

Simulation of Square Root Processes Revisited

Louis Scott

October 2022

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. In derivative pricing, there are many cases which require Monte Carlo simulation of stochastic processes, including the square root process. Fast simulation methods for square root processes have been previously developed in papers by Kahl and Jäckel (2006), Andersen (2008), and Andersen, Jäckel, and Kahl (2010). The purpose of this paper is to revisit the techniques for fast simulation of square root processes over small time periods for use in dynamic financial models, and to introduce an alternative method developed as an extension of the methods developed by Andersen, Jäckel, and Kahl.

Section 1 reviews the square root process and the simulation methods based on the non-central chi squared distribution, which is the exact distribution for the process over discrete time intervals. Simulation methods for the non-central chi squared distribution are relatively slow and are not well suited for parameter calibration and parallel processing. As discussed in section 1, non-central chi squared simulation requires acceptance/rejection methods so that it is difficult to predict the number of simulations for each time step and control the number of simulations on each simulation path. Section 2 covers alternative approximations methods and develops an extension of existing methods. These alternative simulation methods require only one simulation per time step and facilitate both parameter calibration and parallel processing. Section 3 provides tests of the alternative simulation methods, including applications with parallel processing on a GPU. The test results indicate that the approximation methods generally converge and significantly reduce computing time when parallel processing is applied.

I. Simulating Square Root Processes

The square root process is a diffusion process with the following stochastic differential equation.

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

where dz is the change in a Brownian motion. The distribution of $x(t)$, conditional on $x(0)$ for $t > 0$, is non-central chi squared. The parameters κ , θ , and σ are greater than zero, and λ

is a risk premium parameter that can be positive or negative. θ is the long run mean and κ determines the rate of mean reversion for the “real world” process ($\lambda = 0$). $\kappa + \lambda$ determines the rate of mean reversion under the risk neutral measure. 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 applications 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.

One method for simulating x is to use the exact distribution and simulate a non-central chi squared variate over discrete time intervals. The variable $c x(t)$, conditional on $x(0)$, is distributed as a non-central chi squared with v degrees of freedom and non-centrality parameter λ_{nc} defined as follows:

$$v = \frac{4\kappa\theta}{\sigma^2}, \quad \lambda_{nc} = cx(0)e^{-(\kappa+\lambda)t}, \quad \text{and } c = \frac{4(\kappa + \lambda)}{\sigma^2(1 - e^{-(\kappa+\lambda)t})}.$$

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 v plus 2 times the Poisson simulation. 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 v degrees of freedom and non-centrality parameter λ_{nc} :

$$\chi'^2(v, \lambda_{nc}) = (Z + \sqrt{\lambda_{nc}})^2 + \chi^2(v - 1)$$

where Z is a standard normal random variate and $\chi^2(v - 1)$ is a chi squared with $v - 1$ degrees of freedom. $\chi'^2(v, \lambda_{nc})$ is used to indicate a non-central chi squared distribution. This latter method requires $v \geq 1$. The first method using a Poisson simulation and a chi squared simulation can be used for any $v > 0$. The exact method requires either a Poisson simulation plus a chi squared simulation or a normal simulation plus a chi squared simulation. Poisson simulations and normal simulations can be calculated very quickly, but chi squared simulation is more complex and requires acceptance-rejection methods. These methods are exact, but require significantly more computing time, relative to the simulation of random variables with normal or exponential distributions. 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 of that a diffusion process is approximately normal over small time intervals.

Computing technology has improved dramatically in recent years so that financial models requiring Monte Carlo simulation can be run very quickly with parallel processing

using either multiple threads on CPU's or the multiple streaming processors on GPU's. This is the case for derivative pricing in finance, where fast methods are necessary for pricing in real time. Parallel processing introduces additional requirements for the simulation of random numbers, as most random number generators (RNG) are sequential. The pseudo RNG's have initial seeds or states that can be used to control the sequences of random numbers. For some of these RNG's, there are quick methods for calculating the initial seeds necessary for subsequent simulations across multiple processors, which will run in parallel. These step-ahead techniques make it possible to control the initial seeds and the random number simulations necessary for testing and for parameter calibration. Many of the derivative pricing models require parameter calibration in which model parameters are adjusted to fit model prices to observed market prices. Parameter calibration has typically been a challenge for models that must be solved by simulation. In the past, the parameter calibration was performed with approximation models, as Monte Carlo methods were too slow. This is no longer the case as massive parallel processing with GPU's facilitates fast simulation so that one can now perform real time pricing and parameter calibration with simulation models. Parameter calibration requires recalculations of the simulation model with the same random number simulations as the model parameters are changed. The set of parameters that optimizes an objective function for the fit of model prices to market prices changes if the simulated random numbers change. For this reason, it is important to run an optimization algorithm with a fixed set of random numbers. If one is not able to control the random simulations, the optimization algorithm for parameter calibration will be chasing optimal parameter sets that vary. 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 model parameters are adjusted. Likewise, one can control a sequence of normal or exponential random numbers by simulating uniform random numbers and using the inverse of the cumulative distribution function. To control simulations across parallel processing, it is simply a matter of counting the number of simulations on each simulation path and applying the appropriate step-ahead method to calculate the initial seeds for the RNG on each processor. In this manner, one can use parallel processing and reproduce a set of simulations for uniform, exponential, and normal random numbers for parameter calibration. Simulation using the non-central chi squared distribution uses acceptance/rejection methods so that the number of simulations for each time step vary and one cannot predict or control the number of simulations required for each simulation path. To perform parameter calibration, testing, and real time pricing 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 simulations for a selected RNG.

2. Alternative Approximation Methods

Stochastic differential equations are approximated by simulating over small discrete time intervals. Financial models for equity prices and FX rates, for example, typically use

simulations over daily time intervals. The simulation of an exact non-central chi squared over a small time interval does not take advantage of the property that diffusion processes are functions of a Brownian motion, which has a normal distribution. 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 simulation or a simulation based on a mixture of an exponential with a distribution concentrated at zero. The mixture of the exponential simulation and the simulation of zeros 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 the model 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 this implicit Milstein scheme is similar, but not identical, to the quadratic simulation in 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 diffusion, 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 .

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

The second term 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:

$$y(t) = e^{-\frac{1}{2}(\kappa+\lambda)t} y(0) + \frac{\sigma}{2} \int_0^t e^{-\frac{1}{2}(\kappa+\lambda)(t-s)} dz_s .$$

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:

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

which is

$$y(t) = \sqrt{e^{-(\kappa+\lambda)t}y^2(0) + \left(\kappa\theta - \sigma^2/4\right)\left(\frac{1 - e^{-(\kappa+\lambda)t}}{\kappa + \lambda}\right)}.$$

This leads to the following proposed approximation:

$$y(t) = \sqrt{e^{-(\kappa+\lambda)t}y^2(0) + \left(\kappa\theta - \sigma^2/4\right)\left(\frac{1 - e^{-(\kappa+\lambda)t}}{\kappa + \lambda}\right)} + \frac{\sigma}{2} \int_0^t e^{-\frac{1}{2}(\kappa+\lambda)(t-s)} dz_s.$$

$y(0) = \sqrt{x(0)}$ is the boundary condition, and $x(t) = y^2(t)$ so that the solution can be expressed in terms of x as follows:

$$x(t) = \left(\sqrt{e^{-(\kappa+\lambda)t}x(0) + \left(\kappa\theta - \sigma^2/4\right)\left(\frac{1 - e^{-(\kappa+\lambda)t}}{\kappa + \lambda}\right)} + \frac{\sigma}{2} \int_0^t e^{-\frac{1}{2}(\kappa+\lambda)(t-s)} dz_s \right)^2. \quad (2)$$

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,

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

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:

$$E(\Delta x) = (e^{-(\kappa+\lambda)t} - 1)x(0) + \kappa\theta \left(\frac{1 - e^{-(\kappa+\lambda)t}}{\kappa + \lambda} \right) \quad (3)$$

$$\text{Var}(\Delta x) = \sigma^2 \left(\frac{1 - e^{-(\kappa+\lambda)t}}{\kappa + \lambda} \right) \left(e^{-(\kappa+\lambda)t}x(0) + \frac{1}{2}\kappa\theta \left(\frac{1 - e^{-(\kappa+\lambda)t}}{\kappa + \lambda} \right) \right). \quad (4)$$

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.

$$x(t) = e^{-(\kappa+\lambda)t}x(0) + \kappa\theta \left(\frac{1 - e^{-(\kappa+\lambda)t}}{\kappa + \lambda} \right) \\ + \sqrt{e^{-(\kappa+\lambda)t}x(0) + \left(\kappa\theta - \sigma^2/4\right)\left(\frac{1 - e^{-(\kappa+\lambda)t}}{\kappa + \lambda}\right)} \sigma \int_0^t e^{-\frac{1}{2}(\kappa+\lambda)(t-s)} dz_s$$

$$-\frac{\sigma^2}{4} \left(\frac{1 - e^{-(\kappa+\lambda)t}}{\kappa + \lambda} \right) + \left(\frac{\sigma}{2} \int_0^t e^{-\frac{1}{2}(\kappa+\lambda)(t-s)} dz_s \right)^2$$

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

$$\begin{aligned} \text{Var}(\Delta x) &= E \left(\sqrt{e^{-(\kappa+\lambda)t} x(0) + \left(\kappa\theta - \sigma^2/4 \right) \left(\frac{1 - e^{-(\kappa+\lambda)t}}{\kappa + \lambda} \right)} \int_0^t \sigma e^{-\frac{1}{2}(\kappa+\lambda)(t-s)} dz_s \right. \\ &\quad \left. - \frac{\sigma^2}{4} \left(\frac{1 - e^{-(\kappa+\lambda)t}}{\kappa + \lambda} \right) + \left(\frac{\sigma}{2} \int_0^t e^{-\frac{1}{2}(\kappa+\lambda)(t-s)} dz_s \right)^2 \right)^2 \\ &= \sigma^2 \left(\frac{1 - e^{-(\kappa+\lambda)t}}{\kappa + \lambda} \right) \left(e^{-(\kappa+\lambda)t} x(0) + \left(\kappa\theta - \frac{\sigma^2}{4} \right) \left(\frac{1 - e^{-(\kappa+\lambda)t}}{\kappa + \lambda} \right) \right) \\ &\quad + 2 \left(\frac{\sigma^2}{4} \left(\frac{1 - e^{-(\kappa+\lambda)t}}{\kappa + \lambda} \right) \right)^2 \\ &= \sigma^2 x(0)t + o(t) \end{aligned}$$

The variance for the approximation model in (2) 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 proposed simulation method in (2) matches the first 2 moments for the square root diffusion process for sufficiently small time intervals. For the case $4\kappa\theta = \sigma^2$, the approximation model is an exact simulation of the non-central chi squared distribution.

