

SIMULATION OF SQUARE-ROOT PROCESSES

LEIF B.G. ANDERSEN, PETER JÄCKEL, AND CHRISTIAN KAHL

ABSTRACT. We discuss methods for time-discretization and simulation of square-root SDEs, both in isolation (CIR process) and as part of vector-SDEs modeling stochastic volatility (Heston model). Both exact and biased discretization methods are covered.

1. INTRODUCTION

Square-root diffusion processes are popular in many branches of quantitative finance. Guaranteed to stay non-negative, yet almost as tractable as a Gaussian process, the mean-reverting square-root process has found applications ranging from short-rate modeling of the term structure of interest rates, to credit derivative pricing, and to stochastic volatility models, just to name a few.

A thorough description of the theoretical properties of square-root processes as well as their generalization into multi-factor affine jump-diffusion processes can be found in [7] where a full literature survey is also available. While we shall rely on some of these results in the remainder of this article, our focus here is on the problem of generating Monte Carlo paths for the one-factor square-root process, first in isolation and later in combination with a log-normal asset process (as required in most stochastic volatility applications). As we shall see, such path-generation can, under many relevant parameter settings, be surprisingly challenging. Indeed, despite the popularity and the longevity of the square-root diffusion – the first uses in finance date back several decades – it is only in the last few years that a truly satisfactory palette of algorithms has been established.

2. PROBLEM DEFINITION AND KEY THEORETICAL RESULTS

Let $x(t)$ be a scalar random variable satisfying a stochastic differential equation (SDE) of the mean-reverting square-root type, that is

$$(1) \quad dx(t) = \kappa(\theta - x(t)) dt + \epsilon\sqrt{x(t)} dW(t), \quad x(0) = x_0,$$

where κ, θ, ϵ are positive constants and $W(t)$ is a Brownian motion in a given probability measure. Applications in finance of (1) includes the seminal CIR model for interest rates (see **eqf11-025**) and the Heston stochastic volatility model (see **eqf08-018**). In practical usage of such models (e.g. to price options) we are often faced with the problem of generating Monte Carlo paths of x on some discrete timeline. To devise a simulation scheme, it suffices to contemplate the more fundamental question of how, for an arbitrary increment Δ , to generate a random sample of $x(t + \Delta)$ given $x(t)$; repeated application of the resulting one-period scheme will produce a full paths of x on an arbitrary set of discrete dates.

Key words and phrases. CIR process, Heston process, Monte Carlo simulation, Euler scheme, Milstein scheme, moment matching, SDE discretization.

To aid in the construction of simulation algorithms, let us quickly review a few well-known theoretical results for (1).

Proposition 2.1. *Let $F(z; \nu, \lambda)$ be the cumulative distribution function for the non-central chi-square distribution with ν degrees of freedom and non-centrality parameter λ :*

$$(2) \quad F(z; \nu, \lambda) = e^{-\lambda/2} \sum_{j=0}^{\infty} \frac{(\lambda/2)^j}{j! 2^{\nu/2+j} \Gamma(\nu/2+j)} \int_0^z y^{\nu/2+j-1} e^{-y/2} dy .$$

For the process (1) define

$$(3) \quad d = 4\kappa\theta/\epsilon^2 ; \quad n(t, T) = \frac{4\kappa e^{-\kappa(T-t)}}{\epsilon^2 (1 - e^{-\kappa(T-t)})} , \quad T > t .$$

Let $T > t$. Conditional on $x(t)$, $x(T)$ is distributed as $e^{-\kappa(T-t)}/n(t, T)$ times a non-central chi-square distribution with d degrees of freedom and non-centrality parameter $x(t)n(t, T)$. That is,

$$(4) \quad \Pr(x(T) < x|x(t)) = F\left(\frac{x \cdot n(t, T)}{e^{-\kappa(T-t)}}; d, x(t) \cdot n(t, T)\right) .$$

From the known properties of the non-central chi-square distribution, the following Corollary easily follows.

