}, \tag{8}$$

$$D(\tau;\phi) = \frac{b_j - \rho\sigma\phi i + d}{\sigma^2}\left[\frac{1 - e^{d\tau}}{1 - ge^{d\tau}}\right], \tag{9}$$

where

$$g = \frac{b_j - \rho\sigma\phi i + d}{b_j - \rho\sigma\phi i - d}, \tag{10}$$

$$d = \sqrt{(\rho\sigma\phi i - b_j)^2 - \sigma^2(2u_j\phi i - \phi^2)}, \tag{11}$$

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

$$P_j(x, v, T; \ln[K]) = \frac{1}{2} + \frac{1}{\pi}\int_0^\infty \mathrm{Re}\left[\frac{e^{-i\phi\ln[K]} f_j(x, v, T; \phi)}{i\phi}\right] d\phi. \tag{12}$$

The equations (6), (7) and (12) give the solution for European call options as first introduced by Heston in [5]. The integrand in equation (12) is a smooth function that decays rapidly and presents no difficulties as was pointed out also in [5].

2.1 The little Heston trap

The characteristic function solution (7) was later the subject of the paper called "The little Heston trap" by Albrecher et al. [19], who investigated in detail the properties and relations between the original formulation given by Heston and other formulation presented for example in Gatheral [20], which is identical to the Heston formulation, but causes fewer numerical problems in the implementation of the model. Change of term (10) is desired and the new form is

$$g' = \frac{b_j - \rho\sigma\phi i - d}{b_j - \rho\sigma\phi i + d}, \tag{13}$$

as well as change of formulation of terms $C(\tau; \phi)$ given by (8) and $D(\tau; \phi)$ given by (9). The new formulations read

$$C(\tau; \phi) = r\phi i\tau + \frac{a}{\sigma^2}\left\{(b_j - \rho\sigma\phi i - d)\tau - 2\ln\left[\frac{1 - ge^{-d\tau}}{1 - g}\right]\right\}, \tag{14}$$

$$D(\tau; \phi) = \frac{b_j - \rho\sigma\phi i - d}{\sigma^2}\left[\frac{1 - e^{-d\tau}}{1 - ge^{-d\tau}}\right]. \tag{15}$$

These changes produce an equivalent formulation to the original Heston formulation, but regarding properties of both formulations Albrecher et al. [19] provided proofs that the Heston formulation is unstable under certain conditions and the alternative formulation is stable for all considered parameter ranges. They also established a threshold maturity from which the Heston formulation suffers from instability. When the Feller condition is exactly satisfied, they encountered no problems in any version. The paper of Kahl and Jäckel [21] also concerns itself with this problem and uses the rotation count algorithm to overcome the discontinuities in the original Heston formulation.

2.2 Lewis formula

In the book by Lewis [22], the author presents the so called fundamental transform approach for option valuation. The pricing formula by Lewis is well-behaved compared to the formula by Albrecher [19] and it has a numerical advantage in the sense that we have to calculate only one numerical integral for each price of the option. A price of the European call option can be expressed as

$$V(S, v, \tau) = S - K\, e^{-r\tau}\frac{1}{\pi}\int_{0+i/2}^{+\infty+i/2} \underbrace{e^{-ikX}\frac{\hat{H}(k, v, \tau)}{k^2 - ik}}_{f(k)} dk, \tag{16}$$

where $X = \ln(S/K) + r\tau$ and

$$\hat{H}(k, v, \tau) = \exp\left(\frac{2\kappa\theta}{\sigma^2}\left[q\, g - \ln\left(\frac{1 - he^{-\xi q}}{1 - h}\right)\right] + vg\left(\frac{1 - e^{-\xi q}}{1 - he^{-\xi q}}\right)\right),$$

where

$$
\begin{aligned}
g &= \frac{b-\xi}{2}, \quad h = \frac{b-\xi}{b+\xi}, \quad q = \frac{\sigma^2\tau}{2},\\
\xi &= \sqrt{b^2 + \frac{4(k^2 - ik)}{\sigma^2}},\\
b &= \frac{2}{\sigma^2}\Big(ik\rho\sigma + \kappa\Big).
\end{aligned}
$$

To show that formulas (6) and (16) are equivalent, we refer to the recent paper by Baustian et. al. [23], where authors also extended the Lewis's approach to models with jumps. Yet another formula was later obtained by Attari [24], who uses the Euler's identity to represent separately the real and imaginary part of the integrand that in Attari's representation decays faster than in the original Heston's (or Albrecher's) case. Since Attari's version does not bring any extra advantage over the Lewis formula, and the Lewis approach is more general, we further omit the Attari formula from our tests.

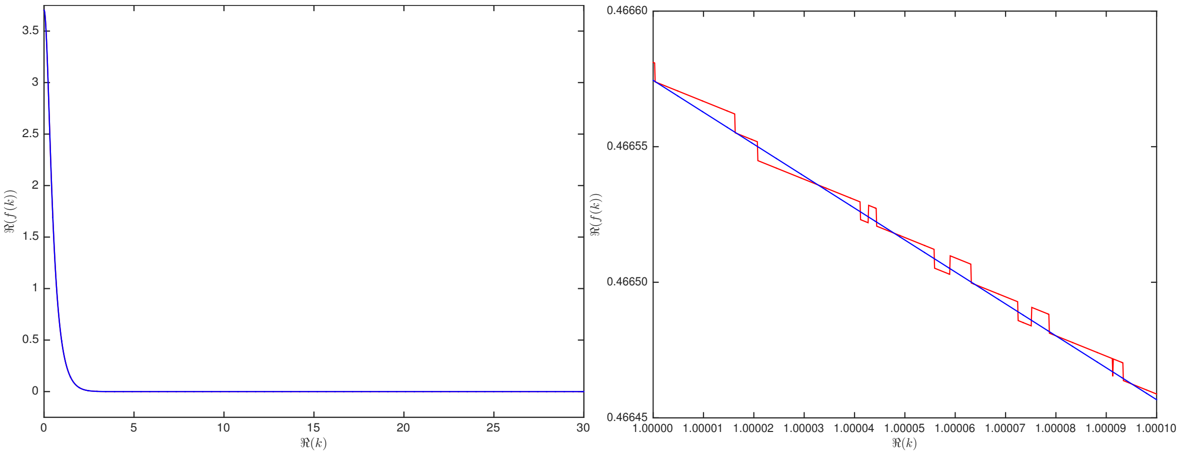
All pricing formulas that are expressed as the inverse Fourier transform integral must be evaluated numerically with caution as is pointed out by Daněk and Pospíšil in [25]. They showed that inaccurately evaluated integrand causes serious problems to adaptive numerical quadratures that consequently can lead to significant differences in the option price, even in the order of hundreds of dollars. To overcome the problems with inaccurately evaluated integrands, Daněk and Pospíšil suggest to use the high precision arithmetic in problematic cases. In MATLAB, there exists a possibility of defining variables and perform numerical calculations in high precision arithmetic using the `vpa` (variable precision arithmetic) bundle that is part of the Symbolic Math Toolbox. In [25] authors provided a fast switching regime algorithm between the standard `double` arithmetic and the high precision `vpa` arithmetic. If not stated otherwise, we use this algorithm in the implementation of the Lewis formula (16).

In Figure 1 we consider market data (ODAX, March 18, 2013, see also section 3.4) and parameter values that were obtained in one of the upcoming calibration trials:

$$
\begin{aligned}
(\tau, K, r, S) &= (1.75616, 7400, 0.00206, 7953.7),\\
(v_0, \kappa, \theta, \sigma, \rho) &= (0.08316721, 33.10986083, 0.53084828, 0.00009564, -0.76040362).
\end{aligned}
$$

Although the global view of the integrand $f(k)$ does not indicate any troubles, a more detailed analysis reveals the unexpected discontinuities in the integrand that should be smooth. For more details about this phenomenon we refer the reader to [25].

Fig. 1. Global and local view to the integrand $f(k)$ in (16). The inaccurately enumerated values were obtained using `double` precision arithmetic (plotted in red), the smooth values are in `vpa` (blue). $\Re(z)$ denotes the real part of the complex variable z.



2.3 Approximation formulas

Numerical evaluation of the inverse Fourier transform integral in (12) or (16) can be rather time consuming, which is most vivid in the calibration process that can require thousands of such evaluations. To find a suitable fast approximation is therefore of interest. Instead of the characteristic function solution, one can use for example the so called Edgeworth expansion of the European call option price (6), where probabilities P_1 and P_2 are approximated using the so called Sartorelli approximations, see Ribeiro and Poulsen [26]. This approach leads to a fast method for option pricing (according to Ribeiro and Poulsen a seventh order expansion is about three times faster than the Heston integration and it is working reasonably well in the range 80-120% moneyness).

Another approach to approximation was used by Forde and Jacquier, who derived both a large-maturity approximation [27] as well as a small-maturity expansion formula [28]. The large-maturity approximation is derived using the tools from large deviations theory and it provides similar results as the volatility of volatility series expansion derived also by Lewis [22]. The small-maturity approximation formula is then derived using the so called saddle point expansion and although the approximation is reasonable good, its usage is rather impractical, because it requires a numerical solution of nonlinear equations that is rather time consuming (speed-up factor is not so significant comparing to the numerical integration in above mentioned formulas).

