UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

Efficient, almost exact simulation of the Heston stochastic volatility model

van Haastrecht, A.; Pelsser, A.

Publication date
2008

Link to publication

Citation for published version (APA):
van Haastrecht, A., & Pelsser, A. (2008). *Efficient, almost exact simulation of the Heston stochastic volatility model*. Faculteit Economie en Bedrijfskunde.
http://ssrn.com/abstract=1131137

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:25 Jul 2022

Efficient, almost exact simulation of the Heston stochastic volatility model

Alexander van Haastrecht[1] [2] and Antoon Pelsser[3].

First version: September 9, 2007
This version: November 17, 2008

Abstract

We deal with several efficient discretization methods for the simulation of the Heston stochastic volatility model. The resulting schemes can be used to calculate all kind of options and corresponding sensitivities, in particular the exotic options that cannot be valued with closed-form solutions. We focus on to the (computational) efficiency of the simulation schemes: though the Broadie and Kaya (2006) paper provided an exact simulation method for the Heston dynamics, we argue why its practical use might be limited. Instead we consider efficient approximations of the exact scheme, which try to exploit certain distributional features of the underlying variance process. The resulting methods are fast, highly accurate and easy to implement. We conclude by numerically comparing our new schemes to the exact scheme of Broadie and Kaya, the almost exact scheme of Smith, the Kahl-Jäckel scheme, the Full Truncation scheme of Lord et al. and the Quadratic Exponential scheme of Andersen.

Keywords: Stochastic volatility, Simulation, Heston, Non-central chi-squared inversion, Control variate.

1 Introduction

The behavior of financial derivatives is usually modeled by stochastic differential equations that (jointly) describe the movements of the underlying financial assets such as the stock prices, stock variances, interest rates or currencies. Though some models yield closed-form solutions for certain derivatives, the fast majority of the exotic options cannot be priced in closed-form. Especially for (forward) path-dependent options, the Monte Carlo approach yields a popular and flexible pricing alternative. Because of the increasingly computational power combined with the use of modern day variance reduction techniques, we expect Monte Carlo techniques to become even more widely applicable in the near future.

Since the introduction of the Black and Scholes (1973) model and in particular since the equity crash of the late eighties a battery of complex models has been proposed to relax some misspecifications of the model. Though the Black and Scholes (1973) model has theoretical and practical appealing

[0] The authors would like to thank Frank de Jong, Richard Plat and participants at the University of Tilburg/Netspar's seminar series for their comments and suggestions.

[1] Netspar/University of Amsterdam, Dept. of Quantitative Economics, Roetersstraat 11, 1018 WB Amsterdam, The Netherlands, e-mail: a.vanhaastrecht@uva.nl

[2] Delta Lloyd Insurance, Risk Management, Spaklerweg 4, PO Box 1000, 1000 BA Amsterdam

[3] Netspar/University of Amsterdam, Dept. of Quantitative Economics, Roetersstraat 11, 1018 WB Amsterdam, The Netherlands, e-mail: a.a.j.pelsser@uva.nl

properties, most of its assumptions, like constant volatility or constant interest rates, do not find justification in the financial markets; one class of models relaxes the constant volatility assumption and incorporates a financial phenomena know as volatility clustering, i.e. they make volatility stochastic. Within this class are the stochastic volatility models of Hull and White (1987), the Stein and Stein (1991) and the Schöbel and Zhu (1999) model. However by far the most popular model stochastic volatility model is the Heston (1993) model, mainly caused by the fact that this model, until the introduction of the Schöbel and Zhu (1999) model, was the only stochastic volatility model that allowed for flexibility over the leverage effect, yet also yielded a closed-form solution for call/put options in terms of one numerical integral[4]. With such a closed-form solution the computation of vanilla European option prices can be done in fast and stable fashion, hence allowing for efficient calibrations to market option data.

Literature review

Despite the fact that the Heston model was already introduced in 1993, there has been relatively little research on efficient discretization methods of its continuous time dynamics. This is in particularly remarkable if one considers that most practical applications of such models, e.g. the pricing and hedging of path-dependent derivatives, practically always involve Monte carlo methods. Only recently a few papers on efficient discretization methods appeared;

first of all a bias-free (discretization) method was introduced in Broadie and Kaya (2006), who developed a scheme that could simulate the Heston process (i.e. stock and variance) from its exact distribution. Though the paper is elegant, it's practical use is unfortunately rather limited: first the variance process is simulated based on an acceptance-rejection method, and secondly the algorithm requires Fourier inversion of the conditional characteristic function of the integrated variance process. Next to the fact the inversion is time-consuming, it is also complex and can lead to numerical errors (e.g. truncation). Moreover the use of acceptance and rejection sampling will scramble random paths when parameters are perturbated, the algorithm results in a low correlation in pre- and post perturbation paths and hence introduces large monte carlo bias in sensitivity analysis (e.g. see Glasserman (2003)). For this reason (popular) low-discrepancy numbers cannot not be applied in conjunction with the Broadie and Kaya (BK) scheme.

Lord et al. (2008) consider different Euler schemes, in particular they investigate how to deal with negative values of the variance process that occur when one uses a direct discretization. The fix that empirically seems to work best is denoted by the Full Truncation (FT) scheme. Though the fix is highly heuristic and uses no known analytical properties of the variance process, the scheme seems to work surprisingly well in comparison to most other schemes. The authors even show that the computational efficiency of this simple Euler scheme outperforms the more advanced BK and the higher order Milstein schemes. However it should be noted that for many relevant parameter configurations[5] the discretization error is still quite high for a practical number of time steps. Hence though the computational efficiency of the FT scheme is better than the BK scheme, the discretization grid still needs to be rather small to obtain an accurate scheme.

Some approximations to the exact schemes are considered in Smith (2008) and Andersen (2007).

[4] The method of the original Heston paper required the calculation of two numerical integrals, whereas some more recent methods require only the evaluation of one numerical integral, e.g. see Carr and Madan (1999), Lord and Kahl (2008) or Lee (2004).

[5] Heston models which are calibrated to main derivative markets, usually have parameter configurations such that variance process has a relatively high probability of reaching the origin. This is often needed a level of skew or kurtosis that is often present in market option prices.

Smith approximates the Fourier inversions required to simulate the integrated variance process. Andersen however focusses on the variance process and develops two efficient schemes which are based on moment-matching techniques. Essentially he approximates the non-centrally chi-squared distribution by a related distribution whose moments are (locally) matched with those of the exact distribution. Since the sampling from the approximated distribution only requires (affine) transformations to uniform and normal draws, his schemes can be implemented quite efficiently and without scrambling random numbers. Next to this approximation Andersen also uses drift interpolation (instead of Fourier inversion) to approximate the integrated variance process. The resulting moment-matched schemes basically outperform all existing schemes in terms of computational efficiency. Especially the Quadratical Exponential (QE(-M)) scheme, turns out to be fast and highly accurate for most practical sizes of the time step.
Though we are aware of the fact that the schemes presented so far certainly do not contain a comprehensive list of the all the available schemes, we feel that the schemes mentioned so far stand out for particular reasons: the BK scheme for its exactness, the Euler scheme with FT fix for its simplicity and the QE-M scheme for its efficiency. For some alternative schemes we refer to Andersen and Brotherton-Ratcliffe (2001) and Glasserman (2003) and the references therein.
Our contribution to the existing literature is twofold. First, we introduce a new and efficient simulation scheme for the Heston stochastic volatility model. We show that the sampling from the (exact) variance process can be done accurately and efficiently: in particular we show that sampling from the non-centrally chi-squared distribution (i.e. the variance process) can effectively be reduced from a three to a one-dimensional inversion procedure. Hence in such a case the inverse distribution function can efficiently be precomputed before the monte carlo run. After discussing the schemes and the benchmark approach, we perform some numerical tests against exact (semi-analytical) option values. Though results can be well compared to those reported in Broadie and Kaya (2006), Andersen (2007) and Lord et al. (2008), we slightly modify the benchmark approach by removing a significant part of the Monte Carlo noise with a simple variance reduction technique, i.e. the asset price as control variate. This enables us to strengthen the numerical results about the accuracy of the simulation schemes.
The setup of the paper is as follows: we first discuss some of the existing Heston simulation schemes. We then introduce a new scheme, based on an efficient sampling method from Heston's variance process. We show numerical tests in which we compare our scheme with existing schemes and finally we conclude.

2 Heston simulation schemes: Euler, Milstein and exact method

To be clear about notations, we shortly formulate the Heston dynamics:

$$\frac{dS(t)}{S(t)} = r(t)dt + \sqrt{v(t)}\,dW^S(t), \qquad S(0) := S_0 \geq 0, \tag{1}$$

$$dv(t) = \kappa(\theta - v(t))dt + \xi\sqrt{v(t)}\,dW^V(t), \qquad v(0) := v_0 \geq 0, \tag{2}$$

with (W_1, W_2) a two-dimensional Brownian motion under the risk-neutral measure Q with instantaneous correlation ρ, i.e.

$$dW^S(t)dW^V(t) = \rho dt. \tag{3}$$

Hence the model parameters are the initial variance $v_0 > 0$, the long run variance $\theta \geq 0$, the mean reversion rate $\kappa \geq 0$, the volatility of the variance $\xi \geq 0$ and the leverage parameter $|\rho_{\mathrm{Sv}}| \leq 1$. Typically, one finds $-1.0 < \rho_{\mathrm{SV}} < -0.6$ implying that the Heston dynamics correlate a down move in the stock

with an up move in the volatility (a phenomenon known as leverage effect). For simplicity we here assume that $r(t)$ is non-stochastic, hence from now on we will write $r(t) \equiv r$.
Since the characteristic function of the log-asset price is known in closed-form for the Heston model, the calibration to vanilla call options can be done efficiently using Fourier inversion, e.g. see Carr and Madan (1999). Please note that in the literature there exists two (theoretically equivalent) formulations of the Heston characteristic function, however as shown in Albrecher et al. (2005) one formulation (as in Heston (1993)) leads to a numerical difficulty called branch cutting, while the other formulation does not have this problem.

2.1 Analytical properties of the variance process

The square root variance dynamics of the Heston was first introduced in a finance (i.e. interest rates) context in Cox et al. (1985); there exist several analytical results for the Feller/CIR/square-root process of (2), for example the variance process is guaranteed to always be greater or equal to zero. Specifically, if $2\kappa\theta \geq \xi$ the Feller condition states that the process can never reach zero (a condition which is however hardly ever satisfied in calibrations to real market data) and for $2\kappa\theta < \xi$ we have that the origin is accessible and strongly reflecting.
The distribution of the variance process is also known; conditional on $v(s)$ $(s < t)$, we have that the variance process is distributed as a constant C_0 times a non-central chi-squared distribution with d degrees of freedom and non-centrality parameter λ, i.e.

$$\mathbb{P}(v(t) \leq x | v(s)) = F_{\chi^2_d(\lambda)}(\frac{x}{C_0}), \tag{4}$$

where $F_{\chi^2_d(\lambda)}(\frac{x}{C_0})$ represents the cumulative distribution of the non-central chi-squared distribution, i.e.

$$F_{\chi^2_d(\lambda)}(z) = \sum_{i=0}^{\infty} \frac{e^{-\frac{\lambda}{2}}(\frac{\lambda}{2})^i}{i!} \frac{\int_0^z z^{\frac{d}{2}} e^{-\frac{u}{2}} du}{\Gamma(i + \frac{k}{2})}, \tag{5}$$

with

$$C_0 := \frac{\xi^2(1 - e^{-\kappa\Delta t})}{4\kappa}, \quad d := \frac{4\kappa\theta}{\xi^2}, \quad \lambda := \frac{4\kappa e^{-\kappa\Delta t} v(s)}{\xi^2(1 - e^{-\kappa\Delta t})} \quad \text{and: } \Delta t := t - s. \tag{6}$$

Hence not from (5) that the non-central chi-squared distribution is equivalent to an ordinary chi-squared G with $d + 2N$ degrees of freedom, where N is a Poisson-distribution with mean $\frac{1}{2}\lambda$. The cumulative distribution of (5) thus can be written in the following form

$$F_{\chi^2_d(\lambda)}(z) = \sum_{i=0}^{\infty} \mathbb{P}(N = i) G_{\chi^2}(z, d + 2i), \tag{7}$$

which will be an important expression in the remainder of this paper. From known properties of the non-central chi-squared distribution (e.g. see Cox et al. (1985) or Abramowitz and Stegun (1964)) we then have that the mean m and variance s^2 of $v(t)$ conditional on $v(s)$ are given by

$$m := \theta + (v(s) - \theta)e^{-\kappa\Delta t}, \tag{8}$$

$$s^2 := \frac{v(s)\xi^2 e^{-\kappa\Delta t}}{\kappa}(1 - e^{-\kappa\Delta t}) + \frac{\theta\xi^2}{2\kappa}(1 - e^{-\kappa\Delta t})^2. \tag{9}$$

While some discretization schemes of the Heston dynamics heavily rely on these properties (e.g. see Broadie and Kaya (2006),Andersen (2007) and Smith (2008)), other schemes do not incorporate the specific distributional properties (e.g. see the Euler and Milstein schemes of Lord et al. (2008) and Kahl and Jäckel (2006)).

2.2 (Log-)Euler scheme

Probably the simplest way to discretize the variance dynamics is by using a first-order Euler scheme. One should however take care on how to fix negative values of the variance process; the handling negative values in the wrong way leads to extremely biased schemes, whereas using the right fix leads to an Euler scheme that outperforms almost all existing schemes in terms of computational efficiency, e.g. see Lord et al. (2008). Since not all literature sources use the proper fix when comparing their scheme with an Euler scheme and the scheme provides a good intuition behind the difficulties of the simulation of the Heston model, we explicitly discuss the Euler scheme here.