Corollary 2.2. *For $T > t$, $x(T)$ has the following first two conditional moments:*

$$(5) \quad \mathbb{E}(x(T)|x(t)) = \theta + (x(t) - \theta) e^{-\kappa(T-t)} ;$$

$$(6) \quad \text{Var}(x(T)|x(t)) = \frac{x(t)\epsilon^2 e^{-\kappa(T-t)}}{\kappa} (1 - e^{-\kappa(T-t)}) + \frac{\theta\epsilon^2}{2\kappa} (1 - e^{-\kappa(T-t)})^2 .$$

Proposition 2.3. *Assume that $x(0) > 0$. If $2\kappa\theta \geq \epsilon^2$ then the process for x can never reach zero. If $2\kappa\theta < \epsilon^2$, the origin is accessible and strongly reflecting.*

The condition $2\kappa\theta \geq \epsilon^2$ in Proposition 2.3 is often known as the *Feller condition* (see [12]) for (1). When (1) is used as a model for interest rates or credit spreads, market-implied model parameters are typically such that the Feller condition is satisfied. However, when (1) represent a stochastic variance process (as in Section 4), the Feller condition rarely holds. As it turns out, a violation of the Feller condition may increase the difficulty of Monte Carlo path generation considerably.

3. SIMULATION SCHEMES

3.1. Exact Simulation. According to Proposition 2.1, the distribution of $x(t+\Delta)$ given $x(t)$ is known in closed form. Generation of a random sample of $x(t+\Delta)$ given $x(t)$ can therefore be done entirely bias-free by sampling from a non-central chi-square distribution. Using the fact that a non-central chi-square distribution can be seen as a regular chi-square distribution with Poisson-distributed degrees of freedom (see [9]), the following algorithm can be used.

- (1) Draw a Poisson random variable N , with mean $\frac{1}{2}x(t)n(t, t+\Delta)$.
- (2) Given N , draw a regular chi-square random variable χ_v^2 , with $v = d + 2N$ degrees of freedom.
- (3) Set $x(t+\Delta) = \chi_v^2 \cdot \exp(-\kappa\Delta) / n(t, t+\Delta)$.

Steps 1 and 3 of this algorithm are straightforward, but Step 2 is somewhat involved. In practice, generation of chi-squared variables would most often use one of several available techniques for the gamma-distribution, a special case of which is the chi-square distribution. A standard algorithm for the generation of Gamma variates of acceptance-rejection type is the Cheng-Feast algorithm [5], and a number of others are listed in [9], though, direct generation by the aid of the inverse cumulative distribution function [6] is also a practically viable option.

We should note that if $d > 1$, it may be numerically advantageous to use a different algorithm, based on the convenient relation

$$(7) \quad \chi_d^2(\lambda) \stackrel{d}{=} \left(Z + \sqrt{\lambda} \right)^2 + \chi_{d-1}^2, \quad d > 1,$$

where $\stackrel{d}{=}$ denotes equality in distribution, $\chi_d^2(\lambda)$ is a non-central chi-square variable with d degrees of freedom and non-centrality parameter λ , and Z is an ordinary $\mathcal{N}(0, 1)$ Gaussian variable. We trust that the reader can complete the details on application of (7) in a simulation algorithm for $x(t + \Delta)$.

One might think that the existence of an exact simulation-scheme for $x(t + \Delta)$ would settle once and for all the question of how to generate paths of the square-root process. In practice, however, several complications may arise with the application of the algorithm above. Indeed, the scheme is quite complex compared with many standard SDE discretization schemes and may not fit smoothly into existing software architecture for SDE simulation routines. Also, computational speed may be an issue, and the application of acceptance-rejection sampling will potentially cause a “scrambling effect” when process parameters are perturbed, resulting in poor sensitivity computations. While caching techniques can be designed to overcome some of these issues, storage, look-up, and interpolation of such a cache pose their own challenges. Further, the basic scheme above provides no explicit link between the paths of the Brownian motion $W(t)$ and that of $x(t)$, complicating applications in which, say, multiple correlated Brownian motions need to be advanced through time.

