Alternative Methods for Simulating Square Root Processes

Louis O. Scott

Rose Creek Market Risk Consulting, U.S.A.

Louis Scott
1270 Rose Creek
Greensboro, GA 30642
Email: LOScott@outlook.com

Short Biography
Louis Scott spent 17 years in investment banking, with Morgan Stanley and UBS, and the most recent 10 years with the Federal Reserve Bank of New York. His various roles included quantitative analytics, derivative pricing, market risk models, and market risk management. He also served as an adjunct professor for the Financial Engineering programs at NYU and Fordham. Prior to moving into industry, he was a finance professor at University of Illinois and University of Georgia.

Web Link for C++ and Cuda C code:
`https://github.com/LOScott-Models/SimulateSquareRootProcess`

Alternative Methods for Simulating Square Root Processes

December 2025

Abstract

Square root processes are used in quantitative finance to model state variables that are nonnegative. Applications include models for stochastic volatility, stochastic credit spreads, and strictly nonnegative interest rates. Applications for derivative pricing typically require simulation of stochastic processes, including the square root process. The purpose of this paper is to revisit methods for fast simulation of square root process to be applied with parallel processing. The paper reviews the square root process and a variety of simulation methods, including the simulation of the non-central chi squared distribution, the exact distribution for the process over discrete time intervals. Simulation methods for the non-central chi squared distribution are relatively slow and not well suited for simulation in dynamic financial models. Alternative approximation methods are developed, and a battery of goodness-of-fit statistical tests are applied to the alternative methods. The paper includes discussion of random number generation for the purpose of simulating square root processes across parallel processors. The alternative simulation methods require only one uniform random number generator per time step and facilitate hedging and parameter calibration with parallel processing. The test results indicate that the approximation methods generally converge and significantly reduce computing time when parallel processing is applied on a GPU.

Key Words

Derivative pricing, simulation, diffusion approximation, parallel processing, high performance computing, GPU computing

1 Introduction

Square root processes are used in quantitative finance to model state variables that are nonnegative. Applications include models for stochastic volatility and stochastic credit spreads, as well as the Cox-Ingersoll-Ross (1985) model which uses the square root process to model strictly nonnegative interest rates. This process produces models that are analytically tractable and useful for modeling financial time series. The purpose of this paper is to reexamine methods for simulating square root processes over small time periods in dynamic financial models for applications with parallel processing. The square root process is a diffusion process with the following stochastic differential equation for x:

$$dx = (\kappa\theta - (\kappa + \lambda)x)\, dt + \sigma\sqrt{x}\, dz, \tag{1}$$

where $\kappa > 0$ is the rate of mean reversion for the real world process ($\lambda = 0$), $\theta > 0$ is the long run mean, $\sigma > 0$ is the volatility parameter, and dz is the change in a Brownian motion. The parameter λ is a risk premium that can be positive or negative. The rate of mean reversion under the risk neutral measure is determined by $\kappa+\lambda$. If $\kappa+\lambda \leq 0$, the risk-neutral process for x is not a stationary time series, but one can evaluate expectations for asset pricing using the distribution conditional on a starting value. This process is used in quantitative finance for random variables that are restricted to be nonnegative. Various special cases include Heston's (1993) stochastic volatility model, the Cox-Ingersoll-Ross (CIR, 1985) model for interest rates, and the stochastic volatility LIBOR market models developed in Andersen and Brotherton-Ratcliffe (2005), Wu, and Zhang (2006), and Kiesel and Lutz (2011). Duffie, Pan, and Singleton (2000) developed an exponential affine model for option prices in which an independent jump process is added to the square root process for the stochastic volatility. More recent applications of stochastic volatility in models for foreign exchange rates and earthquakes are in Ahlip, Park, and Prodan (2017), Ascione, Mehrdoust, Orlando, and Samimi (2023), Orlando and Bufalo (2022).

Dynamic financial models include stochastic processes for asset prices, interest rates, and key state variables. The following stochastic differential is an example of a diffusion model applied in finance:

$$\mathbf{dy} = \boldsymbol{\mu}(\mathbf{y}, t)\, dt + \boldsymbol{\sigma}(\mathbf{y}, t)\, \mathbf{dz},$$

where $\mathbf{y}$ is a vector of length k containing state variables which include asset prices and interest rates, $\mathbf{dy}$ is the stochastic differential for $\mathbf{y}$, $\boldsymbol{\mu}(\mathbf{y}, t)$ is a vector of length k, $\boldsymbol{\sigma}(\mathbf{y}, t)$ is a k by m matrix, and $\mathbf{dz}$ is the change for a vector of m independent Brownian motion processes, each with mean zero and variance dt. $\boldsymbol{\mu}(\mathbf{y}, t)$, and $\boldsymbol{\sigma}(\mathbf{y}, t)$ are functions of $\mathbf{y}$, t, and fixed parameters. One or more of the state variables in $\mathbf{y}$ can be a square root processes. Simulations of the state variables typically use daily time steps and extend for months or many years, depending on the application and the time horizon. In derivative pricing models, the stochastic processes are approximated over small discrete time intervals, and Monte Carlo simulations are used to estimate expected values for valuation, as well as partial derivatives which are the deltas used for hedging. Simulation can also be used to calibrate the initial state variables and model parameters to fit model valuations to market prices. Computing technology has improved dramatically in recent years so that financial models requiring Monte Carlo simulation can run very quickly with parallel processing across multiple threads on CPUs, GPUs, or special purpose processing chips. In the case of Monte Carlo simulation for derivative pricing, fast methods are necessary to produce valuations and hedges in real time as markets change. Parallel processing introduces additional requirements for the generation of random numbers, as most random number generators (RNG) are sequential. The pseudo RNGs have initial seeds or states that control the sequences of uniform random numbers. For some of

these RNGs, there are quick methods, known as step-ahead techniques, for calculating the initial seeds necessary to generate sequences of uniform random numbers across multiple processors which will run in parallel. This is the case for the pseudo RNG's covered in L'Ecuyer (1999), L'Ecuyer and Simard (2007), and Salmon, Moraes, Dror, and D. E. Shaw (2011). Pseudo RNG's from these references are used for parallel processing in the statistical libraries available in python and Cuda C. The step-ahead techniques facilitate control of the initial seeds and the sequences of uniform random numbers across parallel processors.

If the simulation of a financial model can be performed by using the inverse cumulative distribution function (CDF) with uniform random numbers, one can easily control the sequences of random variates across parallel processors. It is simply a matter of counting the number of time steps and random uniform variates required on each simulation path and applying the appropriate step-ahead techniques. In addition, one can apply the concept of common random numbers covered in Asmussen and Glynn (2007) and L'Ecuyer (2023). For example, deltas for hedging are typically calculated by valuing the derivative and repeating the valuation using a small change in an initial state variable. If the same sequence of uniform random variates is used to estimate the valuation with and without the small change, the simulation error for the derivative can be significantly reduced. The conditions for the variance reduction are covered in L'Ecuyer (2023, 6.4) and are typically satisfied in derivative pricing models if common random numbers are used. Parameter calibration has typically been a challenge for models that must be solved by simulation. In the past, parameter calibration has been performed with approximation models, as Monte Carlo methods were too slow. This is no longer the case as massive parallel processing with GPUs facilitates fast simulation so that real time pricing and parameter calibration with simulation models can be performed. The set of parameters that optimizes an objective function for the fit of model prices to market prices can change if the simulated random numbers are not repeated. For this reason, it is important to run an optimization algorithm with common random numbers. Otherwise, the optimization algorithm will be chasing optimal parameter sets that vary across simulations. This is the "scrambling" effect mentioned in Andersen, Jäckel, and Kahl (2010). If one uses the same starting seeds for a uniform RNG, then the same random numbers can be reproduced as initial state variables and model parameters are adjusted. The contribution of this paper is the development of new approximation methods for simulating the square root process that satisfy the properties of common random numbers. Bounds on the approximation errors are not developed, but empirical tests of the alternative methods are presented. The proposed approximation methods are useful in a variety of applications in quantitative finance, including applications across parallel processors.

Section 2 reviews the simulation methods based on the non central chi squared distribution, which is the probability distribution for the the square root process over discrete time intervals. Simulation methods for this distribution are relatively slow and are not well suited for parameter calibration and parallel processing. As discussed in section 2, non central chi squared simulation is typically based on acceptance/rejection methods so that it is difficult to predict the number of uniform random numbers for each time step and control the number of uniform random numbers on each simulation path. Section 3 covers alternative approximation methods for simulating square root processes and develops extensions of existing methods. The alternative simulation methods require only one uniform random number generation per time step and facilitate parallel process and parameter calibration. Section 4 provides tests of the alternative simulation methods, including applications with parallel processing on a GPU. The test results show that the alternative methods approximate the distribution for the square root process and significantly reduce computing time when parallel processing is applied.

2 Simulating Square Root Processes

One method for simulating x in equation (1) is to use the exact distribution and simulate a scaled non central chi squared variate over discrete time intervals. The variable cx_t, conditional on x_0, is distributed as a non central chi squared with ν degrees of freedom and non centrality parameter λ_{nc} defined as follows:

$$\nu = \frac{4\kappa\theta}{\sigma^2}, \quad c = \frac{4(\kappa+\lambda)}{\sigma^2(1-e^{-(\kappa+\lambda)t})}, \quad \text{and} \quad \lambda_{nc} = cx_0e^{-(\kappa+\lambda)t}. \tag{2}$$

There are several methods available for simulating non central chi squared variates. The distribution is a mixture of the chi squared distribution with the degrees of freedom generated randomly by a Poisson distribution, so that one method is to simulate a Poisson variate with parameter $\frac{1}{2}\lambda_{nc}$ followed by a simulation of a chi squared with degrees of freedom equal to ν plus 2 times the Poisson random variate. Another method covered in Johnson & Kotz (1995, Ch. 29) and Glasserman (2003, Ch. 3), is to simulate a normal variate and a chi squared variate as follows, with ν degrees of freedom and non centrality parameter λ_{nc}:

$$(\tilde{Z} + \sqrt{\lambda_{nc}})^2 + \tilde{X},$$

where $\tilde{Z}$ is a standard normal random variate and $\tilde{X}$ is a chi squared random variare with $\nu - 1$ degrees of freedom. This latter method requires $\nu \geq 1$. The first method using a Poisson random variate and a chi squared random variate can be used for any $\nu > 0$. Poisson and normal random number generation can be calculated very quickly, but chi squared random number generation is more complex and requires either acceptance-rejection methods or an inversion of the gamma CDF. These methods are exact, but require significantly more computing time, relative to the random number generation of random variables with normal or exponential distributions. The number of random number generations varies at each time step with acceptance-rejection methods so that one cannot predict or control the number of random number generations required for each simulation path. In addition, this method does not produce a sequence of common random numbers. The exact simulation applied to the square root process is appropriate for any chosen time interval, but it does not take advantage of the property that a diffusion process is approximately normal over small time intervals. To perform parameter calibration, testing, real time pricing, and hedging with the simulation of square root processes, it is necessary to have a fast method which includes control over the initial seeds and the number of uniform random number generations.

3 Alternative Approximation Methods

Stochastic differential equations are approximated by simulating over small discrete time intervals. Financial models for equity prices and FX rates typically use simulations over daily time intervals. The papers by Kahl and Jäckel (2006), Andersen (2008), and Andersen, Jäckel, and Kahl (2010) develop several alternative methods for the simulation of square root processes. Two of their methods are particularly useful for simulating square root processes: Andersen's (2008) quadratic-exponential method (QE) and the implicit Milstein scheme in equation (11) of Andersen, Jäckel, and Kahl (2010), which was originally developed in Kahl and Jäckel, (2006). The QE method uses either a quadratic function of a normal random variate or a random variate based on a mixture of an exponential distribution and a distribution concentrated at zero. The mixture of the exponential random variate and a zero is used to capture the behavior of the non-central chi squared distribution for values close to zero when the degrees of freedom parameter is small. The QE method is a moment matching

method in which two parameters are set to match the mean and variance of the diffusion over a small time interval. The implicit Milstein scheme is useful when the degrees of freedom parameter is greater than or equal to one. It should be noted that the implicit Milstein scheme is similar, but not identical to Andersen's QE method.