If $4\kappa\theta \geq \sigma^2$, and $x(0) \geq 0$, the term inside the square root is always positive and the requirement that x be nonnegative is preserved with the simulation methodology. If $4\kappa\theta < \sigma^2$, the term inside the square root can become negative for small values of $x(0)$. 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 a large number of 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 cumulative distribution function is 0 for $x=0$, or $F(0) = 0$, with the probability density function going to infinity as x approaches 0. The slope of $F(x)$ at $x = 0$ is infinite. An alternative is to follow Andersen's (2008) suggestion in Appendix B and develop a mixture of the exponential distribution with a distribution that has this property. One natural choice is to use a mixture with a beta distribution.

An alternative methodology developed here is to use a mixture of the non-central chi squared distribution with 1 degree of freedom with a beta distribution as follows.

$$F(cx) = p (\min [cx, 1])^{\nu^*/2} + (1 - p)\chi^{2'}(1, \lambda_{nc}, cx/\beta) \quad \text{for } x \geq 0 \quad (5)$$

$\chi^{2'}(1, \lambda_{nc}, cx/\beta)$ is the cumulative distribution function for the non-central chi squared distribution with one degree of freedom, non-centrality parameter λ_{nc} , and input argument cx/β . The beta distribution is similar to a chi squared with degrees for 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 simulation of a chi squared when the degrees of freedom is small. The non-central chi squared with one degree of freedom is used because it is easy to simulate (one normal simulation) and it provides a good approximation as ν decreases below one. The beta parameter ν^* is scaled down from ν as follows: $\nu^* = \nu / 2.0$. Simulation is performed by using the inverse function method: simulate a uniform random variable for $F(cx)$ and invert to get the simulated value for x . The parameters p and β should be set to match the mean and variance for the square root process as follows. First, define M and S^2 to be the mean and variance for cx , and define m_1 and m_2 to be the mean and 2nd moment for the Beta distribution as follows:

$$\begin{aligned} M &= c \left((e^{-(\kappa+\lambda)t} - 1)x(0) + \kappa\theta \left(\frac{1 - e^{-(\kappa+\lambda)t}}{\kappa + \lambda} \right) \right) \\ S^2 &= c^2 \sigma^2 \left(\frac{1 - e^{-(\kappa+\lambda)t}}{\kappa + \lambda} \right) \left(e^{-(\kappa+\lambda)t}x(0) + \frac{1}{2}\kappa\theta \left(\frac{1 - e^{-(\kappa+\lambda)t}}{\kappa + \lambda} \right) \right) \\ m_1 &= \frac{\nu^*}{\nu^* + 2} \quad \text{and} \quad m_2 = \frac{\nu^*}{\nu^* + 4}. \end{aligned}$$

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

$$\begin{aligned} \lambda_{nc}^* &= \frac{\lambda_{nc}^2 + 6\lambda_{nc} + 3}{(\lambda_{nc} + 1)^2} \\ a_1 &= m_1^2 \lambda_{nc}^* - m_2, \quad a_2 = S^2 + M^2 + m_2 - m_1 \lambda_{nc}^* M, \quad a_3 = M^2 (\lambda_{nc}^* - 1) - S^2 \\ p &= \frac{-a_2 + \sqrt{a_2^2 - 4a_1 a_3}}{2 a_1} \\ \beta &= \frac{M - pm_1}{(1 - p)(\lambda_{nc} + 1)} \end{aligned}$$

To simulate the mixture distribution, simulate 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 and use the inverse for standard normal distribution to simulate the normal random variate. If

the Beta distribution is selected, create a second uniform random number as follows to simulate the Beta variate:

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

If the non-central chi squared is selected, $U_1 > p$, calculate the 2nd uniform random number as follows to simulate the normal variate:

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

Alternatively, one can use a second independent simulation for U_2 . The proposed simulation methodology uses a quadratic function of a normal simulation or a mixture of a beta and a non-central chi squared with one degree of freedom. Hence, we use the name quadratic-beta or QB. In summary for cases where $\nu \geq 1$, the QB methodology uses the normal simulation in the quadratic method in equation (2). If $\nu < 1$, $\lambda_{nc} > 4$, and

$$x(0) \geq -e^{(\kappa+\lambda)t} \left(\kappa\theta - \sigma^2/4 \right) \left(\frac{1 - e^{-(\kappa+\lambda)t}}{\kappa + \lambda} \right),$$

use the quadratic method in equation (2). If these conditions are not satisfied, switch to the mixed beta non-central chi squared simulation in equation (5). The parameter λ_{nc} controls the probability for simulating values, $\exp(-\lambda_{nc})$, close to zero for a non-central chi squared. In cases where λ_{nc} is small (< 4), the probability of generating small values is larger and the mixture of the beta and the non-central chi squared provides a better approximation for the distribution.

3. Tests of the Simulation Methodologies, with Parallel Processing

A variety of comparisons and tests are presented in this section. We begin by comparing plots of the cumulative distribution functions for the QB and QE approximation methods with the exact distribution for a non-central chi squared with degrees of freedom equal to 0.08, 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 with the model parameters set as follows: $\kappa = 0.5, \theta = 0.04, \sigma = 1.0, \lambda = 0$, and $t = 0.1$, with 3 different values for $x(0)$: 0.01, 0.09, and 0.04. The distribution functions for the 2 approximation methods 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 $t = 1/365$ for a calendar day, or with $t = 1/252$ for a trading day. The graphs in Figure 2 contain plots of the distribution functions for daily time steps with $t = 1/365$, 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 the exception of the case in Panel B in which the QE method which produces a high probability for $x = 0$. This is a characteristic of the QE approximation method.

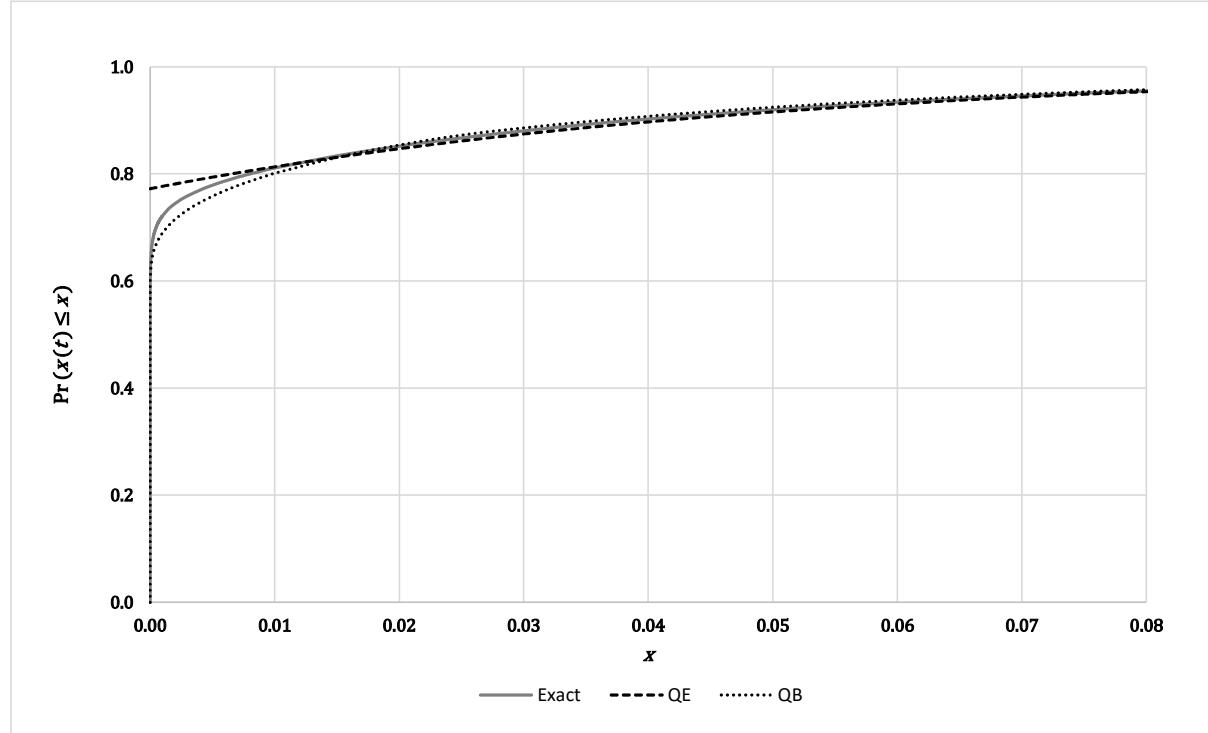
The next set of tests are tests of the simulated distributions generated by the two approximation methods, compared to the corresponding exact non-central chi squared distribution. The Euler methodology is included in these tests as a “straw man” for comparison purposes. The Euler method is simple and easy to implement, and in the cases tested, it requires less compute time. The following equation is used for the Euler methodology:

$$x(t) = x(0) + (e^{-(\kappa+\lambda)\Delta t} - 1)x(0) + \kappa\theta \left(\frac{1 - e^{-(\kappa+\lambda)\Delta t}}{\kappa + \lambda} \right) + \sigma\sqrt{x(0)} \Delta z$$