In light of the discussion above, it seems reasonable to also investigate the application of simpler simulation algorithms. These will typically exhibit a bias for finite values of Δ , but convenience and speed may more than compensate for this, especially if the bias is small and easy to control by reduction of step-size. We proceed to discuss several classes of such schemes.

3.2. Biased Taylor-Type Schemes.

3.2.1. *Euler Schemes.* Going forward, let us use \hat{x} to denote a discrete-time (biased) approximation to x . A classical approach to constructing simulation schemes for SDEs involves the application of Itô-Taylor expansions, suitably truncated. See **eqf13-001**, **eqf13-002**, and **eqf13-23** for details. The simplest such scheme is the *Euler scheme*, a direct application of which would write

$$(8) \quad \hat{x}(t + \Delta) = \hat{x}(t) + \kappa(\theta - \hat{x}(t))\Delta + \epsilon\sqrt{\hat{x}(t)}Z\sqrt{\Delta},$$

where Z is $\mathcal{N}(0, 1)$ Gaussian variable. One immediate (and fatal) problem with (8) is that the discrete process for x can become negative with non-zero probability, making computation of $\sqrt{\hat{x}(t)}$ impossible and causing the time-stepping scheme to fail. To get around this problem, several remedies have been proposed in the literature, starting with the suggestion in [13] that one simply replace $\sqrt{\hat{x}(t)}$ in (8)

with $\sqrt{|\hat{x}(t)|}$. [14] reviews a number of such “fixes”, concluding that the following works best:

$$(9) \quad \hat{x}(t + \Delta) = \hat{x}(t) + \kappa(\theta - \hat{x}(t)^+) \Delta + \epsilon \sqrt{\hat{x}(t)^+} Z \sqrt{\Delta},$$

where we use the notation $x^+ = \max(x, 0)$. In [14] this scheme is denoted “full truncation”; its main characteristic is that the process for \hat{x} is allowed to go below zero, at which point the process for x becomes deterministic with an upward drift of $\kappa\theta$.

3.2.2. Higher-Order Schemes. The scheme (9) has first-order weak convergence, in the sense that expectations of functions of x will approach their true values as $\mathcal{O}(\Delta)$. To improve convergence, it is tempting to apply a *Milstein scheme*, the most basic of which is

$$(10) \quad \hat{x}(t + \Delta) = \hat{x}(t) + \kappa(\theta - \hat{x}(t)) \Delta + \epsilon \sqrt{\hat{x}(t)} Z \sqrt{\Delta} + \frac{1}{4} \epsilon^2 \Delta (Z^2 - 1).$$

As was the case for (8), this scheme has a positive probability of generating negative values of \hat{x} and therefore cannot be used without suitable modifications. [11] list several other Milstein-type schemes, some of which allow for a certain degree of control over the likelihood of generating negative values. One particularly appealing variation is the *implicit Milstein scheme*, defined as

$$(11) \quad \hat{x}(t + \Delta) = \frac{\hat{x}(t) + \kappa\theta\Delta + \epsilon \sqrt{\hat{x}(t)} Z \sqrt{\Delta} + \frac{1}{4} \epsilon^2 \Delta (Z^2 - 1)}{1 + \kappa\Delta}.$$

It is easy to verify that this discretization scheme will result in strictly positive paths for the x process if $4\kappa\theta > \epsilon^2$. For cases where this bound does not hold, it will be necessary to modify (11) to prevent problems with the computation of $\sqrt{\hat{x}(t)}$. For instance, whenever $\hat{x}(t)$ drops below zero, we could use (9) rather than (11).

Under certain sufficient regularity conditions, Milstein schemes have second-order weak convergence. Due to the presence of a square-root in (1), these sufficient conditions are violated here, and one should not expect (11) to have second-order convergence for all parameter values, even the ones that satisfy $4\kappa\theta > \epsilon^2$. Numerical tests of Milstein schemes for square-root processes can be found in [11] and [9]; overall these schemes perform fairly well in certain parameter regimes, but are typically less robust than the Euler scheme.