An alternative approach is to transform the diffusion equation and use a solution to a differential equation that has a constant variance for the diffusion term. For the square root process, one can apply the following transformation to x: $y = \sqrt{x}$. An application of Ito's lemma produces the following stochastic differential equation for y:

$$dy = \tfrac{1}{2}\left(\frac{\kappa\theta - \frac{1}{4}\sigma^2}{y} - (\kappa+\lambda)y\right)dt + \tfrac{1}{2}\sigma dz_t\,.$$

The last term, $\frac{1}{2}\sigma dz_t$ is a normal random variable, and an exact solution for y must satisfy this stochastic differential equation. If $4\kappa\theta = \sigma^2$, the differential equation can be solved as follows:

$$y_t = e^{-\frac{1}{2}(\kappa+\lambda)t}y_0 + \tfrac{1}{2}\sigma\int_0^t e^{-\frac{1}{2}(\kappa+\lambda)(t-s)}dz_s\,,$$

and the integral is a stochastic integral. The solution for $4\kappa\theta \neq \sigma^2$, is more complex and does not lead to a tractable solution for simulating y or ultimately x. An approximate solution can be found by working with the solution to the following ODE:

$$dy = \tfrac{1}{2}\left(\frac{\kappa\theta - \frac{1}{4}\sigma^2}{y} - (\kappa+\lambda)y\right)dt\,,$$

which is

$$y_t = \sqrt{e^{-(\kappa+\lambda)t}y_0^2 + \left(\kappa\theta - \frac{\sigma^2}{4}\right)\left(\frac{1-e^{-(\kappa+\lambda)t}}{\kappa+\lambda}\right)}\,.$$

This leads to the following proposed approximation:

$$y_t = \sqrt{e^{-(\kappa+\lambda)t}y_0^2 + \left(\kappa\theta - \frac{\sigma^2}{4}\right)\left(\frac{1-e^{-(\kappa+\lambda)t}}{\kappa+\lambda}\right)} + \frac{\sigma}{2}\int_0^t e^{-\frac{1}{2}(\kappa+\lambda)(t-s)}dz_s\,.$$

For $y(0) = \sqrt{x_0}$ and $x_t = y^2(t)$, the solution can be expressed in terms of x as follows:

$$x_t = \left(\sqrt{e^{-(\kappa+\lambda)t}x_0 + \left(\kappa\theta - \frac{\sigma^2}{4}\right)\left(\frac{1-e^{-(\kappa+\lambda)t}}{\kappa+\lambda}\right)} + \frac{\sigma}{2}\int_0^t e^{-\frac{1}{2}(\kappa+\lambda)(t-s)}dz_s\right)^2\,. \tag{3}$$

It should be noted that this simulation method is a quadratic function of a normal simulation, which is similar to the implicit Milstein scheme and the quadratic part of the QE method described above. The integral of the diffusion process is simulated by simulating a standard normal random variate and multiplying by the square root of the variance,

$$\mathrm{Var}\left(\int_0^t e^{-\frac{1}{2}(\kappa+\lambda)(t-s)}dz_s\right) = \frac{1-e^{-(\kappa+\lambda)t}}{\kappa+\lambda}\,.$$

The simulation method in (3) requires that the term inside the square root be nonnegative and this is guaranteed if $4\kappa\theta \geq \sigma^2$, or equivalently $\nu \geq 1$.

The mean and variance of $\Delta x \equiv x_t - x_0$ for the square root process, for a given time interval t, are as follows:

$$E(\Delta x) = \left(e^{-(\kappa+\lambda)t} - 1\right) x_0 + \kappa\theta \left(\frac{1 - e^{-(\kappa+\lambda)t}}{\kappa + \lambda}\right) \tag{4}$$

$$\mathrm{Var}(\Delta x) = \sigma^2 \left(\frac{1 - e^{-(\kappa+\lambda)t}}{\kappa + \lambda}\right) \left(e^{-(\kappa+\lambda)t} x_0 + \tfrac{1}{2}\kappa\theta \left(\frac{1 - e^{-(\kappa+\lambda)t}}{\kappa + \lambda}\right)\right). \tag{5}$$

These two moments follow from the non-central chi squared distribution for x_t, conditional on x_0. The evaluation of the mean and variance for the approximation model follows from an analysis of the following terms:

$$\begin{aligned} x_t = {} & e^{-(\kappa+\lambda)t} x_0 + \kappa\theta \left(\frac{1 - e^{-(\kappa+\lambda)t}}{\kappa + \lambda}\right) \\ & + \sqrt{e^{-(\kappa+\lambda)t} x_0 + \left(\kappa\theta - \frac{\sigma^2}{4}\right) \left(\frac{1 - e^{-(\kappa+\lambda)t}}{\kappa + \lambda}\right)} \left(\sigma \int_0^t e^{-\frac{1}{2}(\kappa+\lambda)(t-s)} dz_s\right) \\ & - \frac{\sigma^2}{4} \left(\frac{1 - e^{-(\kappa+\lambda)t}}{\kappa + \lambda}\right) + \left(\frac{\sigma}{2} \int_0^t e^{-\frac{1}{2}(\kappa+\lambda)(t-s)} dz_s\right)^2. \end{aligned}$$

The approximation model for the simulation of Δx has a mean that matches the mean for the square root process. The variance for Δx in the approximation model is

$$\begin{aligned} \mathrm{Var}(\Delta x) = {} & E\left(\sqrt{e^{-(\kappa+\lambda)t} x_0 + \left(\kappa\theta - \frac{\sigma^2}{4}\right) \left(\frac{1 - e^{-(\kappa+\lambda)t}}{\kappa + \lambda}\right)} \int_0^t \sigma e^{-\frac{1}{2}(\kappa+\lambda)(t-s)} dz_s \right. \\ & \left. - \frac{\sigma^2}{4} \left(\frac{1 - e^{-(\kappa+\lambda)t}}{\kappa + \lambda}\right) + \left(\frac{\sigma}{2} \int_0^t e^{-\frac{1}{2}(\kappa+\lambda)(t-s)} dz_s\right)^2\right)^2 \\ = {} & \sigma^2 \left(\frac{1 - e^{-(\kappa+\lambda)t}}{\kappa + \lambda}\right) \left(e^{-(\kappa+\lambda)t} x_0 + \left(\kappa\theta - \frac{\sigma^2}{4}\right) \frac{1 - e^{-(\kappa+\lambda)t}}{\kappa + \lambda}\right) \\ & + 2\left(\frac{\sigma^2}{4} \left(\frac{1 - e^{-(\kappa+\lambda)t}}{\kappa + \lambda}\right)\right)^2, \end{aligned}$$

and $\mathrm{Var}(\Delta x) = \sigma^2 x_0 t + o(t)$. The variance for the approximation model in (3) does not match the variance for the non-central chi squared distribution, but it does converge to the variance for the square root process over small time intervals as t approaches 0. Hence, this approximation might be unreliable when the time steps, dt, are large. Variance is an important characteristic for a distribution, and it is desirable to have approximation methods that match both the mean and variance over small time steps. The variance bias is easy to compute, and the approximation model can be modified to match the variance, by applying the following additional calculations: subtract the mean from x_t, rescale this residual so that the variance matches $\mathrm{Var}(\Delta x)$, and add back the mean. The rescaling factor is calculated as follows using the the exact variance in (5):

$$\frac{\sqrt{e^{-(\kappa+\lambda)t} x_0 + \frac{1}{2}\kappa\theta \left(\frac{1 - \exp(-(\kappa+\lambda)t)}{\kappa+\lambda}\right)}}{\sqrt{e^{-(\kappa+\lambda)t} x_0 + \left(\kappa\theta - \frac{\sigma^2}{8}\right) \left(\frac{1 - \exp(-(\kappa+\lambda)t)}{\kappa+\lambda}\right)}}.$$

If $4\kappa\theta \geq \sigma^2$, the scaling factor is less than 1 and the resulting calculation for x_t is guaranteed to be nonnegative. For the case $4\kappa\theta = \sigma^2$, the approximation model in (3) is an exact simulation of the non-central chi squared distribution, and the rescaling factor is 1. If $4\kappa\theta < \sigma^2$, the term inside the square root of the denominator can become negative for small values of x_0. In addition, if the square root term is nonnegative, the rescaling to match the variance can produce a negative value for x. In these cases, an alternative simulation method is required. For cases where $4\kappa\theta < \sigma^2$, one can apply Andersen's QE method, but this methodology tends to produce many simulations at exactly 0. In cases where $2\kappa\theta < \sigma^2$, the square root process can touch zero which serves as a reflecting barrier (See Feller (1950)). In these cases, the CDF is 0 for $x = 0$, and the probability density function approaches infinity as x approaches 0. Hence a simulation method for theses case should produce values close to, but not exactly, zero.

The remainder of this section is devoted to the development of two alternative methodologies when $4\kappa\theta < \sigma^2$. Both methods use the quadratic part of Andersen's (2008) QE method, which is calculated as follows. Let x be x_t conditional on x_0, and calculate the mean and variance for cx, as follows with c as defined in (2):

$$M = c\left(e^{-(\kappa+\lambda)t-1}x_0 + \kappa\theta\left(\frac{1-e^{-(\kappa+\lambda)t}}{\kappa+\lambda}\right)\right)$$

$$S^2 = c^2\sigma^2\left(\frac{1-e^{-(\kappa+\lambda)t}}{\kappa+\lambda}\right)\left(e^{-(\kappa+\lambda)t}x_0 + \tfrac{1}{2}\kappa\theta\frac{1-e^{-(\kappa+\lambda t)}}{\kappa+\lambda}\right).$$

Define ψ, b, and a as follows:

$$\psi = \frac{S^2}{M^2}$$

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

$$a = \frac{M}{1+b^2} \quad .$$

If $\psi \leq 2$, $b^2 \geq 0$, and the following quadratic function of a normal random variate is used to simulate x: $cx = a(b+\tilde{Z})^2$. If this condition is not satisfied, the first alternative methodology replaces the mixture of zero and the exponential distribution used in Andersen's QE method with a mixture of a beta distribution and a non-central chi squared distribution with 1 degree of freedom. Define $F_{\chi^2}(y;\nu)$ as the CDF for a χ^2 variate with degrees of freedom ν. Let ν, λ_{nc}, and c be as defined in (2). The distribution for a scalar transformation of the square root process, x, is a non-central chi squared with the following CDF:

$$F_{\chi^{2\prime}}(cx;\nu,\lambda_{nc}) = e^{-\frac{1}{2}\lambda_{nc}}\sum_{j=0}^{\infty}\frac{(\frac{1}{2}\lambda_{nc})^j}{j!}F_{\chi^2}(cx;\nu+2j).$$

With parameters p and β and $\nu^* = \nu/2.0$, the CDF for the mixture of distributions is as follows:

$$F_1(cx) = p\min[cx,1]^{\nu^*/2} + (1-p)F_{\chi^{2\prime}}(cx/\beta,1,\lambda_{nc}). \tag{6}$$

where $\nu^* = \nu/2$ and $F_{\chi^{2\prime}}(cx/\beta,1,\lambda_{nc})$ is the CDF for the non central chi squared distribution with input value cx/β, 1 degree of freedom, and non centrality parameter $\lambda_n c$. The CDF for the beta distribution, $\min[cx,1]^{\nu^*/2}$, mimics a chi squared with degrees of freedom less than 1 as it has the property that the density function goes to infinity as x gets close to zero. The beta distribution is also used in the acceptance/rejection methods for the exact simulation of a chi squared when the

degrees of freedom are small. The non-central chi squared with one degree of freedom is used in the mixture of distributions because it is easy to simulate (one normal simulation) and it provides a good approximation for values of ν below, but near one. Simulation is performed by first generating a $U(0,1)$ random variable and choosing the beta distribution if $U(0,1) \leq p$ or choosing the non-central chi squared distribution if $U(0,1) > p$. A second random variate is simulated and the inverse CDF method is used to generate either the beta random variate or a normal random variate for the non central chi squared variate with 1 degree of freedom. The parameters p and β are set to match the mean and variance for the square root process. See Appendix A for the calculations of p and β. This simulation methodology uses a quadratic function of a normal simulation if $\psi \leq 2$ or a mixture of a beta and a non-central chi squared with one degree of freedom if $\nu < 1$. Hence, the name Quadratic–Beta–Non-Central Chi Squared or QBNC1 is applied. In addition, a second random simulation is not required, as the first simulation can be rescaled using the intervals between 0, p, and 1 to produce the second $U(0,1)$ random number for the inverse CDF method. The treatment for the second simulation is covered in section A of the Appendix. It should be noted that this methodology is not the inverse CDF method, but only one uniform random variate is required for each time step.

The second alternative method uses the quadratic part of Andersen's QE method and replaces the mixture of zero and the exponential distribution with an approximation for the CDF of the non central chi squared. The CDF approximation uses a beta distribution and 2 exponential distributions, hence the name Quadratic–Beta–Double Exponential or QB2Exp is applied. The CDF for $F_{\chi^{2\prime}}(cx;\nu,\lambda_{nc})$ can be used to derive the CDF for $x = x_t$, for the square root process. The PDF for x is

$$f_{\chi^{2\prime}}(x;\nu,\lambda_{nc}) = e^{-\frac{1}{2}\lambda_{nc}} \sum_{j=0}^{\infty} \frac{(\frac{1}{2}\lambda_{nc})^j}{j!} c\,(cx)^{\nu/2+j-1}\, e^{-\frac{1}{2}cx} \frac{2^{\nu/2+j}}{\Gamma(\nu/2+j)}.$$

When $\psi > 2$, the quadratic method cannot be applied, and the non-centrality parameter λ_{nc} is small. In this case, the CDF and the PDF can be approximated with the first term of the summation, $j = 0$. For cx sufficiently small, $\exp(-0.5\,cx)$ is approximately equal to 1, and the following approximation can be used for the CDF:

$$e^{-\frac{1}{2}\lambda_{nc}} \frac{2^{\nu/2}}{\Gamma(\frac{\nu}{2})} (cx)^{\nu/2}.$$

A critical value, x_1, is selected so that $\exp(-0.5\,cx_1)$ is approximately one. x_1 is set so that the second term in the expansion of $\exp(-0.5\,cx)$ is equal to -0.1. A second critical value is selected so that x_2 is equal to x_1 plus the mean for x. Define F_2 as the CDF and f_2 as the PDF for the approximating distribution, with β_1 and β_2 as constants that are to be set to match the mean and variance.