Conditional on time s a naive Euler discretization of the variance process for $t > s$ (with $\Delta t := t - s$) reads

$$v(t) = v(s) + \kappa\Delta t(\theta - v(s)) + \xi\sqrt{v(s)}Z_v\sqrt{\Delta t}, \tag{10}$$

with Z_v standard normal distributed. The main source of difficulty in above scheme is that the variance can become negative, explicitly the probability of $v(t)$ becoming negative is

$$\mathbb{P}(v(t) < 0) = \mathbb{P}\Big(Z_v < \frac{-v(s) - \kappa\Delta t(\theta - v(s))}{\xi\sqrt{v(s)}\sqrt{\Delta t}}\Big) = \Phi\Big(\frac{-v(s) - \kappa\Delta t(\theta - v(s))}{\xi\sqrt{v(s)\Delta t}}\Big). \tag{11}$$

Notice that though this probability decays to zero as Δt becomes smaller, it will be strictly positive for any choice of the time step Δt (unless $\xi = 0$). Hence if one does not want the variance process to cross over to the imaginary domain, one has to decide what to do if the variance process turns negative in an Euler scheme. Several ad-hoc fixes for this exist in the literature, for example by making zero an absorbing or reflecting boundary for the variance process. Lord et al. Lord et al. (2008) unify several Euler schemes in the following framework:

$$v(t) = f_1(v(s)) + \kappa\Delta t(\theta - f_2(v(s))) + \xi\sqrt{f_3(v(s))}Z_v\sqrt{\Delta t}, \tag{12}$$

where all schemes should satisfy $f_i(x) = x$ for $x \geq 0$ and $f_3(x) \geq 0$ for all $x \in \mathbb{R}$. This translates into the natural conditions that for positive values of the variance the regular Euler scheme should be employed and that strictly negative values hereof are transformed into positive ones. The most sensible choices for $f_i(x)$ are the identity function ($f(x) = x$), absorption ($f(x) = x^+$) or reflection ($f(x) = |x|$). Since all schemes coincide and are be bias-free as $\Delta t \to 0$, the choice of the fix seems innocent and almost indifferent. 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 not too extreme parameter configurations. The fix that seems to work the best is produced by the so-called Full Truncation (2007) scheme and chooses $f_1(x) := x$, $f_2(x) = f_3(x) := \max(x, 0) = x^+$, see Lord et al. (2008). The resulting Euler scheme reads

$$v(t) = v(s) + \kappa\Delta t(\theta - v(s)^+) + \xi\sqrt{v(s)^+}Z_V\sqrt{\Delta t}. \tag{13}$$

Hence provided with a discretization scheme for the variance process, we also need to specify the simulation schemes of the corresponding asset price process. The most straightforward choices would be to either directly apply an Euler discretization scheme to the stock price process of equation (1) or to simulate the stock price from its exact (conditional) distribution. Direct discretization yields the following **Euler scheme**

$$S(t) = S(s)\Big(1 + r\Delta t + \sqrt{f_5(v(t))}Z_S\sqrt{\Delta t}\Big) \tag{14}$$

and does entails some discretization error of the exact process. Here Z_S is a normal distributed random variable (with correlation ρ to Z_V) and $f_5(x)$ should be chosen non-negative.

Alternatively one can also use the exact solution of the stock price dynamics (1), which by an application of Ito's lemma is given by

$$S(t) = S(s) \exp\Big[\int_s^t [r - \frac{1}{2}v(u)]du + \int_s^t \sqrt{v(u)}\, dW_S(u)\Big]. \tag{15}$$

Hence taking logarithms and discretizing in an Eulerly fashion, one obtains the following **log-Euler scheme**

$$\log(S(t)) = \log(S(s)) + \Big[r - \frac{1}{2}f_4(v(s))\Big]\Delta t + \sqrt{f_5(v(s))}\, Z_S \sqrt{\Delta t}. \tag{16}$$

The above described log-Euler scheme does not entail any discretization error in the stock price direction, of course the scheme usually does show biases in the Euler discretization the variance process (and thus in resulting stock prices). Following Lord et al. (2008) we choose to set $f_3(x) = f_4(x) = f_5(x) = x^+$, which seem to be the most logical choices, since the Ito correction term of equation (16) is then consistent with the corresponding the volatility of the stock price, hence this implies the martingale condition of the stock price process is preserved in the discretization. In an implementation the correlated standard normal random variables Z_V and Z_S can (for example) be generated with the use of a Cholesky decomposition: with a (instantaneous) correlation of ρ between the driving Brownian motions this can be done 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.

Note that the pure Euler scheme (14) can be seen as a first order approximation of above log-Euler scheme. Since the log-Euler scheme entails no discretization error in the stock price direction, we prefer to work under this log transformation when employing an Euler scheme, e.g. see also Lord et al. (2008)). Additionally since the full truncation scheme seems to have the smallest bias among all Euler schemes, we adopt this fix for possible negative values of the variance process when using an Euler scheme. The main advantage of the Euler scheme lies its simplicity and speed: little code and computing time is needed to compute one iteration in the scheme. Additionally the use of the scheme is not restricted to the Heston model, but can also be applied to all kind models, for example to the family of CEV-processes Lord et al. (2008). Its generality also implies its weakness: the Euler scheme doesn't use any information of known analytical properties of the square root variance process.

Full truncation algorithm

Using a log-Euler scheme for the stock price process, the full truncation scheme for the Heston can be summarized by the following algorithm:

1. Generate a random sample Z_1 from the standard normal distribution[6] and set $Z_v := Z_1$.

2. Given $v(s)$, compute $v(t)$ from equation (13).

3. Generate a random sample Z_2 from the standard normal distribution and set

$$Z_S := \rho_{SV} Z_V + \sqrt{1-\rho_{SV}^2}\, Z_2. \tag{17}$$

4. Given $\log(S(s))$, compute $\log(S(t))$ using equation (16).

[6] It may be advisable to use an inversion method for generating of normal samples, since then also a quasi random generator can be used. This inversion over an uniform random variable with the ('approximated') inverse standard normal distribution function can for example be done using Wichura's method, see Wichura (1998).

2.3 Kahl-Jäckel Scheme

A generic implicit Milstein scheme for the variance process in combination with an alternative discretization for the stock price was suggested in Kahl and Jäckel (2006), i.e. the following discretization scheme was proposed:

$$\begin{aligned} v(t) &= \frac{v(s) + \kappa\theta\Delta t + \xi\sqrt{v(s)}Z_V\sqrt{\Delta t} + \frac{1}{4}\xi^2\Delta t(Z_V^2 - 1)}{1 + \kappa\Delta t} && (18) \\ \log(S(t)) &= \log(S(s)) + \Big[r - \frac{(v(s) + v(t))}{4}\Big]\Delta t + \rho\sqrt{v(s)}Z_V\sqrt{\Delta t} \\ &\quad + \frac{1}{2}(\sqrt{v(s)} + \sqrt{v(t)})(Z_S + \rho Z_V)\sqrt{\Delta t} + \frac{\rho\xi\Delta t}{2}(Z_V^2 - 1) && (19) \end{aligned}$$

Kahl-Jäckel show that this scheme results in positive paths for the variance process for $2\kappa\theta > \xi$, a condition which is hard to meet in practice. Hence in many practical implementations of the above dynamics, one has to decide on how to fix negative values of the variance process. Since Kahl and Jäckel (2006) do not specifically tackle this issue, we follow Andersen (2007) who adopts the same fix as Lord et al. (2008) use in the full truncation Euler scheme. That is, whenever the variance process drops below zero, we use (13) rather than (18) and take $v(s)^+$ and $v(t)^+$ rather than using $v(s)$ and $v(t)$ in (19). The resulting algorithm is similar to the ft-algorithm 2.2: one just replaces the variance and asset process from (2) and (4) with the above defined discretizations for the variance and asset process.

2.4 Exact scheme of Broadie and Kaya

In an elegant paper, Broadie and Kaya (2006) worked out an exact simulation scheme for the Heston model. Though theoretically the method is exact, its practical use is limited; the scheme suffers from a lack of speed, it is complex and sensitivity calculations (often used for risk management) are hard since the scheme relies on acceptance and rejection sampling techniques. For example, the numerical tests in Lord et al. (2008) show that for most practical situations even a simple Euler scheme outperforms the exact scheme in terms of computational efficiency. [7]

Though in most practical situations a direct implementation of the exact scheme is probably not the best available option (see section 2.5), there are some approximations or computational tricks that can be made to improve upon the computational efficiency. For example, Andersen (2007) and Smith (2008) both use the exact scheme as starting point and from there on try to improve upon some of the incorporated bottlenecks.

We will first discuss the exact method and its incorporated difficulties: by using the explicit solution (15) of the asset price process and consecutively using Ito's lemma and using a Cholesky decomposi-

[7] Note that in the numerical results of Broadie and Kaya (2006) and Smith (2008), an Euler scheme is used that handles negative values of the variance in a suboptimal way. However as shown in Lord et al. (2008) the choice on how to cope with negative values of the variance process is extremely important for the quality (i.e. bias) of the scheme. Because the (semi-)exact schemes in Broadie and Kaya (2006) and Smith (2008) are benchmarked against a suboptimal Euler scheme, this leads them to a false conclusion in comparing their schemes against 'the' Euler scheme. This was point was first noted in Lord et al. (2008) and later on in Andersen (2007). From their results in can for example be seen that the Euler scheme (with the 'right' fix) outperforms the exact and Kahl-Jäckel scheme in terms of computational efficiency, whereas in Broadie and Kaya (2006) and in Smith (2008) (who use suboptimal fixes) this is just the other way around.

tion one obtains

$$\begin{aligned}\log(S(t)) &= \log(S(s)) - \frac{1}{2}\int_s^t v(u)du \\ &\quad +\rho\int_s^t \sqrt{v(u)}\,dW_v(u) + \sqrt{1-\rho^2}\int_s^t \sqrt{v(u)}\,dW(u),\end{aligned} \tag{20}$$

where $W(u)$ is a Brownian motion independent of $W_v(u)$. Integrating the square-root variance process of equation (2) gives the following solution:

$$v(t) = v(s) + \int_s^t \kappa(\theta - v(u))du + \xi\int_s^t \sqrt{v(u)}\,dW_v(u), \tag{21}$$

or equivalently

$$\int_s^t \sqrt{v(u)}\,dW_v(u) = \xi^{-1}(v(t) - v(s) - \kappa\theta\Delta t + \kappa\int_s^t v(u)du). \tag{22}$$

In Broadie and Kaya (2006), it is then noticed that one can substitute equation (22) into the solution (20) to arrive at

$$\begin{aligned}\log(S(t)) &= \log(S(s)) + \frac{\kappa\rho}{\xi}\int_s^t v(u)du - \frac{1}{2}\int_s^t v(u)du \\ &\quad +\frac{\rho}{\xi}(v(t) - v(s) - \kappa\theta\Delta t) + \sqrt{1-\rho^2}\int_s^t \sqrt{v(u)}\,dW(u),\end{aligned} \tag{23}$$

hence an exact simulation involves sampling from the following three stochastic quantities:

1. $v(t)|v(s)$: from (4) and (6) one use that $v(t)|v(s)$ is distributed as a constant C_0 times a non-central chi-squared distribution with d degrees of freedom and non-centrality parameter λ.

2. $\int_s^t v(u)du|v(s), v(t)$: Broadie and Kaya (2006) derive the characteristic function

$$\begin{aligned}\Psi\big(a, v(s), v(t)\big) &= \mathbb{E}\Big[\exp(ia\int_s^t v(u)du)\Big|v(s), v(t)\Big] \\ &= \frac{\gamma(a)e^{\frac{1}{2}(\gamma(a)-\kappa)(t-s))}(1-\exp(-\kappa(t-s))}{\kappa(1-e^{-\gamma(a)(t-s)})} \\ &\quad \times\exp\Big[\frac{v(s)+v(t)}{\xi^2}\Big(\frac{\kappa(1+e^{-\kappa(t-s)}}{1-e^{-\kappa(t-s)}} - \frac{\gamma(a)(1+e^{-\gamma(a)(t-s)})}{1-e^{-\gamma(a)(t-s)}}\Big)\Big] \\ &\quad \times\frac{I_{\frac{1}{2}d-1}\Big[\sqrt{v(s)v(t)}\,4\gamma(a)e^{-\frac{\gamma(a)}{2}(t-s)}/\xi^2(1-e^{-\gamma(a)(t-s)})\Big]}{I_{\frac{1}{2}d-1}\Big[\sqrt{v(s)v(t)}\,4\gamma(a)e^{-\frac{\kappa}{2}(t-s)}/\xi^2(1-e^{-\kappa(t-s)})\Big]},\end{aligned} \tag{24}$$

   with $\gamma(a) := \sqrt{\kappa^2 - 2\xi^2 ia}$, $d := \frac{4\kappa\theta}{\xi^2}$ and where $I_\nu(x)$ is the modified Bessel function of the first kind. Hence the characteristic function (24) can numerically be inverted to obtain the value of the distribution function $G(x)$ for a certain point $x \in \Omega$, i.e.

$$G(x, v(s), v(t)) = \frac{2}{\pi}\int\limits_0^\infty \frac{\sin(ax)}{x}\mathrm{Re}\Big[\Psi(a, v(s), v(t))\Big]da. \tag{25}$$

Finally to generate sample from $\int_s^t v(u)du|v(s), v(t)$ one can use

$$G(\int_s^t v(u)du|v(s), v(t)) = U, \tag{26}$$

and invert G over a uniform random sample U to find $x_i : x_i = G^{-1}(U, v(s), v(t))$, e.g. by a Newton-Raphson root search of $G(x_i, v(s), v(t)) - U = 0$. Note that such a root finding procedure involves multiple Fourier inversions: one for each evaluation of $G(x_i, v(s), v(t))$.

3. $\int_s^t \sqrt{v(u)}dW(u)| \int_s^t v(u)du$: since $v(u)$ is independent of $W(u)$, it directly follows that the this expression is distributed as $N(0, \int_s^t v(u))$. Hence this sampling can be done easily and efficient by sampling from a normal distribution.

Broadie and Kaya algorithm

Exact simulation of (23) is feasible and can be performed by the following algorithm:

1. Conditional on $v(s)$, use the definitions of (6) to generate a sample of $v(t)$ by sampling from a constant times a non-central chi-squared distribution with d degrees of freedom and non-centrality parameter λ.

2. Conditional on $v(s)$ and $v(t)$, generate a sample of $\int_s^t v(u)du$ by a numerical inversion of the distribution function G of $\left(\int_s^t v(u)du \,\middle|\, v(s), v(t)\right)$ over a uniform sample U, for example by a root search $G(x_i, v(s), v(t)) - U = 0$. Since the distribution function G is not known is closed form, $G(x_i, v(s), v(t))$ has to be obtained by Fourier inverting the characteristic function of $\int_s^t v(u)du|v(s), v(t)$.

3. Use (22) to set:

$$\int_s^t \sqrt{v(u)}dW_v(u) = \xi^{-1}(v(t) - v(s) - \kappa\theta\Delta t + \kappa \int_s^t v(u)du) \tag{27}$$

4. Generate an independent random sample Z_S from the standard normal distribution and use the fact that $\int_s^t \sqrt{V(u)}dW(u)$ is normally distributed with mean zero and variance $\int_s^t V(u)du$ and thus can be generated as

$$\int_s^t \sqrt{V(u)}dW(u) \sim Z_S \sqrt{\Delta t \int_s^t V(u)du}\,, \tag{28}$$

5. Given $\log(S(s))$,$\int_s^t \sqrt{v(u)}dW_v(u)$, $\int_s^t \sqrt{V(u)}dW(u)$ and $\int_s^t V(u)du$ use (23) to compute $\log(S(t))$.