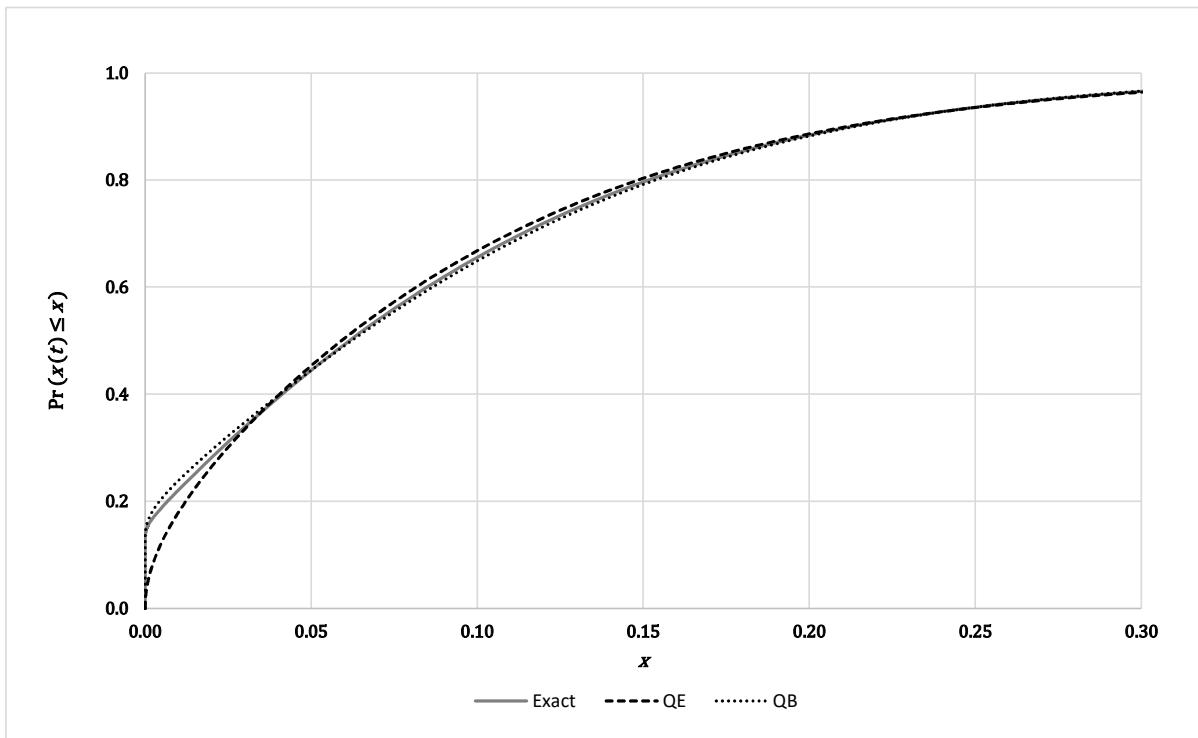
where Δz is a normal random variable with mean zero and variance Δt , and $x(t)$ is set to 0 if a negative value is simulated. When the degrees of freedom of parameter is less than 2, the Euler method produces negative values. When the degrees of freedom are less than 1, the method produces simulations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

Figure 1. Cumulative Distribution Functions for Approximation Methods

Panel A. $x(0) = 0.01$



Panel B. $x(0) = 0.09$



Panel C. $x(0) = 0.04$

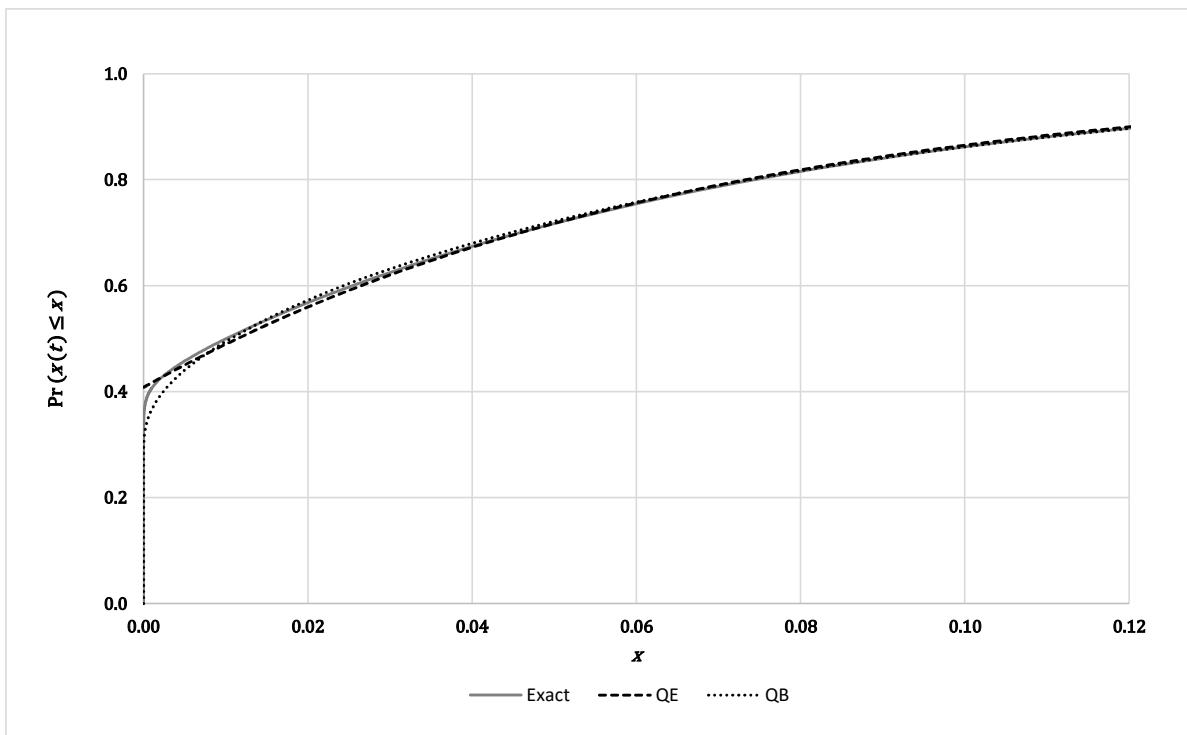
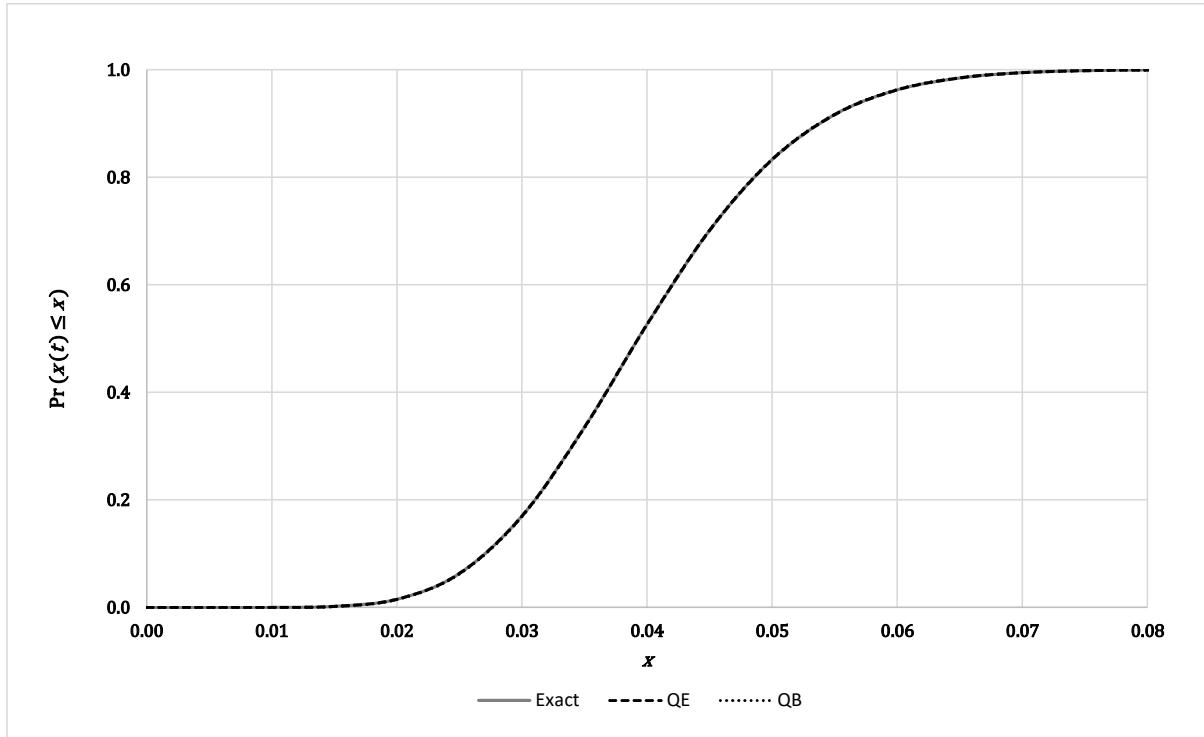
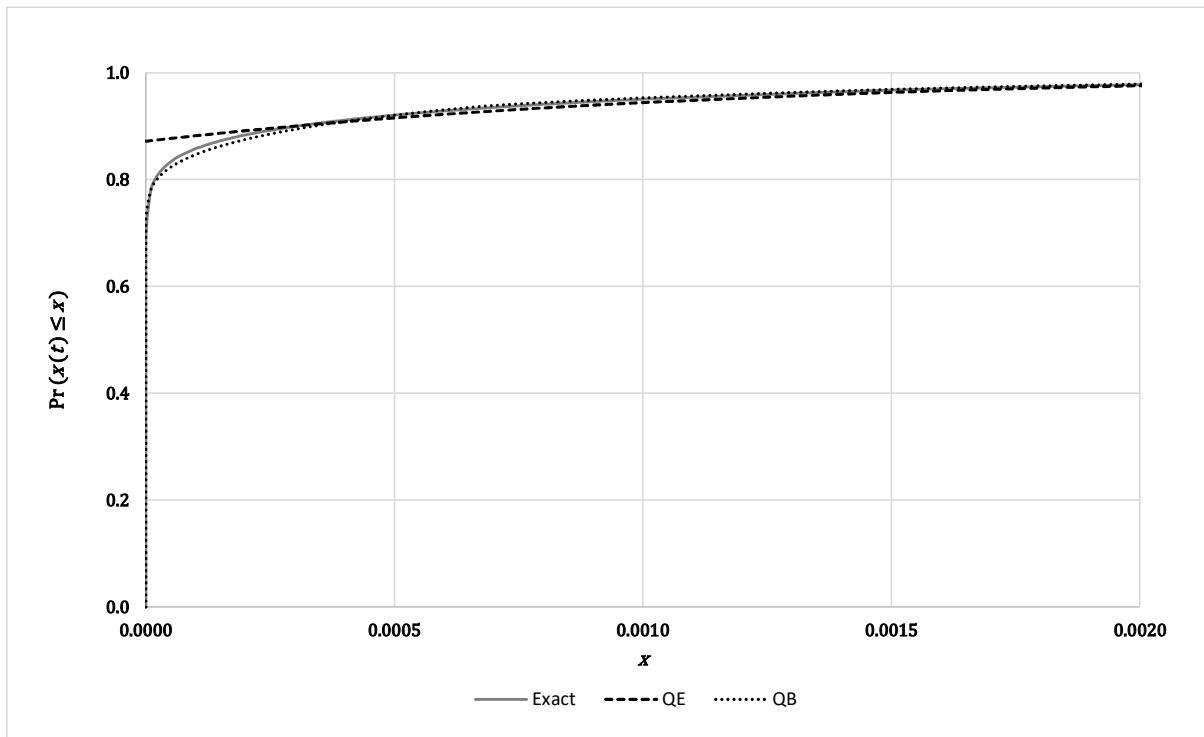