For $x \leq x_1$:

$$F_2(x) = e^{-\frac{1}{2}\lambda_{nc}} \frac{2^{\nu/2}}{\Gamma(\frac{\nu}{2})} (cx)^{\nu/2} = \frac{2^{\nu/2}}{A^*} (cx)^{\nu/2}$$

$$f_2(x) = c\frac{A^*}{2^{\nu/2}} (cx)^{\nu/2-1}$$

with

$$A^* = e^{-\frac{1}{2}\lambda_{nc}} \frac{2^{\nu/2}}{\Gamma(\frac{\nu}{2})}$$

The CDF for $x > x_1$ is approximated with 2 exponential distributions as follows.

For $x_1 < x \leq x_2$:

$$F_2(x) = F_2(x_1) + (1 - F_2(x_1))(1 - e^{-\beta_1(x-x_1)})$$
$$f_2(x) = (1 - F_2(x_1))\beta_1 e^{-\beta_1(x-x_1)}$$

For $x > x_2$:

$$F_2(x) = F_2(x_2) + (1 - F_2(x_2))(1 - e^{-\beta_2(x-x_2)})$$
$$f_2(x) = (1 - F_2(x_2))\beta_2 e^{-\beta_2(x-x_2)}$$

The parameters β_1 and β_2 are set to match the mean and variance of the approximation model to the mean and variance for the square root process. The detailed calculations for the mean of x, the variance of the change in x, β_1, and β_2 are in Appendix B. The solutions for β_1 and β_2 that match the mean and variance require an iterative search on the value for β_1. The acronym QB2Exp is applied to this Beta Double Exponential method. This method is based on a piecewise approximation of the non central chi squared CDF when the moment matching conditions for the quadratic method are not satisfied. The iterative search procedure and the inversion from the approximation CDF's can be calculated very quickly to produce model simulations that approximate the square root process. In summary, the quadratic solution in (3) with rescaling for the variance is recommended for simulating the square roor process if $\nu \geq 1$. If $\nu < 1$, the QBNC1 and the QB2Exp methods are recommended and both methods provide good approximations to the non central chi squared distribution. The next section provides tests of these proposed simulations methods along with several alternatives.

4 Tests of the Simulation Methodologies

A variety of comparisons and tests are presented in this section beginning with plots of the CDFs for the alternative approximation methods with the exact distribution for a non-central chi squared with degrees of freedom equal to 0.08. This is the case covered in Figures 1 and 2 in Andersen (2008), and in Figure 1 in Andersen, Jäckel, and Kahl (2010). Figure 1 here contains plots of the distribution functions for a single time step with the same model parameters set as follows: $\kappa = 0.5$, $\theta = 0.04$, $\sigma = 1.0$, $\lambda = 0$, $\nu = 0.08$, $t = 0.1$, and $x_0 = 0.01$. In this case, an alternative distribution is used to approximate the non-central chi squared because the quadratic approximation cannot be used. The distribution functions for the approximation methods QB2Exp and QE are close to the exact distribution functions, but there are some differences, notably the concentration of the QE distribution function at $x = 0$. The approximation methods improve as the size of the time interval is reduced. As previously noted, derivative pricing models typically use daily time steps so that the size of the time interval is either $t = \frac{1}{365}$ for a calendar day, or $t = \frac{1}{252}$ for a trading day. The graphs in Figure 2 contain plots of the distribution functions for one daily time step with $t = \frac{1}{365}$, and the model parameters used in Figure 1. The distribution functions for both approximation methods are much closer to the exact distribution when the time interval is reduced to one day, with one exception: the case in Panel B in which the QE method produces a high probability for $x = 0$. This is a characteristic of the QE approximation method.

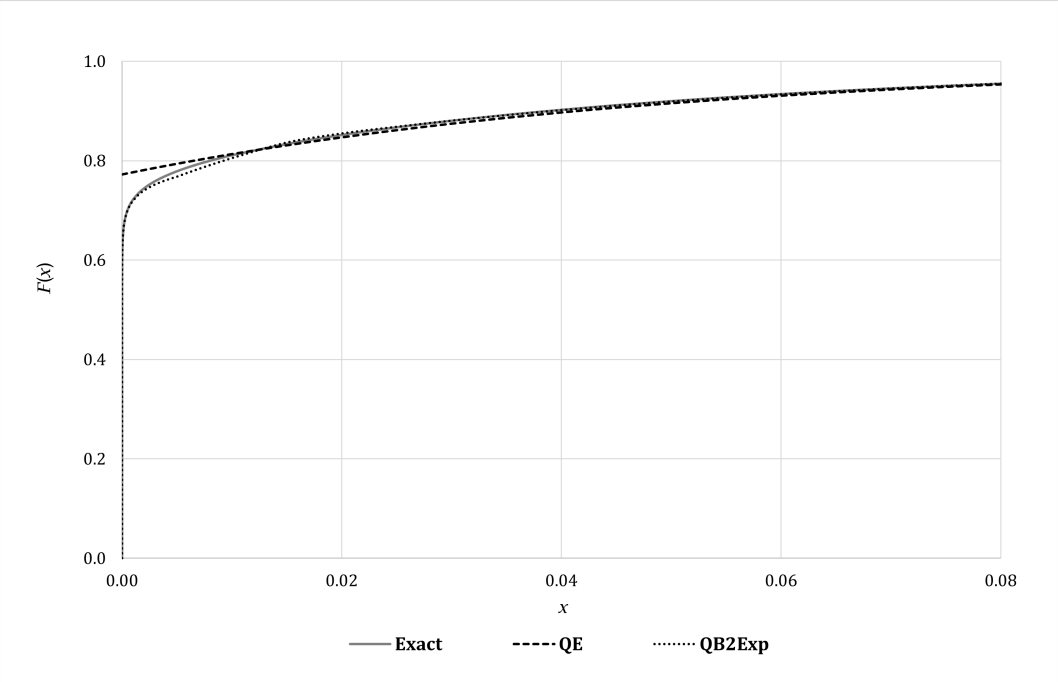
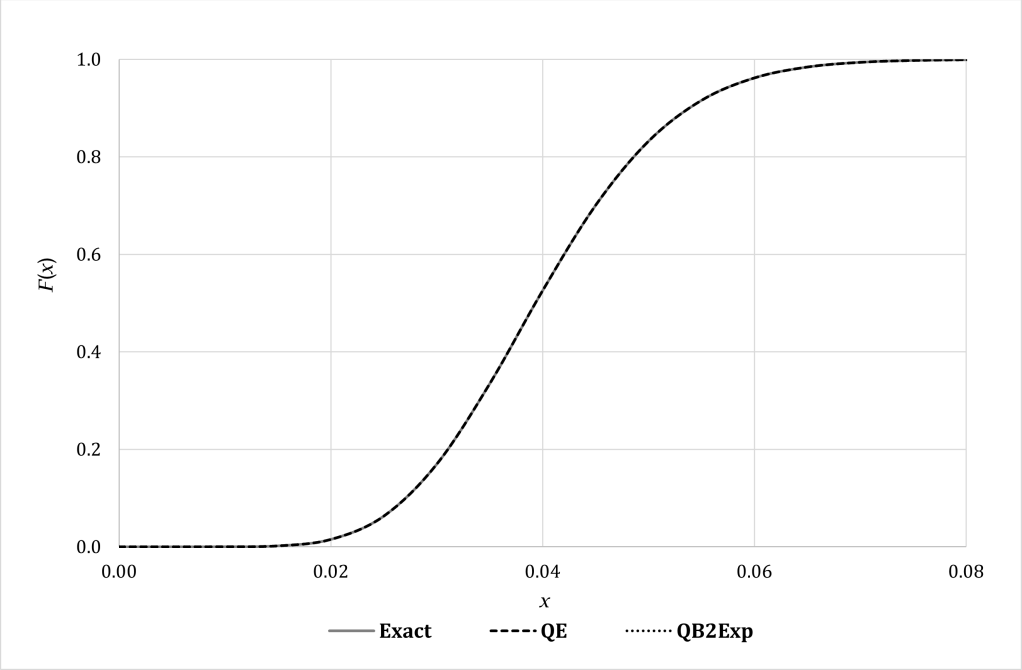


Figure 1: Cumulative Distribution Functions for Approximation Models

with $x_0 = 0.04$



with $x_0 = 0.0001$

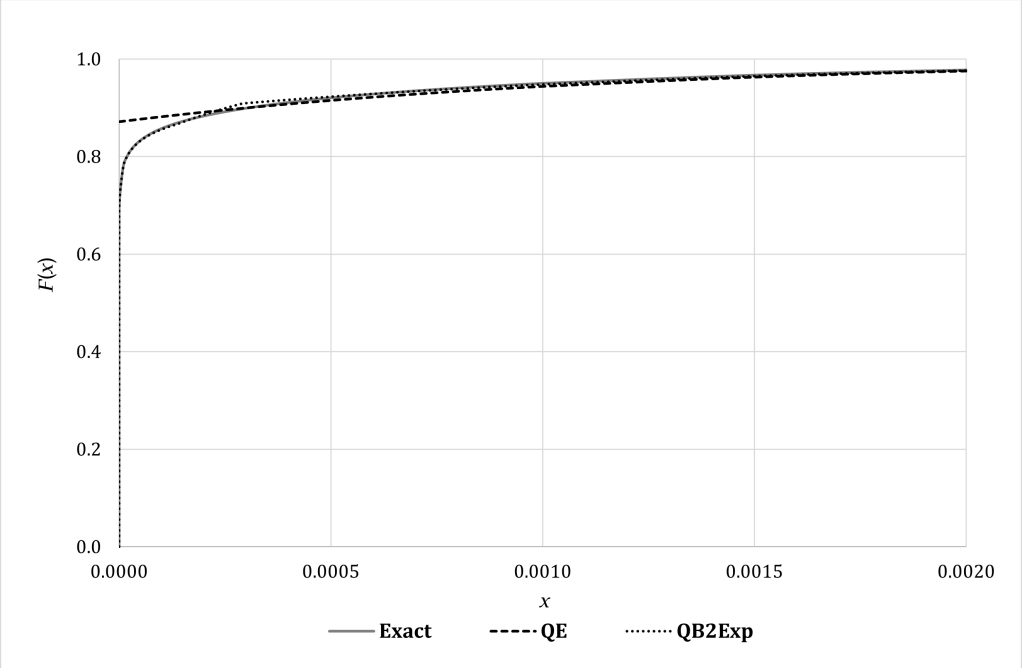


Figure 2: Cumulative Distribution Functions for a Daily Time Interval

4.1 Tests of the Terminal Distribution Over Different Time Horizons

The first set of tests are tests of the simulated distributions generated by the three approximation methods, as approximations for the non-central chi squared distribution. The Euler methodology is included in these tests as a "straw man" for comparison purposes. The Euler method is simple and easy to implement, and in the cases tested, it requires less computing time than the other methods. The following equation is used for the Euler methodology:

$$x_t = x_0 + \left(e^{-(\kappa+\lambda)\Delta t} - 1\right) x_0 + \kappa\theta \left(\frac{1 - e^{-(\kappa+\lambda)\Delta t}}{\kappa + \lambda}\right) + \sigma x_0 \sqrt{x_0} \Delta z$$

where Δz is a normal random variable with mean zero and variance Δt, and x_t is set to 0 if a negative value is simulated. When the degrees of freedom of parameter are less than 2, the Euler method produces negative values. When the degrees of freedom are less than 1, the method produces simulations with significant biases.

The simulation methods are run with the following model parameters for daily time steps: $\kappa = 0.25$, $\theta = 0.04$, $\lambda = -.125$, and $\Delta t = \frac{1}{365} = 0.00273973$, with different values for x_0: 0.01 and 0.04. The values for σ are varied to produce a range of values for the degrees of freedom, ν. The tests are run on simulations of multiple time steps over 91 daily time intervals to test the quality of the resulting terminal distribution. 91 days corresponds to a quarterly time interval. The tests are summarized in Table 1. Table 2 contains a second set of tests run for a much longer time horizon, 3,650 days, or 10 years. In both sets of tests 1,000,000 independent simulation paths are run. The computing times in Table 1 are reported for simulations run separately on a CPU and on a graphical processing unit (GPU) to show the benefits of parallel processing with a GPU. There are two numbers for the GPU compute times, with the first number being the time with double precision calculations on the GPU, and the second being the time with single precision calculations. The initial seeds for each simulation path on the GPU are set using the step-ahead method, so that the GPU simulations calculated with double precision match exactly the CPU simulations that are run sequentially. The single precision calculations on the GPU use the same sequences of uniform random variates, and the calculations are close to, but do not match, the double precision calculations. The computing times with single precission are much smaller. Simulation on a GPU is faster if single precision arithmetic and single precision functions in the Cuda C Math Library are used. The tests in Table 2 are run on a GPU only because the computing times are much longer on a CPU without parallel processing. The same tests have been applied to simulations using the exact methods for simulating non-central chi squared variates described in section 2. The exact simulations, one set using small daily time steps and one using a single time step over the full time period (91 days or 10 years), produce simulations that pass all of the statistical tests in Tables 1 and 2. The results for the exact simulation are not reported in Tables 1 and 2. The computing times for the exact method over daily time steps are roughly 5 to 10 times the computing times for the approximation methods. Simulation of the exact method over a single time step is, however, much faster, but most pricing model applications of the square root process require simulation over the full path of the process.