3.3. Moment Matching Schemes.

3.3.1. Log-Normal Approximation. The simulation schemes introduced in Section 3.2 all suffer to various degrees from an inability to keep the path of x non-negative throughout. One, rather obvious, way around this is to draw $\hat{x}(t + \Delta)$ from a user-selected probability distribution that i) is reasonably close to the true distribution of $x(t + \Delta)$; and ii) is certain not to produce negative values. To ensure that i) is satisfied, it is natural to select the parameters of the chosen distribution to match one or more of the true moments for $x(t + \Delta)$, conditional upon $x(t) = \hat{x}(t)$. For instance, if we assume that the true distribution of $x(t + \Delta)$ is well approximated by a log-normal distribution with parameters μ and σ , we write (see [2])

$$(12) \quad \hat{x}(t + \Delta) = e^{\mu + \sigma Z},$$

where Z is a Gaussian random variable, and μ, σ are chosen to satisfy

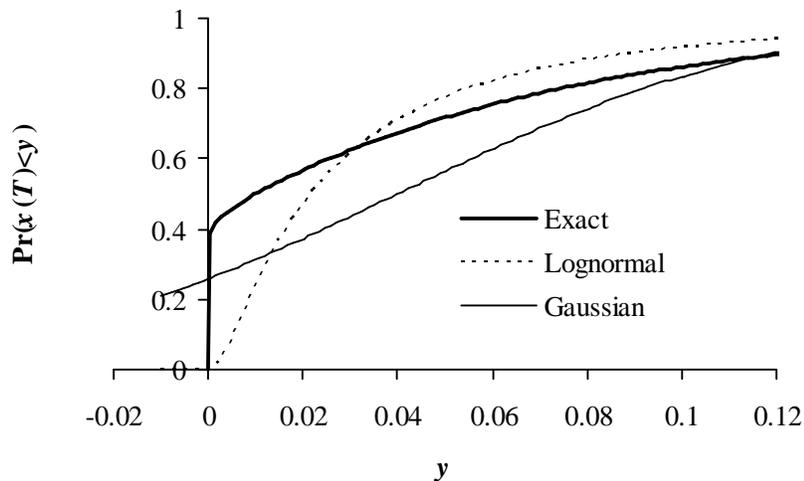
$$(13) \quad e^{\mu + \frac{1}{2}\sigma^2} = \mathbb{E}(x(t + \Delta) | x(t) = \hat{x}(t)) ,$$

$$(14) \quad e^{2(\mu + \frac{1}{2}\sigma^2)} (e^{\sigma^2} - 1) = \text{Var}(x(t + \Delta) | x(t) = \hat{x}(t)) .$$

The results in Corollary 2.2 can be used to compute the right-hand sides of this system of equations, which can then easily be solved analytically for μ and σ .

As is the case for many other schemes, (12) works best if the Feller condition is satisfied. If not, the lower tail of the log-normal distribution is often too thin to capture the true distribution shape of $\hat{x}(t + \Delta)$ – see Figure 1.

FIGURE 1. Cumulative Distribution of x



Notes: The figure shows the cumulative distribution function for $x(T)$ given $x(0)$, with $T = 0.1$. Model parameters were $x(0) = \theta = 4\%$, $\kappa = 50\%$, and $\epsilon = 100\%$. The lognormal and Gaussian distributions in the graph were parameterized by matching mean and variances to the exact distribution of $x(T)$.

