Management Science and Financial Engineering
Vol 20, No 1, May 2014, pp.27-32
ISSN 2287-2043 │ EISSN 2287-2361 │
http://dx.doi.org/10.7737/MSFE.2014.20.1.027

Simulation of the Shifted Poisson Distribution with an Application to the CEV Model

Chulmin Kang*
Center for Applications of Mathematical Principles, National Institute for Mathematical Sciences

(Received: April 30, 2014 / Revised: May 19, 2014 / Accepted: May 22, 2014)

ABSTRACT

This paper introduces three different simulation algorithms of the shifted Poisson distribution. The first algorithm is the inverse transform method, the second is the rejection sampling, and the third is gamma-Poisson hierarchy sampling. Three algorithms have different regions of parameters at which they are efficient. We numerically compare those algorithms with different sets of parameters. As an application, we give a simulation method of the constant elasticity of variance model.

Keywords: Simulation, Shifted Poisson Distributions, Inverse Transform Methods, Rejection Sampling

* Corresponding Author, E-mail: ckang@nims.re.kr

1. INTRODUCTION

A non-negative integer valued random variable N is said to be a shifted Poisson random variable with the shift parameter $v \geq 0$ and the rate parameter $\lambda > 0$ if

$$p_n = \mathbf{P}(N = n) = \frac{\Gamma(v)}{\gamma(v;\lambda)} \frac{\lambda^{v+n}}{\Gamma(v+n+1)} e^{-\lambda}$$

for $n = 0, 1, 2, \cdots$, where $\gamma(v, \lambda)$ and $\Gamma(v)$ are the incomeplete gamma function and the (complete) gamma function defined by

$$\gamma(v;\lambda) = \int_0^{\lambda} t^{v-1} e^{-t} dt,$$

$$\Gamma(v) = \gamma(v;\infty) = \int_0^{\infty} t^{v-1} e^{-t} dt.$$

We shall write $N \sim \mathrm{SP}(v;\lambda)$ if N has a shifted Poisson distribution with parameters v and λ. We recall that $\lim_{v \to 0} \Gamma(v)/\gamma(v;\lambda) = 1$. In particular, $\mathrm{SP}(0;\lambda) \equiv \mathrm{Poisson}(\lambda)$. In this sense, the shifted Poisson distribution is a generalization of the Poisson distribution. On the other hand, we can get the shifted Poisson distribution by *shifting* the Poisson distribution (or conditioning on a large outcome, see Lemma 2.4).

The shifted Poisson distribution arises naturally in the study of several diffusion processes including the squared Bessel processes and the constant elasticity of variance (henceforth, CEV) processes, see (Göing-Jaeschke and Yor, 2003) and (Makarov and Glew, 2010). Particularly, simulation methods, developed in this paper, can be used in the generation of sample paths of CEV processes. Therefore, our methods would be building blocks for Monte Carlo pricing under the CEV model.

This paper introduces three different simulation methods of shifted Poisson distributions. The first algorithm is the inverse transform method, the second is the rejection sampling, and the third is gamma-Poisson hierarchy sampling. We compare them from a computational point of view by analyzing the expected number of iterations, and we perform numerical experiments with different sets of parameters. As a result, we find that three algorithms have different regions of parameters at which they are efficient.

The rest of this paper is organized as follows. Section 2 studies basic properties of the shifted Poisson distribution. Section 3 introduces three simulation methods for the distribution. Section 4 presents an application to CEV model. Section 5 provides numerical results. Section 6 concludes this paper.

2. BASIC PROPERTIES

In this section, we investigate some basic properties of the shifted Poisson distribution. First, we recall the holomorphic expansion of the incomplete gamma function, e.g. see (Abramowitz and Stegun, 1964)

$$\gamma(v;\lambda)=\lambda^{v}\Gamma(v)e^{-\lambda}\sum_{n=0}^{\infty}\frac{\lambda^{n}}{\Gamma(v+n+1)}.$$

Using the above representation, we can find the moment generating function of the shifted Poisson distribution.