2.5 Disadvantages of the exact scheme

Though the Broadie and Kaya scheme is theoretically appealing (and this was probably also the primary objective of their paper), we will discuss in the following section why its practical use might be limited. That is, we discuss some practical implementation issues that incorporated with the use of the exact scheme;

first of all, (2.4-1) requires that the variance process $v(t)|v(s)$ has to be sampled from a constant C_0 times a non-central chi-squared distribution with d degrees of freedom and non-centrality parameter λ (see):

$$v(t) \overset{d}{=} C_0 \chi_d^2(\lambda), \tag{29}$$

For simulation purposes one can use the following representations of the non-central chi-squared distribution (see Johnson et al. (1994) and Glasserman (2003)):

$$\chi_d^2(\lambda) \overset{d}{=} \begin{cases} (Z + \sqrt{\lambda})^2 + \chi_{d-1}^2 & \text{for } d > 1, \\ \chi_{d+2N}^2 & \text{for } d > 0, \end{cases} \tag{30}$$

with $Z \sim N(0,1)$, χ_ν^2 an ordinary chi-squared distribution with ν degrees of freedom and where N is Poisson distributed with mean $\mu := \frac{1}{2}\lambda$. Since in most practical applications $d << 1$, one is usually forced to work with the second representation of the non central chi-squared distribution[8]; hence exact sampling from the variance process can be done by first conditioning on a Poisson variate and consecutively generating a sample from a chi-squared or gamma distribution[9]. Since direct inversion of the gamma distribution is relatively slow, Broadie and Kaya (2006) suggest to use an acceptance and rejection method to generate gamma variates. Though such sampling can be done fairly quick (e.g. by making use of some recent advances of Marsaglia and Tsang (2000)), the methods are still relatively slow in comparison to sampling methods for normal variates.

Moreover the main disadvantage of acceptance and rejection techniques is that the (number of) samples depend on the specific Heston parameters. As a consequence the total drawings of random numbers cannot be predetermined and sample paths will show a rather small correlation coefficient for different parameter inputs. These properties are usually inconvenient in financial applications, since both perturbation analysis[10] (to calculate model sensitivities with respect to different parameters) as well as the use of quasi random numbers generator becomes extremely hard, not to say practically almost impossible.

Another practical difficulty of the scheme lies in step (2.4-1), where one has to generate a sample of $\int_s^t v(u)du|v(s), v(t)$ by numerically inverting the distribution function of $(\int_s^t v(u)du|v(s), v(t))$ over an uniform random variable u, by a root search of $G(x_i, v(s), v(t)) - U = 0$. However because the distribution function G is not known is closed form, it has to be obtained by Fourier inverting the characteristic function (24), which contains two modified Bessel functions that each represent an infinite series. The root-finding procedure (and the involved Fourier inversions) has to repeated several times until a tolerance level ε is reached for a guess x_i, such that $G(x_i, v(s), v(t)) - U < \varepsilon$. Next to the fact that both in the evaluation of (24) as well as the required Fourier inversions require a great computational effort, the implementation of this step also has to be done with great care to avoid noticeable biases from the numerical inversions.

[8] Otherwise, if $d > 1$, one might want to use the first representation, since sampling from the normal distribution is usually more efficient than sampling from a Poisson distribution.

[9] The Chi-squared distribution is a special case of the gamma distribution, $\chi_\nu^2 \overset{d}{=} \text{gamma}(\frac{\nu}{2}, 2)$, where gamma$(k, \theta)$ is a gamma distribution with shape k and scale θ.

[10] The efficiency in the calculation of model sensitivities crucially depends on the size of the correlation coefficient between pre- and post perturbation paths.

3 Approximations to the exact scheme

As elaborated in section 2.5 the exact scheme has some practical disadvantages. However it does provide an extremely well base to construct some approximate schemes which might be more practically and computationally more efficient. A few authors have already tried to improve the bottlenecks in simulating the variance and/or integrated variance process, e.g. see Andersen (2007) and Smith (2008). In the remainder of this section we will unify and discuss the two methodologies that can improve upon the performance of the Broadie and Kaya scheme. That is, we consider approximations of:

1. The integrated variance process.
2. The variance process itself.

Moreover we will look at schemes that combine the latter approximations.

Approximating the integrated variance distribution

As elaborated in section 2.5, a huge bottle neck of the simulation scheme is the sampling of the conditional integrated variance process. There are however several ways to approximate a sample from the integrated variance process $\int_s^t v(u)du \,\big|\, v(s), v(t)$.

1. **Drift interpolation:** Without using any specific information of the integrated variance process, one can use a drift interpolation method to approximate the integrated variance process, i.e.

   $$\int_s^t v(u)du \,\Big|\, v(s), v(t) \;\approx \gamma_1 v(s) + \gamma_2 v(t), \tag{31}$$

   for some constants γ_1, γ_2, which can be set in several ways: an Euler-like setting would read $\gamma_1 = 1, \gamma_2 = 0$, while a mid-point rule corresponds to the predictor-corrector setting $\gamma_1 = \gamma_2 = \frac{1}{2}$.

2. **Approximate the Fourier inversion:** One can also try to approximate the Fourier inverted sampling of the integrated variance process. For example Smith Smith (2008) tries to speed up the inversion of the characteristic function (24) by caching values of a projected version hereof. Though such a method might speed the inversion, one still has to use a rather time-consuming Fourier inversion combined with a root finding procedure to draw a sample of the integrated process. Alternatively one can try to use the first moments of the conditional integrated variance process (which can be obtained by differentiating the cf. of (24)) to develop a moment-matched sampling method.

Approximating the variance process

Another (practical) disadvantage of the exact scheme is the use of acceptance rejection sampling method for the non-central chi-squared distributed variance process (see section 2.5). Hence we consider two methods that can be used to approximate the variance distribution.

1. **Moment-matching:** Andersen (2007) suggests to approximate the variance process by related distributions whose first two moments are (locally) matched with those of the true variance distribution. Moreover, since the distributions can be analytically inverted, the methods can be directly used in conjunction with perturbation and low-discrepancy methods by straightforward inversions a uniform random variates.

2. **Direct inversion:** To overcome the acceptance and rejection sampling method, one can also use direct inversion of the non-central chi-squared distribution to generate a sample of the variance process. However since no analytical expression exists for this inverse, one has to use a (time-consuming) root finding procedure to numerical invert the distribution. We will show however, that can efficiently create a cache of inverses from which a sample of the non-central chi-squared distribution can be generated (i.e. looked up from a table) in two simple steps.

3.1 Broadie and Kaya Drift Interpolation scheme

Probably the easiest way to give the exact scheme a performance boost is to approximate the Fourier inverted sampling of the integrated variance process by the simple drift interpolation method of equation (31). Moreover since the sampling of the integrated variance process is most time-consuming step of the exact scheme, one can expect a large efficiency gain. The simulation of the Broadie and Kaya Drift Interpolation (BK-DI) scheme is straightforward; in the exact scheme of 2.4, one only has to replace the sampling of the integrated variance process in step 2 by the drift interpolation rule (31). Hence though the resulting method is simple and reasonable efficient, sampling from the variance process is still performed by an acceptance-rejection method, which (as discussed in section 2.5) is rather inconvenient for most financial applications. We also like to note that though for reasonable time-spacings the drift approximation error is usually rather small, one does slightly violate the discrete-time no-arbitrage condition, i.e. the discretized stock price is not exactly a martingale. In section 4.1 we show how one can enforce this condition with the above discussed discretization method.

3.2 Almost Exact Fourier inversion scheme

Smith (2008) tries to speed up the inversion of the characteristic function (24) by caching values of a projected version hereof. The core of the almost exact simulation method (AESM) in Smith (2008) is to project the exact characteristic function $\Psi(a, v(s), v(t))$, which depends on $v(s)$ and $v(t)$ via the arithmetic and geometric mean $\frac{1}{2}(v(s) + v(t))$ and $\sqrt{v(s)v(t)}$, onto a function $\widetilde{\Psi}(a, z)$ in which the dependency on the means is approximated by the combination

$$z = \omega \frac{1}{2}(v(s) + v(t)) + (1 - \omega) \sqrt{v(s)v(t)}, \quad 0 \leq \omega \leq 1, \tag{32}$$

for a suitable choice of ω. Hence the arithmetic and geometric mean, which are similar in expectation, are replaced by a weighted average of the two. In this way the three-dimensional function $\Psi(a, v(s), v(t))$ is approximated by the two-dimensional function

$$\begin{aligned} \widetilde{\Psi}(a, z) &= \frac{\gamma(a) e^{\frac{1}{2}(\gamma(a)-\kappa)(t-s))}(1 - \exp(-\kappa(t-s))}{\kappa(1 - e^{-\gamma(a)(t-s)})} \\ &\quad \times \exp\Big[\frac{2z}{\xi^2}\Big(\frac{\kappa(1 + e^{-\kappa(t-s)}}{1 - e^{-\kappa(t-s)}} - \frac{\gamma(a)(1 + e^{-\gamma(a)(t-s)})}{1 - e^{-\gamma(a)(t-s)}}\Big)\Big] \\ &\quad \times \frac{I_{\frac{1}{2}d-1}\Big[z 4\gamma(a) e^{-\frac{\gamma(a)}{2}(t-s)} / (\xi^2(1 - e^{-\gamma(a)(t-s)}))\Big]}{I_{\frac{1}{2}d-1}\Big[z 4\gamma(a) e^{-\frac{\kappa}{2}(t-s)} / (\xi^2(1 - e^{-\kappa(t-s)}))\Big]}, \end{aligned} \tag{33}$$

which can then be cached on a sufficiently small discrete (two-dimensional) grid of a and z-points. Though Smith claims that the approximation works well, the implementation still requires a time-consuming root search of Fourier inversions for each time step. Hence though the evaluation of the characteristic can be approximated in an computationally efficient way, the root search and inversion are still rather time-consuming in comparison with a simple drift interpolation method. Additionally the total algorithm has to be implemented with great care to avoid numerical truncation and discretization errors.

3.3 Quadratic Exponential scheme

In the Quadratic Exponential (QE) scheme, Andersen (2007) suggests to approximate the sampling from the non-central chi-squared distribution is approximated by a draw from a related distribution, which is moment-matched with the first two (conditional) moment of non-central chi-squared distribution. The choice of distribution is split up into two parts, which are based on the following observations with respect to the size of the non-centrality parameter (e.g. see Abramowitz and Stegun (1964)):

1. For a moderate of high non-centrality parameter the non-central chi-squared can be well represented by a power function applied to a Gaussian variable (which is equivalent to a non-central chi-squared distribution with one degree of freedom). For sufficiently high values of $v(s)$, a sample of $v(t)$ hence can be generated by

$$v(t) = a(b + Z_v)^2, \tag{34}$$

   where Z_v is standard normal distributed random variable and a and b are constants to be determined by moment-matching.

2. For sufficiently low values of $v(s)$, the density of $v(t)$ can (asymptotically) be approximated by

$$\mathbb{P}\Big(v(t) \in [x, x + dx]\Big) \approx (p\delta(0) + \beta(1 - p)e^{-\beta x})dx, \quad x \geq 0, \tag{35}$$

   where δ represents Dirac's delta function, and p and β are non-negative constants.

The constants a,b,p,β can (locally) be chosen such that the first two moments of the approximate distribution matches those of the exact one. These constants depend on the size of the time-step Δt,$v(s)$, as well as on Heston's model parameters. Sampling from these distributions can be done in a simple and efficient way:

- From the first distribution one only has to draw a standard normal random variable and apply the quadratic transformation of equation (34).
- Sampling according to equation (35) can be done by inversion of the distribution function; The distribution function is obtained by integrating the probability density function, and can consecutively be inverted to obtain the following inverse distribution function:

$$L^{-1}(u) = \begin{cases} 0 & \text{if } 0 \leq u \leq p, \\ \beta^{-1}\log(\frac{1-p}{1-u}) & \text{if } p < u \leq 1. \end{cases} \tag{36}$$

  Using the inverse distribution function sampling method, one obtains an easy and efficient sampling scheme by first generating a uniform random number U_v and then setting

$$v(t) = L^{-1}(U_v) \tag{37}$$

Together, the equations (34) and (37) define the Quadratic Exponential (QE) discretization scheme. What yet remains is the determination of the moment-matching constants a,b,p and β, as well as a rule that defines 'high and low values' of the non-centrality parameter, i.e. a rule that determines when to switch from (34) and (37). We first discuss the latter:

recalling that the conditional mean and variance of the square-root process are given by m and s^2 as defined in equations (8) and (9). Andersen then bases the switching rule on the value of ψ

$$\psi := \frac{s^2}{m^2} = \frac{\frac{v(s)\xi^2 e^{-\kappa\Delta t}}{\kappa}(1-e^{-\kappa\Delta t}) + \frac{\theta\xi^2}{2\kappa}(1-e^{-\kappa\Delta t})^2}{(\theta + (v(s)-\theta)e^{-\kappa\Delta t})^2} = \frac{C_1 v(s) + C_2}{(C_3 v(s) + C_4)^2}, \tag{38}$$

with

$$C_1 = \frac{\xi^2 e^{-\kappa\Delta t}}{\kappa}(1-e^{-\kappa\Delta t}), \qquad C_2 = \frac{\theta\xi^2}{2\kappa}(1-e^{-\kappa\Delta t})^2, \qquad C_3 = e^{-\kappa\Delta t}, \qquad C_4 = \theta(1-e^{-\kappa\Delta t}).$$

Note that ψ is inversely related to the size of the non-centrality parameter. It can be shown that for $\psi \leq 2$ the quadratic transformation (34) can be moment-matched with the exact distribution and for $\psi \geq 1$ the exponential one of (37). Thus for $\psi \leq 2$, we can moment match the quadratic sampling scheme (34) and for $\psi \geq 1$ and we can moment match the exponential scheme (37). Since these domains overlap, at least one of the two methods is applicable. A natural procedure is then to introduce some critical level $\psi_c \in [1, 2]$, and use (34) if $\psi \leq \psi_c$ and (37) otherwise. Following Andersen, who notes that the exact choice of ψ_c has a relatively small impact on the quality of the overall simulation scheme, we use $\psi_c = 1.5$ in the numerical tests.

Notice though ψ (locally) has to be calculated for every step in a simulation and contains 'computational expensive' components (e.g. the exponent $\exp(-\kappa\Delta t)$) many of these terms only depend on the size of time step. From efficiency considerations it is therefore advisable to pre-cache the static constants $C_1, \ldots, C_4$ before the Monte Carlo simulation starts. In the case one uses a non-equidistant time grid different constants of course need to be cached for every applicable size of the time step.