Among all approximation formulas for the Heston model, we would like to highlight the formula obtained by Alòs, de Santiago and Vives [15]. The approximation of the Heston European call price is

$$\hat{V}_t := BS(t, X_t, \tilde{v}_t) + H(t, X_t, \tilde{v}_t)U_t + K(t, X_t, \tilde{v}_t)R_t, \tag{17}$$

where $BS(t, x, \sigma)$ denotes the Black-Scholes price at t of a European call option with constant volatility σ, current log-stock price x, time to maturity $\tau = T - t$, strike price K and interest rate r, $X_t = \ln(S_t/K)$, $D_t = \exp(-\kappa(T-t))$,

$$\tilde{v}_t = \sqrt{\frac{1}{T-t}\int_t^T \mathbb{E}_t[v_s^2]ds} = \sqrt{\theta + \frac{v_t^2 - \theta}{\kappa\tau}(1 - D_t)}$$

denotes the future average volatility,

$$\begin{aligned}
U_t &= \frac{\rho\sigma}{2\kappa^2}\left\{\theta\kappa(T-t) - 2\theta + v_t^2 + D_t\left(2\theta - v_t^2\right) - \kappa\,(T-t)\,D_t(v_t^2 - \theta)\right\},\\
R_t &= \frac{\sigma^2}{8\kappa^2}\left\{\theta(T-t) + \frac{v_t^2 - \theta}{\kappa}(1 - D_t) - \frac{2\theta}{\kappa}(1 - D_t) - 2(v_t^2 - \theta)(T-t)D_t\right.\\
&\qquad\left. + \frac{\theta}{2\kappa}(1 - D_t^2) + \frac{v_t^2 - \theta}{\kappa}(D_t - D_t^2)\right\},\\
H(t, x, \sigma) &= \frac{e^x}{\sigma\sqrt{2\pi(T-t)}}\exp\left(-\frac{d_+^2(\sigma)}{2}\right)\left(1 - \frac{d_+(\sigma)}{\sigma\sqrt{T-t}}\right),\\
K(t, x, \sigma) &= \frac{e^x}{\sigma\sqrt{2\pi(T-t)}}\exp\left(-\frac{d_+^2(\sigma)}{2}\right)\frac{d_+^2(\sigma) - \sigma d_+(\sigma)\sqrt{T-t} - 1}{\sigma^2(T-t)},\\
d_+(\sigma) &= \frac{1}{\sigma\sqrt{T-t}}\left[X_t + \left(r + \frac{\sigma^2}{2}\right)(T-t)\right],\\
d_-(\sigma) &= d_+(\sigma) - \sigma\sqrt{T-t}.
\end{aligned}$$

Alòs, de Santiago and Vives showed that the approximation is especially good for small σ, in particular that the error is of order $O(\sigma^2(\rho + \sigma))$. The biggest advantage of this approximation is however in its speed of calculation which makes it suitable for fast pre-calibration that we plan to show later on.

We would like to close this section by mentioning also the approximation by the usage of fast Fourier transform (FFT) as was suggested by Carr and Madan [29]. Although the FFT methods, including the fractional FFT modification [30, 31], cosine FFT method [32] or methods based on wavelets [33], are fast in calculating an approximation of the inverse Fourier transform integral in many discrete points at once, however, in option pricing problem many values are calculated redundantly and, moreover, with relatively low precision that should be in modern financial applications considered unsatisfactory. Heston model in particular was studied from the FFT perspectives by Zhylyevskyy [34, 35].

3 Heston model calibration

Market prices are used to calibrate an option pricing model, which can then be used to compute prices of more complex (exotic) options or to compute hedge ratios. Determining the model parameters to match the market prices of a set of options is what is commonly referred to as *model calibration* problem, it is in fact the inverse problem to the option pricing problem. The price to pay for more realistic model such as the Heston model is the increased complexity of model calibration. As pointed out by Jacquier and Jarrow [36], the estimation method becomes as crucial as the model itself.

In order to compute option prices using the Heston model, we need input parameters that are not observable from the market data, therefore we can not use empirical estimates. Bakshi et al. [37] documented that the implied structural parameters differ significantly from their time-series estimated counterparts. For instance, the magnitudes of time-series correlation coefficient of the asset return and its volatility estimated from the daily prices were much lower than their option-implied counterparts.

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

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

$$\inf_{\Theta} G(\Theta), \qquad G(\Theta) = \sum_{i=1}^{N} w_i |C_i^{\Theta}(t, S_t, T_i, K_i) - C_i^*(T_i, K_i)|^2, \tag{18}$$

where N denotes the number of options used, w_i is a weight, $C_i^*(T_i, K_i)$ is the market price of the call option observed at time t. C^{Θ} denotes the model price computed using vector of model parameters $\Theta = (v_0, \kappa, \theta, \sigma, \rho)$.

Thinking of G as an objective function of Θ, the optimization problem (18) is not an easy one. Function G is not convex and is not of any particular structure. Moreover, it may have more than one global minimum and even if it had not, it is not clear whether the unique minimum can be reached by gradient based algorithm.

Local deterministic algorithms can be used, but there is always the possibility to end up in a local minimum, one has to choose an initial guess, which is "good enough". Due to high sensitivity to initial parameters various stochastic algorithms are suggested. Nevertheless, for example Mikhailov and Nögel [10] use built-in Excel solver, which is a local optimizer that uses generalized reduced gradient method.

Another method used to calibrate the model is regularisation. This method adds to the objective function (18) penalization function, e.g. $f(\Theta)$ such that

$$\inf_{\Theta} G(\Theta) + \alpha f(\Theta)$$

is convex, hence gradient based optimizing procedure can be used. This yields another parameter to be estimated α, which is called regularisation parameter. For further details, see Hamida and Cont [38].

3.1 Considered optimization techniques

Following the methodology and results obtained by the authors in [13, 14], we describe in more details the behaviour of all optimization techniques. We consider functions from MATLAB's Optimization Toolbox, genetic algorithm (GA) and simulated annealing (SA) as well as MATLAB's `lsqnonlin` using trust-region-reflective algorithm. In addition to the MATLAB's functions we also test performance of the MS Excel's solver, adaptive simulated annealing (ASA) (available at ingber.com as a standalone C code and library with MATLAB gateway available) and modified sequential quadratic programming (modSQP) method suggested by Kienitz and Wetterau [11]. Although the ASA is not implemented in MATLAB directly and the available gateway has its limits in vector calculations, the re-annealing algorithm in ASA is faster than the standard annealing in SA and based on the performance on a set of artificially generated prices we abandon SA results from the tables below. Although modSQP is not designed for nonlinear least squares minimization problems of the type (18), it surprisingly works well on small examples, but it is not applicable for larger number of options, therefore we omit modSQP as well.

3.2 Considered errors

To evaluate the performance of all optimization methods, we measure the following errors:

$$\text{Maximum absolute relative error: } \mathrm{MARE}(\Theta) = \max_i \frac{|C_i^\Theta - C_i^*|}{C_i^*}, \quad i = 1, \dots, N,$$

$$\text{Average absolute relative error: } \mathrm{AARE}(\Theta) = \frac{1}{N}\sum_{i=1}^{N} \frac{|C_i^\Theta - C_i^*|}{C_i^*},$$

$$\text{Root-mean-square error: } \mathrm{RMSE}(\Theta) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(C_i^\Theta - C_i^*)^2}.$$

3.3 Considered weights

As defined above, w_i stands for a particular weight. In practice the weights are put where the most liquid quotes are on the market. This is usually around ATM. With our market data we use the bid ask spreads.

We aim to have the model prices as close to the Mid price as possible. Thus we put weight on the small spreads. According to U. Nögel, everybody does his own fine tuning on that, depending on the market and the products. We decided not to pick up just one choice for the weight function, but rather test more of these choices and see if there is a significant influence on the results caused by the particular choice of the weight function. For one particular option we denoted the weights by capital letters A, B and C, where weight A denotes $\frac{1}{|\mathrm{Spread}|}$, weight B denotes $\frac{1}{(\mathrm{Spread})^2}$ and weight C denotes $\frac{1}{\sqrt{\mathrm{Spread}}}$. The bigger the spread the less weight is put on the particular difference between the model price and the market price during the calibration process. In [14], it was shown that weights calculated from the Black-Scholes vega greeks (denoted by weights D) can be the least suitable and hence we abandoned their usage as well as we have not normalized the weights and omitted also the equal weights E.

3.4 Considered real market data

In order to calibrate Heston model to data obtained from real market we use Bloomberg's Option Monitor to obtain data for two consecutive days for ODAX call options. The following Table 1 sums up the data we use to calibrate the Heston model.

Table 1. Real market data

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

As a risk free interest rate r we use the corresponding 3 month EURIBOR rate, S_0 is the current DAX value at the time of obtaining the data.

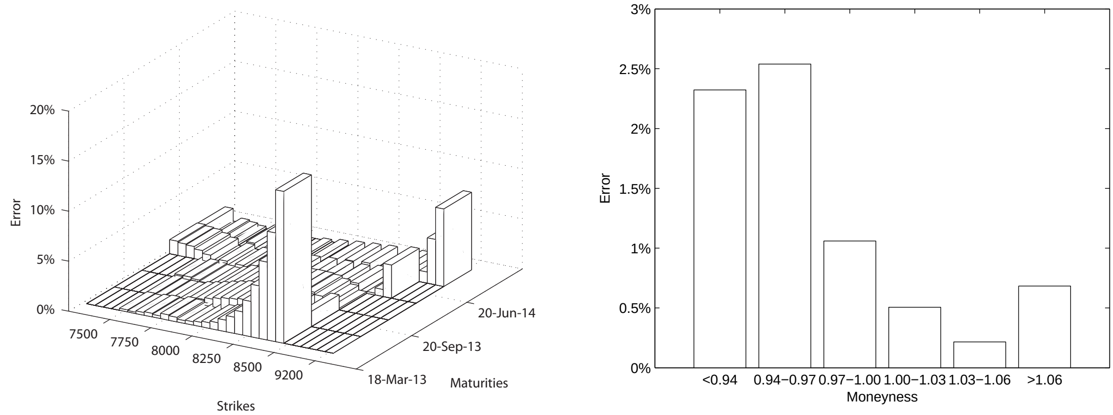
3.5 Calibration results – March 18, 2013

In this section we present the results of calibration of the Heston model to the first set of data from March 18, 2013. We were using GA and ASA as global optimizers and MS Excel's Solver and MATLAB's `lsqnonlin` as local optimizers. Global optimizers can find a solution on their own even without any initial guess, local optimizers on the other hand need a suitable starting point. Hence, the combinations of global and local optimizers also follow. In all algorithms we set the function tolerance stopping criteria to be `TolFun=1e-9`, i.e. the algorithm stops if the change in the best utility function value (i.e. the change to the minimal value found in previous iterations) is less than or equal to `TolFun`. In global optimizers this tolerance represents the average relative change of the best value of the utility function over the so called stalled iterations, with the maximum of stalled generations set to `StallGenLimit=50`. In local optimizers, an additional termination tolerance `TolX=1e-9` for change in the solution (point of the minima) is considered.