Figure 2. Cumulative Distribution Functions for a Daily Time Interval

Panel A. $x(0) = 0.04$



Panel B. $x(0) = 0.0001$



The simulation methods are run with the following model parameters for daily time steps: $\kappa = 0.25$, $\theta = 0.04$, $\lambda = -0.125$, and $t = \frac{1}{365} = 0.00273973$, 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, v . The tests are run on simulations of multiple time steps over 91 daily time tests 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 simulation paths are run. The compute 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. The tests in Table 2 are run on a GPU only because the compute times are much longer on a CPU without parallel processing. The same tests were applied to simulations using the exact methods for simulating non-central chi squared variates described in section 2. The exact simulation produces simulations that pass all of the tests in Tables 1 and 2, and the computing times are roughly 10 times the computing times for the approximation methods.

A variety of standard statistical tests are calculated for the simulated distributions. The mean and variance of the simulated distributions 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 do serve as tests 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 close to zero, and a simulation of zero produces an infinite value for this statistic. For this reason, there are cases with the QE method and the Euler method that 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 about the mean for the simulated values. The critical values for the Kolmogorov-Smirnov statistic have been calculated by running the program provided by Simard & 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 test are from Table 1, p. 229, in Csorgo & 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 the statistic do vary with the degrees of freedom. The model simulations indicate that the critical values for the cases in Table 1 are generally 2.3 to 2.5 for the 95th percentile and 3.7 to 4.2 for the 99th percentile.

Finally, a few comments regarding the simulation of the models and the calculation of the non-central chi squared distribution are necessary. The QB and QE simulation methods and the Euler method have been programmed in C++. Additional simulations of the exact non-central chi squared distribution have been programmed, but the results are not reported as the exact methodology produces simulations that pass all statistical tests. The non-central chi squared cumulative distribution is calculated using the function available in the Boost library. The model simulations for normal, beta, and exponential distributions are based on the inverse method applied using the pseudo RNG for a uniform distribution, MRG32ka, which is a combined multiple recursive RNG that was developed by L'Ecuyer (1999). This RNG has a very long period and it 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 method for the MRG32k3a RNG has been developed in the online paper by Bradley, du Toit, Giles, Tong, and Woodhams, 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 the GPU match those that are run sequentially on a CPU. The counter based RMG'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 compute times reported in Tables 1 and 2 have been performed with an Intel i7-5930K CPU, which has a maximum turbo frequency of 3.70 gigahertz, and an Nvidia RTX Titan GPU. The Nvidia RTX Titan is an older technology with 4,608 cuda cores and 576 tensor cores. Current technology includes CPU's with speeds above 4 gigahertz and GPU's with more cuda cores and tensor

cores that run at faster speeds. The compute times would be significantly less on Nvidia's A100 or H100 GPU's. (The C++ code will be available on GitHub.)

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 show results when the ν is either 4.0 or 1.0. The QB and QE perform well in these cases, as all statistical tests are passed, with the exception of the Anderson-Darling test for the QE method when $\nu = 1$. In these cases, the QE method is simulating terminal values that are exactly zero. It should be noted that if one adjusts the switch threshold in the QE method 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 2 methods. 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 of the statistical tests. For $\nu < 1$, the QE method passes the tests for the mean and variance, but it fails on almost all of the tests on the simulated distributions. In these cases, the QB method passes all of the tests for the mean, but it fails on the t test for the variance at the 95% level when $\nu \leq 0.25$, as the variance is less than the variance for a non-central chi squared. The QB method also fails on most of the statistical tests on the simulated distributions, but the test statistics are much smaller than those produced by the QE method.

Table 2 contains results for simulations over a much longer time period, 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 QB, 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 tests statistic. Again, 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. These tests were not run on a CPU as the compute times would have been much longer. The compute times in Table 2 are between 1 and 5 seconds and show that one can run these simulation methods with daily time steps for a large number of simulations, 1 million, over a long time horizon, 10 years, with a relatively quick response time.

Table 1. Tests of the Simulation Methods with parameters: $\kappa = 0.25, \theta = 0.04, \sigma = 1.0, \lambda = -0.125$ using 1,000,000 simulation paths over 91 days with 1 time step per day

Panel A. $x(0) = 0.04, \sigma = 0.1, \nu = 4.0$

	QB	QE	Euler	Analytic
Mean	0.041219	0.041219	0.041219	0.041227
t test (Mean)	-0.08	-0.08	-0.08	
Std. Deviation	0.009901	0.009900	0.009901	0.009909
t test (Variance)	-1.03	-1.20	-1.08	
Kolmogorov-Smirnov	0.0006250	0.0006479	0.0004902	
Cramer-von Mises	0.0716	0.0741	0.0322	
Anderson-Darling	0.4238	0.4475	0.3578	
CPU Run Time (sec.)	2.281	3.785	1.854	
GPU Run Time (sec.)	0.056	0.069	0.052	

Panel B. $x(0) = 0.01, \sigma = 0.1, \nu = 4.0$

	QB	QE	Euler	Analytic
Mean	0.012143	0.012143	0.012143	0.012148
t test (Mean)	-0.06	-0.06	-0.06	
Std. Deviation	0.005174	0.005174	0.005170	0.005178
t test (Variance)	-0.78	-1.35	-1.79	
Kolmogorov-Smirnov	0.0006662	0.0007213	0.0004603	
Cramer-von Mises	0.0769	0.0783	0.0525	
Anderson-Darling	0.4516	0.4752	0.6961	
CPU Run Time (sec.)	2.191	3.909	1.724	
GPU Run Time (sec.)	0.056	0.066	0.051	

Panel C. $x(0) = 0.04, \sigma = 0.2, \nu = 1.0$

	QB	QE	Euler	Analytic
Mean	0.041210	0.041210	0.041210	0.041227
t test (Mean)	-0.13	-0.13	-0.12	
Std. Deviation	0.019797	0.019797	0.019799	0.019818
t test (Variance)	-1.29	-1.29	-1.18	
Kolmogorov-Smirnov	0.0006367	0.0006367	0.0007256	
Cramer-von Mises	0.0729	0.0729	0.0647	
Anderson-Darling	0.4389	∞	∞	
CPU Run Time (sec.)	2.298	3.883	1.726	
GPU Run Time (sec.)	0.055	0.074	0.053	

Panel D. $x(0) = 0.01, \sigma = 0.2, \nu = 1.0$

	QB	QE	Euler	Analytic
Mean	0.012137	0.012137	0.012144	0.012148
t test (Mean)	-0.10	-0.10	-0.04	
Std. Deviation	0.010340	0.010340	0.010330	0.010355
t test (Variance)	-1.39	-1.39	-2.32*	
Kolmogorov-Smirnov	0.0006385	0.003430	0.004808	
Cramer-von Mises	0.0735	0.0977	0.3998	
Anderson-Darling	0.4579	∞	∞	
CPU Run Time (sec.)	2.119	3.980	1.865	
GPU Run Time (sec.)	0.055	0.071	0.053	

Panel E. $x(0) = 0.04, \sigma = 0.25, \nu = 0.64$

	QB	QE	Euler	Analytic
Mean	0.041204	0.041204	0.041206	0.041227
t test (Mean)	-0.15	-0.15	-0.14	
Std. Deviation	0.024742	0.024744	0.024745	0.024772
t test (Variance)	-1.41	-1.31	-1.25	
Kolmogorov-Smirnov	0.0006336	0.0006350	0.0009796	
Cramer-von Mises	0.0742	0.0722	0.1404	
Anderson-Darling	0.4698	∞	∞	
CPU Run Time (sec.)	2.283	3.922	1.690	
GPU Run Time (sec.)	0.058	0.074	0.053	

Panel F. $x(0) = 0.01, \sigma = 0.25, \nu = 0.64$

	QB	QE	Euler	Analytic
Mean	0.012135	0.012134	0.012179	0.012148
t test (Mean)	-0.11	-0.12	0.28	
Std. Deviation	0.012917	0.012922	0.012885	0.012944
t test (Variance)	-1.67	-1.38	-3.69***	
Kolmogorov-Smirnov	0.002333***	0.03069***	0.02908***	
Cramer-von Mises	0.3099***	10.9194***	39.3712***	
Anderson-Darling	18.5551**	∞	∞	
CPU Run Time (sec.)	2.378	3.759	1.785	
GPU Run Time (sec.)	0.107	0.061	0.038	

Panel G. $x(0) = 0.04, \sigma = 0.4, \nu = 0.25$

	QB	QE	Euler	Analytic
Mean	0.041186	0.041186	0.041268	0.041227
t test (Mean)	-0.21	-0.21	0.20	
Std. Deviation	0.039558	0.039578	0.039508	0.039636
t test (Variance)	-1.77	-1.32	-2.91**	
Kolmogorov-Smirnov	0.010335***	0.05953***	0.04249***	
Cramer-von Mises	2.7972***	70.9749***	51.6433***	
Anderson-Darling	291.22**	∞	∞	
CPU Run Time (sec.)	2.512	3.812	1.765	
GPU Run Time (sec.)	0.0968	0.077	0.051	