A few comments regarding the simulation of the models and the calculation of the non-central chi squared distribution are necessary. The QB2Exp, QBNC1, and QE approximation methods and the Euler method have been programmed in C++. A separate function for the simulation of the model in (3) with rescaling to match the variance has been programmed for the case when the degrees of freedom, ν, are greater than or equal to 1. This model is labeled as Eq. (3) in the tables. If $\nu < 1$, the QB2Exp and QBNC1 methods use Andersen's quadratic method that matches the mean and variance with ψ_c set to 2. To improve computing performance, the functions for the different methods have been programmed with upfront calculations for model parameters, where

possible. The non-central chi squared CDF is calculated using the function available in the Boost library. Simulations for the normal, beta, and exponential distributions are based on the inverse CDF method using the pseudo RNG for a uniform distribution, MRG32ka, which is a combined multiple recursive RNG developed by L'Ecuyer (1999). This RNG has a very long period and passes the tests for RNG's (Crush and Big Crush) in L'Ecuyer and Simard (2007). It is used in statistical packages (Python, R, Matlab) as well as the Curand library for parallel processing on Nvidia GPU's. The step ahead methods for the MRG32k3a RNG are covered in L'Ecuyer, Simard, Chen and Kelton (2002) and Bradley, du Toit, Giles, Tong, and Woodhams (2011), and this method is used to set starting seeds for the RNG on each processor when running on a GPU. The step ahead methodology ensures that the seeds and the simulations on a GPU match those that are run sequentially on a CPU. The counter based RNG's developed in Salmon, Moraes, Dror, and Shaw (2011) are also potential candidates for use in parallel processing. These RNG's, also known as Philox, pass the tests for RNG's, have step-ahead methods necessary for parallel processing, and are available in the Curand library and most statistical packages. The model calculations and computing times reported in Tables 1 and 2 have been performed with an AMD Ryzen Threadripper PRO 5955WX CPU, which has a maximum turbo frequency of 4.50 gigahertz, and an Nvidia RTX 4070 GPU. Performance results will vary across different processors.

A variety of standard statistical tests are calculated for the simulated distributions using the simulations of the terminal values for the square root process. The mean and variance of the simulated terminal values are tested with t statistics, which are distributed approximately as standard normal for the large number of simulations. In addition, the Kolmogorov-Smirnov, Cramer-von Mises, and Anderson-Darling statistics are calculated for the simulated distributions. These statistics, including their large sample distributions, are covered in the following references: Anderson and Darling (1952, 1954), Csorgo and Faraway (1996), and Simard and L'Ecuyer (2011). These 3 statistics are designed to test whether an empirical distribution has been drawn from a specific distribution. The approximation methods are approximations for the non-central chi squared distribution, so that there is an expectation that these tests are likely to show rejection that the simulated distributions are non-central chi squared. The tests serve as measures for the goodness of fit for the different simulation methods; cases where the test statistics are less than the corresponding critical values indicate that the simulated distribution is close to the non-central chi squared distribution. It should be noted that the Anderson-Darling test uses weights based on the underlying distribution, under the null hypothesis, and places more weight on the tails of the distribution. The Anderson-Darling statistic is particularly sensitive to simulations in the tails, including those close to zero; a simulation of a zero value produces an infinite value for the Anderson-Darling statistic. For this reason, there are cases in which the QE method and the Euler method produce infinite Anderson-Darling statistics.

The t tests for means and variances are two sided tests so that the critical values for 1 million simulations are ± 1.96 for the 95% level, ± 2.58 for the 99% level, and ± 3.29 for the 99.9% level. The t test for the mean is standard. The t test for the variance is calculated by dividing the difference between the simulated variance and the variance for the non-central chi squared by the square root of the variance of the simulated variance. The variance of the simulated variance is calculated by using a sample (simulated) estimate of the 4th moment.

$$\frac{E(x_t - \text{Mean})^4 - \text{Var}^2}{N}$$

where Mean and Var represent the analytic mean and variance for the non-central chi squared, and N is 1 million for the number of simulations. This variance uses a calculation of the 4th moment around the mean for the simulated values. The critical values for the Kolmogorov-Smirnov statistic

have been calculated by running the program provided by Simard and L'Ecuyer (2011). The critical values for this statistic, with a sample size of 1 million, are as follows: 0.001358 for $p = 0.95$, 0.001627 for $p = 0.99$, and 0.001949 for $p = 0.999$. The critical values for the Cramer-von Mises tests are from Table 1, p. 229, in Csorgo and Faraway (1996). For a sample size of 1 million, the critical values of the Cramer-von Mises statistic are as follows: 0.4612 for $p = 0.95$, 0.7430 for $p = 0.99$, and 1.1616 for $p = 0.999$. The distribution of the Anderson-Darling statistic depends on the underlying distribution, under null hypothesis. To establish critical values for the Anderson-Darling statistic, the exact non-central chi squared distribution has been simulated to produce repeated simulations of this test statistic. The critical values for this test statistic do vary with the degrees of freedom parameter. Model simulations indicate that the critical values for the cases included in the tests below are generally 2.3 to 2.5 for the 95th percentile and 3.7 to 4.2 for the 99th percentile.

The statistical tests in Table 1 are organized in panels to show the results across different values for the degrees of freedom, ν, and the initial value x_0. As noted, the degrees of freedom are controlled by adjusting the variance parameter, σ. Panels A to D contain the results for Equation (3) with the variance rescaled, the QE method, and the Euler method when ν is either 4.0 or 1.0. Equation (3) and the QE approximation perform well in these cases, as all statistical tests are passed, except for the Anderson-Darling test using the QE method when $\nu = 1$. In these cases, the QE method simulates terminal values that are exactly zero. If the switch threshold in the QE method is adjusted from $\psi_c = 1.5$ to $\psi_c = 2.0$, the QE method will always use the quadratic function of a normal simulation when $\nu \geq 1$, so that zeros are not simulated. The Euler method simulates terminal values of zero, and it fails the t test for the simulated variance when $\nu = 1$ and $x_0 = 0.01$, which is the smaller value for x_0. It should be noted that the Euler method performs well when $\nu > 2$, and it runs faster than the other methods. In summary, the quadratic approximation methods perform well when $\nu \geq 1$. The results for the statistical tests start to deteriorate as ν is decreased below 1 in panels E to J. When ν is either 0.25 or 0.1111, the Euler method produces significant biases for the mean and variance, and it fails all statistical tests. For $\nu < 1$, the QB2Exp, QBNC1, and QE methods pass the tests for the mean and variance, but these approximation methods fail on the other statistical tests. The goodness of fit statistics for all three approximation methods become worse as the ν parameter decreases. The goodness of fit is worst for the QE method and the Euler method. The QB2Exp approximation method provides a better fit than the QBNC1 method in most cases. The statistical tests have been run with the exact simulation method described in section 2 and simulation based on the inverse CDF method for the non central chi squared distribution. The exact methods pass all of the statistical tests, but are much slower if simulated over small time steps. The QB2Exp requires more computing time with double precision on a CPU or a GPU, but less time with single precision on a GPU. All methods have been optimized for faster performance with single precision arithmetic on a GPU by using single precision functions in the Cuda C Math Library.

Table 2 contains results for simulations over a much longer time horizon, which is set at 3,650 days or 10 years. Panels A to D in this table contain simulation tests for $\nu = 2.0$ and $\nu = 0.64$, with values of x_0 set to 0.04 or 0.01. The QB2Exp, QBNC1, QE, and Euler methods all perform well in the tests when $\nu = 2.0$, with the exception that the Euler method does produce some zero terminal values which result in an infinite value for the Anderson-Darling test statistics. When $\nu = 0.64$, the test statistics indicate failures similar to those observed in Table 1. Table 2 includes the run times on a GPU for both double and single precision. These tests were not run on a CPU as the computing times would have been much longer. The computing times for single precision in Table 2 are less than 2 seconds for the QB2Exp, QBNC1 and QE methods and show that one can run these simulation methods with daily time steps for large numbers of simulations, 1 million, over a long-time horizon, 10 years, with a relatively quick response time.

The statistical tests for the simulated distributions indicate that the approximation methods

Table 1: Tests of the Simulation Methods

with parameters: $\kappa = 0.25$, $\theta = 0.04$, $\lambda = -.125$

Panel A: $x_0 = 0.04$, $\sigma = 0.1$, $\nu = 4.0$				
	Eq. (3)	QE	Euler	Analytic
Mean	0.041236	0.041236	0.041236	0.041227
t test (Mean)	0.09	0.09	0.09	
Std. Deviation	0.0099068	0.0099068	0.0099068	0.0099090
t test (Variance)	-0.29	-0.29	-0.16	
Kolmogorov-Smirnov	0.0009275	0.0009275	0.0010556	
Cramer-von Mises	0.1158	0.1158	0.1880	
Anderson-Darling	0.8297	0.8297	1.1171	
CPU Run Time (sec.)	2.864	3.327	2.776	
GPU Run Time (sec.)	0.051 / 0.005	0.060 / 0.012	0.040 / 0.003	
Panel B: $x_0 = 0.01$, $\sigma = 0.1$, $\nu = 4.0$				
	Eq. (3)	QE	Euler	Analytic
Mean	0.012152	0.012152	0.012152	0.012148
t test (Mean)	0.06	0.06	0.06	
Std. Deviation	0.0051767	0.0051767	0.0051749	0.0051775
t test (Variance)	-0.19	-0.19	-0.61	
Kolmogorov-Smirnov	0.0008953	0.0008956	0.0011123	
Cramer-von Mises	0.1107	0.1108	0.2874	
Anderson-Darling	0.7762	0.7762	1.6528	
CPU Run Time (sec.)	2.856	3.324	2.776	
GPU Run Time (sec.)	0.051 / 0.005	0.062 / 0.011	0.043 / 0.003	
Panel C: $x_0 = 0.04$, $\sigma = 0.2$, $\nu = 1.0$				
	Eq. (3)	QE	Euler	Analytic
Mean	0.041244	0.041244	0.041244	0.041227
t test (Mean)	0.12	0.12	0.12	
Std. Deviation	0.019814	0.019814	0.019816	0.019818
t test (Variance)	-0.22	-0.22	-0.10	
Kolmogorov-Smirnov	0.0009495	0.0009495	0.0012406	
Cramer-von Mises	0.1170	0.1170	0.3251	
Anderson-Darling	0.8436	∞	∞	
CPU Run Time (sec.)	2.858	3.326	2.782	
GPU Run Time (sec.)	0.051 / 0.005	0.062 / 0.012	0.043 / 0.003	
Panel D: $x_0 = 0.01$, $\sigma = 0.2$, $\nu = 1.0$				
	Eq. (3)	QE	Euler	Analytic
Mean	0.012155	0.012155	0.012162	0.012148
t test (Mean)	0.07	0.07	0.14	
Std. Deviation	0.010353	0.010353	0.010343	0.010355
t test (Variance)	-0.15	-0.15	-1.08	
Kolmogorov-Smirnov	0.0009491	0.003415^{***}	0.005012^{***}	
Cramer-von Mises	0.1150	0.1400	0.8718^{**}	
Anderson-Darling	0.8409	∞	∞	
CPU Run Time (sec.)	2.853	3.323	2.776	
GPU Run Time (sec.)	0.051 / 0.005	0.065 / 0.014	0.040 / 0.003	