3.5.1 Genetic algorithm

We were able to obtain a negative correlation rho between the underlying asset and volatility using GA with different weights. Using weights A and B resulted in similar set of parameters, the best result in terms of absolute relative errors was obtained by using weights C. However, we can still observe higher absolute relative errors in Figure 2, for out of the money options, especially those maturing on June 21, 2013 represented by the first row.

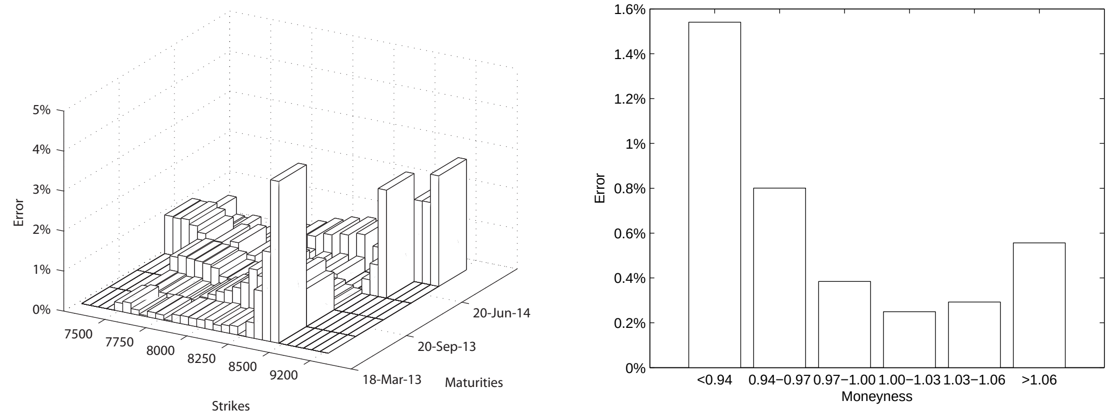
Fig. 2. Results for genetic algorithm. The left picture portrays the absolute relative errors. The right picture shows the average absolute relative errors in terms of moneyness.



3.5.2 Adaptive simulated annealing

As a result of the calibration using ASA with different weights we obtained three quite different sets of parameters. All three contained negative correlation parameter rho for the underlying asset and volatility, but other than that the parameters quite varied. Using weights B produced very low absolute relative errors. Only deep out of money options suffered higher than 0.8% average absolute relative error. This was the best result obtained so far and absolute relative errors can be seen in Figure 3.

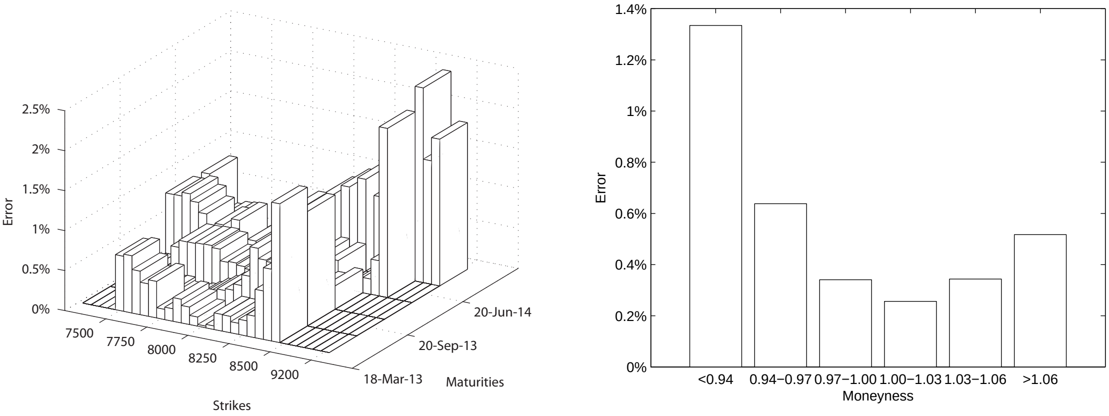
Fig. 3. Results for ASA. The left picture portrays the absolute relative errors. The right picture shows the average absolute relative errors in terms of moneyness.



3.5.3 Deterministic grid for `lsqnonlin`

When calibrating the model using `lsqnonlin`, the result can be extremely sensitive to the provided initial value. To overcome this we divided the state space of possible parameters' values in order to obtain a set of initial values. We used these as initial values and adaptively refined the results by repeating the process for regions where further improvement was possible. This method gave us the best results in terms of absolute relative errors, also the maximum relative error was in line with the rest of the errors. We do not observe high values for the out of the money options with short maturities in Figure 4. We used weights B to do so. Nonetheless, this method is rather time consuming and therefore we adopted another approach described in the next section in order to match the magnitude of the errors for the paramaters obtained by this method.

Fig. 4. Results for `lsqnonlin`. The left picture portrays the absolute relative errors. The right picture shows the average absolute relative errors in terms of moneyness. Initial values were provided by deterministic grid.

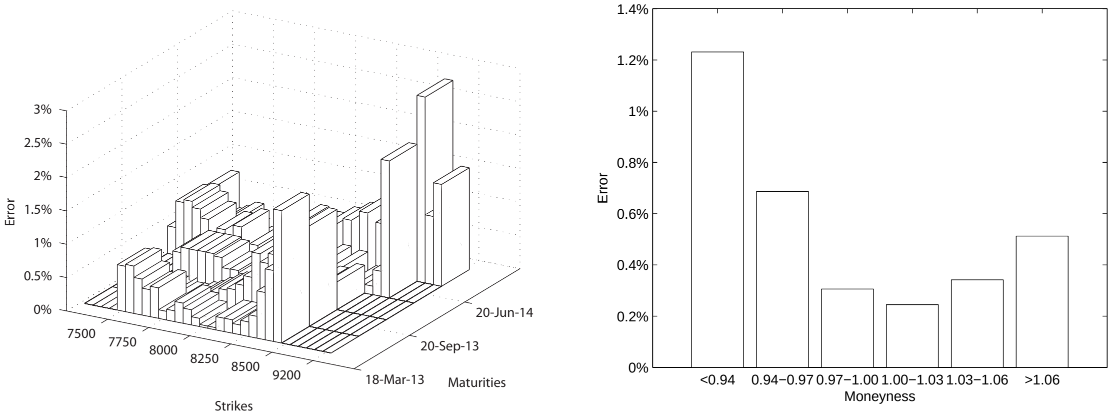


3.5.4 Excel's solver

Having observed quite high absolute relative errors for out of the money options with short maturities when using global optimizers ASA and GA, we tried to eliminate these errors by applying the local optimizers.

At first we employed MS Excel's solver and provided it with the initial guesses from the global optimizers (GA, ASA) and the results were as follows. Except for when we were using weights C, Excel was able to refine the values of parameters when starting from initial values provided by GA. For the choice of weights B we were able to match the best result in terms of average absolute relative error. This can be partly observed in Figure 5, also we were able to obtain maximum absolute relative error under 3%.

Fig. 5. Results for Excel's solver. The left picture portrays the absolute relative errors. The right picture shows the average absolute relative errors in terms of moneyness. The initial point was provided by GA.



When starting from point provided by ASA, although the average absolute error was already at the low value of 0.58%, we were able to refine the parameters using MS Excel's Solver. The lowest average absolute error was obtained using weights A. Also the maximum absolute relative error was lowered to 3.53%. Similar results were obtained using weights C. Using weights B, Excel's solver did not refine the parameters significantly.

Although the MS Excel's solver behaved surprisingly well, it still has its limitations both in terms of speed and precision. Standard Excel installations for example do not support high precision arithmetic and one has to use third party products such as xlPrecision. Their usage was beyond the scope of this paper, i.e. the implementation of all codes in Excel use standard double arithmetic only.

3.5.5 `lsqnonlin`

We applied the same approach for `lsqnonlin` as previously for MS Excel's solver, thus starting from initial points provided by GA nad ASA. `lsqnonlin` was able to refine the parameters using all the weights and starting from both.

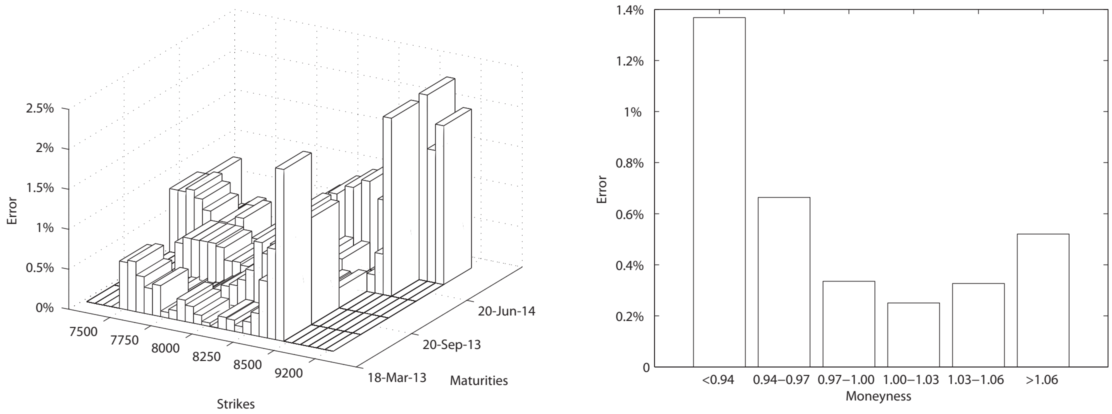
Using weights B and starting from the initial point provided by GA we were able to calibrate the model with average absolute relative error of 0.52%, also we obtained the lowest maximum absolute error of 2.33%, see Figure 6.

The best result, when starting from point provided by ASA, was also obtained using weights B and that was in terms of both maximum absolute relative error and average absolute relative error. Nevertheless, it was not better than what we were able to obtain when using initial guess from GA.