Panel H. $x(0) = 0.01, \sigma = 0.4, \nu = 0.25$

	QB	QE	Euler	Analytic
Mean	0.012125	0.012128	0.012701	0.012148
t test (Mean)	-0.16	-0.14	3.84***	
Std. Deviation	0.020631	0.020664	0.020538	0.020710
t test (Variance)	-2.15*	-1.27	-4.69***	
Kolmogorov-Smirnov	0.04681***	0.2643***	0.1987***	
Cramer-von Mises	236.12***	6199.6***	5890.7***	
Anderson-Darling	6145.6**	∞	∞	
CPU Run Time (sec.)	3.495	3.403	1.855	
GPU Run Time (sec.)	0.151	0.079	0.044	

Panel I. $x(0) = 0.04, \sigma = 0.6, \nu = 0.1111$

	QB	QE	Euler	Analytic
Mean	0.041159	0.041165	0.041994	0.041227
t test (Mean)	-0.28	-0.25	3.14***	
Std. Deviation	0.059268	0.059336	0.058960	0.059454
t test (Variance)	-2.13*	-1.37	-5.64***	
Kolmogorov-Smirnov	0.06142***	0.2859***	0.2054***	
Cramer-von Mises	497.30***	7798.6***	4460.3***	
Anderson-Darling	10553.5**	∞	∞	
CPU Run Time (sec.)	3.111	3.512	1.724	
GPU Run Time (sec.)	0.142	0.060	0.052	

Panel J. $x(0) = 0.01, \sigma = 0.6, \nu = 0.1111$

	QB	QE	Euler	Analytic
Mean	0.012130	0.012127	0.014428	0.012148
t test (Mean)	-0.10	-0.12	12.94***	
Std. Deviation	0.030888	0.030962	0.031500	0.031065
t test (Variance)	-2.32*	-1.36	5.77***	
Kolmogorov-Smirnov	0.1202***	0.5541***	0.4082***	
Cramer-von Mises	3745.8***	56757.2***	35084.6***	
Anderson-Darling	43173.7**	∞	∞	
CPU Run Time (sec.)	5.037	2.811	1.689	
GPU Run Time (sec.)	0.174	0.077	0.053	

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

Table 2. Tests of the Simulation Methods with parameters: $\kappa = 0.25, \theta = 0.04, \sigma = 1.0, \lambda = -0.125$ using 1,000,000 simulation paths over 3,650 days (10 years) with 1 time step per day

Panel A. $x(0) = 0.04, \sigma = 0.14142136, \nu = 2.0$

	QB	QE	Euler	Analytic
Mean	0.068583	0.068583	0.068591	0.068540
t test (Mean)	0.17	0.17	0.20	
Std. Deviation	0.067664	0.067660	0.067668	0.067575
t test (Variance)	0.96	0.96	1.00	
Kolmogorov-Smirnov	0.0007422	0.0007598	0.007849	
Cramer-von Mises	0.0715	0.0754	0.0877	
Anderson-Darling	0.6243	0.4983	∞	
GPU Run Time (sec.)	1.459	1.946	1.367	

Panel B. $x(0) = 0.01, \sigma = 0.14142136, \nu = 2.0$

	QB	QE	Euler	Analytic
Mean	0.059978	0.059978	0.059991	0.059945
t test (Mean)	0.14	0.14	0.19	
Std. Deviation	0.059959	0.059955	0.059963	0.059876
t test (Variance)	0.98	0.93	1.02	
Kolmogorov-Smirnov	0.0006165	0.0006383	0.0008228	
Cramer-von Mises	0.0567	0.0589	0.0851	
Anderson-Darling	0.6057	0.4322	∞	
GPU Run Time (sec.)	1.403	1.946	1.367	

Panel C. $x(0) = 0.04, \sigma = 0.25, \nu = 0.64$

	QB	QE	Euler	Analytic
Mean	0.0685978	0.068594	0.070743	0.068540
t test (Mean)	0.11	0.16	6.37***	
Std. Deviation	0.119751	0.119760	0.120551	0.119457
t test (Variance)	1.12	1.15	4.16***	
Kolmogorov-Smirnov	0.002831***	0.03740***	0.03949***	
Cramer-von Mises	0.3567*	19.7995***	389.07***	
Anderson-Darling	26.1480**	∞	∞	
GPU Run Time (sec.)	4.079	2.593	1.378	

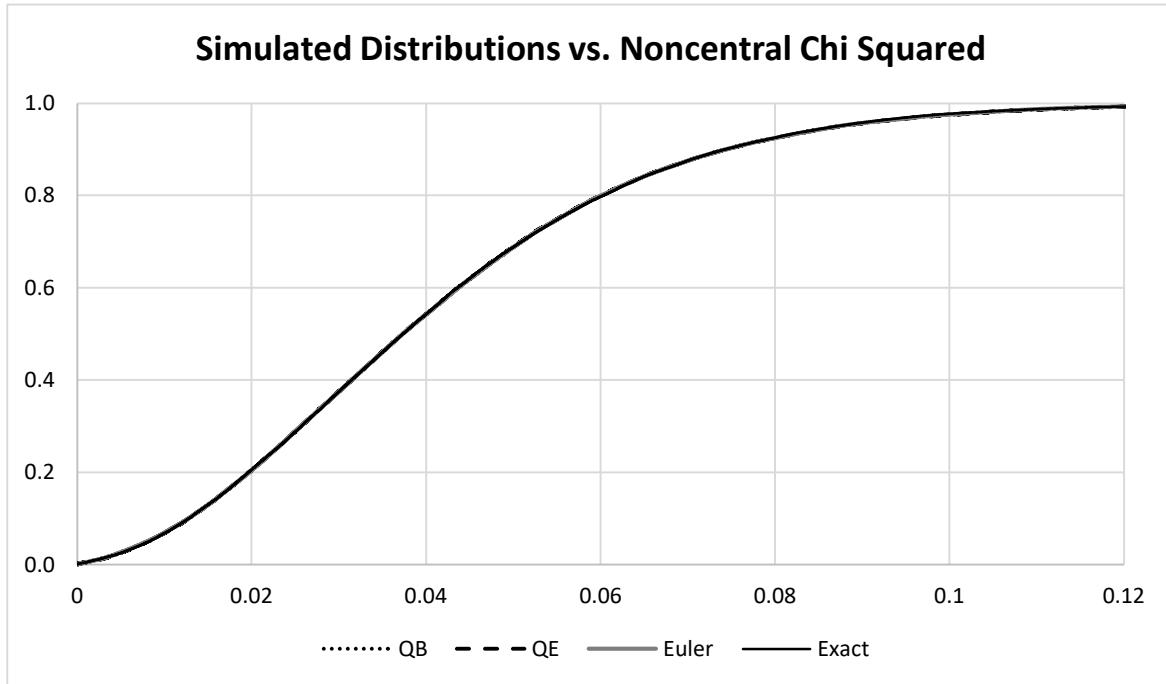
Panel D. $x(0) = 0.01, \sigma = 0.25, \nu = 0.64$

	QB	QE	Euler	Analytic
Mean	0.059968	0.059992	0.062574	0.059945
t test (Mean)	0.07	0.14	8.08***	
Std. Deviation	0.106096	0.106132	0.107660	0.105847
t test (Variance)	1.03	1.18	7.55***	
Kolmogorov-Smirnov	0.002982***	0.03920***	0.04147	
Cramer-von Mises	0.4171*	22.7842***	474.59***	
Anderson-Darling	29.1040**	∞	∞	
GPU Run Time (sec.)	4.377	2.668	1.382	

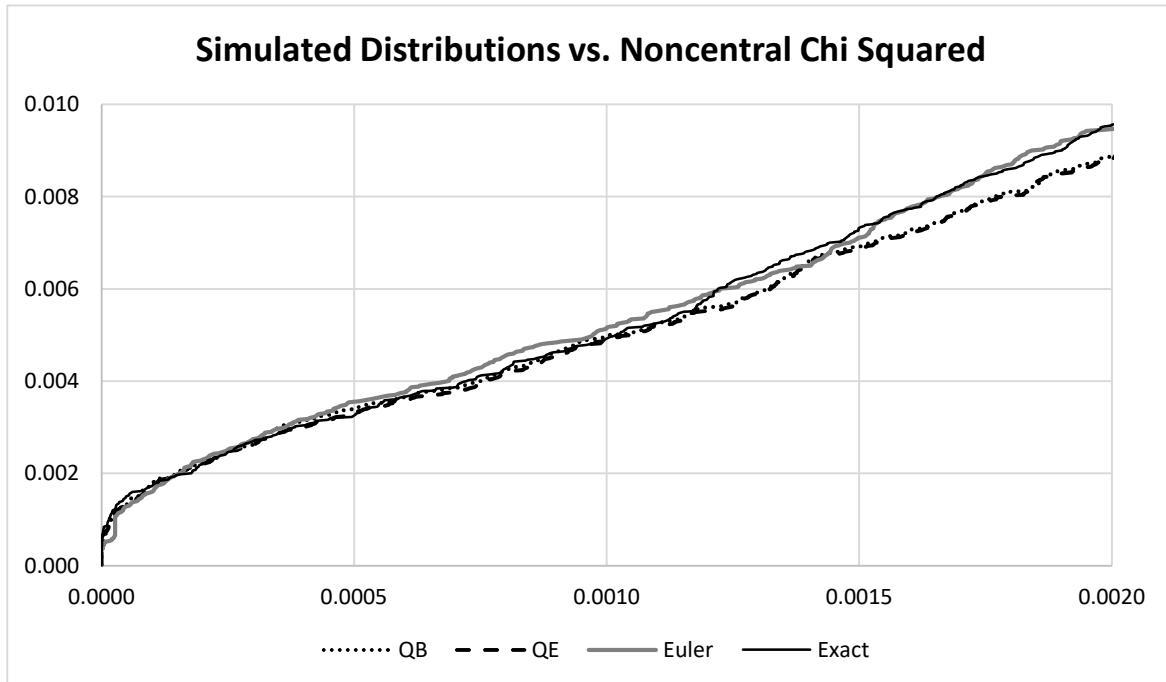
The statistical tests on the simulated distributions indicate that the approximation methods do not reproduce the non-central chi squared distribution exactly, but the tests alone do not isolate the approximation errors for the simulated distributions. The approximation methods are generally accurate for most of the distribution for the square root process, 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 Figure 3. The graphs include graphs of the simulated distributions for $x(t)$ versus the non-central chi squared using the simulations of the terminal values for 91 days, in which t approximately equal to 0.25. The model parameters are set so that $\nu = 0.64$ and $x(0)$ is either 0.04 or 0.01. The statistical tests for the goodness of fit use the non-central chi squared distribution function evaluated with the simulated $x(t)$ values, and the simulated values of $F(x(t))$ using the non-central chi squared distribution should plot along a 45-degree line when sorted. Figure 3 includes these graphs labeled as tests for simulations of cumulative distribution functions. Additional graphs are included to show the behavior in the lower tail. The approximation models generally provide good fits across most of the distribution; in most cases, the simulated distributions are indistinguishable from the non-central chi squared distribution when the entire range of the distribution is plotted. The approximation methods perform well across most of the distribution, but there are significant approximation errors in the lower tail. In the graphs of the lower tail, one can see the simulation of zero values with the QE method and the Euler method. There are approximation errors with the QB method in the lower tail, but it generally provides a better fit than either the QE method or the Euler method.