Lemma 2.1: Let $N\sim\mathrm{SP}(v;\lambda)$. The moment generating function $m(s)$ of N is given by

$$\begin{aligned}m(s)&=\mathrm{E}[e^{sN}]\\&=\frac{\gamma(v;\lambda e^{s})}{\gamma(v;\lambda)}\exp(\lambda(e^{s}-1)-vs).\end{aligned}$$

Proof: We use the holomorphic expansion to have

$$\begin{aligned}m(s)&=\sum_{n=0}^{\infty}e^{sn}p_n\\&=\frac{\Gamma(v)}{\gamma(v;\lambda)}\lambda^{v}e^{-\lambda}\sum_{n=0}^{\infty}\frac{(\lambda e^{s})^{n}}{\Gamma(v+n+1)}\\&=\frac{\gamma(v;\lambda e^{s})}{\gamma(v;\lambda)}\lambda^{v}e^{-\lambda}(\lambda e^{s})^{-v}\exp(\lambda e^{s})\\&=\frac{\gamma(v;\lambda e^{s})}{\gamma(v;\lambda)}\exp(\lambda(e^{s}-1)-vs).\end{aligned}$$

□

Then we find the mean, variance, and mode of the shifted Poisson distribution.

Lemma 2.2: Let $N\sim\mathrm{SP}(v;\lambda)$. The mean and variance of N are given by

$$\begin{aligned}\mathrm{E}[N]&=\lambda-v(1-p_0)\\&=\lambda-\frac{\gamma(v+1;\lambda)}{\gamma(v;\lambda)},\ \text{ and}\\\mathbf{V}(N)&=\lambda+v^{2}p_0(1-p_0)+v(v+1)p_1-2v\lambda p_0.\end{aligned}$$

The random variable N has a mode at $\max(\lfloor\lambda-v\rfloor,0)$ if $\lambda-v\notin N$, and it has modes at $\max(\lambda-v,0)$ and $\max(\lambda-v-1,0)$ if $\lambda-v\in N$.

Proof: Notice that $(v+n)p_n=\lambda p_{n-1}$ for all $n=1,2,\cdots$. This implies that

$$\begin{aligned}\mathrm{E}[v+N]&=vp_0+\sum_{n=1}^{\infty}(v+n)p_n\\&=vp_0+\lambda\sum_{n=0}^{\infty}p_n\\&=\lambda+vp_0.\end{aligned}$$

That is, $\mathrm{E}[N]=\lambda-v(1-p_0)$. Using integration by parts, we can easily see that

$$\gamma(v+1;\lambda)=v\gamma(v;\lambda)-\lambda^{v}e^{-\lambda}.$$

It follows that

$$\begin{aligned}v(1-p_0)&=\frac{v\gamma(v;\lambda)-\lambda^{v}e^{-\lambda}}{\gamma(v;\lambda)}\\&=\frac{\gamma(v+1;\lambda)}{\gamma(v;\lambda)}.\end{aligned}$$

Similarly, $(v+n)(v+n-1)p_n=\lambda^{2}p_{n-2}$ for all $n=2,3,\cdots$. So, we have

$$\begin{aligned}&\mathrm{E}[(v+N)(v+N-1)]\\&\quad=v(v-1)p_0+v(v+1)p_1\\&\qquad+\sum_{n=2}^{\infty}(v+n)(v+n-1)p_n\\&\quad=v(v-1)p_0+v(v+1)p_1+\lambda^{2}.\end{aligned}$$

Therefore, the variance is

$$\begin{aligned}\mathbf{V}(N)&=\mathbf{V}(v+N)\\&=\mathrm{E}[(v+N)^{2}]-(\mathrm{E}[v+N])^{2}\\&=(v^{2}p_0+v(v+1)p_1+\lambda^{2}+\lambda)\\&\quad-(v^{2}p_0^{2}+2\lambda vp_0+\lambda^{2})\\&=\lambda+v^{2}p_0(1-p_0)+v(v+1)p_1-2v\lambda p_0.\end{aligned}$$

From the recursive relation $p_n=\dfrac{\lambda}{v+n}p_{n-1}$, we can easily see that p_n decreases if $\lambda-v\le n$, and it increases otherwise. □

In particular, the shifted Poisson distribution has a unique mode at 0 if $v\ge\lambda$.

Corollary 2.3: We denote by $\mathrm{E}[N|v,\lambda]$ the expectation of $N\sim\mathrm{SP}(v;\lambda)$. Then

$$\lim_{v\to\infty}\mathrm{E}[N|v,\lambda]=0,$$

for all $\lambda>0$.