3.5.6 Comparison

Table 2 sums up the results comparing the global optimizers ASA and GA on the real set of data. However, using the `lsqnonlin` and starting from predetermined number of initial points we were able to calibrate the model with

Fig. 6. Results for `lsqnonlin`. The left picture portrays the absolute relative errors. The right picture shows the average absolute relative errors in terms of moneyness. The initial point was provided by GA.



average absolute relative error 0.51% and maximum absolute relative error of 2.44%. The illustrations are given in Figure 4.

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

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

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

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

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

It is very likely, based on the results in Table 2, that weights B suit `lsqnonlin` the best in terms of both average absolute relative error and maximum absolute relative error.

3.5.7 Speed-up using approximations

A tremendous speed-up of the calibration process can be achieved if the optimization is run against the approximation formula (17) rather than the exact formula. For example local optimization using `lsqnonlin` can give a solution for any initial guess in computational time less than a second (reference PC with 1x Intel i7-3520M 2.9 GHz CPU and 8 GB RAM). To get a more precise solution, we take this result as a starting point for another `lsqnonlin` run now with the exact formula. If we have no initial guess available at all and we need to run the global optimization first, we can actually use the approximation formula in the global optimization as well. Moreover, it is not necessary to wait for the global optimization to finish, but several iterations are sufficient to give a meaningful result that can serve as the initial guess for the first local optimization. Based on empirical observations we designed the following calibration algorithm:

Step 1: Run 10 iterations (generations) of the GA optimization for utility (fitness) function (18) that uses approximation formula (17), GA is run with population size 50, let us denote the result Θ_{GA}.

Step 2: Run the first `lsqnonlin` optimization for utility function (18) that uses approximation formula (17), the optimization is run with the initial guess Θ_{GA}, let us denote the result as Θ_{LSQ_1}.

Step 3: Run the second `lsqnonlin` optimization for utility function (18) that uses exact Lewis formula (16), the optimization is run with the initial guess Θ_{LSQ_1} and the result Θ_{LSQ_2} is taken as the solution of the whole calibration process.

If run repeatedly, the average computational times for steps 1, 2 and 3 are 550 msec, 120 msec and 950 msec respectively, i.e. giving us the average calibration time 1.62 sec. However, if we skip the second step, i.e. if we run only one `lsqnonlin` optimization for utility function that uses exact Lewis formula (16) and the optimization is run with the initial guess Θ_{GA}, the computational times increase. The average time for this single step is 2.83 sec. The most significant difference is however in computational time of the first step. If it is run with the exact formula (16), the average time of this step is 4.15 sec. It is worth to mention that due to the random nature of GA, the variance of computational times can be large. Computational times of the GA optimization can especially grow, if the numerical evaluation of the integral in (16) requires to switch to the high precision arithmetic as we discussed in section 2.2 above. From this perspective, using the approximation formula in the global optimization is very efficient. We can conclude that incorporating the approximation formula into the calibration process brought us a speed-up factor ca 4.3, i.e. the calibration is now 4.3 times faster than the calibration procedure presented in [14]. In fact, the speed-up is even bigger, since in [14] the GA was run for more iterations than 10 and with population size 100, but to provide the meaningful comparison, in both tests we fixed the number of iterations in the first step of the calibration to be 10 and the population size to be 50 which seems to be sufficient for problems with five variables.

3.6 Calibration results – March 19, 2013

We did find out in the previous section that the global optimizers GA and ASA are comparable, although it seemed that GA provided us with a better starting point for local optimizers (we remind that in standalone GA or ASA optimizations the maximum number of iterations is not limited, the algorithms terminate if the step size or function tolerance stopping criteria is met). MS Excel's Solver was competing very well against MATLAB's `lsqnonlin`. We repeated the process on the data from the following day March 19, 2013. This time we had 10 more data available with one more maturity.

Although performing well for the set of data from the previous day, this time Excel failed to significantly refine the initial values provided by the global optimizers. On the other hand, using MATLAB's `lsqnonlin` we were able to refine the initial set of parameters provided by GA and obtain average absolute relative error in Figure 8 comparable to average absolute relative error obtained by the method using deterministic grid as the initial value for `lsqnonlin` in Figure 7.

When comparing the global otimizers, GA provided better results than ASA. A solution from GA was then used as the initial starting point for `lsqnonlin` that improved the result even further, producing the lowest maximum absolute relative error of 2.22%, which can be observed in Figure 8.

Interesting fact is that as far as the weights are concerned, we can not say which works the best for GA and ASA. Different weights yielded best results for both GA and ASA when compared to the previous set of data. On the contrary, according to the results, `lsqnonlin` seems to be favoring weights B. All the results discussed above and more can be seen in Table 3.

Fig. 7. Results for `lsqnonlin`. The left picture portrays the absolute relative errors. The right picture shows the average absolute relative errors in terms of moneyness. Initial values were provided by deterministic grid.

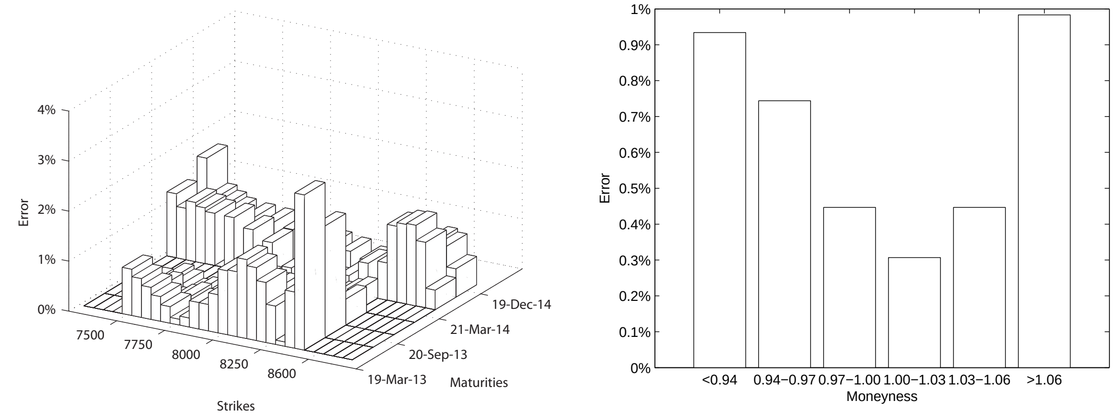
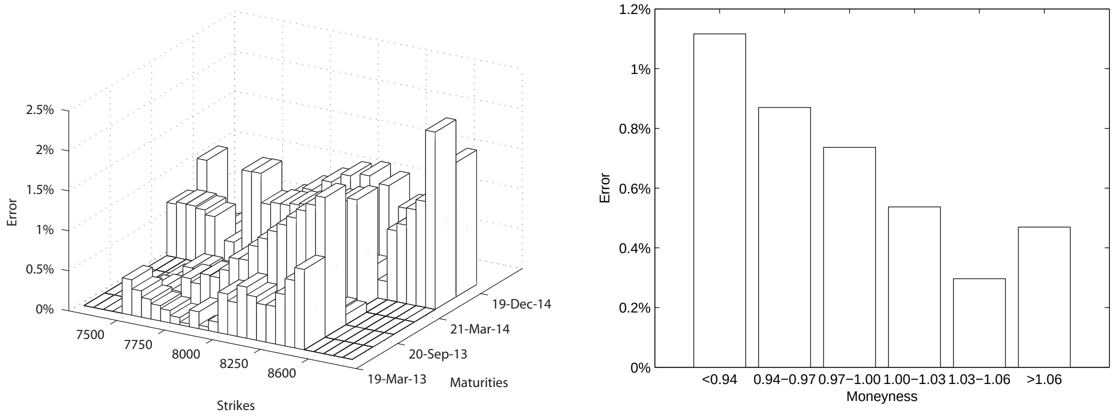


Fig. 8. Results for `lsqnonlin`. The left picture portrays the absolute relative errors. The right picture shows the average absolute relative errors in terms of moneyness. The initial point was provided by GA.



4 Heston model simulation

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

Let us now consider the system of stochastic differential equations (1), (2), (3) with initial conditions (4) and (5).

4.1 Euler scheme

Let us now apply the Euler scheme to the variance process (2). A naive Euler discretization of the variance process reads

$$v_{n+1} = v_n + \kappa(\theta - v_n)\Delta_n + \sigma\sqrt{v_n}Z_v\sqrt{\Delta_n} \tag{19}$$

where Z_v is a standard normal random variable and Δ_n denotes the step size. The main problem when using the above scheme is the fact, that the scheme does not preserve the positivity of the process. The discrete process v_n can

Table 2. Calibration results for March 18, 2013. * denotes initial guess provided by GA. ** indicates initial guess provided using ASA. *** is used for results obtained by deterministic grid.

Algorithm	W.	AARE	RMSE	MARE	v0	kappa	theta	sigma	rho
GA	A	2.00%	10.40	20.70%	0.03226	0.07065	0.73827	0.81988	-0.52083
GA	B	2.07%	14.04	15.13%	0.03193	0.07747	0.73826	0.85729	-0.55003
GA	C	1.24%	5.76	15.17%	0.03035	0.55662	0.11191	0.71420	-0.55050
ASA	A	1.19%	6.12	14.52%	0.03219	1.12162	0.08278	0.96401	-0.54227
ASA	B	0.58%	3.83	4.04%	0.02845	1.26339	0.06718	0.67255	-0.62816
ASA	C	2.55%	11.19	33.54%	0.04111	0.80249	0.13210	1.55269	-0.47895
lsqnonlin***	B	0.51%	3.67	2.44%	0.02741	1.18184	0.06586	0.57479	-0.66686
Excel*	A	0.65%	3.49	3.86%	0.02683	0.66747	0.08426	0.46984	-0.67899
Excel*	B	**0.51%**	3.48	**2.79%**	0.02746	1.12422	0.06762	0.57479	-0.66342
Excel*	C	1.24%	5.76	15.17%	0.03035	0.55663	0.11192	0.71417	-0.55050
Excel**	A	0.55%	3.46	3.53%	0.02745	1.09385	0.06818	0.57187	-0.64966
Excel**	B	0.58%	3.82	3.95%	0.02843	1.26363	0.06716	0.67246	-0.62834
Excel**	C	0.56%	3.43	3.51%	0.02729	1.06117	0.06852	0.55391	-0.65495
lsqnonlin*	A	0.55%	3.46	3.42%	0.02747	1.09567	0.06829	0.57399	-0.65043
lsqnonlin*	B	**0.52%**	3.68	**2.33%**	0.02760	1.20011	0.06601	0.59282	-0.65886
lsqnonlin*	C	0.58%	3.38	4.19%	0.02732	0.97657	0.07120	0.54564	-0.65127
lsqnonlin**	A	0.55%	3.48	3.39%	0.02750	1.11668	0.06781	0.57870	-0.64958
lsqnonlin**	B	0.54%	3.96	2.68%	0.02786	1.24433	0.06596	0.62264	-0.64732
lsqnonlin**	C	0.58%	3.37	4.10%	0.02730	0.97637	0.07113	0.54339	-0.65279

Table 3. Calibration results for March 19, 2013. * denotes initial guess provided by GA. ** indicates initial guess provided using ASA. *** is used for results obtained by deterministic grid.