Table 1: Tests of the Simulation Methods, continued

Panel E: $x_0 = 0.04$, $\sigma = 0.25$, $\nu = 0.64$					
	QB2Exp	QBNC1	QE	Euler	Analytic
Mean	0.041247	0.041247	0.041247	0.041248	0.041227
t test (Mean)	0.13	0.13	0.13	0.13	
Std. Deviation	0.024768	0.024768	0.024768	0.024770	0.024772
t test (Variance)	-0.21	-0.21	-0.21	-0.13	
Kolmogorov-Smirnov	0.0009531	0.0009531	0.0009531	0.0012770	
Cramer-von Mises	0.1185	0.1185	0.1186	0.4402	
Anderson-Darling	0.8704	0.8666	∞	∞	
CPU Run Time (sec.)	3.512	3.375	3.338	2.828	
GPU Run Time (sec.)	0.068 / 0.005	0.064 / 0.009	0.061 / 0.012	0.039 / 0.004	
Panel F: $x_0 = 0.01$, $\sigma = 0.25$, $\nu = 0.64$					
	QB2Exp	QBNC1	QE	Euler	Analytic
Mean	0.012156	0.012156	0.012156	0.012201	0.012148
t test (Mean)	0.07	0.07	0.07	0.46	
Std. Deviation	0.012941	0.012940	0.012941	0.012904	0.012944
t test (Variance)	-0.18	-0.17	-0.19	-2.51^{**}	
Kolmogorov-Smirnov	0.001945^{**}	0.002366^{***}	0.03070^{***}	0.02909^{***}	
Cramer-von Mises	0.2357	0.2663	11.063^{***}	40.486^{***}	
Anderson-Darling	6.818^{**}	12.109^{**}	∞	∞	
CPU Run Time (sec.)	4.024	3.444	3.327	2.825	
GPU Run Time (sec.)	0.314 / 0.013	0.091 / 0.023	0.067 / 0.022	0.039 / 0.004	
Panel G: $x_0 = 0.04$, $\sigma = 0.4$, $\nu = 0.25$					
	QB2Exp	QBNC1	QE	Euler	Analytic
Mean	0.041255	0.041255	0.041255	0.041336	0.041227
t test (Mean)	0.14	0.14	0.14	0.55	
Std. Deviation	0.039628	0.039628	0.039629	0.039559	0.039636
t test (Variance)	-0.18	-0.18	-0.17	-1.76^{*}	
Kolmogorov-Smirnov	0.002295^{***}	0.009704^{***}	0.05942^{***}	0.04254^{***}	
Cramer-von Mises	0.3224	2.3451^{***}	70.656^{***}	53.361^{***}	
Anderson-Darling	6.785^{**}	258.14^{**}	∞	∞	
CPU Run Time (sec.)	3.798	3.433	3.301	2.826	
GPU Run Time (sec.)	0.218 / 0.010	0.093 / 0.019	0.063 / 0.019	0.040 / 0.003	
Panel H: $x_0 = 0.01$, $\sigma = 0.4$, $\nu = 0.25$					
	QB2Exp	QBNC1	QE	Euler	Analytic
Mean	0.012162	0.012154	0.012162	0.012732	0.012148
t test (Mean)	0.10	0.04	0.10	4.06^{***}	
Std. Deviation	0.020695	0.020696	0.020695	0.020572	0.020710
t test (Variance)	-0.41	-0.40	-0.41	-3.83^{***}	
Kolmogorov-Smirnov	0.007771^{***}	0.04484^{***}	0.2639^{***}	0.1985^{***}	
Cramer-von Mises	9.525^{***}	208.31^{***}	6174.3^{***}	5887.0^{***}	
Anderson-Darling	73.15^{**}	5659.9^{**}	∞	∞	
CPU Run Time (sec.)	6.020	3.889	2.979	2.827	
GPU Run Time (sec.)	0.698 / 0.027	0.139 / 0.040	0.077 / 0.034	0.040 / 0.004	

Table 1: Tests of the Simulation Methods, continued

Panel I: $x_0 = 0.04$, $\sigma = 0.6$, $\nu = 0.1111$					
	QB2Exp	QBNC1	QE	Euler	Analytic
Mean	0.041265	0.041262	0.041264	0.042089	0.041227
t test (Mean)	0.15	0.14	0.15	3.53^{***}	
Std. Deviation	0.059429	0.059427	0.059429	0.059058	0.059454
t test (Variance)	-0.29	-0.31	-0.29	-4.58^{***}	
Kolmogorov-Smirnov	0.004956^{***}	0.06094^{***}	0.2862^{***}	0.2054^{***}	
Cramer-von Mises	3.560^{***}	488.2^{***}	7818.3^{***}	4461.5^{***}	
Anderson-Darling	25.78^{**}	10428.0^{**}	∞	∞	
CPU Run Time (sec.)	4.543	3.687	3.011	2.827	
GPU Run Time (sec.)	0.438 / 0.018	0.136 / 0.030	0.071 / 0.028	0.040 / 0.003	
Panel J: $x_0 = 0.01$, $\sigma = 0.6$, $\nu = 0.1111$					
	QB2Exp	QBNC1	QE	Euler	Analytic
Mean	0.012156	0.012156	0.012156	0.014469	0.012148
t test (Mean)	0.05	0.05	0.05	13.17^{***}	
Std. Deviation	0.031006	0.031006	0.031005	0.031566	0.031065
t test (Variance)	-0.77	-0.77	-0.78	-6.53^{***}	
Kolmogorov-Smirnov	0.008191^{***}	0.1192^{***}	0.5537^{***}	0.4076^{***}	
Cramer-von Mises	17.79^{***}	3625.1^{***}	56628.9^{***}	35017.0^{***}	
Anderson-Darling	72.65^{**}	42261.9^{**}	∞	∞	
CPU Run Time (sec.)	6.668	4.338	2.287	2.825	
GPU Run Time (sec.)	0.767 / 0.030	0.147 / 0.043	0.078 / 0.037	0.040 / 0.004	

Notes: significance levels are indicated with asterisks: * for the 95% level, ** for the 99% level, and *** for the 99.9% level.

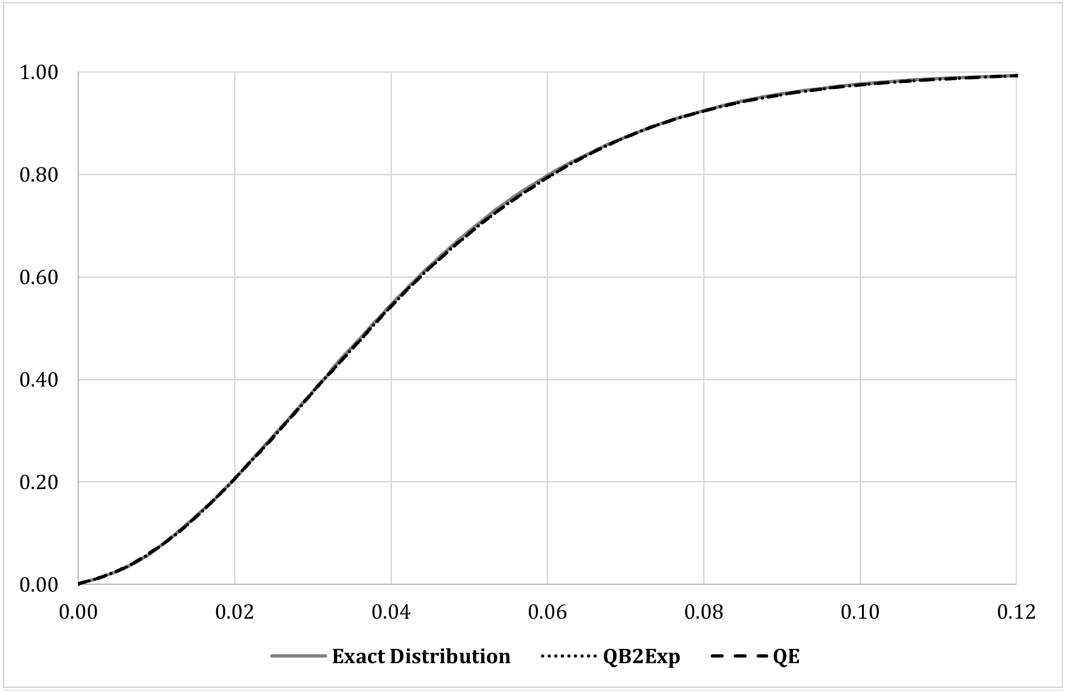
do not approximate the non-central chi squared distribution well when ν is significantly less than 1. The approximation methods are generally accurate for most of the distribution, but there are significant approximation errors in the lower tail when $\nu < 1$. This observation is shown in the plots of the simulated distributions in Figures 3-6 for the cases when ν is either 0.64 or 0.25. The graphs include plots of the simulated distributions for x_t, versus the exact non-central chi squared using the simulations of the terminal values for 91 days, in which t approximately equal to 0.25. Figures 3 and 5 includes plots of the simulated distribution with the non-central chi squared distribution function. The plots contain the cumulative distribution function on the y axis with the terminal x value on the x axis. One can also examine the empirical distributions by calculating the simulated values of the distribution function, $F[x_t]$, using the non-central chi squared distribution evaluated with the simulations of x_t for the different approximation methods. The simulated distribution functions should plot along a 45-degree line when sorted in Figures 4 and 6. Additional graphs are included to show the behavior in the lower tails. The approximation models generally provide good fits across most of the distribution, as the simulated distributions are indistinguishable from the non-central chi squared distribution when the entire range of the distribution is plotted. There are, however, significant approximation errors in the lower tail. The graphs of the lower tail reveal the simulation of zero values with the QE method. There are approximation errors with the QB2Exp method in the lower tail, but it provides a much better fit than the QE method.

Table 2: Tests of the Simulation Methods

with parameters: $\kappa = 0.25$, $\theta = 0.04$, $\lambda = -.125$

Panel A: $x_0 = 0.04$, $\sigma = 0.14142136$, $\nu = 2.0$				
	Eq. (3)	QE	Euler	Analytic
Mean	0.068583	0.068583	0.068591	0.068540
t test (Mean)	0.17	0.17	0.20	
Std. Deviation	0.067660	0.067660	0.067668	0.067575
t test (Variance)	0.92	0.92	1.00	
Kolmogorov-Smirnov	0.0007577	0.0007598	0.0007849	
Cramer-von Mises	0.0753	0.0754	0.0877	
Anderson-Darling	0.4989	0.4983	∞	
GPU Run Time (sec.)	1.755 / 0.115	2.197 / 0.425	1.513 / 0.114	
Panel B: $x_0 = 0.01$, $\sigma = 0.14142136$, $\nu = 2.0$				
	Eq. (3)	QE	Euler	Analytic
Mean	0.059978	0.059978	0.059991	0.059945
t test (Mean)	0.14	0.14	0.19	
Std. Deviation	0.059955	0.059955	0.059963	0.059876
t test (Variance)	0.93	0.93	1.02	
Kolmogorov-Smirnov	0.0006372	0.0006383	0.0008228	
Cramer-von Mises	0.0589	0.0589	0.0851	
Anderson-Darling	0.4326	0.4322	∞	
GPU Run Time (sec.)	1.750 / 0.114	2.191 / 0.425	1.508 / 0.115	
Panel C: $x_0 = 0.04$, $\sigma = 0.25$, $\nu = 0.64$				
	QB2Exp	QBNC1	QE	Euler	Analytic
Mean	0.068593	0.068596	0.068594	0.070743	0.068540
t test (Mean)	0.15	0.16	0.16	6.37^{***}	
Std. Deviation	0.119757	0.119771	0.119760	0.120551	0.119457
t test (Variance)	1.14	1.19	1.15	4.16^{***}	
Kolmogorov-Smirnov	0.002230^{***}	0.002981^{***}	0.03740^{***}	0.03949^{***}	
Cramer-von Mises	0.2762	0.3417	19.800^{***}	389.07^{***}	
Anderson-Darling	8.689^{**}	18.158^{**}	∞	∞	
GPU Run Time (sec.)	25.68 / 0.889	5.048 / 1.530	3.021 / 1.309	1.530 / 0.114	
Panel D: $x_0 = 0.01$, $\sigma = 0.25$, $\nu = 0.64$				
	QB2Exp	QBNC1	QE	Euler	Analytic
Mean	0.059991	0.059991	0.059992	0.062574	0.059945
t test (Mean)	0.14	0.14	0.14	8.08^{***}	
Std. Deviation	0.106126	0.106119	0.106132	0.107660	0.105847
t test (Variance)	1.15	1.12	1.18	7.55^{***}	
Kolmogorov-Smirnov	0.002338^{***}	0.003030^{***}	0.03920^{***}	0.04147^{***}	
Cramer-von Mises	0.3292	0.3772	22.7842^{***}	474.59^{***}	
Anderson-Darling	9.800^{**}	19.622^{**}	∞	∞	
GPU Run Time (sec.)	29.32 / 1.032	5.398 / 1.678	3.115 / 1.419	1.529 / 0.115	

Notes: significance levels are indicated with asterisks: * for the 95% level, ** for the 99% level, and *** for the 99.9% level.



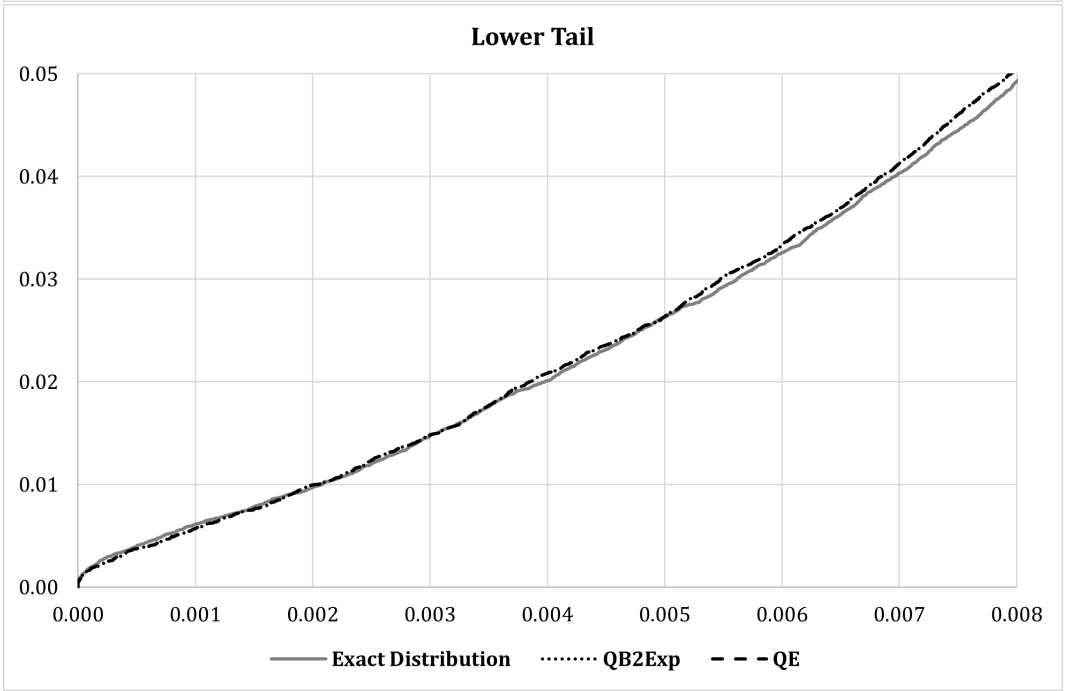


Figure 3: Simulated Distributions vs. Non Central Chi Squared

Terminal distributions after 91 days, t approximately 0.25, with parameters: $\kappa = 0.25$, $\theta = 0.04$, $\lambda = -.125$, $x_0 = 0.04$, and $\sigma = 0.25$ so that $\nu = 0.64$