Proof: We recall a series expansion of the incomplete gamma function, e.g. see (6.5.33) of (Abramowitz and Stegun, 1964)

$$\gamma(v;\lambda)=\sum_{n=0}^{\infty}\frac{(-1)^{n}\lambda^{v+n}}{(v+n)n!}.$$

It follows that

$$\frac{\gamma(v+1;\lambda)}{\gamma(v;\lambda)} = \lambda\frac{\sum_{n=0}^{\infty}\frac{(-1)^n\lambda^{v+n}}{(v+1+n)n!}}{\sum_{n=0}^{\infty}\frac{(-1)^n\lambda^{v+n}}{(v+n)n!}}$$

$$= \lambda\frac{\sum_{n=0}^{\infty}\frac{v}{v+1+n}\frac{(-1)^n\lambda^{v+n}}{n!}}{\sum_{n=0}^{\infty}\frac{v}{v+n}\frac{(-1)^n\lambda^{v+n}}{n!}}.$$

By the dominated convergence theorem, we have $\lim_{v\to\infty}\gamma(v+1;\lambda)/(v;\lambda)=\lambda$. Hence, by Lemma 2.2, the mean converges to zero as v tends to infinity. □

The next lemma explains why we call it the *shifted* Poisson distribution.

Lemma 2.4: Let $N \sim \mathrm{SP}(v;\lambda)$. Then

$$(N-m\,|\,N\geq m)\sim \mathrm{SP}(v+m;\lambda),$$

for all nonnegative integer m.

Proof: Since the normalizing constant is uniquely determined, it suffices to observe that

$$\mathbf{P}(N-m=n\,|\,N\geq m)\sim\frac{\lambda^{v+m+n}}{\Gamma(v+m+n+1)},$$

for all $n=0,1,2,\cdots$. □

The next lemma gives an upper bound which will be used in our acceptance-rejection algorithm.

Lemma 2.5: Let p_n and q_n be the probability mass function of $\mathrm{SP}(v;\lambda)$ and $\mathrm{Poisson}(\lambda)$, respectively. Then

$$\frac{p_n}{q_n}\leq\frac{\lambda^v}{v\gamma(v;\lambda)},$$

for all $n=0,1,2,\cdots$.

Proof: Firstly, we observe that

$$\frac{\Gamma(n+1)}{\Gamma(v+n+1)}=\frac{1}{\Gamma(v+1)}\prod_{k=1}^{n}\frac{k}{v+k}$$

$$\leq\frac{1}{\Gamma(v+1)},$$

for all $n=0,1,2,\cdots$. So we have

$$\frac{p_n}{q_n}=\frac{\Gamma(v)}{\gamma(v;\lambda)}\lambda^v\frac{\Gamma(n+1)}{\Gamma(v+n+1)}$$

$$\leq\frac{\Gamma(v)}{\gamma(v;\lambda)}\lambda^v\frac{1}{\Gamma(v+1)}$$

$$=\frac{\lambda^v}{v\gamma(v;\lambda)}.$$

□

We give another construction of the shifted Poisson distribution using a gamma random variable and a Poisson random variable. This lemma will be used to develop an algorithm for generating a shifted Poisson random number, namely Algorithm 3 in Section 3.

Lemma 2.6: Let X be a gamma random variable with shape parameter v and scale parameter 1 conditional on $X\leq\lambda$. That is,

$$\mathbf{P}(X\leq x)=\frac{\int_0^x t^{v-1}e^{-t}dt}{\int_0^\lambda t^{v-1}e^{-t}dt}=\frac{\gamma(v;x)}{\gamma(v;x)},$$

for $0\leq \mathrm{x}\leq\lambda$. If $N\sim\mathrm{Posson}(\lambda-X)$, then the distribution of N is $\mathrm{SP}(v;\lambda)$.

Proof: Observe that

$$\begin{aligned}
&\mathbf{P}(N=n)\\
&=\mathbf{E}[\mathbf{P}(N=n\,|\,X)]\\
&=\frac{1}{\gamma(v;\lambda)}\int_0^\lambda\frac{(\lambda-t)^n}{n!}e^{-(\lambda-t)}t^{v-1}e^{-t}dt\\
&=\frac{e^{-\lambda}}{n!\gamma(v;\lambda)}\int_0^\lambda(\lambda-t)^n t^{v-1}e^{-t}dt\\
&=\frac{\lambda^{v+n}e^{-\lambda}}{n!\gamma(v;\lambda)}\int_0^1(1-u)^n u^{v-1}du\\
&=\frac{\lambda^{v+n}e^{-\lambda}}{n!\gamma(v;\lambda)}B(n+1,v),
\end{aligned}$$

where $B(\cdot,\cdot)$ is the beta function. Recall that

$$B(n+1,v)=\frac{\Gamma(n+1)\Gamma(v)}{\Gamma(v+n+1)}=\frac{n!\Gamma(v)}{\Gamma(v+n+1)}.$$

Hence, we have

$$\mathbf{P}(N=n)=\frac{\Gamma(v)}{\gamma(v;\lambda)}\frac{\lambda^{v+n}}{\Gamma(v+n+1)}e^{-\lambda}=p_n,$$

for all $n=0,1,2,\cdots$. □

3. SIMULATION METHODS

We shall introduce three simulation methods of the shifted Poisson distribution.