Algorithm	W.	AARE	RMSE	MARE	v0	kappa	theta	sigma	rho
GA	A	1.25%	6.26	12.46%	0.02897	0.68921	0.10313	0.79492	-0.53769
GA	B	2.10%	14.94	13.80%	0.03073	0.06405	0.94533	0.91248	-0.53915
GA	C	1.70%	8.30	18.35%	0.03300	0.83930	0.10826	1.14674	-0.49923
ASA	A	2.26%	11.79	19.51%	0.03876	0.80811	0.13781	1.63697	-0.46680
ASA	B	2.62%	13.80	28.65%	0.03721	1.45765	0.09663	1.86941	-0.37053
ASA	C	1.73%	8.42	19.82%	0.03550	1.22482	0.09508	1.44249	-0.49063
lsqnonlin***	B	0.58%	4.95	3.10%	0.02382	1.75680	0.04953	0.42134	-0.84493
Excel*	A	1.25%	6.27	12.46%	0.02897	0.68922	0.10314	0.79490	-0.53769
Excel*	B	1.25%	6.27	12.46%	0.02896	0.68921	0.10314	0.79492	-0.53769
Excel*	C	1.25%	6.23	12.66%	0.02903	0.68932	0.10294	0.79464	-0.53763
Excel**	A	1.73%	8.42	19.82%	0.03550	1.22482	0.09509	1.44248	-0.49062
Excel**	B	1.78%	9.79	18.18%	0.03439	1.22399	0.09740	1.43711	-0.49115
Excel**	C	1.73%	8.42	19.82%	0.03550	1.22482	0.09509	1.44248	-0.49062
lsqnonlin*	A	0.67%	3.49	3.07%	0.02491	0.82270	0.07597	0.48665	-0.67099
lsqnonlin*	B	0.65%	4.27	2.22%	0.02497	1.22136	0.06442	0.55993	-0.66255
lsqnonlin*	C	0.68%	3.45	3.66%	0.02486	0.75195	0.07886	0.46936	-0.67266
lsqnonlin**	A	1.73%	8.42	19.82%	0.03550	1.22482	0.09508	1.44249	-0.49063
lsqnonlin**	B	1.71%	9.10	19.48%	0.03511	1.22672	0.09636	1.44194	-0.49089
lsqnonlin**	C	1.73%	8.42	19.82%	0.03550	1.22482	0.09508	1.44249	-0.49063

become negative with non-zero probability,

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

as Z_v is the standard normal variable, Φ denotes the standard normal cumulative distribution function. This makes the computation of $\sqrt{v_n}$ impossible and the time-stepping scheme fails. To avoid this, the practitioners adopted two ways of fixing it. First approach is referred to as the absorption. This approach follows the rule that process equals to zero whenever it attains a negative value (i.e. if $v < 0$ then let $v := 0$). The other approach is called reflection. The process is being reflected by its origin (i.e. if $v < 0$ then let $v := -v$), see Gatheral [20].

The various schemes can be unified in the following framework

$$v_{n+1} = f_1(v_n) + \kappa(\theta - f_2(v_n))\Delta_n + \sigma\sqrt{f_3(v_n)}Z_v\sqrt{\Delta_n}, \tag{20}$$

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

The particular choice for the "fixing" function $f_i(x)$ in the literature is the identity function ($f(x) = x$), for the absorption we have $f(x) = x^+$, where $x^+ = \max(x, 0)$, and for reflection $f(x) = |x|$. In the paper by Higham and Mao [6] the scheme uses the following configuration $f_1(x) = f_2(x) := x$ and $f_3(x) := |x|$, whereas for example in paper by Deelstra and Delbaen [39] $f_3(x) := x^+$ is used. All schemes coincide and are to be bias-free as $\Delta_n \to 0$ and so the choice of the fix functions may seem almost indifferent, but the contrary is true. While some schemes are extremely biased for practical sizes of the time step, others turn out to be almost bias-free for not "too extreme parameter configurations" as pointed out in Van Haastrecht and Pelsser [40].

According to the paper by Lord et al. [7] a scheme called the full truncation scheme produces the smallest bias, closely followed by the moment-matching, Euler scheme using log-normal approximation by Andersen and Brotherton-Ratcliffe [41] and the partial truncation scheme by Deelstra and Delbaen [39]. The full truncation scheme uses $f_1(x) := x$ and $f_2(x) = f_3(x) := x^+$. The resulting Euler scheme reads

$$v_{n+1} = v_n + \kappa(\theta - v_n^+)\Delta_n + \sigma\sqrt{v_n^+}Z_v\sqrt{\Delta_n}, \tag{21}$$

where x^+ =max$(x; 0)$. The simulation schemes of the corresponding asset price process (1) is then

$$S_{n+1} = S_n + rS_n\Delta_n + S_n\sqrt{v_n^+}Z_S\sqrt{\Delta_n}, \tag{22}$$

where Z_S is a standard normal random variable with correlation ρ to Z_v. This pair of correlated standard normal random variables Z_v and Z_S with a correlation of ρ between the driving Brownian motions using Cholesky decomposition can be simulated by setting $Z_v := Z_1$ and $Z_S := \rho Z_v + \sqrt{1-\rho^2}Z_2$, where Z_1 and Z_2 are two independent draws from the standard normal distribution.

Alternatively, the process for $\ln S(t)$ can be simulated, and then exponentiated. If we discretize log-stock price S rather than the stock price S, there are no higher order corrections to the Euler discretization for details, see Gatheral [20]. The log-Euler scheme

$$\ln(S_{n+1}) = \ln(S_n) + [r - \frac{1}{2}v_n^+]\Delta_n + \sqrt{v_n^+}Z_S\sqrt{\Delta_n}, \tag{23}$$

as noted by van Haastrecht and Pelsser [40] does not entail any discretization error in the stock price direction. On the other hand, the scheme usually shows biases in the Euler discretization of the variance process (and thus in resulting stock prices).

4.2 Milstein scheme

Milstein scheme is a refinement to the Euler scheme produced by adding an extra term using the Itô's lemma. As pointed out by Gatheral [20], applying this scheme to the variance process yields reduced frequency with which the process goes negative relative to the Euler scheme. Nevertheless, negative values still appear and we have to adopt similar fixes in (24). The Milstein scheme reads

$$v_{n+1} = v_n + \kappa(\theta - v_n^+)\Delta_n + \sigma\sqrt{v_n^+}Z_v\sqrt{\Delta_n} + \frac{1}{4}\sigma^2(Z_v^2 - 1)\Delta_n, \tag{24}$$

where Z_v stands for standard normal random variable. Similarly, we can apply the Milstein scheme to the process $S(t)$, which produces

$$S_{n+1} = S_n + rS_n\Delta_n + S_n\sqrt{v_n^+}Z_S\sqrt{\Delta_n} + \frac{1}{2}S_n v_n^+(Z_S^2 - 1)\Delta_n, \tag{25}$$

where Z_S is a standard normal random variable with correlation ρ to Z_v.

4.3 Kahl-Jäckel scheme

Introduced in Kahl and Jäckel [42] as IJK scheme, it follows

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

This scheme was developed as a *drop-in* replacement for the standard log-Euler procedure since it requires no additional random numbers per step or any adaptive refinement. Authors Kahl and Jäckel [42] claim the IJK scheme is not superior in convergence order to the log-Euler scheme, but they praise the fact that it excels over other equally simple methods by showing a superior convergence behaviour that is faster by a multiplicative factor.

4.4 van Haastrecht and Pelsser revised

In [40], van Haastrecht and Pelser commented on Kahl-Jäckel scheme implemented in the following way. Conditional on time s a discretization of the log Stock price process S for $t > s$ (with $\delta t := t - s$) reads exactly

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

We left the original notation as well. This however does not match our implementation of the scheme. The Heston log price S comes from the formula by Kahl and Jäckel [42], which is exactly

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

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

Clearly the sign in the term $(Z_S + \rho Z_V)$ is wrong and should be $(Z_S - \rho Z_V)$ instead. Another mistake is in the last term, where there is $\frac{1}{2}$ instead of $\frac{1}{4}$ compared to our equation. b_n is ξv_n^p for Heston in the original article by Kahl and Jäckel [42]. This implies $b(s) = \xi v(s)^{1/2}$, which cancels out with $v(s)^{-1/2}$ and leaves ξ in the notation of Haastrecht and Pelsser. This is probably due to the fact that the IJK scheme in Kahl-Jäckel [42] was not stated explicitly for Heston model. The wrongly interpreted scheme published in Haastrecht and Pelsser [40] can be found in some other papers as well.