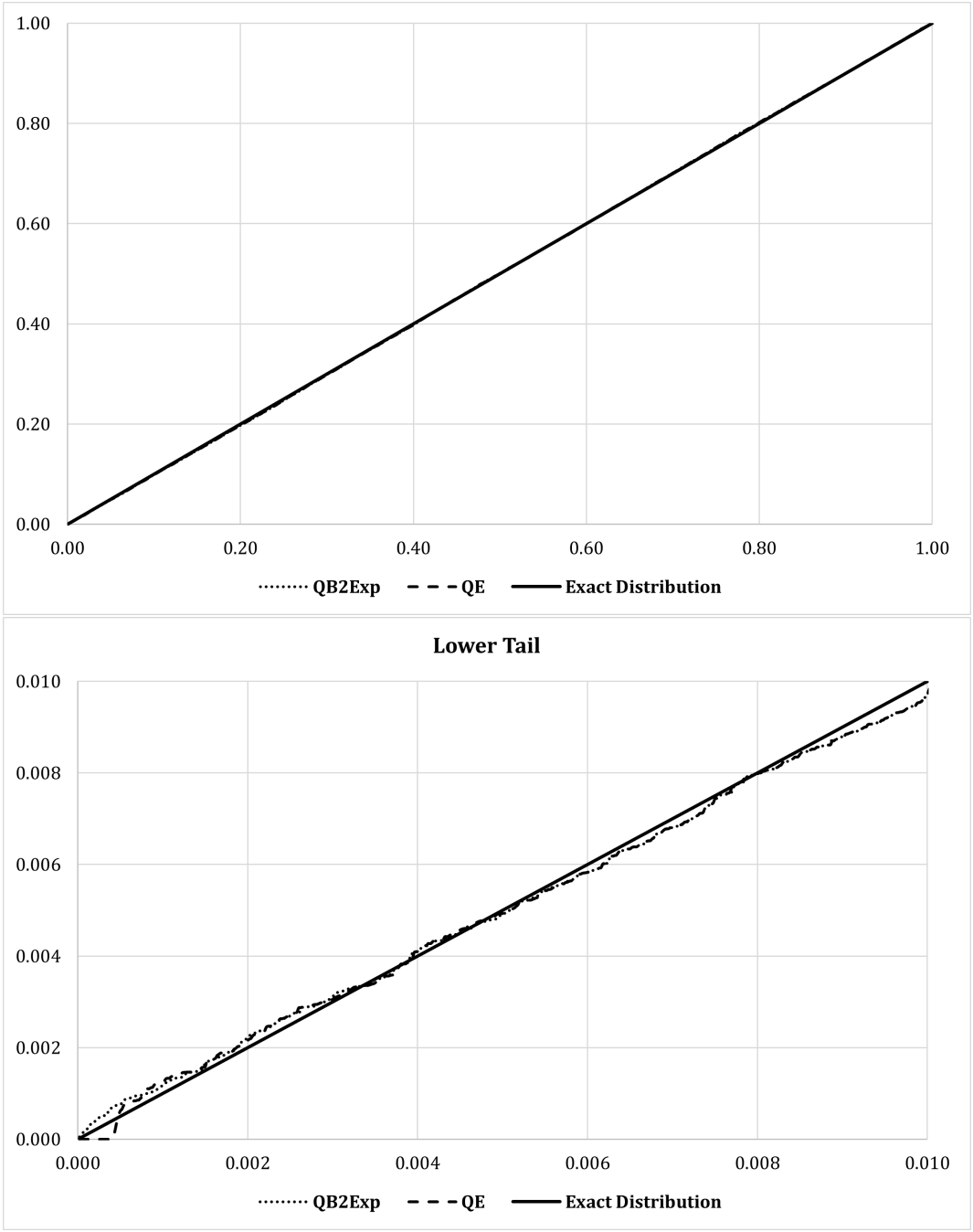
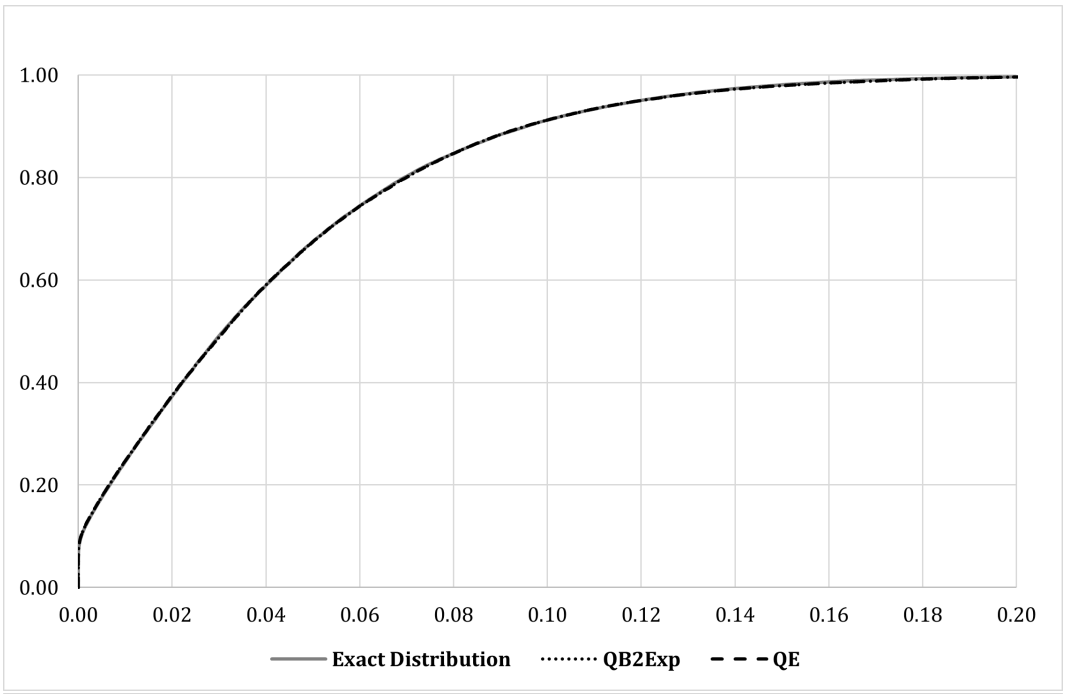


Figure 4: Simulated Cumulative Distribution Functions vs. Non Central Chi Squared

Terminal distributions after 91 days, t approximately 0.25, with parameters: $\kappa = 0.25$, $\theta = 0.04$, $\lambda = -.125$, $x_0 = 0.04$, and $\sigma = 0.25$ so that $\nu = 0.64$



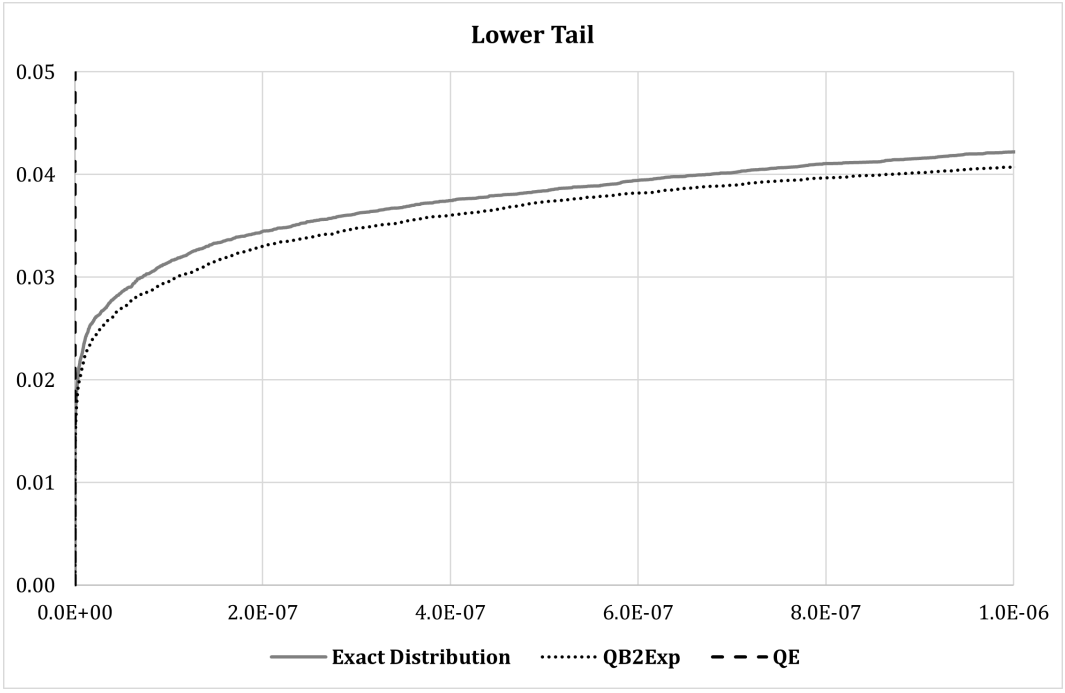


Figure 5: Simulated Distributions vs. Non Central Chi Squared

Terminal distributions after 91 days, t approximately 0.25, with parameters: $\kappa = 0.25$, $\theta = 0.04$, $\lambda = -.125$, $x_0 = 0.04$, and $\sigma = 0.4$ so that $\nu = 0.25$

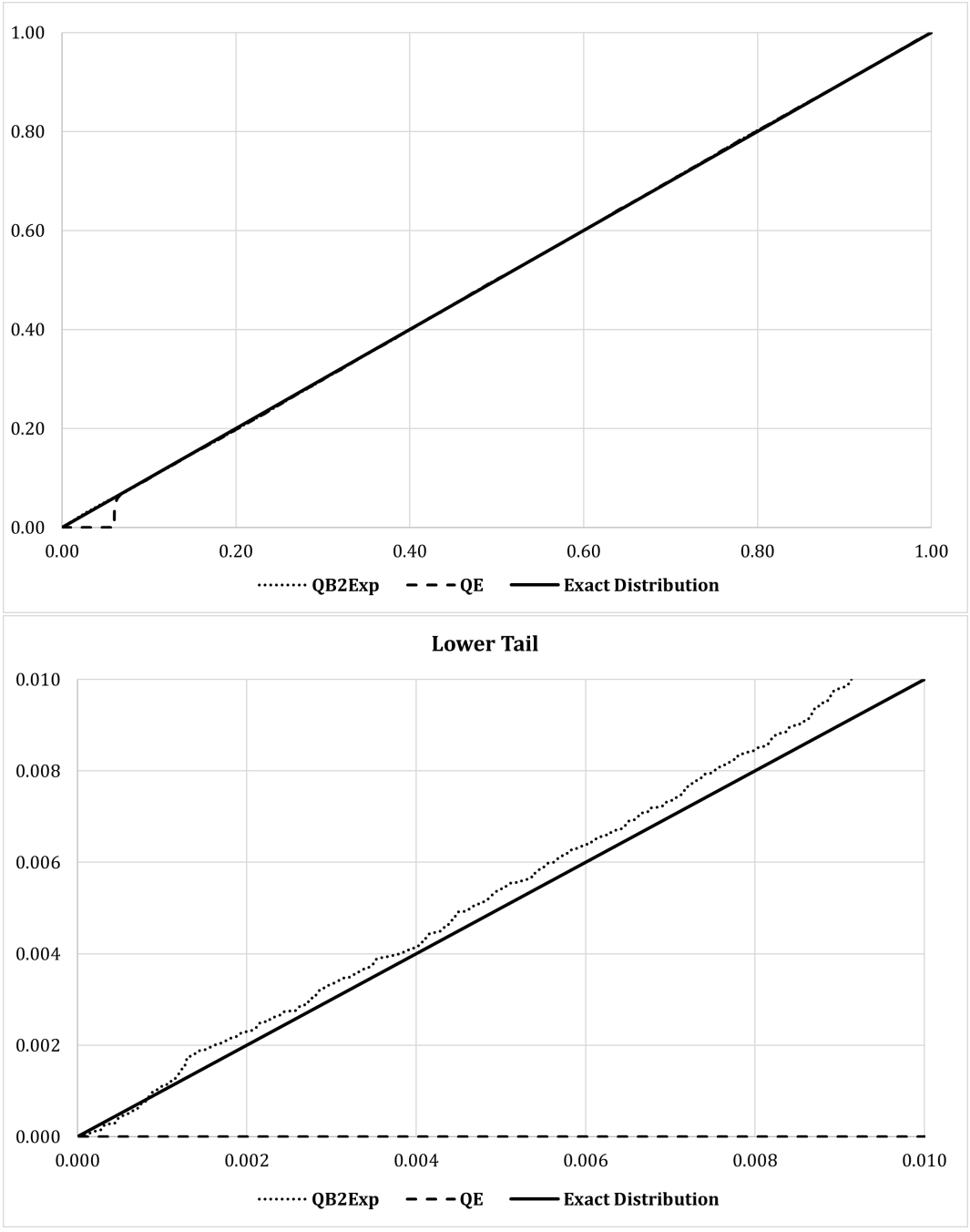


Figure 6: Simulated Cumulative Distribution Functions vs. Non Central Chi Squared