The first algorithm is the inverse transform method. We use the following recursion formula

$$p_0 = \frac{\lambda^v}{v\gamma(v;\lambda)} e^{-\lambda}, \quad \text{and}$$
$$p_n = \frac{\lambda}{v+n} p_{n-1}, \quad n = 1, 2, \cdots.$$

Data: $v \ge 0, \lambda > 0$
Result: $N \sim \mathrm{SP}(v;\lambda)$
Generate $U \sim \mathrm{unif}(0,1)$;
$$p \leftarrow \frac{\lambda^v}{v\lambda(v;\lambda)} e^{-\lambda}, \ s \leftarrow p, \ n \leftarrow 0;$$
while $U \ge s$ **do**
  $n \leftarrow n+1, \ \ p \leftarrow \frac{\lambda}{v+n} p, \ \ s \leftarrow s+p;$
end
  return *n*

Algorithm 1. Inverse transform method

Algorithm 1 requires a single computation of the incomplete gamma function. We use the algorithm of (Press *et al*., 2007) in our numerical experiments. The expected number of iterations of Algorithm 1 is $\mathbf{E}[N] = \lambda - \frac{\gamma(v+1;\lambda)}{\gamma(v;\lambda)}$. Therefore, by Corollary 2.3, it is efficient for large *v*.

Our second algorithm is a rejection sampling using Lemma 2.5. We shall write

$$c_0 = 0, \quad \text{and}$$
$$c_n = -\log\left(\frac{v\gamma(v;\lambda)}{\lambda^v} \frac{p_n}{q_n}\right) = \sum_{k=1}^{n} \log\left(1 + \frac{v}{k}\right), \quad n = 1, 2, \cdots.$$

Data: $v \ge 0, \lambda > 0$
Result: $N \sim \mathrm{SP}(v;\lambda)$
reject $\leftarrow$ true;
while reject **do**
  Generate $E \sim \exp(1)$;
  Generate $N \sim \mathrm{Poisson}(\lambda)$;
 If $E > c_N$ **then**
  reject $\leftarrow$ false;
 end
end
return *N*

Algorithm 2. Rejection sampling

Algorithm 2 avoids any computations of incomplete or complete gamma functions. But it requires generations of Poisson random numbers. There are many well-known algorithms for generating Poisson random numbers. Among them, the algorithm of (Devorye, 1986) is a uniformly fast for all $\lambda > 0$. In our numerical experiments, we used his algorithm for generating Poisson random numbers. The expected number of iterations is the upper bound for p_n / q_n, i.e. $\lambda^v/(v\gamma(v;\lambda))$. Notice that this upper bound grows very rapidly as λ increases for $v \gg 1$. Therefore, we recommend the reader to use Algorithm 2 for only very small values of λ for $v \gg 1$.

Our third algorithm is based on Lemma 2.6. We first generate a gamma random variable and then we generate a Poisson random variable conditional on the gamma random variable.

Algorithm 3 requires generations of gamma random variables and a Poisson random variable. We use Algorithm GKM1 if $v > 1$ and Algorithm GS* if $0 < v \le 1$ from (Fishman, 1996) for generating gamma random variables. The expected number of iterations is $1/p$ where $p = \mathbf{P}(X \le \lambda)$. Therefore, this algorithm is efficient when λ is sufficiently larger than *v*.

Data: $v \ge 0, \lambda > 0$
Result: $N \sim \mathrm{SP}(v;\lambda)$
  reject $\leftarrow$ true;
while reject **do**
  Generate $X \sim \Gamma(v, 1)$;
 If $X \le \lambda$ **then**
  reject $\leftarrow$ false;
 end
end
  Generate $N \sim \mathrm{Poisson}(\lambda - X)$;
return *N*

Algorithm 3. Gamma-Poisson hierarchy sampling

4. CEV MODEL

Several authors studied the exact simulation of the constant elasticity of variance model. For instance, (Makarov and Glew, 2010) provided a mixture sampling method with a shifted Poisson mixing variable, and (Lindsay and Brecher, 2012) suggested an inverse transform method based on a root-finding algorithm. In this section, we review the exact simulation method of (Makarov and Glew, 2010).