4.5 Exact scheme

Broadie and Kaya [8] devised an exact simulation algorithm for the Heston model. In numerical comparisons of their algorithm to an Euler discretisation with the absorption fix, they find that for the pricing of European options in the Heston model and variations thereof, the exact algorithm compares favourably in terms of root-mean-squared (RMS) error. According to Broadie and Kaya [8] the volatility at time t, given the values of v_u and for $u < t$, can be written as

$$v_t = v_u + \kappa\theta(t - u) - \kappa\int_u^t v_s ds + \sigma\int_u^t \sqrt{v_s}dW_s^{(1)} \tag{26}$$

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

$$\ln(S_t) = \ln(S_u) + r(t-u) - \frac{1}{2}\int_u^t v_s ds + \rho \int_u^t \sqrt{v_s} dW_s^{(1)} + \sqrt{1-\rho^2}\int_u^t \sqrt{v_s} dW_s^{(2)}, \tag{27}$$

where $W^{(1)}$ and $W^{(2)}$ are uncorrelated Brownian motions. Substituting $\int_u^t \sqrt{v_s} dW_s^{(1)}$ from (26) into (27) yields

$$\ln(S_t) = \ln(S_u) + r(t-u) + \left(\frac{\kappa\rho}{\sigma} - \frac{1}{2}\right)\int_u^t v_s ds + \frac{\rho}{\sigma}(v_t - v_u - \kappa\theta(t-u)) + \sqrt{1-\rho^2}\int_u^t \sqrt{v_s} dW_s^{(2)}. \tag{28}$$

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

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

Knowing all the terms in (26) except for $\sigma \int_u^t \sqrt{v_s} dW_s^{(1)}$, the value of $\sigma \int_u^t \sqrt{v_s} dW_s^{(1)}$ can be easily obtained from (26). Furthermore, they use the fact that v_t is independent of the Brownian motion $W_t^{(2)}$ and therefore the distribution of $\int_u^t \sqrt{v_s} dW_s^{(2)}$ given the path generated by v_t is normal with mean 0 and variance $\int_u^t \sqrt{v_s} ds$. This scheme gives an unbiased estimator of the derivative price.

The algorithm is however very time-consuming, because of the application of the numerical inversion of a cumulative distribution using the characteristic function involving modified Bessel function of the first kind. Also we found out that with finer discretization the modified Bessel function of the first kind takes too large values (in particular, unrepresentable values). Therefore it is not recommended for the pricing of options when requiring the value of the asset price on a larger number of time instants. Nonetheless, (28) is used in the next sections.

4.6 QE scheme

The Quadratic Exponential scheme, also known as QE scheme, was introduced by Andersen [9]. It exploits the fact that the value v_{n+1} conditional on value v_n follows a non-central chi-square distribution. It uses two different approximations to the non-central chi-square distribution, depending on the values of the variance.

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

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

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

1. If $\psi \leq \psi_C$ we set $v_{n+1} = a(b + Z_v)^2$, where

$$a = \frac{m}{1+b^2}, \qquad b^2 = 2\psi^{-1} - 1 + \sqrt{2\psi^{-1}}\sqrt{2\psi^{-1} - 1} \geq 0 \qquad \text{and} \qquad Z_v = \Phi^{-1}(U_v).$$

2. If $\psi > \psi_C$ we set $v_{n+1} = \Psi^{-1}(U_v; p, \beta)$, where

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

$$p = \frac{\psi - 1}{\psi + 1} \qquad \text{and} \qquad \beta = \frac{1-p}{m}.$$

As stated above the value of ψ_C indicates which approximation to use. The first approximation can be only used for $\psi \leq 2$. For higher values of ψ, that correspond to low values of v_n, the scheme will fail. For $\psi \geq 1$ the second one can be used. Therefore $\psi_C \in [1,2]$. Andersen [9] states that the exact choice for ψ_C appears to have relatively small effects on the quality of the overall simulation scheme and proposes $\psi_C = 1.5$.

This scheme can not be used with the Euler discretization of $S(t)$, since it would result in too feeble effective correlation and consequently, paths of S with poor distribution tails. Instead Andersen used the exact scheme from Broadie and Kaya (28) and tried to preserve the key driver of the correlation between S_{n+1} and v_{n+1}, which is $\frac{\rho}{\sigma}v_{n+1}$. Instead of sampling the integral $\int_u^t v_s ds$ he used an approximation $\int_u^t v_s ds \approx \Delta_n[\gamma_1 v_n + \gamma_2 v_{n+1}]$, where setting $\gamma_1 = 1, \gamma_2 = 0$ yields Euler like setting. For the last integral the same approach as in (27) is applied, thus

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

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

$$\ln(S_{n+1}) = \ln(S_n) + K_0 + K_1 v_n + K_2 v_{n+1} + \sqrt{K_3 v_n + K_4 v_{n+1}} \cdot Z_S, \tag{29}$$

where $K_0, ..., K_4$ are given by:

$$K_0 = -\frac{\rho\kappa\theta}{\sigma}\Delta_n, \quad K_1 = \gamma_1 \Delta_n \left(\frac{\kappa\rho}{\sigma} - \frac{1}{2}\right) - \frac{\rho}{\sigma}, \quad K_2 = \gamma_2 \Delta_n \left(\frac{\kappa\rho}{\sigma} - \frac{1}{2}\right) + \frac{\rho}{\sigma},$$
$$K_3 = \gamma_1 \Delta_n (1-\rho^2), \quad K_4 = \gamma_2 \Delta_n (1-\rho^2)$$

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

Equation (29) is the exact proposal as can be found in Andersen [9]. However, we also include the drift, thus the resulting scheme we use for $S(t)$ reads

$$\ln(S_{n+1}) = \ln(S_n) + r\Delta_n + K_0 + K_1 v_n + K_2 v_{n+1} + \sqrt{K_3 v_n + K_4 v_{n+1}} \cdot Z_S. \tag{30}$$

QE scheme appears to be the most popular one among the recent propositions. A couple years later, a high order discretization schemes for the variance process were presented in the paper by Alfonsi [43], but rigorous error analysis for Heston model was regarded as an open issue. Possible extensions of the QE scheme include GammaQE scheme and Double Gamma scheme introduced by Chan and Joshi [44]. These two methods use a long-time stepping discretization for the variance process. The Double Gamma scheme is an exact simulation scheme and the GammaQE scheme is an almost exact simulation scheme for the variance process where the accuracy improves with the step length. Both methods are based on the combined compound Poisson and Gamma distribution representation of the variance process (cf. with Cox et. al. [17]). Proposed schemes then use precalculated values of the inverse of the gamma distribution and the compound Poisson random variables are approximated using a mixture of exponential distribution and quadratic Gaussian distribution. There are some similarities to the non-central chi-squared inversion scheme presented by van Haastrech and Pelsser [40]. None of these methods are, however, widely accepted by practitioners and further numerical analysis of these schemes is an ongoing research.

4.7 E+M scheme

We also consider a combination of the exact scheme for the underlying asset process with Milstein scheme for the volatility process. The motivation for this is the fact that the frequency of negative values of the variance process decreases for the Milstein scheme (24) compared to Euler, as pointed out for example in Gatheral [20]. The idea behind the use of the exact scheme by Broadie and Kaya (28) is that it can be used for discretizations on a larger number of time instants when not trying to sample the integral, but using an approximation instead.

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

4.8 MC simulations

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

We use the set of parameters listed in Table 4. The model implied parameters were obtained by genetic algorithm and then refined using nonlinear least-squares method. Using the model with this set of parameters we were able to obtain prices with mean absolute relative error of 0.65% and maximum absolute relative error of 2.22%.

Table 4. Set of used parameters from March 19, 2013.

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

However, as we stated above, the choice of the parameters can cause problems. The following Figure 9 shows the mean path of the underlying asset for the examined schemes. We used $\Delta = 2^{-6}$ and $T = 1$. Number of simulations was $nPaths = 100\,000$.

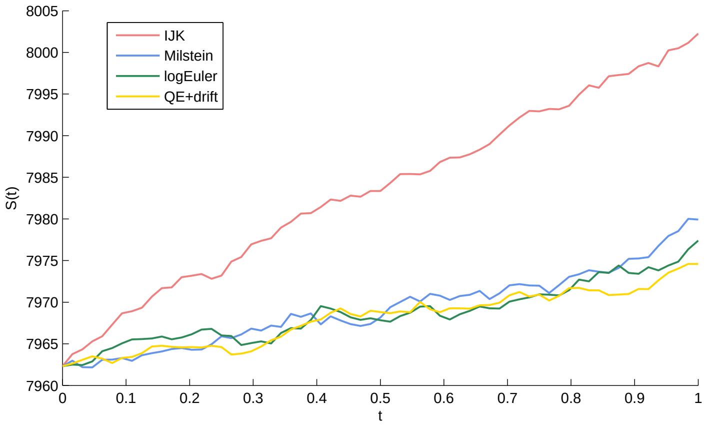


Fig. 9. Mean of 100000 paths using parameters from Table 4, $\Delta = 2^{-6}$ and $T = 1$.

It is clear in Figure 9 that the paths simulated by IJK scheme differ significantly. This behaviour is invariable over changing number of simulation and the step size Δ and results in great errors when trying to price the options using Monte-Carlo method. IJK scheme clearly does not work for this set of parameters and thus we do not consider it as a competing scheme. Nonetheless the results for IJK scheme are presented for illustration.