3.3.2. *Truncated Gaussian.* Figure 1 demonstrates that the density of $\hat{x}(t + \Delta)$ may sometimes be nearly singular at the origin. To accommodate this, one could contemplate inserting an actual singularity through outright truncation at the origin of a distribution that may otherwise go negative. Using a Gaussian distribution for this, say, one could write

$$(15) \quad \hat{x}(t + \Delta) = (\mu + \sigma Z)^+ ,$$

where μ and σ are determined by moment-matching, along the same lines as in Section 3.3.1 above. While this moment-matching exercise cannot be done in entirely analytical fashion, a number of caching tricks outlined in [3] can be used to make the determination of μ and σ essentially instantaneous. As documented in [3], the scheme (15) is robust and generally has attractive convergence properties when applied to standard option pricing problems. Being fundamentally Gaussian when $\hat{x}(t)$ is far from the origin, (15) is somewhat similar to the Euler scheme (9),

although performance of (15) is typically somewhat better than (9). Unlike (9), the truncated Gaussian scheme (15) also ensures, by construction, that negative values of $\hat{x}(t + \Delta)$ cannot be attained.

3.3.3. Quadratic-Exponential. We finish our discussion of biased schemes for 1 with a more elaborate moment-matched scheme, based on a combination of a squared Gaussian and an exponential distribution. In this scheme, for large values of $\hat{x}(t)$, we write

$$(16) \quad \hat{x}(t + \Delta) = a(b + Z)^2 ,$$

where Z is a standard Gaussian random variable, and a and b are certain constants, to be determined by moment-matching. a and b will depend on the time-step Δ and $\hat{x}(t)$, as well as the parameters in the SDE for x . While based on well-established asymptotics for the non-central chi-square distribution (see [3]), formula (16) does not work well for low values of $\hat{x}(t)$ – in fact, the moment-matching exercise fails to work – so we supplement it with a scheme to be used when $\hat{x}(t)$ is small. [3] shows that a good choice is to approximate the density of $\hat{x}(t + \Delta)$ with

$$(17) \quad \Pr(\hat{x}(t + \Delta) \in [x, x + dx]) \approx (p\delta(x) + \beta(1 - p)e^{-\beta x}) dx , \quad x \geq 0 ,$$

where δ is a Dirac delta-function, and p and β are non-negative constants to be determined. As in the scheme in Section 3.3.2, we have a probability mass at the origin, but now the strength of this mass (p) is explicitly specified, rather than implied from other parameters. The mass at the origin is supplemented with an exponential tail. It can be verified that if $p \in [0, 1]$ and $\beta \geq 0$, then (17) constitutes a valid density function.

Assuming that we have determined a and b , Monte Carlo sampling from (16) is trivial. To draw samples in accordance with (17), we can generate a cumulative distribution function

$$(18) \quad \Psi(x) = \Pr(\hat{x}(t + \Delta) \leq x) = p + (1 - p)(1 - e^{-\beta x}) , \quad x \geq 0 ,$$

the inverse of which is readily computable, allowing for efficient generation of random draws by the inverse distribution method.

What remains is the determination of the constants a , b , p , and β , as well as a rule for when to switch from (16) to sampling from (18). The first problem is easily settled by moment-matching techniques.

Proposition 3.1. *Let $m \triangleq \mathbb{E}(x(t + \Delta)|x(t) = \hat{x}(t))$ and $s^2 \triangleq \text{Var}(x(t + \Delta)|x(t) = \hat{x}(t))$ and set $\psi = s^2/m^2$. Provided that $\psi \leq 2$, set*

$$(19) \quad b^2 = 2\psi^{-1} - 1 + \sqrt{2\psi^{-1} - 1} \sqrt{2\psi^{-1} - 1} \geq 0$$

and

$$(20) \quad a = \frac{m}{1 + b^2} .$$

Let $\hat{x}(t + \Delta)$ be as defined in (16); then $\mathbb{E}(\hat{x}(t + \Delta)) = m$ and $\text{Var}(\hat{x}(t + \Delta)) = s^2$.

Proposition 3.2. *Let m , s , and ψ be as defined in Proposition 3.1. Assume that $\psi \geq 1$ and set*

$$(21) \quad p = \frac{\psi - 1}{\psi + 1} \in [0, 1] ,$$

and

$$(22) \quad \beta = \frac{1-p}{m} = \frac{2}{m(\psi+1)} > 0.$$

Let $\hat{x}(t+\Delta)$ be sampled from (18); then $E(\hat{x}(t+\Delta)) = m$ and $\text{Var}(\hat{x}(t+\Delta)) = s^2$.