The CEV model is an option pricing model which generalizes the celebrated Black-Scholes model. The stock price is described by the following stochastic differential equation

$$dS_t = rS_t dt + \sigma S_t^{\beta/2} dW_t, \ S_0 = s,$$

where $r > 0$ is the instantaneous interest rate and $\beta \in (0, 2)$ is the elasticity of variance. This model is in-troduced by (Cox and Ross, 1976).

According to (Delbaen and Shirakawa, 2002), S_t can be represented as follows

$$S_t = e^{rt}(X_{\tau_t \wedge \zeta})^{1/(2-\beta)} \text{ in distribution,}$$

where X_t is the solution of

$$dX_t = \delta dt + 2\sqrt{|X_t|}\, dW_t,\ X_0 = x,$$

with $\delta = (2-2\beta)/(2-\beta),\ x = s^{2-\beta}$,

$$\tau_t = \frac{\sigma^2}{2r(2-\delta)}\left(1-\exp\left\{-\frac{2rt}{2-\delta}\right\}\right), \text{ and}$$
$$\zeta = \inf\{t \ge 0;\ X_t = 0\}.$$

The distribution of X_t is well-known in the literature, e.g. see (Göing-Jaeschke and Yor, 2003). For $\delta > 0$ (or $\beta \in (0, 1)$), X_t has a scaled non-central chi square distribution. Now, we consider the case $\delta < 0$ (or $\beta \in (1, 2)$). We shall write $v = 1 - \delta/2$. In this case,

$$\mathbf{P}(\zeta > t) = \frac{\gamma(v; x/2t)}{\Gamma(v)},\ t > 0,$$

and the transition density of X_t conditional on $\zeta > t$ is given by

$$\begin{aligned} q_t(x, y) &= \frac{\Gamma(v)}{\gamma\left(v; \dfrac{x}{2t}\right)} \\ &\times \frac{1}{2t}\left(\frac{x}{y}\right)^{\frac{v}{2}} \exp\left(-\frac{x+y}{2t}\right) I_v\left(\frac{\sqrt{xy}}{t}\right) \\ &= \frac{\Gamma(v)}{\gamma(v; x/2t)} \\ &\times \sum_{n=0}^{\infty} \frac{(x/2t)^{v+n} e^{-x/2t}}{\Gamma(v+n+1)} \frac{y^n e^{-y/2t}}{n!(2t)^{n+1}}. \end{aligned}$$

Therefore, $(X_t | \zeta > t)$ has a gamma mixture distribution with a shifted Poisson mixing variable. We summarize these observations in Algorithm 4.

Data: $s > 0, t \ge 0, r > 0, \sigma > 0, \beta \in (1, 2)$
Result: S_t from the CEV model
$$\delta \leftarrow \frac{2-2\beta}{2-\beta},\ v \leftarrow 1-\delta/2,\ x \leftarrow s^{2-\beta};$$
$$\tau \leftarrow \frac{\sigma^2}{2r(2-\delta)}\left(1-\exp\left\{-\frac{2rt}{2-\delta}\right\}\right);$$
Generate $G \sim \Gamma(v, 1)$;
if $G \sim x/2\tau$ **then**
  return $S \leftarrow 0$
end
  Generate $N \sim \text{SP}(v; x/2\tau)$;
  Generate $X \sim \Gamma(N+1, 1)$;
return $S \leftarrow e^{rt}(2\tau X)^{1/(2-\beta)}$

Algorithm 4. Simulation of the CEV model

5. NUMERICAL RESULTS

In Section 3, we introduced three different simulation methods and we gave the expected number of iterations for those methods. This section is devoted to numerical comparison of those algorithms.

Table 1 shows the elapsed time in seconds to generate 10^5 samples from $\text{SP}(v=1; \lambda)$ using three different algorithms. We find that Algorithm 2 is very inefficient for large values of λ. As expected, the elapsed time for Algorithm 1 increases as λ increases. Algorithm 3 is almost uniformly fast for those values of λ. The reason is that the expected number of iterations is $1/p$, with $p = \mathbf{P}(X \le \lambda)$, which is almost the same as 1.