The moment-matching constants a,b,p and β of the just defined sampling schemes still have to be specified, and should be chosen such that the first two (conditional) moments are matched with the first and second central moment m and s^2 of the exact non-central chi-squared distribution. The following statements hold regarding the conditional moments of the schemes (34) and (37)

1. For $\psi \leq 2$, setting

$$b^2 = 2\psi^{-1} - 1 + \sqrt{2\psi^{-1}}\sqrt{2\psi^{-1} - 1} \geq 0, \tag{39}$$

$$a = \frac{m}{1+b^2}, \tag{40}$$

assures that the first two moments of the sampling scheme (34) are matched with the exact moments non-central chi-squared distribution, see Andersen (2007), proposition 5, pp.14.

2. For $\psi \geq 1$, setting

$$p = \frac{\psi - 1}{\psi + 1} \in [0, 1), \tag{41}$$

$$\beta = \frac{1-p}{m} = \frac{2}{m(\psi+1)} > 0, \tag{42}$$

assures that the first two moments of the sampling scheme (37) are matched with the exact moments non-central chi-squared distribution, see Andersen (2007), proposition 6, p.15.

QE Algorithm

Assuming that some critical switching level $\psi_c \in [1,2]$ and values for $\gamma_1, \gamma_2 \in [0,1]$ have been selected, the Quadratic Exponential variance sampling can be summarized by the following algorithm:

1. Given $v(s)$, compute m and s^2 and $\psi = \frac{m^2}{s^2}$ using equations (8) and (9).
2. **If** $\psi \leq \psi_c$:
   (a) Compute a and b from equations (40) and (39).
   (b) Generate a random sample Z_v from the standard normal distribution.
   (c) Use (34), i.e. set $v(t) = a(b + Z_v)^2$.

   Otherwise, if $\psi > \psi_c$:

   (a) Compute β and p according to equations (41) and (42).
   (b) Draw a uniform random number U_v.
   (c) Use (37), i.e. set $v(t) = L^{-1}(U_v)$ where L^{-1} is given in (36).

3.4 Non-central Chi-squared Inversion scheme

Instead of using moment-matched schemes, another way to circumvent the acceptance and rejection technique is to use a direct inversion of the Non-central Chi-Squared distribution. We will call this new scheme the Non-central Chi-squared Inversion (NCI) scheme; since direct inversion is too slow, another solution could be to design a three-dimensional cache of the inverse of the non-centrally chi-squared distribution function $F^{-1}(x, v, \lambda)$, which can be created by a root finding procedure of the distribution function. Though this method was already suggested by Broadie and Andersen, they comment that because the parameter space is potentially extremely large in a three-dimensional cache, that such 'brute-force' caching will have its own challenges, like its dimension and the design of inter- and extrapolation rules. Therefore Andersen does not pursue this way, but continues on the development of moment-matched schemes. In the following we however show that the three-dimensional parameter space can effectively be projected onto an one dimensional search space. This one-dimensional cache can then be created and used in an efficient fashion; the overhead of the one-dimensional cache is low, while the simulation of the variance process can be done fast and by simple (linear) interpolation of two values of the cache over an uniform random variable. Moreover since the total number of uniform draws is fixed (and independent of the Heston parameters), this new method can directly be used in conjunction with perturbation analysis and low-discrepancy numbers.

A Poisson conditioned caching method

Recall from (2.5) and (6) that the exact distribution of the variance process is a constant times a non-central chi-squared distribution, for which representation (30) can be used, i.e.

$$v(t)|v(s) \overset{d}{=} C_0 \chi^2_{d+2N} \quad \text{for } d > 0, \tag{43}$$

with and N a Poisson distribution[11] with mean $\mu = \frac{1}{2}\lambda$ and with (see (6)):

$$d = \frac{4\kappa\theta}{\xi^2}, \quad \mu = \frac{1}{2} C_5\, v(s), \quad \text{and } C_5 := \frac{2\kappa e^{-\kappa\Delta t}}{\xi^2(1 - e^{-\kappa\Delta t})}. \tag{44}$$

[11] Recall: $\mathbb{P}(N = n) = \frac{\mu^n e^{-\mu}}{n!}, \quad n = 0, 1, 2, \ldots$.

Thus sampling from the non-centrally chi-squared distributed variance process is equivalent to sampling from a 'Poisson-conditioned' chi-squared distribution. Though this observation was already being used in the Broadie and Kaya scheme, our (yet to be described) sampling method is different. We claim our method is more efficient and better applicable in financial applications; moreover our sampling method for the variance scheme can either be used on its own or can be used as drop in for the variance sampling of the exact or almost exact scheme of Broadie and Kaya (2006) or Smith (2008). In the following sections we first describe the Poisson-sampling method and consecutively show how one can exploit a property of this distribution, which can enable you to create an efficient cache (and corresponding sampling method) of the non-central chi-squared distribution.

Poisson sampling

Notice that the mean μ of the Poisson distribution depends on the size of the time step (through C_5) as well as on the current state of the variance process $v(s)$; for almost all practical model configurations one finds $\mathbb{E}[\mu] << 10$, for which the corresponding Poisson-distribution decays quite rapidly and has rather 'thin' tails.[12] This implies that we can (efficiently) draw a sample N_j from a Poisson distribution with a relatively small mean μ by just inverting its distribution function over an uniform random variable (e.g. see Knuth Knuth (1981) and Ahrens and Dieter (1982)):

1. Draw a uniform random number U_P, set $N_j = 0$ and $P_{N_j} = P_C = \exp(-\mu)$.
2. **While** $P_C \leq U_P$:
   $N_j \rightarrow N_j + 1$, $P_{N_j} \rightarrow P_{N_j} \cdot \frac{\mu}{N_j}$ and $P_C \rightarrow P_C + P_{N_j}$.

Hence for small μ the above inversion algorithm is very efficient, since most of probability mass lies within the first values of the distribution, i.e. see figure 1.

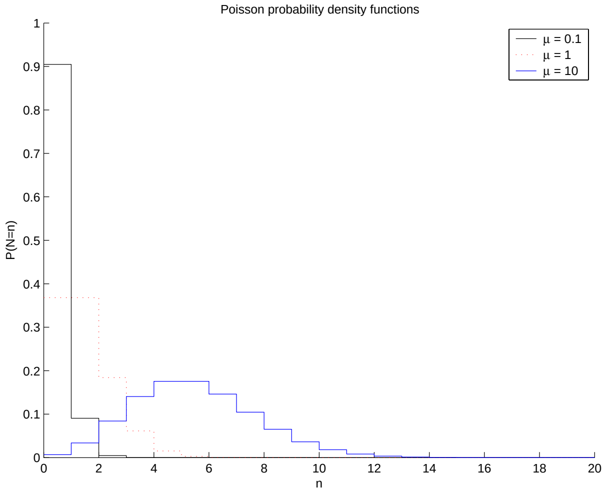

Figure **1**: Poisson probability density functions for different μ.

Next to the fact that the thin tail of the Poisson distribution enables us to efficiently invert the Poisson distribution, it implies that we can create a cache for the non-central chi-squared distribution by precomputing chi-squared distributions for a truncated set Poisson-outcomes; since there is a little

[12] For all $\mu < 10$, $\mathbb{P}(N > 35) < 4.462 \cdot 10^{-11}$.

probability mass in the tail of the Poisson-distributions that one encounters during a simulation, the truncation error usually is negligible.

Caching the Chi-squared distributions

We first introduce some notation: let $N_{\max}$ represent a certain threshold level (e.g. such that $\mathbb{P}(N > N_{\max}) < \varepsilon$) and let

$$\mathcal{N} := \left\{0, \ldots, N_j, \ldots, N_{\max}\right\} \tag{45}$$

represent the set of Poisson-values for which we will cache the inverse of the corresponding conditional chi-squared distributions (i.e. according to 43). Since the inverse lives on the uniform domain, we let $\mathcal{U}_{N_j}$ represent a corresponding sorted set of uniform variables for which we the inverse $(\chi^2_{d+2N_j})^{-1}(\cdot)$ is calculated[13], i.e.

$$\mathcal{U}_{N_j} := \{0, \ldots, 1-\delta\}. \tag{46}$$

Thus we suggest to create a cache of the values of the inverse of the non-central chi-squared distribution function by means of conditioning on a truncated range of Poisson-values and precomputing the corresponding chi-squared distribution functions, i.e. we precompute

$$H^{-1}_{N_j}(U_i) := G^{-1}_{\chi^2_{d+2N_j}}(U_i), \quad \forall N_j \in \mathcal{N}\ \forall U_i \in \mathcal{U}_{N_j}, \tag{47}$$

with $G^{-1}_{\chi^2_{d+2N_j}}$ the inverse chi-squared distribution with $d+2N_j$ degrees of freedom.

Generating a sample from the variance process

From (6) we know that $v(t)|v(s)$ is distributed as a constant C_0 times a non-central chi-squared distribution, we can use the results of the previous subsection and sample from the variance process by first conditioning on a Poisson variable N_j and consecutively inverting the corresponding chi-squared distribution. To invert the chi-squared distribution for $N_j \leq N_{\max}$, we just draw a uniform variate and interpolate between the two values of the inverse distribution cache that surround the uniform numbers. In case $N_j > N_{\max}$ we use the distribution corresponding to $N_{\max}$ and moment-matching techniques which we explain below.

The caching method (47) and the following sampling rule form the core of the NCI scheme. That is, draw a Poisson number N_j and a uniform random number U_V (e.g. $U_i < U_V < U_{i+1}$) then a sample of $v(t)|v(s)$ is generated by

$$v(t) = F^{-1}_{N_j}(U_V), \tag{48}$$

with

$$F^{-1}_{N_j}(U_V) := \begin{cases} C_0 J(U_V) & \text{for } N_j \leq N_{\max}, \\ C_0 F^{-1}_{\chi^2_{d+2N_j}}(U_V) & \text{for } N_j > N_{\max}, \end{cases} \tag{49}$$

with C_0 as defined in (6) and where $J(\cdot)$ represents an interpolation rule. The NCI sampling scheme thus consists of an inversion of the non-central chi-squared distribution for the low and most frequent Poisson-outcomes and of a moment-matching scheme based on the chi-squared distribution for the

[13] Since $\lim_{U\to 1} G^{-1}_{\chi^2_{d+2N_j}}(U) = \infty$, one should use $1-\delta$ instead of 1 to avoid numerical difficulties. Here δ is defined as a small machine number: in C one can for example set $\delta =$ DBL_EPSILON which is defined in the header float.h

rare and high Poisson outcomes: though the probability of $\{N > N_{\max}\}$ is usually small, it will be strictly greater than zero for all $N_{\max}$ and we decide to use direct inversion[14].

Design of the cache: a practical example

As example we will work out a way to implement the Non-Central Chi-Squared Inversion (NCI) scheme, specifically we will explore how to design the cache. A few details still has to be filled in: which value should be chosen for $N_{\max}$ in (45), how should the uniform numbers for a $\mathcal{U}_{N_j}$ in (46) be aligned and which interpolation rule J should be chosen to interpolate between two values of the inverse chi-squared distribution.

For expositional purposes we use the parameter values $v(0) = 0.09$, $\theta = 0.09$, $\kappa = 1.0$, $\xi = 1.0$ for the variance process and we use an equidistant time grid with $\Delta t = 0.25$ and maturity $T = 5$. Using (43), (44) and (48) this then implies that the exact distribution of the variance process equals a constant C_0 ordinary chi-squared distribution with $d + 2N = \frac{4\kappa\theta}{\xi^2} + 2N = 0.36 + 2N$ degrees of freedom, where N (cf. (44)) is Poisson distributed with mean $v(s)C_5 = v(s)7.042$. Using this setting as example we comment on the choice of $N_{\max}$. As shown in table **1**, this choice mainly depends on the mean of Poisson distribution: for the case $s = 0$, we have $v(0) = 0.09$, hence the scheme implies that we need to sample from a Poisson-distribution with mean $\mu = 0.634$ and we could easily use this mean to set a bound for $N_{\max}$. Unfortunately one then ignores the randomness of the mean: even though the means of the stationary and non-stationary distribution can be approximately equal, the randomness significantly increase the mass in the tails of non-stationary distribution function (i.e. based on all Poisson-draws in the simulation) when compared to the 'stationary' Poisson-distribution at time 0. An example of this behavior can be seen in the empirical distribution function as reported in table **1**.

n	$\mathbb{P}(N > n)$	n	$\mathbb{P}(N > n)$	n	$\mathbb{P}(N > n)$
-1	1	4	0.0361	9	0.0075
0	0.2585	5	0.0253	10	0.0058
1	0.1362	6	0.0182	20	0.0007
2	0.0816	7	0.0133	40	$1.26 \cdot 10^{-5}$
3	0.0530	8	0.0099	80	$9.00 \cdot 10^{-7}$

Table **1**: Empirical distribution function based on the Poisson samples that were drawn in 10^7 simulations with the parameters: $v(0) = 0.09$, $\theta = 0.09$, $\kappa = 1.0$, $\xi = 1.0$, $\Delta t = 0.25$ and maturity $T = 5$.