Terminal distributions after 91 days, t approximately 0.25, with parameters: $\kappa = 0.25$, $\theta = 0.04$, $\lambda = -.125$, $x_0 = 0.04$, and $\sigma = 0.4$ so that $\nu = 0.25$

4.2 Tests Based on Valuation

The last tests of the approximation methods are based on simulations for two financial valuation problems which have analyic solutions: the Heston stochastic volatility model and the Cox-Ingersoll-Ross model for the bond price. The Heston model is a stochastic volatility model with the variance process modeled as a square root diffusion:

$$\begin{aligned} d\log S &= \left(r - \tfrac{1}{2}v\right) dt + \sqrt{v}\, dz_1 \\ dv &= (\kappa\theta - (\kappa+\lambda)v)\, dt + \sigma\sqrt{v}\, dz_2 \,. \end{aligned}$$

The model includes a correlation between the Brownian motions: $E(dz_1 dz_2) = \rho\, dt$. The model is simulated by simulating a square root diffusion for v and using the following stochastic integral for the stock price to capture the correlation:

$$\log S_t = \log S_0 + \int_0^t \left(r - \tfrac{1}{2}v_s\right) ds + \int_0^t \sqrt{v_s}\left(\sqrt{1-\rho^2}\, dz_{1s} + \rho\, dz_{2s}\right).$$

The following three methods are used to simulate v_t: QB2Exp, QE, and Euler. Each simulation for v_t is decomposed as follows:

$$v_t = e^{-(\kappa+\lambda)t}v_0 + \kappa\theta\left(\frac{1-e^{-(\kappa+\lambda)t}}{\kappa+\lambda}\right) + \sigma\int_0^t \sqrt{v_s}\, dz_{2s}$$

and

$$\int_0^t \sqrt{v_s}\, dz_{2s} = \frac{1}{\sigma}\left(v_t - e^{-(\kappa+\lambda)t}v_0 - \kappa\theta\left(\frac{1-e^{-(\kappa+\lambda)t}}{\kappa+\lambda}\right)\right).$$

At each time step, the log of the stock price is simulated as follows:

$$\log S_t = \log S_0 + \left(r - \tfrac{1}{4}(v_0+v_t)\right)t + \rho\int_0^t \sqrt{v_s}\, dz_{2s} + \tilde{Z}\sqrt{(1-\rho^2)v_0 t}\ ,$$

where $\tilde{Z}$ is a standard normal random variate. For testing, the rate of mean of reversion, κ, is set at 2.0 and both x_0 and the long run average, θ, are set at 0.04.The instantaneous standard deviation is $\sqrt{0.04} = 0.20$, or 20 percent, which is in the range of a typical volatility for a equity index. The risk premium, λ, is set at -0.25, which is a risk adjustment that reduces the rate of mean reversion and increases the long run average under the risk neutral distribution. Correlations between stock price changes and volatility changes are typically negative and large. The correlation can be estimated by calculating the sample correlation coefficient between changes in the log of the S&P 500 index and changes in the square of the volatility index, VIX, which is a daily times series calculated by the Chicago Board of Options Exchange. This volatility index is used to measure the implied volatility for the S&P 500. The sample correlation for the daily time series is -0.7090 for the period 1990 to 2021. Option calibrations for this correlation can be higher, and ρ is set to -0.90 for the tests. The volatility parameter, σ, controls the volatility of the variance rate and is adjusted to produce different levels for the degrees of freedom. The tests of the convergence of simulated option prices for the Heston model are summarized in Table 3.

The Heston model is used to simulate option prices for strikes from 70 to 125, with the initial stock price at 100. The parameters for the square root process are included in the Table 3. The interest rate has been set at 3.0% per annum. Again, 1 million simulation paths are run for a

time horizon of 91 days (3 months), and the time steps per day are set at 2. Initial runs with only 1 time step per day resulted in some biased simulations for all methods including the exact simulation of the non-central chi squared. The table includes one case with the degrees of freedom, ν, set to 2.0, and all methods, including the Euler method perform well in that the option prices are converging to the analytic solution for the Heston model. The analytic solutions for the option prices in the Heston model are computed separately by using the Fourier inversion method to calculate the relevant probability functions. Table 3 includes 3 additional cases for the degrees of freedom, ν = 0.8889, 0.5, and 0.08. The tests in the table show that both the QB2Exp and the QE methods perform well and generally converge as the degrees of freedom are decreased to 0.08. The same tests have been run with the QBNC1 method; the results are similar to those for the QB2Exp and QE methods, but are not included in the table. In sharp contrast, the Euler method has significant biases and does not converge for many of the option strikes when the degrees of freedom are less than one. Additional tests have been performed by increasing the number of time steps per day and the number of simulation paths to test the convergence of the simulation methods. When the degrees of freedom are less than 1, the QB2Exp and QE methods are converging to the analytic solutions, and the Euler method produces significant biases.

The last set of tests include simulations of an integral of a square root process to estimate the discount function in the Cox-Ingersoll-Ross (CIR) model. The integrals are approximated using daily time steps over 1-year and 5-year time periods. The simulated discount functions should converge to the analytic solution in the CIR model. The results are summarized in Table 4 for two sets of parameter values, one with degrees of freedom at 2.0 and one with degrees of freedom at 0.50. The tests are performed with 1,000,000 simulation paths and the QB2Exp and QE methods show convergence to the corresponding analytic solutions. The Euler method produces simulations that converge when the degrees of freedom are 2.0, but the simulations with degrees of freedom at 0.50 do not converge. The tests in Tables 3 and 4 show that the QB2Exp and QE approximation methods can perform well as approximations for security and derivative valuations, despite the approximation errors highlighted above for the simulated distributions.

Finally, it should be noted the QB2Exp, QBNC1, and QE methods use a single uniform random variate to generate the square root process at each time step, so that simulated payoff functions can be represented as functions of a vector of uniform random varaites. Hence, one can apply the Randomized Quasi Monte Carlo (RQMC) methods developed in Giles and Waterhouse (2009) and L'Ecuyer and Munger (2012) for the valuation examples in Tables 3 and 4, as well as valuations based on more complex payoff functions. The RQMC methods developed in these two papers can be used to signficiantly reduce the simulation error and compute time relative to the MC simulation.

5 Summary

This paper has developed alternative methods to approximate the square root process for simulation in dynamic financial models and other applications that require simulation over many time steps. The tests in section 4.1 examine the terminal distributions generated by the different approximation methods. If $\nu \geq 1$, the quadratic methods, equation (3) with rescaling of the variance and QE, and the Euler method provide good approximations for the terminal distributions. The Euler method requires less computing time, but the quadratic method in equation (3) is close. There are cases where the QE method and the Euler method produce zero values. If $\nu < 1$, the performance of the approximation methods is mixed as the approximations for the terminal distributions deteriorate as ν moves toward zero. The QE method and the Euler method produce too many zero values for the terminal distribution, and all of the methods have approximation errors in the lower tail of the

Table 3: Simulation Tests for Heston Model, with European Option Prices

$\kappa = 2.0, x_0 = \theta = 0.04, \lambda = -0.50, \rho = -0.90, r = 0.03$, using 1,000,000 simulation paths over 91 days with 2 time steps per day

Panel A: $\sigma = 0.4$, $\nu = 2.0$								
Type	Strike	Heston	QB2Exp	t test	QE	t test	Euler	t test
Put	70	0.0301	0.0298	-0.43	0.0298	-0.44	0.0297	-0.62
Put	75	0.0828	0.0827	-0.06	0.0827	-0.06	0.0826	-0.18
Put	80	0.2073	0.2076	0.19	0.2076	0.19	0.2077	0.22
Put	85	0.4761	0.4771	0.42	0.4771	0.42	0.4775	0.63
Put	90	1.0090	1.0106	0.48	1.0106	0.48	1.0121	0.94
Put	95	1.9823	1.9857	0.71	1.9857	0.71	1.9887	1.35
Put	100	3.6206	3.6249	0.66	3.6249	0.66	3.6294	1.37
Cal	100	4.3658	4.3607	-1.00	4.3607	-1.00	4.3645	-0.25
Cal	105	1.9369	1.9334	-1.07	1.9334	-1.07	1.9373	0.14
Call	110	0.5451	0.5440	-0.68	0.5440	-0.68	0.5463	0.79
Call	115	0.0660	0.0667	1.56	0.0667	1.57	0.0671	2.41
Call	120	0.00288	0.00289	0.07	0.00289	0.08	0.00292	0.46
Call	125	0.00007	0.00007	0.10	0.00007	0.10	0.00007	0.18
Panel B: $\sigma = 0.6$, $\nu = 0.8889$								
Type	Strike	Heston	QB2Exp	t test	QE	t test	Euler	t test
Put	70	0.0639	0.0637	-0.31	0.0637	-0.31	0.0635	-0.56
Put	75	0.1429	0.1432	0.22	0.1432	0.22	0.1430	0.11
Put	80	0.2998	0.3003	0.27	0.3003	0.27	0.3005	0.38
Put	85	0.5945	0.5960	0.52	0.5960	0.52	0.5969	0.84
Put	90	1.1206	1.1227	0.54	1.1227	0.54	1.1253	1.21
Put	95	2.0170	2.0210	0.78	2.0210	0.77	2.0261	1.74
Put	100	3.4827	3.4873	0.68	3.4873	0.68	3.4964	2.02
Cal	100	4.2278	4.2232	-1.08	4.2232	-1.08	4.1917	-8.39
Cal	105	1.5922	1.5897	-0.99	1.5897	-0.99	1.5672	-10.11
Cal	110	0.2490	0.2491	0.10	0.2491	0.11	0.2445	-5.04
Cal	115	0.0138	0.0140	1.10	0.0140	1.09	0.0140	0.79
Cal	120	0.00057	0.00053	-1.12	0.00053	-1.12	0.00053	-0.91
Cal	125	0.00002	0.00002	-0.15	0.00002	-0.15	0.00002	0.09
Panel C: $\sigma = 0.8$, $\nu = 0.50$								
Type	Strike	Heston	QB2Exp	t test	QE	t test	Euler	t test
Put	70	0.1046	0.1046	0.04	0.1046	0.04	0.1045	-0.10
Put	75	0.2049	0.2054	0.33	0.2054	0.33	0.2057	0.45
Put	80	0.3830	0.3840	0.42	0.3840	0.42	0.3851	0.91
Put	85	0.6871	0.6891	0.62	0.6891	0.63	0.6923	1.62
Put	90	1.1895	1.1923	0.64	1.1923	0.65	1.2002	2.48
Put	95	1.9983	2.0027	0.77	2.0027	0.77	2.0207	3.97
Put	100	3.2883	3.2934	0.72	3.2936	0.74	3.3369	6.80
Cal	100	4.0335	4.0294	-1.10	4.0292	-1.13	3.8965	-37.98
Cal	105	1.2265	1.2259	-0.30	1.2258	-0.35	1.1616	-33.59
Cal	110	0.1130	0.1136	0.96	0.1135	0.88	0.1103	-4.38
Cal	115	0.0069	0.0072	1.55	0.0072	1.56	0.0071	0.99
Cal	120	0.00046	0.00050	0.94	0.00050	0.93	0.00051	1.04
Cal	125	0.00003	0.00005	1.03	0.00005	1.03	0.00005	1.10

Table 3: Simulation Tests for Heston Model, continued

Panel D: $\sigma = 2.0$, $\nu = 0.08$								
Type	Strike	Heston	QB2Exp	t test	QE	t test	Euler	t test
Put	70	0.2907	0.2909	0.10	0.2911	0.16	0.3462	20.14
Put	75	0.4139	0.4144	0.16	0.4145	0.20	0.5041	26.39
Put	80	0.5784	0.5791	0.19	0.5790	0.16	0.7222	34.34
Put	85	0.7966	0.7976	0.22	0.7973	0.16	1.0244	44.82
Put	90	1.0890	1.0905	0.27	1.0900	0.19	1.4519	59.31
Put	95	1.4989	1.5013	0.36	1.5006	0.26	2.0899	80.62
Put	100	2.1743	2.1778	0.47	2.1772	0.38	3.2082	117.91
Cal	l 100	2.9194	2.9183	-0.47	2.9186	-0.34	2.4764	-202.14
Cal	l 105	0.2656	0.2677	1.84	0.2678	1.98	0.3334	53.34
Call	110	0.0386	0.0388	0.36	0.0387	0.26	0.0470	15.29
Call	115	0.0087	0.0089	0.76	0.0088	0.56	0.0099	4.51
Call	120	0.00225	0.00231	0.44	0.00229	0.26	0.00250	1.70
Call	125	0.00063	0.00068	0.62	0.00067	0.48	0.00076	1.49
Call	130	0.00019	0.00054	0.65	0.00053	0.54	0.00061	1.52

Table 4: Simulation Tests for the Discount Function in the CIR Model

$\kappa = 0.5, \theta = 0.04, \lambda = -0.25, x_0 = 0.04$, using 1,000,000 simulation paths with 1 time steps per day

Panel A: $\sigma = 0.20$, $\nu = 2.0$							
Years	Analytic	QB2Exp	t test	QE	t test	Euler	t test
1 Year	0.95659608	0.95660239	0.30	0.95660239	0.30	0.95660233	0.30
5 Years	0.76412413	0.76427294	1.15	0.76427299	1.15	0.76424651	0.95
Panel B: $\sigma = 0.40$, $\nu = 0.50$							
Years	Analytic	QB2Exp	t test	QE	t test	Euler	t test
1 Year	0.95724993	0.95725349	0.06	0.95725294	0.07	0.95688211	-12.44
5 Years	0.79221829	0.79232113	0.52	0.79233675	0.60	0.78150273	-72.61

distribution. For $\nu < 1$, the QB2Exp method produces the best approximations. Equation (3) with rescaling of the variance, QB2Exp, and QE perform well in the tests of valuation models in section 4.2, but the Euler method performs poorly for $\nu < 1$. Equation (3) is recommended for $\nu \geq 1$, and the QB2Exp method is recommended for $\nu < 1$. Because these two approximation methods require the generation of only one uniform random variate at each time step and satisfy the properties of common random numbers, they are useful for a variety of applications in quantitative finance running on parallel processors.