Table 1. Elapsed time in seconds to generate 10^5 samples from $\text{SP}(v=1; \lambda)$

Algorithm	λ			
	1	10	100	500
1	0.0321	0.0263	0.0667	0.2547
2	0.0170	0.4170	19.0477	412.2158
3	0.0139	0.0288	0.0384	0.0370

Table 2 shows the elapsed time in seconds to generate 10^5 samples from $\text{SP}(v; \lambda = 0.1)$ using three different algorithms. In this case, Algorithm 2 is almost uniformly fast for v. The reason is that the expected number of iteration $\lambda^v/(v\gamma(v;\lambda))$ is almost 1. On the other hand, Algorithm 3 failed to generate any sample for $v \ge 10$. The acceptance probability $\mathbf{P}(X \le 0.1)$ is about 2.52×10^{-17} for $v = 10$.

Table 2. Elapsed time in seconds to generate 10^5 samples from $\text{SP}(v; \lambda = 0.1)$

Algorithm	v			
	1	10	100	500
1	0.0285	0.0252	0.0241	0.0305
2	0.0058	0.0057	0.0057	0.0057
3	0.0715	∞	∞	∞

We close this section with an application to the discrete monitoring asian call option pricing under the CEV model. The payoff is given by

$$V = \left(\frac{1}{n+1}\sum_{j=0}^{n} S_{t_j} - K\right)^+,$$

where $0 \le t_0 < \cdots < t_n \le T$ are monitoring dates. Then the risk neutral price is $\Pi = \mathbf{E}[e^{-rT}V]$. We can estimate the price using the Monte-Carlo method, i.e.

$$\Pi \approx \frac{1}{N}\sum_{i=1}^{N} V_i,$$

where $V_1, \cdots, V_n$ are independent samples of the payoff.

Table 3 shows Monte-Carlo estimates of the option under the CEV model. We found that prices of out-of-the-money (in-the-money) options under the CEV model are higher (lower) than those of the Black-Scholes model.

Table 3. Monte-Carlo estimates of discrete monitoring asian call option prices under the CEV model; parameters are set to $r = 0.05$, $\sigma = 0.05$, $T = 1$, $K = 1$, $t_j = j/10$, $j = 0, \cdots, 10$; 10^6 independent samples are used to estimate.

β	S_0				
	0.6	0.8	1.0	1.2	1.4
1.2	0.0071	0.0394	0.1207	0.2544	0.4238
1.4	0.0065	0.0388	0.1210	0.2551	0.4247
1.6	0.0059	0.0380	0.1207	0.2560	0.4262
1.8	0.0054	0.0374	0.1210	0.2568	0.4269
BS	0.0049	0.0368	0.1201	0.2569	0.4266

6. CONCLUDING REMARKS

We introduced three different algorithms for generating shifted Poisson random variables. The first is the inverse transform, the second is the rejection sampling, and the third is gamma-Poisson hierarchy sampling. We provided the expected number of iterations for each method, and conducted numerical experiments to compare efficiency of three algorithms. We recommend the reader to use Algorithm 2 for small λ, say $\lambda \le 1$, and Algorithm 3 for large λ, say $\lambda \ge \nu$. Otherwise, the reader may use Algorithm 1.

As an application, we gave a simulation method for the CEV model with an example of discrete monitoring asian call option pricing. Numerical experiments revealed that the prices of out-of-the-money (in-the-money) options under the CEV model are higher (lower) than that of the Black-Scholes model.

REFERENCES

Abramowitz, M. and I. A. Stegun, *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, U.S. Government Printing Office, Washington, D.C., 1964.

Cox, J. C. and S. A. Ross, "The valuation of options for alternative stochastic processes," *Journal of Financial Economics* 3, 1/2 (1976), 145-166.

Delbaen, F. and H. Shirakawa, "A note on option pricing for the constant elasticity of variance model," *Asia-Pacific Financial Markets* 9, 2 (2002), 85-99.

Devroye, L., *Non-uniform random variate generation*, Springer-Verlag, New York, 1986.

Fishman, G. S., *Monte Carlo: concepts, algorithms, and applications*, Springer Series on Operations Research, Springer-Verlag, New York, 1996.

Göing-Jaeschke, A. and M. Yor, "A survey and some generalizations of bessel processes," *Bernoulli* 9, 2 (2003), 313-349.

Lindsay, A. and D. Brecher, "Simulation of the CEV process and the local martingale property," *Mathematics and Computers in Simulation* 82, 5 (2012), 868-878.

Makarov, R. and D. Glew, "Exact simulation of bessel diffusions," *Monte Carlo Methods and Applications* 16, 3/4 (2010), 283-306.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: the art of scientific computing*, 3rd edition, Cambridge University Press, Cambridge, 2007.