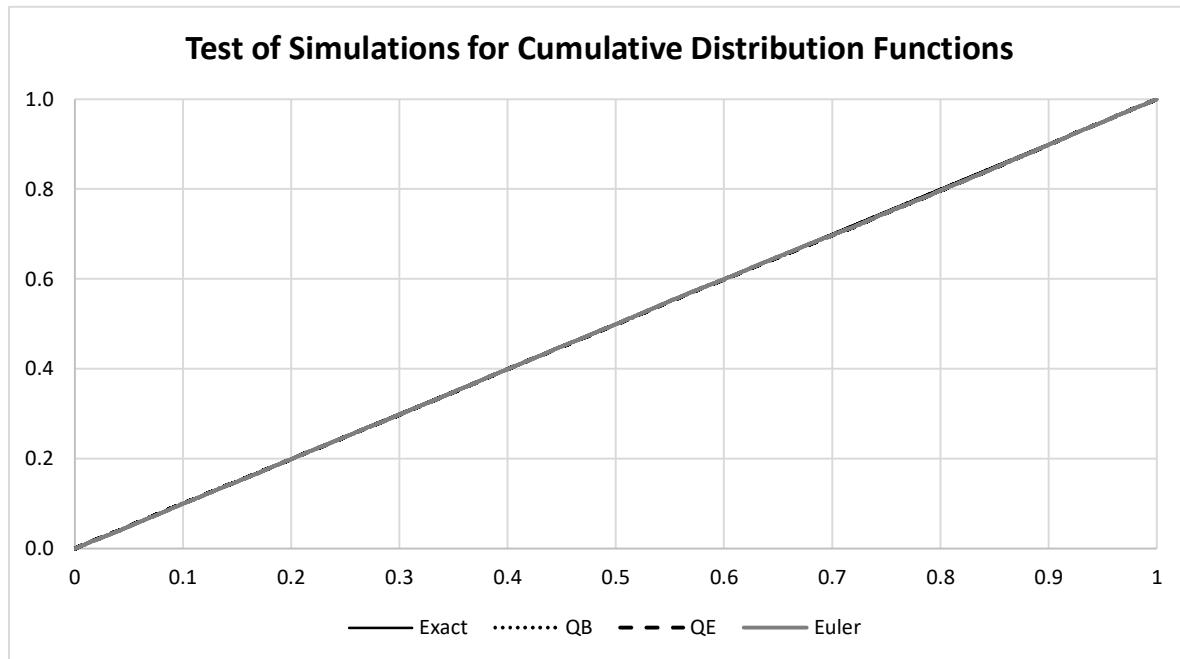
Figure 3. Simulated Distributions

$x(0) = 0.04, \sigma = 0.25, \nu = 0.64$

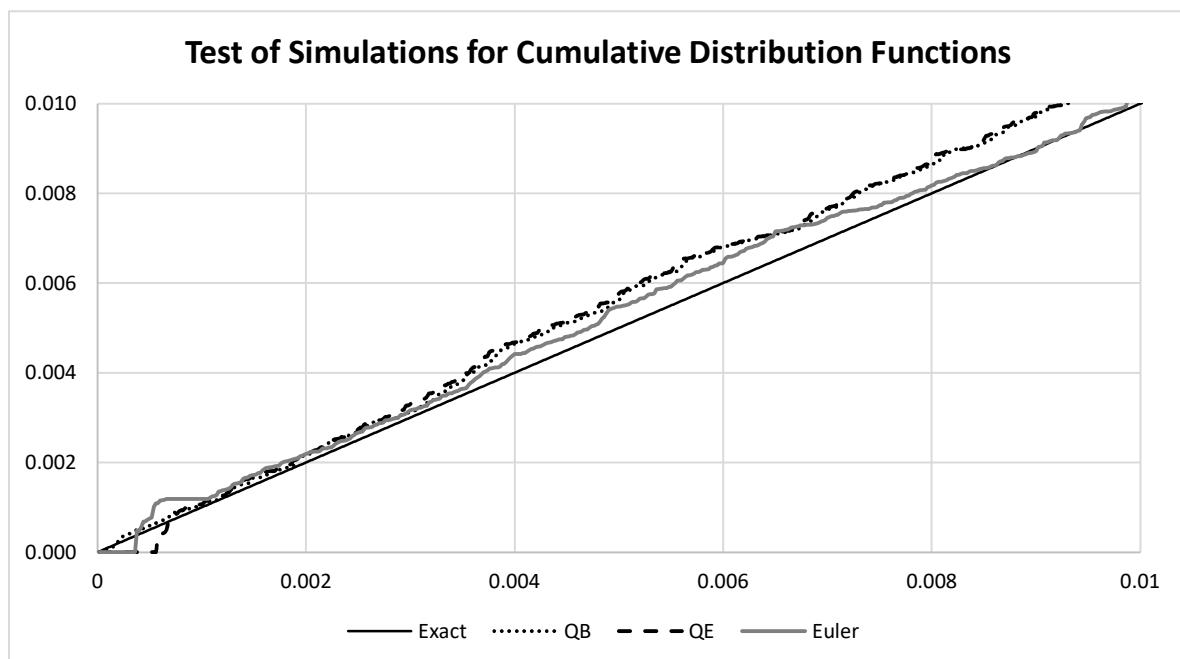


Lower Tail

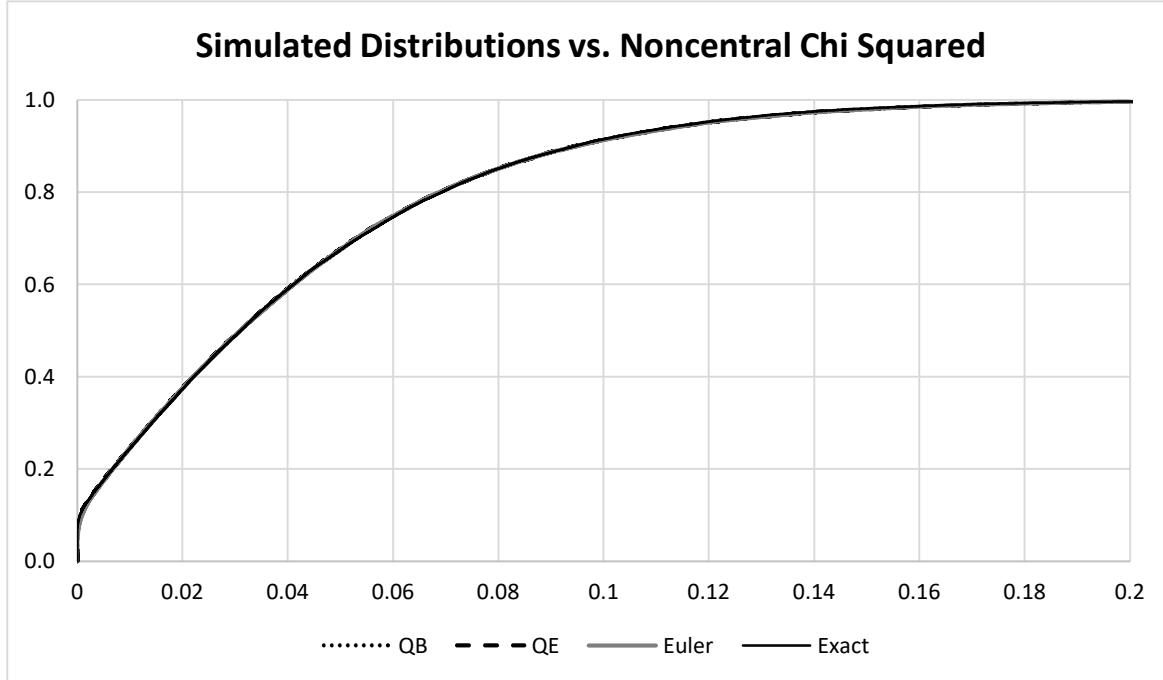




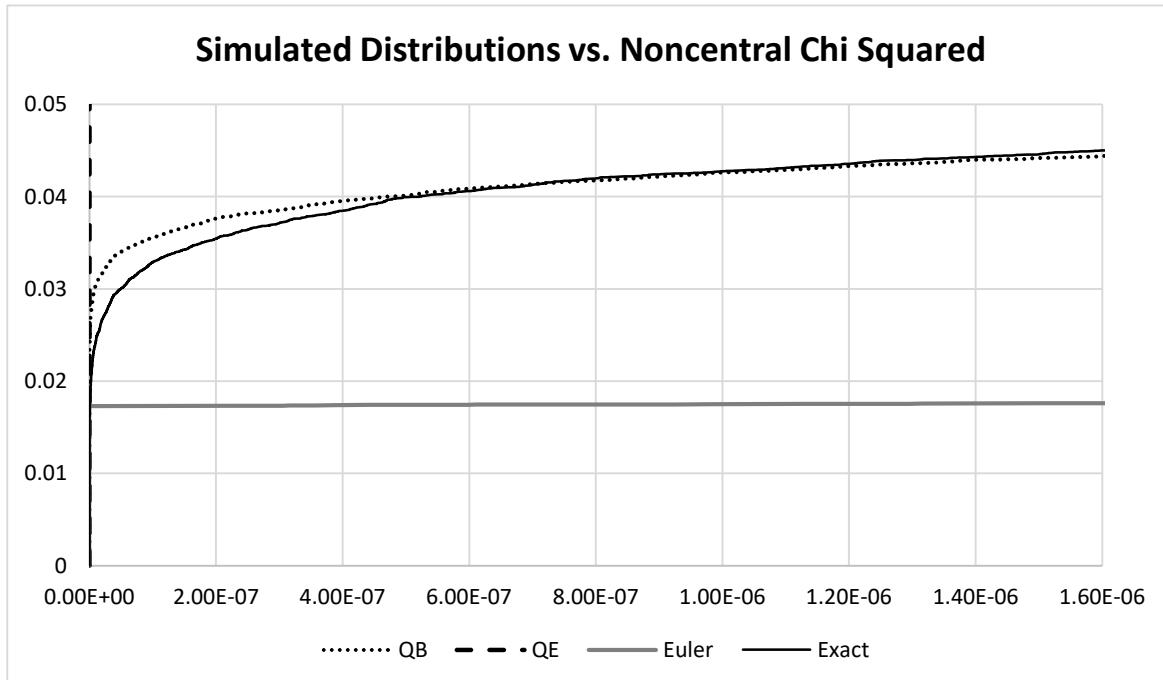
Lower Tail



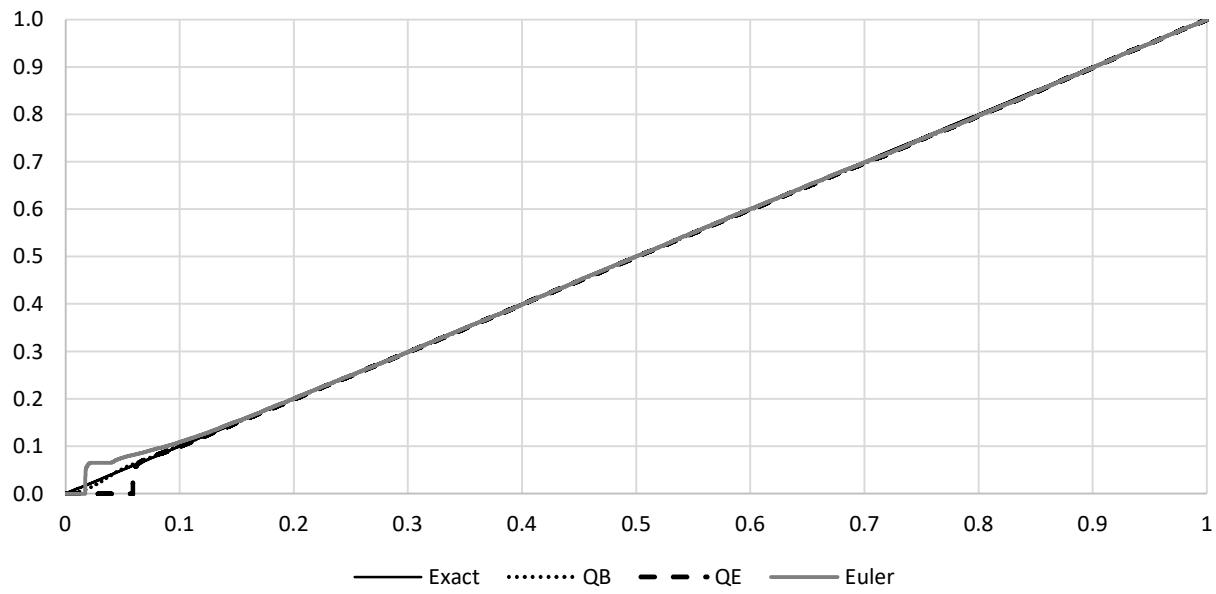
$$x(0) = 0.04, \sigma = 0.4, \nu = 0.25$$



Lower Tail



Test of Simulations for Cumulative Distribution Functions



Lower Tail

Test of Simulations for Cumulative Distribution Functions

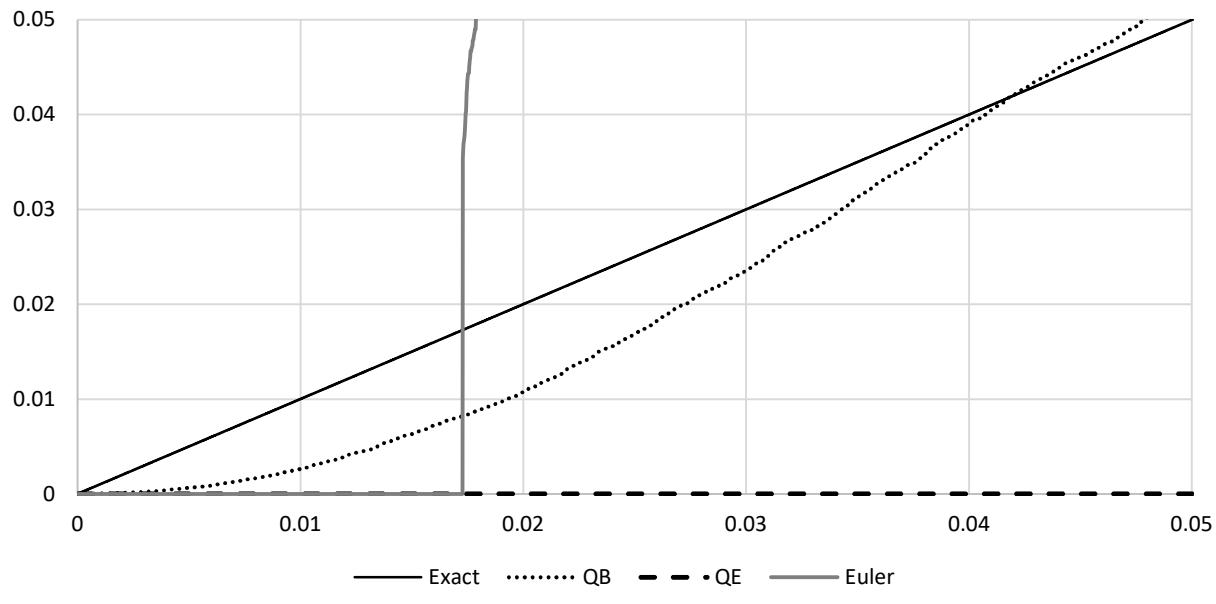


Table 3 contains additional tests of the convergence of simulated option prices in the Heston model. The Heston model is a stochastic volatility model with the variance process modeled as a square root diffusion.

$$d \log S = \left(r - \frac{1}{2} v \right) dt + \sqrt{v} dz_1$$

$$dv = (\kappa\theta - (\kappa + \lambda)v) dt + \sigma\sqrt{v} dz_2$$

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.

$$\log S(t) = \log S(0) + \int_0^t \left(r - \frac{1}{2} v(s) \right) ds + \int_0^t \sqrt{v(s)} \left(\sqrt{(1 - \rho^2)} dz_1(s) + \rho dz_2(s) \right)$$

The simulation for $v(t)$ is decomposed as follows.

$$v(t) = e^{-(\kappa+\lambda)t} v(0) + \kappa\theta \left(\frac{1 - e^{-(\kappa+\lambda)t}}{\kappa + \lambda} \right) + \sigma \int_0^t \sqrt{v(s)} dz_2(s)$$

and

$$\int_0^t \sqrt{v(s)} dz_2(s) = \frac{\left(v(t) - e^{-(\kappa+\lambda)t} v(0) - \kappa\theta \left(\frac{1 - e^{-(\kappa+\lambda)t}}{\kappa + \lambda} \right) \right)}{\sigma}$$

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

$$\log S(t) = \log S(0) + \left(r - \frac{1}{4} (v(0) + v(t)) \right) t + \int_0^t \rho \sqrt{v(s)} dz_2(s) + \sqrt{(1 - \rho^2)v(0)t} \tilde{Z}$$

where \tilde{Z} is a standard normal simulation.

The simulations in Table 3 are simulations of the 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. Again, the degrees of freedom are adjusted by resetting the σ parameter. The interest rate has been set at 3.0% per annum and the correlation parameter has been set at -0.90 to capture the negative correlation typically observed between volatility and stock prices. Again, 1 million simulation paths are run for a time period 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, v , 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, $\nu = 0.8889$, 0.5 , and 0.08 . Panel E of the table includes one example with a much smaller value for the rate of mean reversion, $\kappa = 0.5$ and $\lambda = -0.25$, and $\nu = 0.5$. The tests in the table show that both the QB and the QE methods perform well and generally converge as the degrees of freedom are reduced down to 0.08 . 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 QB 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 approximate 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 QB 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.

Clean up code for CPU/GPU and set up for Windows and Linux, include executables.

Store on GitHub, the CPU code only.