One then has to decide which size, alignment and interpolation rule one uses on the grid of uniform numbers in the cache(s) of the inverse of the chi-squared distribution with $d + 2N_j$ degrees of freedom; first notice that, as shown in table **1**, the number of draws from the corresponding chi-squared distributions differs significantly across the bins, e.g. in the example more than 74% of the drawings end up in the first bin. Another point to take into account for interpolation rule, is that the inverse distribution function is a monotone function, hence we would like the interpolation rule to preserve the monotonicity in the cached data points. A third point might be that some areas (i.e. in the tails of

[14] Since for most parameter configurations and a reasonable choice of $N_{\max}$, the probability of $N_{\max}$ we only need to use a direct inversion a very limited amount of times, the computational overhead of direct inversion will be relatively small. Alternatively one can also opt to use an approximation for a chi-squared distribution with moderate to large degrees of freedom, e.g. see Abramowitz and Stegun (1964).

the distribution) are 'more difficult' to interpolate, which might plea for using a non-equidistant alignment. To keep the mapping of a uniform sample to the corresponding cached value straightforward, we simply opt to use a equidistant grid. Hence given a uniform drawing U_V and say $1, \dots, K$-values in the cache, we the lower index i can then be easily located by evaluating $l := \text{floor}(U_V * K)$.
For the interpolation rules, we suggest two rules that preserve the monotonicity of the date and are relatively easy to implement. The first one is linear interpolation: given $U_i < U_V < U_{i+1}$ the linear interpolation rule $J(\cdot)$ is given by

$$J(U_V) := \frac{U_{i+1} - U_V}{U_{i+1} - U_i} H_j^{-1}(U_i) + \frac{U_V - U_i}{U_{i+1} - U_i} H_j^{-1}(U_{i+1}). \tag{50}$$

This rule has the advantage that is fast to execute, but in our experience some more points has to be cached in comparison to higher order interpolation rules to obtain full accuracy. Hence it might depend on the situation (e.g. on the number of paths) whether linear interpolation is suited. Alternatively we therefore suggest a monotone cubic Hermite spline interpolation which rule is defined as follows.

$$J(U_V) := h_{00}(t) H_j^{-1}(U_i) + h_{01}(t) H_j^{-1}(U_{i+1}) + \Delta_i(m_i h_{10}(t) + m_{i+1} h_{11}(t)), \tag{51}$$

with $t := \frac{U_V - U_i}{U_{i+1} - U_i}$ and where the corresponding definitions of the Hermite polynomials $h_{00}, h_{01}, h_{10}, h_{11}$ and the weights m_i are given in appendix A.1. Though the spline interpolation rule requires a few more (elementary) operations in each step, one can significantly reduce the required number of points in the cache. Finally a suitable number of points and the choice of $N_{\max}$ is both case (e.g. the number of simulations) as well as parameter dependent and basically constitutes a efficiency weigh-off between bias and variance. We do mind the reader to be careful with too coarse grids, since $\mathbb{E}[N] = \text{Var}[N] \to \infty$ as $\Delta t \to 0$ and one might end up with many exceedings of $N_{\max}$ implying a loose of efficiency since in such case direct inversion[15] instead of a look-up value from the cache is being used. It is not very likely that this will cause a problem, since from the numerical results of 5.2 it follows that using only a few steps per year already produces schemes which have a negligible bias. As for a certain product small time steps (e.g. weekly steps) are required and one can for example retain the efficiency of the cache by using the modification as described in section 3.5.

NCI Algorithm

Assuming that a certain threshold level $N_{\max}$, an alignment of the grid $\mathcal{U}_{N_j}$ (46) and interpolation rule $J(\cdot)$ (49) has been chosen, the Non-central Chi-Squared Inversion (NCI) algorithm can be summarized by the following algorithm:

1. For $N_j = 0, \dots, N_{\max}$, use (47) to precompute the inverses of the chi-squared distribution on the grid $\mathcal{U}_{N_j}$ and compute d using (44)

2. Given $\nu(s)$ and μ using (44).

3. Generate a sample from a Poisson distribution with mean μ according to (3.4):

[15] Since $\chi^2_{d+2N_j} \overset{d}{=} \gamma_{\frac{\nu}{2},2} \overset{d}{=} 2\gamma_{\frac{\nu}{2},1}$, the inverse of a chi-squared distribution with ν degrees of freedom can be obtained by taking twice the inverse of gamma distribution with shape $k = \frac{\nu}{2}$ and scale 1. For the inversion of the gamma distribution we use an implementation of Maddock et al. (2008). Their (root search) algorithm first uses the method of Didonato and Morris Didonato and Morris (1986) to find an initial approximation for the inverse and consecutively applies a few Halley iterations to cut off some extra digits. In many cases only a few iterations are needed for the complete algorithm, i.e. in the evaluation of the function `2.0*gamma_p_inv(k,u)`.

(a) Draw a uniform random number U_P.

(b) Set $N_j = 0$ and $P_{N_j} = P_C = \exp(-\mu)$.

(c) **While** $P_C \leq U_P$[16]:
Set $N_j = N_j + 1$, $P_{N_j} = P_{N_j} \cdot \frac{\mu}{N_j}$ and $P_C = P_C + P_{N_j}$.

4. Generate a sample from a chi-squared distribution with $d + 2N_j$ degrees of freedom:

(a) Draw a uniform random variable U_V.

(b) Use (48), i.e. set $v(t) = F^{-1}_{N_j}(U_V)$ where $F^{-1}_{N_j}$ is defined in (49).

3.5 The NCI-QE scheme

An alternative sampling method for the non-central chi-squared distributed variance process can be to combine the NCI and QE approximations of section 3.4 and 3.3. This combination is motivated by the fact that while NCI-scheme is particularly efficient for small non-centrality parameters (i.e. implying a small Poisson-mean and hence a small cache, e.g. see 44), the QE-scheme works especially well for moderate to high non-centrality parameters (i.e. corresponding to the quadratic normal-like approximation (34), see Abramowitz and Stegun (1964) or Andersen (2007)). To see which circumstances correspond to low/high non-centrality parameters, we use the parameter settings of table **2** and show the impact of instantaneous variance and the size of the time step on the value of λ (6). The results are given in figure **2**.

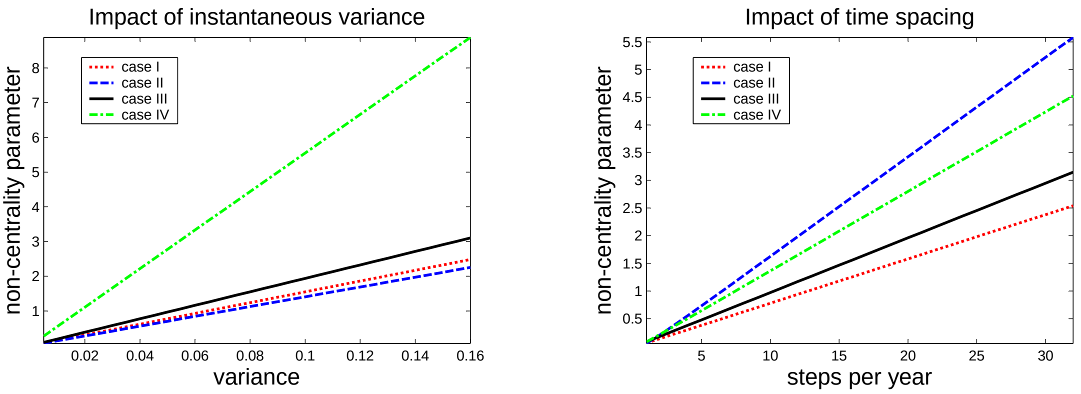

Figure **2**: The non-centrality parameter λ (6) as function of the instantaneous variance and the size of the time step for the parameter settings of table **2**. In the left graph we fixed the number of time steps at 8 steps per year.

From equation (6) and the figure and we can hence see that the non-centrality parameter grows linearly as function of the instantaneous variance and (almost) linear as function of the number of time steps. Hence recalling that the NCI/QE works particularly well for low/high non-centrality, a logical switching rule would try to combine the best of both worlds by using a critical switching value λ_c of the non-centrality parameter. That is, for $\lambda \leq \lambda_c$ one uses the NCI sampling scheme and else the quadratic moment-matched approximation (34) of the QE scheme. The exact choice of the switching value λ_c

[16] To avoid eventual numerical errors, one might define a maximum number of loops for safety.

(provided that the moment-matching condition 38 is satisfied) appears to have a relatively small effect on the quality of the simulation scheme, indicating that the transition between both schemes is rather smooth for moderate degrees of freedom. In our numerical test we use $\lambda_c = 4$ as critical switching value between the schemes; first, for $\lambda > 4$ the quadratic moment-matched scheme is always applicable since then $\psi < 2$ and secondly, $\lambda < 4$ implies for the NCI scheme that we need to draw a sample from a Poisson-conditioned chi-squared distribution with mean $\mu = \frac{1}{2}\lambda < 2$. Since there is little probability mass in the Poisson distribution for small μ[17] we only need to cache a small number of chi-squared distributions (e.g. see (46) and (49)), and at the same time the use of the direct inversion (i.e. when the Poisson-draw exceeds the maximum cache size) will be limited.

By using $\lambda_c = 4$ as critical value, the figure **2** implies that for the considered parameter configurations, the scheme uses the NCI part the majority of the time for a number of time steps that is smaller than $20-50$ steps per year. The other way around: for larger values of the instantaneous variance and a smaller time-grid the QE scheme will used more extensively. Since in practice $4-8$ time steps a year is already sufficient to produce a scheme with a negligible bias (see section 5), one will probably apply NCI part the majority of the time. Do note that as we make the size of the time step smaller and smaller, the use of the NCI becomes more and more remote.

The simulation algorithm for the NCI-QE scheme works straightforward: in step (2) of the NCI-algorithm 3.4 one evaluates μ: if $\mu < \frac{1}{2}\lambda_c$, one applies (3)-(4) or else one uses the quadratic approximation in step (2) of the QE-algorithm 3.3.

4 Asset price sampling in combination with drift interpolation

Though a sample of the integrated variance process can be generated using the exact method of Broadie and Kaya, or the almost exact method of Smith, it is probably computationally more efficient to use the simple drift interpolation method of (31), for example with the corrector-predictor setting $\gamma_1 = \gamma_2 = 0.5$. Hence by applying the drift interpolation method for the integrated variance process, one can modify the 'Broadie and Kaya' asset price sampling scheme of (23) as follows:

$$\begin{aligned}\log(S(t)) &= \log(S(s)) + r\Delta t + \frac{\kappa\rho}{\xi}(\gamma_1 v(s) + \gamma_2 v(t))\Delta t - \frac{1}{2}(\gamma_1 v(s) + \gamma_2 v(t))\Delta t \\ &\quad + \frac{\rho}{\xi}(v(t) - v(s) - \kappa\theta\Delta t) + \sqrt{1-\rho^2}\sqrt{\gamma_1 v(s) + \gamma_2 v(t)} \cdot Z_s\sqrt{\Delta t}, \\ &= \log(S(s)) + r\Delta t + K_0 + K_1 v(s) + K_2 v(t) + \sqrt{K_3 v(s) + K_4 v(t)} \cdot Z_S \qquad (52)\end{aligned}$$

where Z_s is a standard normal distributed random variable, independent of v, and with

$$\begin{aligned}K_0 &= -\frac{\rho\kappa\theta}{\xi}\Delta t, & K_1 &= \gamma_1\Delta t\Big(\frac{\kappa\rho}{\xi} - \frac{1}{2}\Big) - \frac{\rho}{\xi}, & K_2 &= \gamma_2\Delta t\Big(\frac{\kappa\rho}{\xi} - \frac{1}{2}\Big) + \frac{\rho}{\xi}, \\ K_3 &= \gamma_1\Delta t\Big(1-\rho^2\Big), & K_4 &= \gamma_2\Delta t\Big(1-\rho^2\Big).\end{aligned}$$

Hence this asset price sampling scheme can be used in conjunction with different methods to simulate the variance process. One can for example use this method in conjunction with variance sampling that we have previously considered, that is with the BK-DI, the QE, the NCI and the NCI-QE scheme of sections 3.1-3.5. Please note that the above discretization scheme usually is not completely arbitrage-free, though the bias induced from this approximation is usually rather small and controllable by

[17] For all $\mu < 2$ one has $\mathbb{P}(N > 10) < 8.3 \cdot 10^{-6}$

reducing the size of the time step. However with little effort one can also (locally) enforce an exact no-arbitrage condition, leading to the martingale corrected BK-DI-M, QE-M, NCI-M and NCI-QE-M schemes, see Andersen (2007) and section 4.1.

Asset price sampling scheme

The stock price discretization scheme of the previous section can be summarized as follows:

1. Conditional on $v(s)$, use the BK-DI scheme 3.1, the QE scheme 3.3, the NCI scheme 3.4 or the NCI-QE scheme of 3.5 to generate a sample for $v(t)$.
2. Generate a random sample Z_S from the standard normal distribution.[18]
3. Given $\log(S(s))$, $v(s)$, $v(t)$ and Z_S, compute $\log(S(t))$ from (52).

4.1 Martingale correction

As discussed in Andersen and Piterbarg (2007), the continuous-time asset price process $S(t)$ might not have finite higher moments, but the discounted stock price will always be a martingale under the risk-neutral measure, i.e.

$$\mathbb{E}^Q[e^{-r\Delta t}S(t)\big|\mathcal{F}_s] = S(s) < \infty. \tag{53}$$

If we however takes the exponent of the scheme (52), one usually has that the discretized stock price process (from here on denoted by $\widetilde{S}(t)$)

$$\widetilde{S}(t) = \widetilde{S}(s)\exp[r\Delta t + K_0 + K_1v(s) + K_2v(t) + \sqrt{K_3v(s) + K_4v(t)} \cdot Z_S], \tag{54}$$

does not satisfy the martingale condition, i.e.

$$\mathbb{E}^Q[e^{-r\Delta t}\widetilde{S}(t)\big|\mathcal{F}_s] \neq \widetilde{S}(s). \tag{55}$$

As noted by Andersen (2007) the practical relevance of this is often minor, because the net drift away from the martingale is typically very small and controllable by reducing the size of the time step. Moreover the ability to hit the mean of the distribution does not necessarily leads to better option prices. Following Glasserman (2003) and Andersen (2007), we do discuss the martingale correction method, that is we investigate whether it is possible to modify the NCI scheme such that the discretized option price becomes martingale. Additionally we look at the regularity of the discretization scheme, e.g. we look whether there might parameter values where the $\widetilde{S}$-process might blow up in the sense that $\mathbb{E}[\widetilde{S}(t)|S(s)] = \infty$.

By the tower law of conditional expectations, we have

$$\mathbb{E}^Q[e^{-r\Delta t}\widetilde{S}(t)\big|\mathcal{F}_s] = \mathbb{E}\Big[\mathbb{E}^Q(e^{-r\Delta t}\widetilde{S}(t)\big|\mathcal{F}_s, v(t))\Big], \tag{56}$$

[18] It might be advisable to use an inversion method to generate normal samples, since then a quasi random generator is also applicable. The algorithm described in Wichura (1998) gives an approximation $\widetilde{\Psi}^{-1}$ of Ψ^{-1} with a relative error of

$$\frac{|\widetilde{\Psi}^{-1} - \Psi^{-1}|}{1 - |\Psi^{-1}|} < 10^{-15}.$$

hence for the martingale condition (53) to hold, we need $\widetilde{S}(s)$ to equal the above expectation. Using the moment-generating function of the normal distribution, we find that the following equation needs to be satisfied:

$$\begin{aligned}
\widetilde{S}(t) &= \widetilde{S}(s)\exp[K_0^* + K_1 v(s)]\mathbb{E}\Big[\exp[K_2 v(t)]\mathbb{E}^Q(\exp[\sqrt{K_3 v(s) + K_4 v(t)} \cdot Z_S]\Big|\mathcal{F}_s, v(t))\Big] \\
&= \widetilde{S}(s)\exp[K_0^* + K_1 v(s)]\mathbb{E}\Big[\exp[K_2 v(t) + \frac{1}{2}(K_3 v(s) + K_4 v(t))]\Big] \\
&= \widetilde{S}(s)\exp[K_0^* + (K_1 + \frac{1}{2}K_3)v(s)]\Psi_{v(t)}(A) \qquad (57)
\end{aligned}$$

with

$$A := K_2 + \frac{1}{2}K_4, \qquad (58)$$

and where $\Psi_{v(t)}(x)$ denotes the moment-generating function of the (discretized) variance process $v(t)$ evaluated in the point x. Hence for the martingale condition to hold we need

$$\exp(K_0^* + (K_1 + \frac{1}{2}K_2)v(s))\Psi_{v(t)}(A) = 1, \qquad (59)$$

which (assuming the regularity condition $\Psi_{v(t)}(A) < \infty$) is satisfied by replacing the constant K_0 in (52) by

$$K_0^* := -\log(\Psi_{v(t)}(A)) - (K_1 + \frac{1}{2}K_2)v(s). \qquad (60)$$

Hence what remains is to determine the moment-generating function of the simulated variance processes. Additionally we look at the regularity of the schemes: we check whether there might be parameter configurations for which the moment-generating function of the simulated variance process does not exist.