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

4.9 MC test

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

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

$$(7100, 7200, \dots, 7500, 7550, 7600, \dots, 8350, 8400, 8500, \dots, 9000).$$

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

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

$$\hat{C}(0) = e^{-rT}\mathbb{E}[(\hat{S}(T) - K)^+], \tag{31}$$

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

$$\hat{C}(0) \approx e^{-rT}\frac{1}{nPaths}\sum_{i=1}^{nPaths}(\hat{C}^{(i)}(T) - K)^+. \tag{32}$$

This leads to error

$$e = C(0) - \hat{C}(0). \tag{33}$$

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

$$\hat{\varepsilon}_j = \frac{1}{N}\sum_{n=1}^{N}\frac{|C(0) - \hat{C}(0)|}{C(0)}, \tag{34}$$

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

$$\hat{\varepsilon} = \frac{1}{M}\sum_{j=1}^{M}\varepsilon_j \tag{35}$$

and then use formula

$$\hat{\sigma}_{\varepsilon}^2 = \frac{1}{M-1}\sum_{j=1}^{M}(\hat{\varepsilon}_j - \hat{\varepsilon})^2 \tag{36}$$

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

$$(\hat{\varepsilon} - \Delta\hat{\varepsilon}, \hat{\varepsilon} + \Delta\hat{\varepsilon}), \tag{37}$$

where

$$\Delta\hat{\varepsilon} = t_{1-\alpha,M-1}\sqrt{\frac{\hat{\sigma}_{\varepsilon}^2}{M}} \tag{38}$$

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

4.10 MC test results

Figure 10 corresponds to Table 5 and shows the mean absolute relative errors. They decrease with higher number of simulated paths and the results for the particular schemes are consistent through all the three values. Results show that the QE scheme is the best choice for cases where we do not need small step size Δ. We did not include the

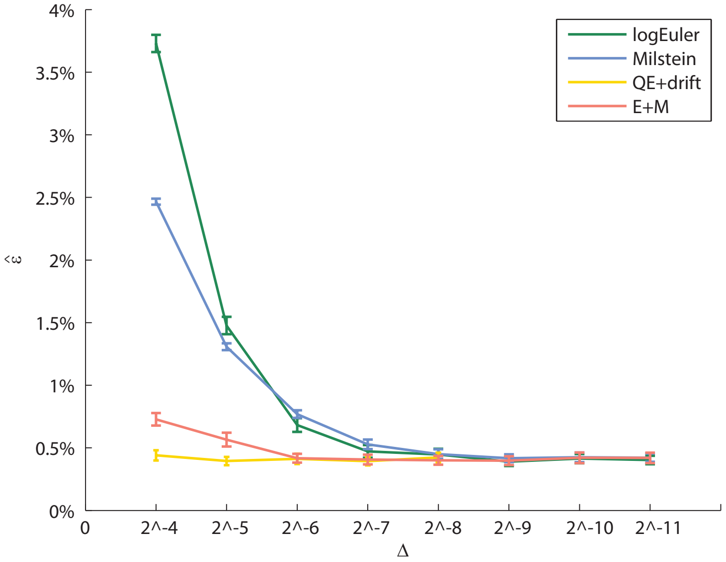


Fig. 10. Results for $\Delta = 2^{-4} - 2^{-11}$, for 100 000 simulated paths, number of batches $M = 100$.

Table 5. Results for $\Delta = 2^{-4} - 2^{-11}$, for 100 000 simulated paths, number of batches $M = 100$. IJK scheme is included.

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

results for Δ smaller than 2^{-8} since the scheme was not improving and it also took too long to simulate the paths this way.

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

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

We also included the results of IJK scheme although it proved to be not useful for this set of parameters. We can see this also in Table 5. The errors are high and do not depend on the time step Δ.

4.11 Variance reduction

In Monte Carlo simulations, reduction of the variance of the simulation results is of interest. One of the simplest variance reduction methods uses the so called antithetic variates. Having one simulated path of considered stochastic process, antithetic path is such a trajectory that is negatively correlated to the original path and both paths together reduce the variance of the sample paths. There are at least two advantages of this technique. At first, antithetic variates reduce the number of underlying random numbers needed to generate $nPath$ trajectories. Moreover, they reduce the variance of the sample paths and consequently improve the accuracy.

Let $\theta = \mathbb{E}[h(X)] = \mathbb{E}[Y]$ and let Y_1 and Y_2 be two generated samples of the random variable Y. Then an unbiased estimate $\hat{\theta}$ of θ is given by $\hat{\theta} = \frac{Y_1+Y_2}{2}$ and its variance

$$\operatorname{Var}(\hat{\theta}) = \frac{\operatorname{Var}(Y_1) + \operatorname{Var}(Y_2) + 2\operatorname{Cov}(Y_1, Y_2)}{4}$$

Obviously the variance is reduced if $\operatorname{Cov}(Y_1, Y_2) < 0$. This principle can be easily adapted to pathwise simulations such as the Euler or Milstein schemes that use the standard normal random variables for simulation of the underlying Wiener process. If $Y_1 \sim \mathcal{N}(\mu, \sigma^2)$ and $Y_2 = 2\mu - Y_1$, then also $Y_2 \sim \mathcal{N}(\mu, \sigma^2)$. This fact actually can be used also in the QE scheme, that behaved best in our tests. We can therefore further improve our Monte Carlo simulations.

Let $\hat{C}_1$ denote the option price estimate (32) using the standard simulation of $nPaths/2$ trajectories of the Heston model over the interval $[0, T]$ and let $v_1 = \operatorname{Var}(\hat{C}_1)$. Let $\hat{C}_2$ denote the antithetic simulation of $nPaths/2$ trajectories (i.e. together we have again $nPaths$ trajectories) and let $v_2 = \operatorname{Var}(\hat{C}_2)$. Then the variance reduction factor

$$\operatorname{Var}_{\text{red}} = \frac{\frac{v_1}{nPaths} - \frac{v_2}{nPaths/2}}{\frac{v_1}{nPaths}} \cdot 100 = \frac{v_1 - 2v_2}{v_1} \cdot 100.$$

In our example of $nPaths = 100\,000$ simulations of the process using the QE scheme with the same values of the parameters and $T = 1$ and with antithetic variates implemented, we get the variance reduction ca 38.2%.

5 Conclusion

We presented several most popular and industry standard methods to simulate the Heston model and gave reasons why we excluded the Exact scheme by Broadie and Kaya [8] from the testing. It was not only because of the computationally expensive numerical inversion but also because of the problems of dividing two modified Bessel functions of the first kind. These two were causing difficulties because of the representation of their rather high values for finer discretizations. Also IJK scheme by Kahl and Jäckel [42], which is supposed to be a *drop-in* method for simple schemes, is tested with parameters obtained from the market and it appears not to be useful for pricing options using Monte Carlo simulations. We also did experiments and proposed a scheme compromising speed and accuracy based on the observed performances of the so far known schemes and we seem to have out-performed the other simple schemes when using the Monte Carlo simulations for pricing options. The proposed scheme uses Milstein scheme for the variance process exploiting the lower frequency of negative values and combines it with the Exact scheme approach by Broadie and Kaya for the asset process and could be a *drop-in* replacement for log-Euler or Milstein scheme if higher precision is required. The best in terms of precision appears to be the QE scheme of Andersen [9], which is the most popular one among the practitioners these days. All the conclusions were made based on the test that involved mimicking real market situation and thus we avoided parameters or range of strikes and maturities that could favour some of the schemes purposely. We could further improve even the QE scheme by some of the variance reduction technique such as by using antithetic variates, but in our testing we avoided variance reduction methods in order to compare the core of all simulation methods.

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

Calibration of the Heston model is not covered in the literature to such extent that could be comparable to the coverage of the simulation schemes. We take the real data from the market and try to calibrate the model to them. This is not an easy task, especially when one uses rather high number of options. We approached it by combining the local optimizers, which tend to be sensitive to an initial guess and thus we think one should avoid using only those, with the global optimizers. We suggested using the genetic algorithm from the MATLAB's Global Optimization Toolbox and we developed what we believe is a stable procedure combining the genetic algorithm providing an initial guess, which is later refined by non-linear least squares method (`lsqnonlin`). Incorporating the usage of approximation formula into the calibration process is especially useful in the global optimization part and we showed that we can speed-up the overall calibration more than four times. Having the opportunity to run long computations in the distributed environment we were able to find benchmark values by creating a grid of the state space for the parameters and find out what the model is capable of. We managed to match the magnitude of the errors by our procedure, but of course in shorter computational time.

We also used adaptive simulated annealing and Excel's solver and compared the results for two consecutive days. Another conclusion is that slight adjustments to weights used during calibration can play a significant role and one can improve the results when experimenting with them.

Many other stochastic volatility models have been introduced since the original Heston paper [5]. First possible extension can involve time-dependent parameters. A model with piece-wise constant parameters is studied by Mikhailov and Nögel [10], a linear time dependent Heston model is studied by Elices [45] and a more general case is introduced by Benhamou et al. [46], who calculate also an approximation to the option price. However, Bayer, Fritz and Gatheral [47] claim that the overall shape of the volatility surface does not change in time and that also the parameters should remain constant. Affine jump-diffusion processes keep original parameters constant and add jumps to the underlying asset process, to volatility or to both, see e.g paper by Duffie, Pan and Singleton [48]. A special attention is nowadays paid to models with fractional volatility, we refer to recent paper by Pospíšil and Sobotka [49] where a new approximative fractional stochastic volatility jump diffusion model is presented and compared to the Heston model. In [50], authors also study the robustness and sensitivity analysis of these two models.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgement: This work is based on the Master's thesis *Heston stochastic volatility model* [12] that was written by Milan Mrázek and supervised by Jan Pospíšil. This work was supported by the GACR Grant 14-11559S Analysis of Fractional Stochastic Volatility Models and their Grid Implementation. Computational resources were provided by the CESNET LM2015042 and the CERIT Scientific Cloud LM2015085, provided under the programme "Projects of Large Research, Development, and Innovations Infrastructures".

References

[1] Black F. and Scholes M.S., *The pricing of options and corporate liabilities.* J. Polit. Econ. 81(3), 1973, 637–654. ISSN 0022-3808. DOI: 10.1086/260062.

[2] Hull J.C. and White A.D., *The pricing of options on assets with stochastic volatilities.* J. Finance 42(2), 1987, 281–300. ISSN 1540-6261. DOI: 10.1111/j.1540-6261.1987.tb02568.x.

[3] Scott L.O., *Option pricing when the variance changes randomly: Theory, estimation, and an application.* J. Financ. Quant. Anal. 22(4), 1987, 419–438. ISSN 0022-1090. DOI: 10.2307/2330793.

[4] Stein J. and Stein E., *Stock price distributions with stochastic volatility: An analytic approach.* Rev. Financ. Stud. 4(4), 1991, 727–752. ISSN 0893-9454. DOI: 10.1093/rfs/4.4.727.

[5] Heston S.L., *A closed-form solution for options with stochastic volatility with applications to bond and currency options.* Rev. Financ. Stud. 6(2), 1993, 327–343. ISSN 0893-9454. DOI: 10.1093/rfs/6.2.327.