References

Ahlip, R., L. A. F. Park, and A. Prodan. March 2017. Pricing currency options in the Heston/CIR double exponential jump-diffusion model. *International Journal of Financial Engineering*. 04(01).

Andersen, L.B.G. 2008. Simple and efficient simulation of the Heston stochastic volatility model. *The Journal of Computational Finance* 11 (3): 1–42.

Andersen, L.B.G., and Brotherton-Ratcliffe, R. 2005. Extended Libor market models with stochastic volatility. *The Journal of Computational Finance* 9 (1): 1-40.

Andersen, L., P. Jäckel, C. Kahl. 2010. Simulation of square-root processes. In *Encyclopedia of Quantitative Finance*. `https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470061602.eqf13009`.

Anderson, T.W., and D. A. Darling. 1952. Asymptotic theory of certain "goodness of fit" criteria based on stochastic processes, *Annals of Mathematical Statistics* 23: 193–212.

Anderson, T.W., and D. A. Darling. 1954. A test of goodness-of-fit. *Journal of the American Statistical Association* 49: 765–769.

Ascione, G. , F. Mehrdoust, G. Orlando, and O.Samimi. JUne 2023. Foreign exchange options on Heston-CIR model under L´evy process framework. *Applied Mathematics and Computation* 446.

Asmussen, S., P.W. Glynn, 2007.*Stochastic Simulation: Algorithms and Analysis*, Springer Verlag, New York.

Boost C++ Library: `https://www.boost.org`.

BT. Bradley, J. du Toit, M.B. Giles, R. Tong, P. Woodhams. 2011. Parallelisation techniques for random number generators. In *GPU Computing Gems, Emerald Edition*, 231-246, Morgan Kaufmann. `https://people.maths.ox.ac.uk/gilesm/files/gems_rng.pdf`.

Cheng, R.C.H., and G. M. Feast. 1979. Some simple gamma variate generators. *Journal of the Royal Statistical Society*, Series C (Applied Statistics), 28 (3): 290-295.

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

Csorgo, S. and J.J. Faraway. 1996. The exact and asymptotic distributions of Cramer-von Mises statistics. *Journal of the Royal Statistical Society* B 58 (1): 221-234.

Duffie, D., J. Pan, and K. Singleton. 2000. Transform analysis and asset pricing for affine jump diffusions. *Econometrica* 68: 1343-1376.

Dyer, J.S., and S. A. Dyer. 2008. Approximations to Inverse Error Functions. *IEEE Instrumentation & Measurement Magazine* (October): 32-36.

Feller, W. 1950. Two Singular Diffusion Problems*. *Annals of Mathematics* 54 (1): 173-182.

Giles, M., and B.Waterhouse. 2009. Multilevel quasi-Monte Carlo path simulation. *Advanced Financial Modelling, Radon Series on Computational and Applied Mathematics*, 165–181.

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

Heston, S.L. 1993. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Review of Financial Studies* 6 (2): 327-343.

Johnson, N., S. Kotz, and N. Balakrishnan. 1995. *Continuous Univariate Distributions*, Vol. 2, 2nd Edition, Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. John Wiley & Sons, Inc., New York.

Kahl, C. and P. Jäckel. 2006. Fast strong approximation Monte-Carlo schemes for stochastic volatility models. *Journal of Quantitative Finance* 6 (6): 513-536.

Kiesel, R., and M. Lutz. 2011. Efficient pricing of constant maturity swap spread options in a stochastic volatility LIBOR market model. *The Journal of Computational Finance* 14 (4): 37-72.

L'Ecuyer, P. 1999. Good parameters and implementations for combined multiple recursive random number generators. *Operations Research* 47(1):159-164.

L'Ecuyer, P. 2023. *Stochastic simulation and Monte Carlo methods*, Draft Textbook, `https://www-labs.iro.umontreal.ca/~lecuyer/ift6561/book.pdf`

L'Ecuyer, P., R. Simard, E. J. Chen, and W. D. Kelton. 2002. An object-oriented random number package with many long streams and substreams. *Operations Research* 50 (6):1073-1075

L'Ecuyer, P., and R. Simard. 2007. TestU01: A C library for empirical testing of random number generators. *ACM Trans. Math. Software* 33 (August).

L'Ecuyer, P., and D. Munger. 2012. Constructing adapted lattice rules using problem-dependent criteria. In *Proceedings of the 2012 Winter Simulation Conference*, IEEE Press, 373–384

Orlando, G., and M. Bufalo. December 2022. A generalized two-factor square-root framework for modeling occurrences of natural catastrophes. *Journal of Forecasting* 41(8): 1608–1622.

Salmon, J. K., M. A. Moraes, R. O. Dror, and D. E. Shaw. 2011. Parallel random numbers: as easy as 1, 2, 3. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC11, New York, NY: ACM). `http://www.thesalmons.org/john/random123/papers/random123sc11.pdf`.

Wallace, N.D. 1974. Computer generation of gamma random variates with non-integral shape parameters. *Communications of the Association for Computing Machinery* 17 (12): .

Wilson, E. B., and M. M. Hilferty. 1931. The distribution of chi-square. *Proceedings of the National Academy of Sciences* 17 (12): 684–688.

Wu, L., and F. Zhang. 2006. LIBOR market model with stochastic volatility. *Journal of Industrial and Management Optimization* 2 (2): 199-227.

Appendix

A. Beta – Non-Central Chi Squared Approximation

This method uses the cumulative distribution function specified in (6), for $x \geq 0$.

$$F_1(cx) = p \min[cx, 1]^{\nu^*/2} + (1-p) F_{\chi^{2\prime}}(cx/\beta, 1, \lambda_{nc}).$$

Using M and S^2 as defined for the mean and variance for cx, define m_1 and m_2 to be the mean and 2nd moment for the Beta distribution as follows:

$$m_1 = \frac{\nu^*}{\nu^* + 2} \quad \text{and} \quad m_2 = \frac{\nu^*}{\nu^* + 4}.$$

The solution to match the mean and variance for the process cx is calculated as follows.

$$\lambda^*_{nc} = \frac{\lambda^2_{nc} + 6\lambda_{nc} + 3}{(\lambda_{nc} + 1)^2}$$

$$a_1 = m_1^2 \lambda^*_{nc} - m_2, \quad a_2 = S^2 + M^2 + m_2 - 2m_1\lambda^*_{nc} M, \quad a_3 = M^2 (\lambda^*_{nc} - 1) - S^2$$

$$p = \frac{-a_2 + \sqrt{a_2^2 - 4a_1 a_3}}{2a_1}$$

If $4a_1a_3$ is small relative to a_2^2, set

$$u = \frac{-4a_1 a_3}{a_2^2}$$

and calculate the probability p as follows with $u \leq 0.0001$:

$$p = \frac{a_2 \left(\frac{1}{2}u - \frac{1}{8}u^2 + \frac{3}{48}u^3 - \frac{5}{128}u^4 \cdots\right)}{2a_1}.$$

Then calculate β as follows:

$$\beta = \frac{M - pm_1}{(1-p)(\lambda_{nc} + 1)}.$$

To simulate the mixture distribution, generate a uniform number, $U_1 = U(0,1)$. If $U_1 \leq p$, choose the beta distribution; otherwise choose the non-central chi squared distribution. If the beta distribution is selected, create a second uniform random number as follows to generate the beta variate:

$$U_2 = \frac{U_1}{p} \quad \text{and} \quad cx = U_2^{2/\nu^*}.$$

If the non-central chi squared is selected, calculate the 2nd uniform random number as follows to simulate the normal variate:

$$U_2 = \frac{1 - U_1}{1 - p} \quad \text{and} \quad cx = \beta \left(\sqrt{\lambda_{nc}} + N(U_2)^{-1}\right)^2.$$

Alternatively, one can generate a second independent simulation for U_2.

B. Beta – Double Exponential Approximation

This method uses the approximation of the CDF for a non central chi squared distribution, specified in $F_2(x)$. The mean and the second moment are evaluated by integrating the PDFs over the different ranges for x. The mean is labeled μ, and the second moment is the sum of μ^2 and the variance of Δx, labeled, $\text{Var}(\Delta x)$.

$$\begin{aligned}\mu = \int_0^\infty x f_2(x)dx &= \int_0^{x_1} xcA^* \left(cx\right)^{\frac{\nu}{2}-1} dx + \int_{x_1}^{x_2} x(1-F_2(x_1))\beta_1 e^{-\beta_1(x-x_1)}dx \\ &\quad + \int_{x_2}^{\infty} x(1-F_2(x_2))\beta_2 e^{-\beta_2(x-x_2)}dx \\ =&\frac{A^*\left(cx_1\right)^{\nu/2+1}}{c\left(\frac{\nu}{2}+1\right)} + (1-F_2(x_1))\left[-(x_2-x_1)e^{-\beta_1(x_2-x_1)} + \left(\frac{1}{\beta_1}+x_1\right)\left(1-e^{-\beta_1(x_2-x_1)}\right)\right] \\ &\quad + (1-F_2(x_2))\left(\frac{1}{\beta_2}+x_2\right)\end{aligned}$$

$$\begin{aligned}\mu^2 + \text{Var}(\Delta x) &= \int_0^\infty x^2 f_2(x)dx \\ &= \int_0^{x_1} x^2 cA^* \left(cx\right)^{\nu/2-1} dx + \int_{x_1}^{x_2} x^2(1-F_2(x_1))\beta_1 e^{-\beta_1(x-x_1)}dx \\ &\quad + \int_{x_2}^{\infty} x^2(1-F_2(x_2))\beta_2 e^{-\beta_2(x-x_2)}dx \\ &= \frac{A^*\left(cx_1\right)^{\nu/2+2}}{c^2\left(\frac{\nu}{2}+2\right)} \\ &\quad + (1-F_2(x_1))\left[-(x_2-x_1)^2 e^{-\beta_1(x_2-x_1)} + x_1^2\left(1-e^{-\beta_1(x_2-x_1)}\right)\right] \\ &\quad + (1-F_2(x_1))\left[-(x_2-x_1)e^{-\beta_1(x_2-x_1)} + \frac{1}{\beta_1}\left(1-e^{-\beta_1(x_2-x_1)}\right)\right] \\ &\quad + (1-F_2(x_2))\left(\frac{2}{\beta_2}\left(\frac{1}{\beta_2}+x_2\right)+x_2^2\right)\end{aligned}$$

For a given value of β_1, set β_2 to match the mean as follows. Calculate $\hat{\beta}_2$ from

$$\frac{1}{\hat{\beta}_2} + x_2 = \frac{\mu - \frac{A^*(cx_1)^{\nu/2+1}}{c\left(\frac{\nu}{2}+1\right)} - (1-F_2(x_1))\left[-\left(x_2-x_1\right)e^{-\beta_1(x_2-x_1)} + \left(\frac{1}{\beta_1}+x_1\right)\left(1-e^{-\beta_1(x_2-x_1)}\right)\right]}{1-F_2(x_2)}$$

The solution $\hat{\beta}_2$ is substituted into the equation for the second moment, and an iterative search is performed on β_1 to find the combination β_1, β_2 that match both the mean and second moment, or variance. The search procedure uses the partial derivative of the second moment with respect to β_1 as well as the partial derivatives of $F_2(x_2)$ and $\hat{\beta}_2$ with respect to β_1. The simulation is then based on the inversion of the approximation CDF, $F_2(x)$.

If $U \leq F_2(x_1)$, use

$$x = \frac{1}{c}\left(\frac{U\nu}{2A^*}\right)^{\frac{2}{\nu}}$$

If $F_2(x_1) < U \leq F_2(x_2)$, use

$$x = x_1 - \frac{1}{\beta_1}\log\left(\frac{1-U}{1-F_2(x_1)}\right)$$

If $U > F_2(x_2)$, use

$$x = x_2 - \frac{1}{\beta_2}\log\left(\frac{1-U}{1-F_2(x_2)}\right)$$

The iterative search for β_1 and β_2 is required only if the uniform simulation, $U > F_2(x_1)$.