4.2 Moment generating function of $v(t)$, Regularity

QE(-M) scheme

Using properties of the non-centrally chi-squared distribution with one degrees of freedom on (34) and direct integration on (37) Andersen (2007) derives the moment-generating functions of his variance schemes. For the regularity conditions to hold and the moment-generating functions to exist, the conditions $aA < \frac{1}{2}$ and $A < \beta$ have be satisfied. As motivated in Andersen (2007) these conditions are often (but not always) satisfied in practice: the most restrictive condition is that for $\rho > 0$ the size of the time step has to be sufficiently small. Assuming regularity, one can define

$$K_0^* = \begin{cases} -\frac{Ab^2a}{1-2Aa} + \frac{1}{2}\log(1-2Aa) - (K_1 + \frac{1}{2}K_3)v(s) & \text{if } \psi \le \psi_c, \\ -\log(p + \frac{\beta(1-p)}{\beta-A}) - (K_1 + \frac{1}{2}K_3)v(s) & \text{if } \psi > \psi_c. \end{cases} \qquad (61)$$

and replace K_0 by K_0^* in the QE scheme of (52) to obtain the martingale-corrected QE(-M) scheme for the discretized stock price process S (see Andersen (2007) proposition 9, pp. 21-22).

NCI(-M) and BK-DI(-M) scheme

Since we know from (6) that the variance process $v(t)$ is distributed as a constant C_0 times a non-central chi-squared distribution with d degrees of freedom and non-centrality parameter λ, we find following expression for $\Psi_{v(t)}(A)$ for the exact variance process [19] $v(t)$

$$\Psi_{v(t)}(A) = \mathbb{E}\Big[\exp(Av(t))\Big] = \frac{\exp(\frac{C_0 A \lambda}{1-2C_0 A})}{(1-2C_0 A)^{\frac{d}{2}}}. \tag{62}$$

For this expectation to exist we need $C_0 A < \frac{1}{2}$, i.e.:

$$\frac{\rho}{\xi}(1+\kappa\gamma_2\Delta t) - \frac{1}{2}\gamma_2 \Delta T \rho^2 < \frac{2\kappa}{\xi^2(1-e^{-\kappa\Delta T})}. \tag{63}$$

To get a grasp at the restrictiveness of this condition, notice that the right hand side is always positive; hence it follows that for $\rho \leq 0$ the condition will always be satisfied. In contrast, for $\rho > 0$, equation (63) imposes a limit on the size of the time step ΔT, roughly $\rho\xi\Delta T < 2$. Assuming that the regularity conditions are satisfied, we can apply (60) and set

$$K_0^* = -\frac{C_0 A\lambda}{1-2C_0 A} + \frac{d}{2}\log(1-2C_0 A) - (K_1 + \frac{1}{2}K_3)v(s) =: C_6\lambda + C_7 + C_8 v(s) \tag{64}$$

to enforce the martingale in the NCI(-M) and BK-DI(-M) discretization scheme. Note that the constants

$$C_6 = -\frac{C_0 A}{1-2C_0 A}, \quad C_7 = \frac{d}{2}\log(1-2C_0 A), \quad C_8 = -(K_1 + \frac{1}{2}K_3), \tag{65}$$

can be precomputed before the Monte Carlo run.

5 Numerical results

To test our new scheme we consider the pricing of an Asian and European call options in the Heston model. Since European call prices belong to the vanilla options and can be calculated with a great accuracy (e.g. see Carr and Madan (1999)), they form a standard test case. Additionally we use an Asian option as test case for a more path-dependent option.

To investigate the efficiency and discretization bias of the NCI scheme, we benchmark it to the full truncation Lord et al. (2008) and the QE-M scheme Andersen (2007). These schemes are to our knowledge the most efficient scheme in most practical situations. Note that we do not incorporate the Broadie and Kaya (2006) scheme in our comparison; first, by definition, the Broadie and Kaya scheme is exact and thus free of discretization bias. Therefore theoretically, Monte Carlo simulations are not needed to test the quality of the Broadie and Kaya scheme. However in practice, some numerical error will certainly be introduced in the required numerical inversions of the algorithm (see section 2.4); moreover Lord et al. even show (Lord et al. (2008) pp.16 table 5) that for equal computational budgets[20] both the right Euler scheme (i.e. the full truncation scheme) as well as a moment-matched scheme completely outperform the exact scheme. Thus besides being theoretically sound, practical use and quality of the Broadie and Kaya scheme might be limited and we decide to

[19] Since by setting $N_{\max}$ sufficiently large we can come as close as we want to the true inverse of the variance process by the inverse as defined in (49), we use the above expression as moment generating function of the NCI variance process $v(t)$

[20] With computational budget we mean CPU time. Note that because different schemes require different computational effort for the same time grid, fixing the computational budget implies that the size time step of the required schemes is adjusted to match the computational budget. Since for a fixed time step an Euler and moment-matched scheme require a smaller computational budget than the exact Broadie and Kaya scheme (that requires two numerical inversion procedures), this implies that with a equal budget, smaller time steps can be taken in the Euler and moment-matched schemes in comparison with the exact scheme.

not include it our comparison. Though the almost exact simulation scheme of Smith (2008) without doubt delivers a speed to the exact scheme, we still argue that a moment-matched scheme and the full truncation Euler scheme are more efficient in most situations[21].

Though our benchmark setup is similar to Lord et al. (2008) and Andersen (2007), we add a control variate to get rid of some extra Monte Carlo noise[22]: both in the analysis of Lord et al. (2008) as well as in Andersen (2007) the 'finite sample noise' of the Monte Carlo plays still plays such a big role, that even a number of 10^6 paths in some cases sometimes is not enough to draw a good comparison both schemes, let alone say something about the true quality and bias of the schemes. Though this certainly says something about the overall quality of the full truncation and QE schemes, the produced estimates still seem somewhat inconsistent and this motivates the use of the control variate. With this simple variance reduction technique we are then able to draw stronger conclusions about the overall and the relative quality of the schemes. Though low-discrepancy numbers also form a natural candidate for faster convergence behavior, we do not use them in the benchmark to keep the comparability with the numerical results in the existing literature. We do want to emphasize that in practice, all the three schemes can be used in conjunction with low-discrepancy numbers, e.g. with the Sobol numbers as described in Jäckel (2002) and Press and Flannery (1992).

5.1 Benchmark setup

To test our discretization schemes, we consider the parameter configurations of table (2). These settings correspond to some different parameter settings which are likely to be encountered in equity, FX or interest rate markets.

Example	**Type**	κ	ξ	ρ	$v(0)$	θ	r
case I	Call-10Y	0.5	1.0	-0.9	0.04	0.04	0.00
case II	Call-5Y	1.0	1.0	-0.3	0.09	0.09	0.05
case III	Call-15Y	0.3	0.9	-0.5	0.04	0.04	0.00
case IV	Asian-4Y	1.0407	0.5196	-0.6747	0.0194	0.0586	0.00

Table **2**: Test cases for the Heston schemes, in all cases $S(0) = 100$.

Notice that for all cases we have $\xi^2 >> 2\kappa\theta$, implying that the origin is accessible. The first and third example are taken from Andersen (2007) and serve to represent long-dated FX options. We can expect that such a setting is difficult, since the corresponding parameters combine a low mean-reversion κ with a high volatility of the variance process ξ; hence the variance process has a relatively high probability of reaching the troubled region near zero. The second case stems from Broadie and Kaya (2006) and can correspond to an equity setting. The last setting is taken from Smith (2008) who considers an Asian option with yearly fixings to test some models for a path-dependent

[21] For example compare the performance of the almost exact scheme with the Kahl-Jäckel scheme in Smith (2008) to the Kahl-Jäckel scheme in Andersen (2007) or Lord et al. (2008). Whereas the almost exact scheme in Smith (2008) performs comparable to the Kahl-Jäckel scheme Kahl and Jäckel (2006), the full truncation scheme as well as the QE-M scheme outperform this scheme in Lord et al. (2008) and Andersen (2007).

[22] For an overview of other variance reduction techniques we refer to Jäckel (2002) and Glasserman (2003). For stochastic volatility models the method of Willard Willard (1997), which entirely eliminates the noise of the stock price, can be rather attractive for options that have a closed form solution under the Black and Scholes (1973) model. We do not use the latter method since we are interested in the bias of the joint simulation of stock and variance.

equity option.

In the test cases we consider the pricing of a (vanilla/asian) call option maturing at time T and with strike K. Hence we are interested in a Monte Carlo valuation of the call option price C and Asian option price A, i.e.

$$C = e^{-rT}\mathbb{E}\Big[(S(T) - K)^+\Big], \quad \text{and:} \quad A = e^{-rT}\mathbb{E}\Big[(\sum_{j=1}^{N}\frac{S(T_j)}{N} - K)^+\Big] \tag{66}$$

Hence with a discretization scheme for the stock price $S(t)$, we can use Monte Carlo methods to approximate these prices by $\widehat{C}$ and $\widehat{A}$. Due to errors introduced by discretization and the Monte Carlo one in general finds that the estimated values are not equal the theoretical values. Hence we define the bias $\widehat{b}$ of the discretization scheme as this difference, i.e.

$$\widehat{b} := C - \widehat{C}. \tag{67}$$

In the following subsection we hence specify how the Monte Carlo estimates for $\widehat{C}$ and $\widehat{A}$ can be obtained.

Control variate estimators

The ordinary Monte Carlo estimator would consist of simulating n independent samples of the required (discounted) payoff(s) and averaging over all paths, which assuming (by the strong law of large numbers) converges to the expected option price $\mathbb{E}[\widehat{C}]$ of the discretized asset price dynamics. Though this estimator is generic and simple, there exist various other estimators (e.g. see Glasserman (2003) or Jäckel (2002)) which are more efficient, c.q. which have a lower variance. To obtain some variance reduction over the ordinary estimator, we therefore suggest to use a control variate estimator which is also quite generic and involves little computational overhead;
For $i = 1, \ldots, n$ we generate Monte Carlo samples of the stock price(s) S_i and the corresponding option prices C_i. We can then use the following control variate estimates to estimate the vanilla/Asian call option prices:

$$\overline{C}(b_C) = \frac{1}{n}\sum_{i=1}^{n}(C_i - b_C(S_i - \mathbb{E}[S])), \qquad C_i = e^{-rT}(S_i(T) - K)^+, \tag{68}$$

$$\overline{A}(b_A) = \frac{1}{n}\sum_{i=1}^{n}(A_i - b_A(S_i - \mathbb{E}[S])), \qquad A_i = e^{-rT}(\sum_{j=1}^{N}\frac{S_i(T_j)}{N} - K)^+, \tag{69}$$

which estimators (again by the strong law of large numbers) also converge with probability one to the expected option prices. To see the effectiveness, i.e. variance reductions, of these control variate estimators over the ordinary Monte Carlo estimators we refer to appendix A.2. To get more reliable estimates, we use the control variate estimators rather than the ordinary Monte Carlo estimator for the first three test cases. Though the technique is also quite effective for the considered Asian option (as can be seen from table 5), we use rather use the ordinary Monte Carlo estimator to make our numeric results comparable with those of Smith (2008).

5.2 Numerical tests

For the numerical results of the first three test cases we use the following discretization schemes: the Euler-FT scheme of (13)-(16), the Kahl-Jäckel 'IM-IJK' scheme of (18)-(19), the QE-M scheme of section 3.3, the NCI-M scheme of section 3.4, the combined NCI-QEM scheme of 3.5 and the Broadie and Kaya 'drift interpolated-martingale corrected' scheme (denoted by "BK-DI-M"). For the latter four schemes we use the corrector-predictor scheme of (52) with the a mid-point rule (i.e. $\gamma_1 = \gamma_2 = \frac{1}{2}$) combined with the martingale corrections of section 4.1 for the discretized stock price. Then in the fourth test case we compare the results of the Euler-FT, the IM-IJK, the QE-M, the NCI-M, the NCI-QE-M and the BK-DI-M scheme against the results of the exact and almost exact scheme for a path-dependent option that can be found in Smith (2008).
To obtain accurate estimates we performed all tests using 10^6 Monte Carlo[23] paths. Moreover in the first three tests we use control variate estimators of (68), for all the schemes for which the control variate is applicable, to reduce the standard error even further. That is, for all the schemes that satisfy the discrete time martingale condition, i.e. all schemes except the IM-IJK. Please note that the control variate only affects the variance and not the (expected) bias of the Monte Carlo estimates. To keep our test comparable with the results in Smith (2008) we just use an ordinary Monte Carlo estimator for the fourth test case, i.e. the Asian option.
In tables **3**-**7** we then report the biases of the Monte Carlo estimates and the corresponding standard errors. We star the biases that are statistically insignificant at a 99% confidence level. That is, when the exact price lies within the 99% confidence window of the Monte Carlo price:

$$\text{exact value } \in \left[\overline{C} - z\frac{\sigma_C}{\sqrt{n}}, \overline{C} + z\frac{\sigma_C}{\sqrt{n}}\right], \tag{70}$$

with $z = 2.576 = \Phi^{-1}(1 - \frac{0.01}{2})$ the corresponding quantile-point.

Results for case I-III

The results for the estimated call option price bias of **case I** can be found below in table **3**: we report the Monte Carlo estimates of the bias (67) as function of the time step Δt, for an at-, out- and in-the-money strike. The first thing to notice are the enormous differences in the magnitude of the biases between the schemes: whilst the first and second order discretization schemes (Euler/IM-IJK) are still quite biased for a 32 time steps a year, all the schemes that are based on approximating the non-central chi-squared distribution are already bias-free for a time-spacing of just 4 steps a year: that is, the biases of QE-M, NCI-M, NCI-QE-M and the BK-DI-M scheme are not significantly different from zero at a 99% confidence level on using 4 time steps a year. When we consider the bias over various strike levels, we see that the bias of the Euler and the martingale-corrected scheme decreases if the strikes goes more into the money. This is expected since all these schemes are constructed to be bias-free for $K = 0$ (i.e. by the martingale construction).
All in all, we can conclude from the table that for practical sizes of the time step, the biases of the QE-M, the NCI-M, the NCI-QE-M and the BK-DI-M are substantially lower than those of the Euler and especially the IM-IJK scheme. Before we can conclude that the former schemes are also more efficient that the latter ones, we of course also need to look at the required computational effort of each discretization schemes, which will be addressed in section 5.3.