[6] Higham D.J. and Mao X., *Convergence of monte carlo simulations involving the mean-reverting square root process*. J. Comput. Finance 8(3), 2005, 35–61. ISSN 1460-1559. DOI: 10.21314/JCF.2005.136.

[7] Lord R., Koekkoek R., and van Dijk D., *A comparison of biased simulation schemes for stochastic volatility models*. Quant. Finance 10(2), 2010, 177–194. ISSN 1469-7688. DOI: 10.1080/14697680802392496.

[8] Broadie M. and Kaya Ö., *Exact simulation of stochastic volatility and other affine jump diffusion processes*. Oper. Res. 54(2), 2006, 217–231. ISSN 0030-364X. DOI: 10.1287/opre.1050.0247.

[9] Andersen L., *Simple and efficient simulation of the Heston stochastic volatility model*. J. Comput. Finance 11(3), 2008, 1–42. ISSN 1460-1559. DOI: 10.21314/JCF.2008.189.

[10] Mikhailov S. and Nögel U., *Heston's stochastic volatility model - implementation, calibration and some extensions.* Wilmott magazine 2003(July), 2003, 74–79.

[11] Kienitz J. and Wetterau D., Financial modelling: Theory, implementation and practice with MATLAB source. The Wiley Finance Series. Wiley, 2012. ISBN 9781118413319.

[12] Mrázek M., Heston stochastic volatility model. Master's thesis, University of West Bohemia in Plzeň, Czech Republic, 2013.

[13] Mrázek M., Pospíšil J., and Sobotka T., *On optimization techniques for calibration of stochastic volatility models.* In Applied Numerical Mathematics and Scientific Computation, pages 34–40. Europment, Athens, Greece. ISBN 978-1-61804-253-8, 2014.

[14] Mrázek M., Pospíšil J., and Sobotka T., *On calibration of stochastic and fractional stochastic volatility models.* European J. Oper. Res. 254(3), 2016, 1036–1046. ISSN 0377-2217. DOI: 10.1016/j.ejor.2016.04.033.

[15] Alòs E., de Santiago R., and Vives J., *Calibration of stochastic volatility models via second-order approximation: The Heston case*. Int. J. Theor. Appl. Finance 18(6), 2015, 1–31. ISSN 0219-0249. DOI: 10.1142/S0219024915500363.

[16] Rouah F.D., The Heston Model and its Extensions in Matlab and C#, + Website. Wiley Finance Series. John Wiley & Sons, Inc., Hoboken, NJ, 2013. ISBN 9781118548257.

[17] Cox J.C., Ingersoll J.E., and Ross S.A., *A theory of the term structure of interest rates*. Econometrica 53(2), 1985, 385–407. ISSN 0012-9682. DOI: 10.2307/1911242.

[18] Feller W., *Two singular diffusion problems*. Ann. Math. 54(1), 1951, 173–182. ISSN 0003-486X. DOI: 10.2307/1969318.

[19] Albrecher H., Mayer P., Schoutens W., and Tistaert J., *The little Heston trap*. Wilmott Magazine 2007(January/February), 2007, 83–92.

[20] Gatheral J., The volatility surface: A practitioner's guide. Wiley Finance. John Wiley & Sons, Hoboken, New Jersey, 2006. ISBN 9780470068250.

[21] Kahl C. and Jäckel P., *Not-so-complex logarithms in the Heston model*. Wilmott Magazine 2005(September), 2005, 94–103.

[22] Lewis A.L., Option valuation under stochastic volatility, with Mathematica code. Finance Press, Newport Beach, CA, 2000. ISBN 9780967637204.

[23] Baustian F., Mrázek M., Pospíšil J., and Sobotka T., *Unifying pricing formula for several stochastic volatility models with jumps*. Appl. Stoch. Models Bus. Ind. ISSN 1524-1904. DOI: 10.1002/asmb.2248. To appear (online first 04/2017).

[24] Attari M., *Option pricing using Fourier transforms: A numerically efficient simplification*, 2004. DOI: 10.2139/ssrn.520042. Available at SSRN: http://ssrn.com/abstract=520042.

[25] Daněk J. and Pospíšil J., *Numerical aspects of integration in semi-closed option pricing formulas for stochastic volatility jump diffusion models*, 2017. Manuscript under review (submitted 01/2017).

[26] Ribeiro A. and Poulsen R., *Approximation behoves calibration*. Quant. Finance Letters 1(1), 2013, 36–40. ISSN 2164-9502. DOI: 10.1080/21649502.2013.803781.

[27] Forde M. and Jacquier A., *The large-maturity smile for the Heston model*. Finance Stoch. 15(4), 2011, 755–780. ISSN 0949-2984. DOI: 10.1007/s00780-010-0147-3.

[28] Forde M., Jacquier A., and Lee R., *The small-time smile and term structure of implied volatility under the Heston model*. SIAM J. Finan. Math. 3(1), 2012, 690–708. ISSN 1945-497X. DOI: 10.1137/110830241.

[29] Carr P. and Madan D.B., *Option valuation using the fast Fourier transform*. J. Comput. Finance 2(4), 1999, 61–73. ISSN 1460-1559. DOI: 10.21314/JCF.1999.043.

[30] Bailey D.H. and Swarztrauber P.N., *The fractional Fourier transform and applications*. SIAM Rev. 33(3), 1991, 389–404. ISSN 0036-1445. DOI: 10.1137/1033097.

[31] Bailey D.H. and Swarztrauber P.N., *A fast method for the numerical evaluation of continuous Fourier and Laplace transforms*. SIAM J. Sci. Comput. 15(5), 1994, 1105–1110. ISSN 1064-8275. DOI: 10.1137/0915067.

[32] Fang F. and Oosterlee C.W., *A novel pricing method for European options based on Fourier-cosine series expansions*. SIAM J. Sci. Comput. 31(2), 2009, 826–848. ISSN 1064-8275. DOI: 10.1137/080718061.

[33] Ortiz-Gracia L. and Oosterlee C.W., *A highly efficient Shannon wavelet inverse Fourier technique for pricing European options*. SIAM J. Sci. Comput. 38(1), 2016, B118–B143. ISSN 1064-8275. DOI: 10.1137/15M1014164.

[34] Zhylyevskyy O., *A fast fourier transform technique for pricing american options under stochastic volatility*. Rev. Deriv. Res. 13(1), 2010, 1–24. ISSN 1380-6645. DOI: 10.1007/s11147-009-9041-6.

[35] Zhylyevskyy O., *Efficient pricing of European-style options under Heston's stochastic volatility model*. Theor. Econ. Letters 2(1), 2012, 16–20. ISSN 2162-2078. DOI: 10.4236/tel.2012.21003.

[36] Jacquier E. and Jarrow R., *Bayesian analysis of contingent claim model error*. J. Econometrics 94(1–2), 2000, 145–180. ISSN 0304-4076. DOI: 10.1016/S0304-4076(99)00020-2.

[37] Bakshi G., Cao C., and Chen Z., *Empirical performance of alternative option pricing models*. J. Finance 52(5), 1997, 2003–2049. ISSN 1540-6261. DOI: 10.1111/j.1540-6261.1997.tb02749.x.

[38] Hamida S.B. and Cont R., *Recovering volatility from option prices by evolutionary optimization*. J. Comput. Finance 8(4), 2005, 43–76. ISSN 1460-1559. DOI: 10.21314/JCF.2005.130.

[39] Deelstra G. and Delbaen F., *Convergence of discretized stochastic (interest rate) processes with stochastic drift term*. Appl. Stoch. Models Data Anal. 14(1), 1998, 77–84. ISSN 1099-0747. DOI: 10/cm9jpb.

[40] van Haastrecht A. and Pelsser A., *Efficient, almost exact simulation of the Heston stochastic volatility model*. Int. J. Theor. Appl. Finance 13(01), 2010, 1–43. ISSN 0219-0249. DOI: http://dx.doi.org/10.1142/S0219024910005668.

[41] Andersen L.B. and Brotherton-Ratcliffe R., *Extended LIBOR market models with stochastic volatility*. J. Comput. Finance 9(1), 2005, 1–40. ISSN 1460-1559. DOI: 10.21314/JCF.2005.127.

[42] Kahl C. and Jäckel P., *Fast strong approximation monte carlo schemes for stochastic volatility models*. Quant. Finance 6(6), 2006, 513–536. ISSN 1469-7688. DOI: 10.1080/14697680600841108.

[43] Alfonsi A., *High order discretization schemes for the CIR process: Application to affine term structure and Heston models*. Math. Comp. 79(269), 2010, 209–237. ISSN 0025-5718. DOI: 10.1090/S0025-5718-09-02252-2.

[44] Chan J.H. and Joshi M., *Fast and accurate long stepping simulation of the Heston stochastic volatility model*, 2010. DOI: 10.2139/ssrn.1617187. Available at SSRN: https://ssrn.com/abstract=1617187.

[45] Elices A., *Models with time-dependent parameters using transform methods: application to Heston's model*, 2008. Available at arXiv: https://arxiv.org/abs/0708.2020.

[46] Benhamou E., Gobet E., and Miri M., *Time dependent Heston model*. SIAM J. Finan. Math. 1(1), 2010, 289–325. ISSN 1945-497X. DOI: 10.1137/090753814.

[47] Bayer C., Friz P., and Gatheral J., *Pricing under rough volatility*. Quant. Finance 16(6), 2016, 887–904. ISSN 1469-7688. DOI: 10.1080/14697688.2015.1099717.

[48] Duffie D., Pan J., and Singleton K., *Transform analysis and asset pricing for affine jump-diffusions*. Econometrica 68(6), 2000, 1343–1376. ISSN 0012-9682. DOI: 10.1111/1468-0262.00164.

[49] Pospíšil J. and Sobotka T., *Market calibration under a long memory stochastic volatility model*. Appl. Math. Finance 23(5), 2016, 323–343. ISSN 1350-486X. DOI: 10.1080/1350486X.2017.1279977.

[50] Pospíšil J., Sobotka T., and Ziegler P., *Robustness and sensitivity analyses for stochastic volatility models under uncertain data structure*, 2016. Manuscript under review (submitted 06/2016).