The terms m, s, ψ defined in the two propositions above are explicitly computable from the result in Corollary 2.2. For any ψ_c in $[1, 2]$, a valid *switching rule* is to use (16) if $\psi \leq \psi_c$ and to sample (18) otherwise. The exact choice for ψ_c is non-critical; $\psi_c = 1.5$ is a good choice.

The quadratic-exponential (QE) scheme outlined above is typically the most accurate of the biased schemes introduced in this article. Indeed, in most practical application the bias introduced by the scheme is statistically undetectable at the levels of Monte Carlo noise acceptable in practical applications; see [3] for numerical tests under a range of challenging conditions. Variations on the QE scheme without an explicit singularity in zero can also be found in [3].

4. STOCHASTIC VOLATILITY SIMULATION

As mentioned earlier, square-root processes are commonly used to model stochastic movements in the volatility of some financial asset. A popular example of such an application is the *Heston model* [10], defined by a vector-SDE of the form¹

$$(23) \quad dY(t) = Y(t)\sqrt{x(t)} dW_Y(t),$$

$$(24) \quad dx(t) = \kappa(\theta - x(t)) dt + \epsilon\sqrt{x(t)} dW(t),$$

with $dW_Y(t) \cdot dW(t) = \rho dt$, $\rho \in [-1, 1]$. For numerical work, it is useful to recognize that the process for $Y(t)$ is often relatively close to geometric Brownian motion, making it sensible to work with logarithms of $Y(t)$, rather than $Y(t)$ itself. An application of Itô's Lemma shows that (23)-(24) is equivalent to

$$(25) \quad d \ln Y(t) = -\frac{1}{2}x(t) dt + \sqrt{x(t)} dW_Y(t),$$

$$(26) \quad dx(t) = \kappa(\theta - x(t)) dt + \epsilon\sqrt{x(t)} dW(t).$$

We proceed to consider the joint simulation of (25)-(26).

4.1. Broadie-Kaya Scheme. As demonstrated in [4], it is possible to simulate (25)-(26) bias-free. To show this, first integrate the SDE for $x(t)$ and rearrange:

$$(27) \quad \int_t^{t+\Delta} \sqrt{x(u)} dW(u) = \epsilon^{-1} \left(x(t+\Delta) - x(t) - \kappa\theta\Delta + \kappa \int_t^{t+\Delta} x(u) du \right).$$

Performing a Cholesky decomposition we can also write

$$(28) \quad d \ln Y(t) = -\frac{1}{2}x(t) dt + \rho\sqrt{x(t)} dW(t) + \sqrt{1-\rho^2}\sqrt{x(t)} dW^*(t),$$

where W^* is a Brownian motion independent of W . An integration yields

$$(29) \quad \ln Y(t+\Delta) = \ln Y(t) + \frac{\rho}{\epsilon} (x(t+\Delta) - x(t) - \kappa\theta\Delta) \\ + \left(\frac{\kappa\rho}{\epsilon} - \frac{1}{2} \right) \int_t^{t+\Delta} x(u) du + \sqrt{1-\rho^2} \int_t^{t+\Delta} \sqrt{x(u)} dW^*(u),$$

¹We assume that Y is a martingale in the chosen measure; adding a drift is straightforward.

where we have used (27). Conditional on $x(t + \Delta)$ and $\int_t^{t+\Delta} x(u) du$, it is clear that the distribution of $\ln Y(t + \Delta)$ is Gaussian with easily computable moments. After first sampling $x(t + \Delta)$ from the non-central chi-square distribution (as described in Section 3.1), one then performs the following steps:

- (1) Conditional on $x(t + \Delta)$ (and $x(t)$) draw a sample of $I = \int_t^{t+\Delta} x(u) du$.
- (2) Conditional on $x(t + \Delta)$ and I , use (29) to draw a sample of $\ln Y(t + \Delta)$ from a Gaussian distribution.