[23] We use the Mersenne Twister as pseudo random number generator in combination with incremental path generations.

	Euler-FT	IM-IJK	QE-M	NCI-M	NCI-QE-M	BK-DI-M
Δt	K = 100					
1	-6.359 (±0.043)	-57.574 (±0.275)	-0.222 (±0.020)	0.246 (±0.022)	0.241 (±0.022)	0.234 (±0.022)
1/2	-3.718 (±0.033)	-32.828 (±0.179)	-0.110 (±0.021)	0.075 (±0.022)	0.073 (±0.022)	0.079 (±0.022)
1/4	-2.048 (±0.027)	-18.457 (±0.120)	-0.008* (±0.022)	0.015* (±0.022)	0.029 (±0.022)	0.013* (±0.022)
1/8	-1.036 (±0.024)	-10.094 (±0.084)	0.023 (±0.022)	0.008* (±0.022)	-0.002* (±0.022)	-0.001* (±0.022)
1/16	-0.537 (±0.023)	-5.325 (±0.063)	0.018* (±0.022)	-0.002* (±0.022)	-0.011* (±0.022)	0.010* (±0.022)
1/32	-0.255 (±0.023)	-2.678 (±0.050)	0.015* (±0.022)	0.005* (±0.022)	-0.007* (±0.022)	0.001* (±0.022)
	K = 140					
1	-4.269 (±0.041)	-51.544 (±0.242)	0.084 (±0.006)	0.029 (±0.006)	0.031 (±0.006)	0.031 (±0.006)
1/2	-1.942 (±0.024)	-28.038 (±0.146)	0.025 (±0.006)	0.011 (±0.006)	0.013 (±0.006)	0.014 (±0.006)
1/4	-0.761 (±0.013)	-14.810 (±0.086)	0.001* (±0.006)	0.002* (±0.006)	0.002* (±0.006)	0.004* (±0.006)
1/8	-0.265 (±0.009)	-7.529 (±0.051)	-0.002* (±0.006)	0.002* (±0.006)	0.003* (±0.006)	0.000* (±0.006)
1/16	-0.100 (±0.007)	-3.617 (±0.030)	0.000* (±0.006)	-0.005* (±0.006)	0.001* (±0.006)	-0.001* (±0.006)
1/32	-0.039 (±0.007)	-1.639 (±0.018)	-0.002* (±0.006)	-0.002* (±0.006)	-0.004* (±0.006)	-0.001* (±0.006)
	K = 60					
1	-3.120 (±0.029)	-50.708 (±0.304)	-0.027 (±0.019)	0.138 (±0.019)	0.127 (±0.019)	0.114 (±0.019)
1/2	-1.747 (±0.025)	-26.749 (±0.208)	0.047 (±0.020)	0.062 (±0.020)	0.059 (±0.020)	0.057 (±0.020)
1/4	-0.938 (±0.022)	-13.603 (±0.148)	0.039 (±0.020)	0.006* (±0.020)	0.017* (±0.020)	0.009* (±0.020)
1/8	-0.460 (±0.021)	-6.720 (±0.112)	0.030 (±0.020)	0.013* (±0.020)	-0.009* (±0.020)	0.009* (±0.020)
1/16	-0.235 (±0.020)	-3.305 (±0.091)	0.023 (±0.020)	0.002* (±0.020)	-0.011* (±0.020)	0.000* (±0.020)
1/32	-0.110 (±0.020)	-1.616 (±0.078)	0.011* (±0.020)	-0.002* (±0.020)	0.007* (±0.020)	0.001* (±0.020)

Table **3**: Estimated call option prices biases for I. Numbers in parentheses are the widths of the confidence interval (70) at a 99% confidence level: we starred the biases that were not significantly different from zero. Exact prices respectively are 13.085, 0.296 and 44.330.

Results for case II and III

The numerical results for **case III** and **case III** can be found in the table **6** and **7**. The results are similar to those of **case III**, but a little less severe. Still the almost exact variance schemes by far outperform the considered Euler (FT) and Milstein (KJ) scheme. The tables can be found in appendix A.3.

5.3 Results for case IV and computational times

In the fourth test case we consider a path-dependent (Asian) option; to make the results comparable with the results as reported by Smith (2008) (table 3) we consider the same parameter settings (see table **2**) and use the root mean square error (RMSE) as error measure for our estimates. Moreover since Smith (2008) also considers an Euler-scheme (Reflection), we can use the computational time from this scheme to scale our computational times with the ones as reported in Smith (2008). Hence we are in particular interested in the relative efficiency of the AESM in comparison to the schemes considered here; we are in particular interested in the time it takes the different schemes to reach a certain RMSE. Supported by the previous numerical results, we use 8 steps a year for the considered drift interpolation schemes and 100 time steps a year for the Euler and Milstein schemes. To obtain accurate estimates for standard Monte Carlo estimates, we use a high number of 2 560 000 paths (using pseudo-random Mersenne-Twister numbers). The results are given in table **4**.

	Euler-R	Euler-FT	IM-IJK	AESM	QE-M	NCI-M	NCI-QE-M	BK-DI-M
Δt	1/100	1/100	1/100	1	1/8	1/8	1/8	1/8
RMSE	0.237	0.010*	0.018*	0.008*	0.009*	0.009*	0.009*	0.011*
Time (in sec.)	158.3	157.5	178.6	1134.4	16.3	22.7	17.2	25.6
Relative time	9.7	9.7	11.0	69.7	1.0	1.4	1.1	1.6

Table **4**: Root mean squared errors (RMSE) for case IV. 'Exact' price 9.712: we starred the root mean squared errors with biases that were not significantly different from zero at a 99%-level. Results for the AESM are taken from Smith (2008), table 3.

The first thing to notice from table **4** are the enormous differences in the required computational budgets; though the AESM method only requires a single step per year and we choose to use 100 time steps per year for the Euler schemes, the latter schemes are still approximately 7 times faster! This difference is even more stunning, if we compare the results for the drift interpolation schemes with the AESM method; Whilst the drift interpolation schemes show statistically indifferent RMSEs as the AESM method, they obtain this result approximately 40 to 70 times faster.

Secondly we would like to comment on the differences between the considered Euler schemes; in Broadie and Kaya (2006) and Smith (2008) some benchmarks are performed against a Euler scheme which fixes negative values in a non-optimal way (i.e. by absorbing of reflection negative values). As shown above (and extensively discussed in Lord et al. (2008)) the choice of the fix is extremely important for the overall quality of the simulation scheme. Hence when using such a non-optimal fix, one cannot conclude that ”the” Euler schemes are less efficient than for example the AESM method: in Smith (2008) the Euler-R scheme is outperformed by the AESM method, while we can see from the above result that the Euler-FT by far outperforms the AESM method in terms of computational efficiency.

From a efficiency viewpoint we conclude that the QE-M scheme performs the best in the above test, followed closely by the other drift-interpolation schemes. The Euler-FT and the IM-IJK scheme follow on some distance, though these schemes share the advantage that they are simpler and more generically applicable, e.g. in other CEV models. The Euler-R scheme (though frequently used in the literature) performs extremely worse, with a large RMSE. Though the RMSE of the AESM is small and indifferent from the drift-interpolation schemes, the time required to obtain this estimate is by far the largest and except for the heavily biased non-optimal Euler-R scheme, the AESM scheme is outperformed by all the considered schemes.

6 Conclusion

Though the exact simulation method of Broadie and Kaya Broadie and Kaya (2006) is theoretically appealing, its practical use might be limited due to its complexity and lack of speed. Moreover the suggested acceptance and rejection technique for the variance process hinders perturbation analysis, let alone the use low-discrepancy numbers. Nevertheless the method also provides an excellent starting point for the development of some more efficient approximate schemes. In fact, several quite different methods can be considered to approximate the the sampling methods of the variance and integrated variance process;

Almost in line with the exact method, Smith Smith (2008) recently suggested to speed up the sampling of the integrated variance with the AESM method. Though his results indeed indicate a significant speed up with respect to the exact scheme, one nonetheless has to perform a Fourier inversion in each simulation step which is still rather time-consuming. Moreover we found with other discretization methods that the largest variations were caused by the simulation of the variance process, rather than the integrated variance process. Hence we found with the BK-DI scheme that an easier and more efficient approximation of the exact scheme is to use a simple drift interpolation method for the variance integral. Though this method does not satisfy the discrete time martingale condition, we showed in section 4.1 how one can locally enforce this no-arbitrage condition. Nevertheless, we think that in financial applications the prime disadvantage for all the just mentioned schemes is the rather inconvenient acceptance and rejection sampling of the variance process.

Hence instead of an exact joint simulation of the variance and the integrated variance process, we rather looked at methods that approximate the variance process and use a simple drift approximation for its integral. The simplest scheme in this category is the Euler Full Truncation[24] scheme of Lord et al. (2008): we found it rather remarkable that this scheme by far outperforms many more complex schemes like the Kahl and Jäckel (2006) scheme and the AESM scheme of Smith (2008) in terms of computational efficiency. Though the Euler Full Truncation method is simple and straightforward to implement, it produces rather biased estimates for the moderate sizes of the time step: for our parameter settings we found that one at least has to use 32 time steps per year to obtain reasonable small biases. In a way this does not come as a surprise since the Euler scheme uses no analytical properties of the non-central chi-squared distribution variance process.

Last we considered a category of discretization methods which explicitly relies on the analytical properties of the variance process. For example the Quadratic Exponential (QE) scheme of Andersen (2007) uses moment-matching techniques to approximate the variance process, we suggested in the new Non-Central Chi-squared Inversion (NCI) method to create a cache of the inverses hereof. A crucial point for the latter scheme is that the variance process can be represented by chi-squared distribution whose degrees of freedom are given by a shifted Poisson random variable which has a rather small mean. Hence by conditioning on the shifted Poisson distribution (which decays fast for small means), we have demonstrated that one can create an efficient cache of the variance process at relatively small computational costs. Moreover both the QE as the NCI scheme share the advantage that their sampling methods are based on (efficient) inversion methods rather than on acceptance rejection techniques. Though both discretization methods slightly violate the no-arbitrage condition, one can (following Glasserman (2003) and Andersen (2007)) easily enforce a local martingale condition which leads to the QE-M and NCI-M schemes.

In the last section we investigated the numerical performance the discretization schemes. To strengthen our analysis, we aimed at accurate Monte Carlo estimates, which we tried to obtain by using a high number of sample paths in combination with a simple, but effective variance reduction technique based on the martingale property of the discretized asset price. We found that the schemes based on drift-interpolation in combination with an approximation based on analytical properties of the variance process, in terms of computational efficiency, by far outperformed the Euler, the Kahl-Jäckel, the (almost) exact simulation method. For the martingale corrected QE-M, NCI-M, NCI-QE-M and BK-DI-M schemes, we found that $2-8$ time steps a year already produces negligible biases, whereas the Euler and the Kahl-Jäckel Milstein scheme are still heavily biased even for 32 steps a year. Though the exact and AESM method are able to produce small biases, they show a lack of speed and are even outperformed by a simple Euler(-Full Truncation) scheme. Finally, we conclude that the QE-M scheme performed most efficiently, followed closely by the NCI-QE-M, NCI-M and the BK-DI-M scheme.

[24] As motivated in Lord et al. (2008) (and supported by our own findings) one should apply the Euler scheme with the full truncation fix for negative values that occur in discretizing the variance, and hence not the reflection/absorption fix that are being applied by Smith (2008) and Broadie and Kaya (2006).

A Appendix

A.1 Monotone cubic Hermite spline interpolation

In this appendix we show how to arrange the data in order to use monotone cubic Hermite spline interpolation on a monotone data set (e.g. in 51) according the algorithm of Fritsch and Carlson Fritsch and Carlson (1980): first, the four Hermite splines that form the basis of the interpolation rule are defined as:

$$h_{00}(t) = 2t^3 - 3t^2 + 1.0, \tag{71}$$

$$h_{10}(t) = t^3 - 2t^2 + t, \tag{72}$$

$$h_{01}(t) = -2t^3 - 3t^2, \tag{73}$$

$$h_{11}(t) = t^3 - t^2. \tag{74}$$

Then given two input vectors of x and y-values $x_i, y_i : i = 0, \ldots, n-1$ (i.e. the U_i's and the $H_i^{-1}(U_i)$'s), the weights $m_i, i = 0, \ldots, n-1$ can be found by the following algorithm:

1. Set $m_0 := \frac{y_1 - y_0}{x_1 - x_0}$, $m_{n-1} := \frac{y_{n-1} - y_{n-2}}{x_{n-1} - x_{n-2}}$ and $\Delta_0 := \frac{y_1 - y_0}{x_1 - x_0}$.
2. For $k = 1$ and while $k < n - 1$:
   (a) Set $m_k := \frac{1}{2}\left(\frac{y_k - y_{k-1}}{x_k - x_{k-1}} + \frac{y_{k+1} - y_k}{x_{k+1} - x_k}\right)$.
   (b) Set $\Delta_k := \frac{y_{k+1} - y_k}{x_{k+1} - x_k}$.
   (c) Let $k = k + 1$.
3. For $k = 0$ and while $k < n - 1$:
   (a) If $\Delta_k = 0$, set $m_k := m_{k+1} := 0$.
   (b) Let $k = k + 1$.
4. For $k = 0$ and while $k < n - 1$:
   (a) Let $a_k = \frac{m_k}{\Delta_k}$ and $b_k = \frac{m_{k+1}}{\Delta_k}$.
   (b) If $a_k^2 + b_k^2 \leq 9$, define $t_k = \Delta_k \frac{3}{\sqrt{a_k^2 + b_k^2}}$ and set $m_k := t_k a_k$, $m_{k+1} := t_k b_k$.
   (c) Let $k = k + 1$.

To use the interpolation rule (51), this data preparation of course only has to be done once (i.e. before running the Monte Carlo).