Table 3. Simulation Tests for Heston Model with $\kappa = 2.0, \theta = 0.04, \lambda = -0.50, \rho = -0.90$
using 1,000,000 simulation paths over 91 days with 2 time steps per day

Panel A. $x(0) = 0.04, \sigma = 0.4, \nu = 2.0$

		Heston	QB	t test	QE	t test	Euler	t test
Put	70	0.0301	0.0299	-0.36	0.0298	-0.44	0.0297	-0.62
Put	75	0.0828	0.0828	0.04	0.0827	-0.06	0.0826	-0.18
Put	80	0.2073	0.2078	0.32	0.2076	0.19	0.2077	0.22
Put	85	0.4761	0.4774	0.57	0.4771	0.42	0.4775	0.63
Put	90	1.0090	1.0112	0.65	1.0106	0.48	1.0121	0.94
Put	95	1.9823	1.9865	0.88	1.9857	0.71	1.9887	1.35
Put	100	3.6206	3.6258	0.82	3.6249	0.66	3.6294	1.37
Call	100	4.3658	4.3620	-0.75	4.3607	-1.00	4.3645	-0.25
Call	105	1.9369	1.9346	-0.71	1.9334	-1.07	1.9373	0.14
Call	110	0.5451	0.5447	-0.23	0.5440	-0.68	0.5463	0.79
Call	115	0.0660	0.0669	1.87	0.0667	1.57	0.0671	2.41
Call	120	0.00288	0.00289	0.16	0.00289	0.08	0.00292	0.46
Call	125	0.00007	0.00007	0.12	0.00007	0.10	0.00007	0.18

Panel B. $x(0) = 0.04, \sigma = 0.6, \nu = 0.8889$

		Heston	QB	t test	QE	t test	Euler	t test
Put	70	0.0639	0.0637	-0.32	0.0637	-0.31	0.0635	-0.56
Put	75	0.1429	0.1432	0.21	0.1432	0.22	0.1430	0.11
Put	80	0.2998	0.3003	0.26	0.3003	0.27	0.3005	0.38
Put	85	0.5945	0.5959	0.51	0.5960	0.52	0.5969	0.84
Put	90	1.1206	1.1227	0.53	1.1227	0.54	1.1253	1.21
Put	95	2.0170	2.0209	0.74	2.0210	0.77	2.0261	1.74
Put	100	3.4827	3.4872	0.66	3.4873	0.68	3.4964	2.02
Call	100	4.2278	4.2221	-1.32	4.2232	-1.08	4.1917	-8.39
Call	105	1.5922	1.5889	-1.32	1.5897	-0.99	1.5672	-10.11
Call	110	0.2490	0.2489	-0.18	0.2491	0.11	0.2445	-5.04
Call	115	0.0138	0.0140	1.13	0.0140	1.09	0.0140	0.79
Call	120	0.00057	0.00053	-1.09	0.00053	-1.12	0.00053	-0.91
Call	125	0.00002	0.00002	-0.18	0.00002	-0.15	0.00002	0.09

Panel C. $x(0) = 0.04, \sigma = 0.8, \nu = 0.50$

		Heston	QB	t test	QE	t test	Euler	t test
Put	70	0.1046	0.1044	-0.15	0.1046	0.04	0.1045	-0.10
Put	75	0.2049	0.2051	0.13	0.2054	0.33	0.2057	0.45
Put	80	0.3830	0.3835	0.21	0.3840	0.42	0.3851	0.91
Put	85	0.6871	0.6884	0.41	0.6891	0.63	0.6923	1.62
Put	90	1.1895	1.1913	0.43	1.1923	0.65	1.2002	2.48
Put	95	1.9983	2.0019	0.63	2.0027	0.77	2.0207	3.97
Put	100	3.2883	3.2931	0.66	3.2936	0.74	3.3369	6.80
Call	100	4.0335	4.0268	-1.78	4.0292	-1.13	3.8965	-37.98
Call	105	1.2265	1.2244	-1.04	1.2258	-0.35	1.1616	-33.59
Call	110	0.1130	0.1134	0.62	0.1135	0.88	0.1103	-4.38
Call	115	0.0069	0.0072	1.66	0.0072	1.56	0.0071	0.99
Call	120	0.00046	0.00050	0.83	0.00050	0.93	0.00051	1.04
Call	125	0.00003	0.00005	1.04	0.00005	1.03	0.00005	1.10

Panel D. $x(0) = 0.04, \sigma = 2.0, \nu = 0.08$

		Heston	QB	t test	QE	t test	Euler	t test
Put	70	0.2907	0.2904	-0.10	0.2911	0.16	0.3462	20.14
Put	75	0.4139	0.4139	0.00	0.4145	0.20	0.5041	26.39
Put	80	0.5784	0.5784	0.02	0.5790	0.16	0.7222	34.34
Put	85	0.7966	0.7972	0.13	0.7973	0.16	1.0244	44.82
Put	90	1.0890	1.0899	0.17	1.0900	0.19	1.4519	59.31
Put	95	1.4989	1.5011	0.33	1.5006	0.26	2.0899	80.62
Put	100	2.1743	2.1814	0.95	2.1772	0.38	3.2082	117.91
Call	100	2.9194	2.9146	-2.02	2.9186	-0.34	2.4764	-202.14
Call	105	0.2656	0.2694	3.36	0.2678	1.98	0.3334	53.34
Call	110	0.0386	0.0387	0.17	0.0387	0.26	0.0470	15.29
Call	115	0.0087	0.0088	0.39	0.0088	0.56	0.0099	4.51
Call	120	0.00225	0.00222	-0.22	0.00229	0.26	0.00250	1.70
Call	125	0.00063	0.00061	-0.25	0.00067	0.48	0.00076	1.49
Call	130	0.00019	0.00019	0.12	0.00023	0.85	0.00027	1.60

Panel E. with $\kappa = 0.5, \theta = 0.04, \lambda = -0.25, \rho = -0.90, x(0) = 0.04, \sigma = 0.4, \nu = 0.5$

		Heston	QB	t test	QE	t test	Euler	t test
Put	70	0.0343	0.0342	-0.18	0.0342	-0.14	0.0339	-0.72
Put	75	0.0907	0.0910	0.32	0.0910	0.37	0.0904	-0.23
Put	80	0.2194	0.2203	0.58	0.2204	0.65	0.2197	0.15
Put	85	0.4901	0.4920	0.83	0.4922	0.91	0.4915	0.58
Put	90	1.0162	1.0192	0.87	1.0195	0.95	1.0194	0.91
Put	95	1.9656	1.9709	1.09	1.9713	1.17	1.9724	1.41
Put	100	3.5571	3.5633	0.96	3.5638	1.04	3.5669	1.52
Call	100	4.3022	4.2996	-0.56	4.3003	-0.41	4.2925	-2.04
Call	105	1.8151	1.8139	-0.40	1.8145	-0.19	1.8083	-2.32
Call	110	0.4181	0.4186	0.41	0.4190	0.69	0.4159	-1.78
Call	115	0.0344	0.0351	2.03	0.0351	2.07	0.0350	1.56
Call	120	0.00140	0.00136	-0.60	0.00137	-0.49	0.00138	-0.30
Call	125	0.00004	0.00004	-0.80	0.00004	-0.76	0.00004	-0.63

Table 4. Simulation Tests for the Discount in the CIR Model

Parameters: $\kappa = 0.5, \theta = 0.04, \lambda = -0.25, x(0) = 0.04$

1,000,000 simulation paths with 1 time step per day

Panel A. $\sigma = 0.20, \nu = 2.0$

Years	Analytic	QB	t test	QE	t test	Euler	t test
1	0.95659608	0.95661204	0.75	0.95661197	0.75	0.95661074	0.69
5	0.76412413	0.76399396	-1.02	0.76399060	-1.04	0.76396190	-1.27

Panel B. $\sigma = 0.40, \nu = 0.50$

Years	Analytic	QB	t test	QE	t test	Euler	t test
1	0.95724993	0.95730417	1.32	0.95730082	1.24	0.95692635	-10.48
5	0.79221829	0.79205712	-0.82	0.79201512	-1.03	0.78124234	-74.92

4. Summary

There are several alternative methods for simulating square root processes. One can always simulate a square root process using the simulation method for a non-central chi squared variate, but this method is not well suited for parameter calibration or parallel processing and it is much slower than alternative methods. The exact simulation of the non-central chi squared requires acceptance-rejection methods in most cases so that the number of simulations over each time step vary and it is difficult to control the simulation with typical RNG's. Several alternative methods, namely the QE and QB methods, produce good approximations for the simulation over small time steps and these methods can be easily applied for both parallel processing and parameter calibration. The various tests in section 3 reveal significant biases if the Euler method is used when the degrees of freedom parameter is less than one. If the degrees of freedom parameter is greater than or equal to one, one should use the quadratic method, which is embedded in the QE, QB, or implicit Milstein methods. In these cases, the quadratic method is almost as fast as the Euler method and it meets the requirement that it never produces negative values. When the degrees of freedom parameter is less than one, the QB and QE methods produce simulations that approximate the square root process over small time intervals and converge for pricing applications. It has been noted that the QE method does produce simulations with zero values, whereas the QB method and the exact simulation method do not. This property leads to larger approximations errors for the QE method in the lower tail of the distribution, but the QE method still performs well for pricing applications. In those cases where the QE method produces simulations at zero, the QB and the exact method generate simulations close to zero. The zero simulations lead to failures of the Anderson-Darling test for the QE method, but this is not necessarily an issue for many of the potential applications, as the differences are essentially the differences between 0 and a small number like 10^{-8} .

References

- 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, "Simulation of Square-Root Processes," chapter in *Encyclopedia of Quantitative Finance*, May 2010, available online at
<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.
- Boost C++ Library: <https://www.boost.org/>
- Bradley, T., J. du Toit, M. Giles, R. Tong, and P. Woodhams, "Parallelisation Techniques for Random Number Generators," available online at
https://www.nag.com/IndustryArticles/gpu_gems_article.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)*, Vol. 28, No. 3 (1979): 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
- Cramer, von Mises
- 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 2008: 32-36.
- 1094-6969/08/\$25.00 ©2008 IEEE
- Feller, W. (1950), "Two Singular Diffusion Problems*," *Annals of Mathematics* 54 (1): 173-182.

- 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): 37-0.
- L'Ecuyer, P. (1999), "Good Parameters and Implementations for Combined Multiple Recursive Random Number Generators," *Operations Research* 47(1):159-164.
- L'Ecuyer, P., and R. Simard, "TestU01: A C library for empirical testing of random number generators," *ACM Trans. Math. Software* 33, August 2007.
- Salmon, J. K., M. A. Moraes, R. O. Dror, and D. E. Shaw, "Parallel Random Numbers: As Easy as 1, 2, 3," Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC11), New York, NY: ACM, 2011, available online at
<http://www.thesalmons.org/john/random123/papers/random123sc11.pdf>.
- Simard, R. and P. L'Ecuyer (2011), "Computing the Two-Sided Kolmogorov-Smirnov Distribution," *Journal of Statistical Software* 39 (11): 1-18.
- Wallace, N.D. (1974), "Computer Generation of Gamma Random Variates with Non-integral Shape Parameters," *Communications of the Association for Computing Machinery* 17 (12), December 1974
- 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.

Web Link to GitHub for C++ and Cuda C code (Windows & Linux)