While execution of the second step is straightforward, the first one is decidedly not, as the conditional distribution of the integral I is not known in closed form. In [4], the authors instead derive a characteristic function, which they numerically Fourier-invert to generate the cumulative distribution function for I , given $x(t + \Delta)$ and $x(t)$. Numerical inversion of this distribution function over a uniform random variable finally allows for generation of a sample of I . The total algorithm requires great care in numerical discretization to prevent introduction of noticeable biases and is further complicated by the fact that the characteristic function for I contains two modified Bessel functions.

The Broadie-Kaya algorithm is bias-free by construction, but its complexity and lack of speed is problematic in some applications. At the cost of introducing a (small) bias, [15] improves computational efficiency by introducing certain approximations to the characteristic function of time-integrated variance, enabling efficient caching techniques.

4.2. Other Schemes.

4.2.1. *Taylor-Type Schemes.* In their examination of “fixed” Euler-schemes, [14] suggests simulation of the Heston model by combining (9) with the following scheme for $\ln Y$:

$$(30) \quad \ln \hat{Y}(t + \Delta) = \ln \hat{Y}(t) - \frac{1}{2} \hat{x}(t)^+ \Delta + \sqrt{\hat{x}(t)^+} Z_Y \sqrt{\Delta},$$

where Z_Y is a Gaussian $\mathcal{N}(0, 1)$ draw, correlated to Z in (9) with correlation coefficient ρ . For the periods where \hat{x} drops below zero in (9), the process for \hat{Y} comes to a stand-still.

[11] consider several alternative schemes for Y , the most prominent being the “IJK” scheme, defined as

$$(31) \quad \begin{aligned} \ln \hat{Y}(t + \Delta) = & \ln \hat{Y}(t) - \frac{\Delta}{4} (\hat{x}(t + \Delta) + \hat{x}(t)) + \rho \sqrt{\hat{x}(t)} Z \sqrt{\Delta} \\ & + \frac{1}{2} \left(\sqrt{\hat{x}(t + \Delta)} + \sqrt{\hat{x}(t)} \right) \left(Z_Y \sqrt{\Delta} - \rho Z \sqrt{\Delta} \right) + \frac{1}{4} \epsilon \rho \Delta (Z^2 - 1). \end{aligned}$$

Here, $\hat{x}(t + \Delta)$ and $\hat{x}(t)$ are meant to be simulated by the implicit Milstein scheme (11); again the correlation between the Gaussian samples Z_Y and Z is ρ .

4.2.2. *Simplified Broadie-Kaya.* We recall from the discussion earlier that the complicated part of the Broadie-Kaya algorithm was the computation of $\int_t^{t+\Delta} x(u) du$, conditional on $x(t)$ and $x(t + \Delta)$. [3] suggests a naive, but effective, approximation, based on the idea that

$$(32) \quad \int_t^{t+\Delta} x(u) du \approx \Delta [\gamma_1 x(t) + \gamma_2 x(t + \Delta)],$$

for certain constants γ_1 and γ_2 . The constants γ_1 and γ_2 can be found by moment-matching techniques (using results from [8], p. 16), but [3] presents evidence that it will often be sufficient to use either an Euler-like setting ($\gamma_1 = 1, \gamma_2 = 0$) or a central discretization ($\gamma_1 = \gamma_2 = \frac{1}{2}$). In any case, (32) combined with (29) gives rise to a scheme for Y -simulation that can be combined with any basic algorithm that can produce $\hat{x}(t)$ and $\hat{x}(t+\Delta)$. [3] contains numerical results for the case where $\hat{x}(t)$ and $\hat{x}(t+\Delta)$ are simulated by the algorithms in Sections 3.3.2 and 3.3.3; results are excellent, particularly when the QE algorithm in Section 3.3.3 is used to sample x .