A.2 Effectiveness of the control variate

Roughly, the control variate estimators exploit information about the observed errors in the stock prices $\overline{S}(T) - \mathbb{E}[S(T)]$ (which should be zero by definition) to reduce the errors in the estimate of the vanilla or Asian call option prices (e.g. see Glasserman (2003)). Hence the stock price serves as a control in estimating the vanilla and Asian call option price. If $b_C, b_A = 0$ the control variate estimates fall back to the ordinary Monte Carlo estimates, which usually is not the optimal choice; specifically, in the case of a vanilla call option, the control variate estimator has variance

$$\mathrm{Var}[C_i(b_C)] = \mathrm{Var}[C_i - b_C(S_i - \mathbb{E}[S])] = \sigma_C^2 - 2b_C\sigma_C\sigma_S + b^2\sigma_S^2, \tag{75}$$

which, if $b_C^2\sigma_S^2 < 2b_C\sigma_C\sigma_S$, has smaller variance than the ordinary estimator. Minimizing the variance (75) over b_C then yields that optimal variance reduction is achieved with coefficient

$$b_C^* = \frac{\sigma_C}{\sigma_S}\rho_{CS} = \frac{\mathrm{Cov}(C,S)}{\mathrm{Var}(S)}. \tag{76}$$

Substituting this optimal coefficient[25] into (75) one finds that the control variate is expected to reduce the variance of the ordinary Monte Carlo estimator by a factor

$$\frac{\mathrm{Var}[\overline{C}]}{\mathrm{Var}[\overline{C} - b^*(\overline{S} - \mathbb{E}[S])]} = \frac{1}{1-\rho_{CS}^2}. \tag{78}$$

Notice hereby that the effectiveness of a control variate (crucially) depends on the correlation between the estimated quantity and the used control;
Since the price of call option with a strike of $K = 0$ equals the current stock price (or future discounted stock price) we have $\rho_{CS} = 1$ for a strike call option with strike $K = 0$ and hence one finds a perfect control variate estimator. The other way around, since for higher strikes the correlation between stock and call option price decreases, we will find less effective control variates. This behavior is confirmed by table **5** where we report the variance reduction factors when comparing the control variate estimates with the ordinary Monte Carlo estimates for the price of the call options.
The table is organized as follows: on the first row one can find equally spaced strikes varying from 50 to 130 and on the first column are the corresponding equally spaced maturities varying from 1 to 5 years. The Monte Carlo results are obtained by using the NCI scheme[26] with 8 time steps a year and 10^6 Monte Carlo paths.

	50	60	70	80	90	100	110	120	130	140
case I	101.6	58.8	37.1	24.9	17.6	13.0	9.9	7.8	6.3	5.3
case II	31.1	18.3	11.8	8.1	5.9	4.4	3.4	2.8	2.3	1.9
case III	54.5	31.7	20.5	14.3	10.6	8.2	6.5	5.4	4.6	3.9
case IV	97.4	38.4	17.8	9.3	5.4	3.4	2.3	1.7	1.4	1.2

Table **5**: Variance reductions for the full truncation scheme for the cases of table **2**. Reported is the fraction between the variance of the control variate (68) and the standard Monte Carlo estimator.

[25] If the optimal coefficient cannot be calculated in closed-form, it can be estimated by estimating the sample covariance and variance estimators in (76), e.g. with

$$\widehat{b_C} = \frac{\sum_{i=1}^{n}(C_i - \overline{C})(S_i - \overline{S})}{\sum_{i=1}^{n}(S_i - \overline{S})^2} \tag{77}$$

One can then replace b_C^* by its estimate $\widehat{b_C}$, which estimation procedure might cause some efficiency loss in the quality of the control variate estimator, see Glasserman (2003).

[26] Results for the other schemes are completely similar and available upon request.

A.3 Numerical results for the second and third test case

For **case II** and **case III**, table **6** and **7** list the Monte Carlo estimates of the bias (67) as function of the time step Δt varying from 1/32 year to 1 year and for the strikes $K = 100, 140, 60$. Numbers in parentheses are the widths of the confidence interval (70) at a 99% confidence level: we starred the biases that were not significantly different from zero.

	Euler-FT	IM-IJK	QE-M	NCI-M	NCI-QE-M	BK-DI-M
Δt	$K = 100$					
1	-2.395 (±0.045)	0.847 (±0.148)	0.084 (±0.040)	0.168 (±0.040)	0.177 (±0.040)	0.177 (±0.040)
1/2	-1.030 (±0.041)	1.265 (±0.142)	0.019* (±0.039)	0.053 (±0.039)	0.056 (±0.039)	0.074 (±0.039)
1/4	-0.420 (±0.040)	1.277 (±0.141)	0.026* (±0.039)	0.037* (±0.039)	0.029* (±0.039)	0.025* (±0.039)
1/8	-0.175 (±0.040)	0.955 (±0.143)	0.015* (±0.039)	-0.011* (±0.039)	0.019* (±0.039)	0.009* (±0.039)
1/16	-0.043 (±0.039)	0.649 (±0.145)	0.020* (±0.039)	-0.001* (±0.039)	-0.014* (±0.039)	-0.004* (±0.039)
1/32	-0.032* (±0.039)	0.398 (±0.147)	-0.003* (±0.039)	-0.010* (±0.039)	0.019* (±0.039)	-0.005* (±0.039)
	$K = 140$					
1	-3.185 (±0.062)	0.184 (±0.126)	0.568 (±0.054)	0.456 (±0.054)	0.479 (±0.054)	0.476 (±0.054)
1/2	-1.379 (±0.056)	1.029 (±0.120)	0.195 (±0.053)	0.151 (±0.053)	0.153 (±0.053)	0.163 (±0.053)
1/4	-0.578 (±0.054)	1.242 (±0.119)	0.051* (±0.053)	0.063 (±0.053)	0.067 (±0.053)	0.043* (±0.053)
1/8	-0.241 (±0.053)	1.008 (±0.121)	0.029* (±0.053)	-0.015* (±0.053)	0.037* (±0.053)	0.020* (±0.053)
1/16	-0.056 (±0.053)	0.711 (±0.123)	0.031* (±0.053)	-0.002* (±0.053)	-0.020* (±0.053)	-0.022* (±0.053)
1/32	-0.041* (±0.053)	0.430 (±0.125)	0.000* (±0.053)	-0.018* (±0.053)	0.016* (±0.053)	0.001* (±0.053)
	$K = 60$					
1	-1.082 (±0.023)	1.851 (±0.167)	-0.128 (±0.020)	-0.045 (±0.020)	-0.045 (±0.020)	-0.046 (±0.020)
1/2	-0.441 (±0.021)	1.659 (±0.161)	-0.035 (±0.020)	-0.011* (±0.020)	-0.009* (±0.020)	0.001* (±0.020)
1/4	-0.166 (±0.020)	1.345 (±0.159)	0.008* (±0.020)	0.009* (±0.020)	-0.002* (±0.020)	0.007* (±0.020)
1/8	-0.072 (±0.020)	0.898 (±0.161)	0.005* (±0.020)	-0.009* (±0.020)	0.008* (±0.020)	-0.003* (±0.020)
1/16	-0.020 (±0.020)	0.580 (±0.162)	0.007* (±0.020)	0.005* (±0.020)	-0.003* (±0.020)	0.005* (±0.020)
1/32	-0.019* (±0.020)	0.354 (±0.165)	-0.009* (±0.020)	0.001* (±0.020)	0.006* (±0.020)	0.000* (±0.020)

Table **6**: Estimated call option price biases for case II. Exact prices respectively are 33.597, 18.157 and 56.575.

	Euler-FT	IM-IJK	QE-M	NCI-M	NCI-QE-M	BK-DI-M
Δt	$K = 100$					
1	-7.035 (±0.074)	-22.466 (±0.188)	0.451 (±0.063)	0.148 (±0.052)	0.100 (±0.064)	0.129 (±0.054)
1/2	-4.073 (±0.063)	-12.305 (±0.148)	0.131 (±0.056)	0.085 (±0.051)	0.044* (±0.052)	0.019* (±0.053)
1/4	-2.208 (±0.057)	-6.535 (±0.137)	0.028* (±0.052)	0.040* (±0.054)	0.039* (±0.051)	0.000* (±0.052)
1/8	-1.126 (±0.054)	-3.257 (±0.116)	0.008* (±0.052)	0.021* (±0.053)	-0.032* (±0.054)	-0.014* (±0.054)
1/16	-0.601 (±0.055)	-1.608 (±0.115)	-0.008* (±0.055)	0.006* (±0.051)	-0.036* (±0.058)	-0.003* (±0.053)
1/32	-0.268 (±0.055)	-0.725 (±0.115)	0.021* (±0.054)	-0.004* (±0.051)	-0.072 (±0.054)	0.023* (±0.053)
	$K = 140$					
1	-6.089 (±0.094)	-17.551 (±0.162)	0.269 (±0.078)	0.103 (±0.059)	0.039* (±0.082)	0.082 (±0.063)
1/2	-3.278 (±0.077)	-8.685 (±0.124)	0.010* (±0.067)	0.073 (±0.057)	0.037* (±0.059)	-0.014* (±0.061)
1/4	-1.618 (±0.067)	-4.073 (±0.117)	-0.002* (±0.059)	0.044* (±0.063)	0.048* (±0.058)	-0.002* (±0.060)
1/8	-0.758 (±0.062)	-1.758 (±0.097)	0.012* (±0.060)	0.017* (±0.061)	-0.022* (±0.063)	-0.006* (±0.063)
1/16	-0.416 (±0.065)	-0.779 (±0.098)	-0.018* (±0.066)	-0.005* (±0.058)	-0.048* (±0.070)	-0.014* (±0.062)
1/32	-0.165 (±0.065)	-0.327 (±0.098)	0.015* (±0.063)	-0.019* (±0.058)	-0.082 (±0.063)	0.003* (±0.061)
	$K = 60$					
1	-3.647 (±0.043)	-18.214 (±0.214)	-0.141 (±0.034)	0.019* (±0.030)	-0.001* (±0.034)	0.009* (±0.031)
1/2	-2.023 (±0.037)	-9.165 (±0.172)	-0.040 (±0.032)	0.025* (±0.030)	0.005* (±0.030)	-0.006* (±0.031)
1/4	-1.061 (±0.034)	-4.504 (±0.158)	0.007* (±0.030)	0.006* (±0.031)	0.009* (±0.030)	-0.013* (±0.031)
1/8	-0.529 (±0.032)	-2.136 (±0.136)	0.020* (±0.031)	0.009* (±0.031)	-0.031* (±0.031)	-0.007* (±0.031)
1/16	-0.276 (±0.032)	-1.084 (±0.134)	0.007* (±0.031)	0.016* (±0.030)	-0.025* (±0.032)	-0.010* (±0.031)
1/32	-0.123 (±0.032)	-0.505 (±0.132)	0.017* (±0.031)	0.007* (±0.030)	-0.029* (±0.031)	0.019* (±0.031)

Table **7**: Estimated call option price biases for case III. Exact prices respectively are 16.649, 5.138 and 45.287.

Bibliography

M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables (9th version)*. Dover (New York), 1964.

J.H. Ahrens and U. Dieter. Computer generation of poisson deviates from modified normal distributions. *ACM Transactions on Mathematical Software*, 8(2):163–179, 1982.

H. Albrecher, P. Mayer, W. Schoutens, and J. Tistaert. The little heston trap. http://perswww.kuleuven.be/%7eu0009713/HestonTrap.pdf, 2005.

L. Andersen. Efficient simulation of the heston stochastic volatility model. http://papers.ssrn.com/sol3/papers.cfm?abstract_id=946405, 2007.

L. Andersen and R. Brotherton-Ratcliffe. Extended libor market models with stochastic volatility. Working paper, Gen Re Securities, 2001.

L. Andersen and V. Piterbarg. Moment explosions in stochastic volatility models. *Finance and Stochastics*, 11(1), 2007. http://ssrn.com/abstract=685084.

F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3), 1973.

M. Broadie and Ö. Kaya. Exact simulation of stochastic volatility and other affine jump diffusion models. *Operations Research*, 54(2), 2006.

P. Carr and D.B. Madan. Option valuation using the fast fourier transform. *Journal of Computational Finance*, 2:61–73, 1999.

J.C. Cox, J.E. Ingersoll, and S.A. Ross. A theory of the term structure of interest rates. *Econometrica*, 53(2):385–407, 1985.

A. R. Didonato and A. H. Morris. Computation of the incomplete gamma function ratios and their inverse. *ACM Trans. Math. Software*, 12:377–393, 1986.

F.N. Fritsch and R.E. Carlson. Monotone piecewise cubic interpolation. *SIAM Journal on Numerical Analysisa*, 17(2):238–246, 1980. http://en.wikipedia.org/wiki/Monotone_cubic_spline.

P. Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer Verlag, 2003.

S.L. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Review of Financial Studies*, 6:327–343, 1993.

J. Hull and A. White. he pricing of options on assets with stochastic volatilities. *Journal of Finance*, 42(2):281–300, 1987.

P. Jäckel. *Monte Carlo Methods in Finance*. Wiley Finance, 2002.

N.L. Johnson, S. Kotz, and N. Balakrishnan. *Continuous univariate distributions*, volume 2. Wiley, second edition, 1994.

C. Kahl and P. Jäckel. Fast strong approximation monte-carlo schemes for stochastic volatility models. *Quantative Finance*, 6(6), 2006.

D.E. Knuth. *The Art of Computer Programming, Vol.2: Seminumerical Algorithms*. Addison-Wesley, Reading, Mass., 1981.

R. Lee. Option pricing by transform methods: extension, unification and error control. *Journal of Computational Finance*, 7:51–86, 2004.

R. Lord and J. Kahl. Optimal fourier inversion in semi-analytical option pricing. *Journal of Computational Finance*, 10(4), 2008.

R. Lord, R. Koekkoek, and D. van Dijk. A comparison of biased simulation schemes for stochastic volatility models. *Journal of Quantitative Finance*, 2008.

J. Maddock, P. A. Bristow, H. Holin, and X. Zhang. Boost math toolkit, 2008. freely available from Boost's C++ library, version 1.35: http://www.boost.org.

G. Marsaglia and W.W. Tsang. A simple method for generating gamma variables. *Transactions on Mathematical Software*, 26(3):363–372, 2000.

W.H. Press and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, second edition, 1992.

R. Schöbel and J. Zhu. Stochastic volatility with an ornstein uhlenbeck process: An extension. *European Finance Review*, 4:23–46, 1999.

R.D. Smith. An almost exact simulation method for the heston model. *Journal of Cumputational Finance*, 11(1):115–125, 2008.

E.M. Stein and J.C. Stein. Stock-price distributions with stochastic volatility: an analytic approach. *Review of Financial Studies*, 4:727–752, 1991.

M.J. Wichura. Algorithm as 241: The percentage points of the normal distribution. *Applied Statistics*, 37:477–484, 1998.

G.A. Willard. Calculating prices and sensitivities for path-independent derivative securities in multifactor models. *The Journal of Derivatives*, 5(1):45–61, 1997.