4.2.3. Martingale Correction. Finally, let us note that some of the schemes outlined above (including (31) and the one in Section 4.2.2)) will generally not lead to martingale-behavior of \hat{Y} ; that is, $E(\hat{Y}(t+\Delta)) \neq E(\hat{Y}(t))$. For the cases where the error $e = E(\hat{Y}(t+\Delta)) - E(\hat{Y}(t))$ is analytically computable, it is, however, straightforward to remove the bias by simply adding $-e$ to the sample value for $\hat{Y}(t+\Delta)$. [3] gives several examples of this idea.

5. FURTHER READING

In this article, we restricted ourselves to the presentation of relatively simple methods which in the two-dimensional Heston model setting only require two variates per time step. Such schemes are often the most convenient in actual trading systems and for implementations that rely on Wiener processes built from low discrepancy numbers. More complicated high-order Taylor schemes, which often require extra variates, are described in [13]. The efficacy of such methods are, however, unproven in the specific setting of the Heston model.

In recent work, [1] constructs second-order scheme for the CIR process, using a switching idea similar to that of the QE scheme. For the Heston process, [1] develops a “second order scheme candidate” involving three variates per time step; the numerical performance of the scheme compares favorably with Euler-type schemes.

6. RELATED EQF ARTICLES

- eqf08-018: Heston model
- eqf11-025: Cox-Ingersoll-Ross (CIR) model
- eqf11-029: Affine models
- eqf13-001: Monte Carlo simulation for stochastic differential equations
- eqf13-002: Scenario simulation for stochastic differential equations
- eqf13-023: Stochastic Taylor expansions

REFERENCES

- [1] Alfonsi, A. (2008), “A second-order discretization scheme for the CIR process: application to the Heston model,” Working Paper, Institut für Mathematik, TU Berlin.
- [2] Andersen, L. and R. Brotherton-Ratcliffe (2005), “Extended Libor Market Models with Stochastic Volatility”, *Journal of Computational Finance*, 9 (1), pp. 1-40.
- [3] Andersen, L. (2008), “Simple and Efficient Simulation of the Heston stochastic Volatility Model”, *Journal of Computational Finance*, 11 (3), pp. 1-42.
- [4] Broadie, M. and Ö. Kaya (2006), “Exact simulation of stochastic volatility and other affine jump diffusion processes”, *Operations Research*, vol. 54, no. 2.
- [5] Cheng, R. and G. Feast (1980), “Gamma Variate Generators with Increased Shape Parameter Range”, *Commun. ACM*, 23 (7), pp. 389-394.
- [6] DiDonato, A. R. and A. H. Morris (1987). “Incomplete gamma function ratios and their inverse”, *ACM TOMS*, 13, pp. 318-319.

- [7] Duffie, D., J. Pan and K. Singleton (2000), “Transform analysis and asset pricing for affine jump diffusions”, *Econometrica*, vol. 68, pp. 1343-1376.
- [8] Dufresne, D. (2001), “The integrated square-root process”, Working paper, University of Montreal.
- [9] Glasserman, P. (2003), *Monte Carlo methods in financial engineering*, Springer Verlag, New York.
- [10] Heston, S.L. (1993), “A closed-form solution for options with stochastic volatility with applications to bond and currency options”, *Review of Financial Studies*, vol. 6, no. 2, pp. 327-343.
- [11] Kahl, C. and P. Jäckel (2006), “Fast strong approximation Monte-Carlo schemes for stochastic volatility models”, *Journal of Quantitative Finance*, vol. 6, no. 6, pp. 513-536.
- [12] Karlin, S and H. Taylor. (1981), *A Second Course in Stochastic Processes*, Academic Press.
- [13] Kloeden, P. and E. Platen (1999), *Numerical solution of stochastic differential equations*, 3rd edition, Springer Verlag, New York.
- [14] Lord, R., R. Koekkoek and D. van Dijk (2006), “A Comparison of biased simulation schemes for stochastic volatility models”, Working Paper, Tinbergen Institute.
- [15] Smith, R. (2007), “An almost exact simulation method for the Heston model”, *Journal of Computational Finance*, 11 (1), pp. 115-125.