FAST AND ACCURATE LONG STEPPING SIMULATION OF THE HESTON STOCHASTIC VOLATILITY MODEL

JIUN HONG CHAN AND MARK JOSHI

Abstract. In this paper, we present three new discretization schemes for the Heston stochastic volatility model - two schemes for simulating the variance process and one scheme for simulating the integrated variance process conditional on the initial and the end-point of the variance process. Instead of using a short time-stepping approach to simulate the variance process and its integral, these new schemes evolve the Heston process accurately over long steps without the need to sample the intervening values. Hence, prices of financial derivatives can be evaluated rapidly using our new approaches.

1. Introduction

The use of stochastic volatility models to evaluate prices of financial derivatives among market practitioners has increased in the past few years. The main virtue of these models is that they provide a better calibration to the market implied volatility smiles and skews. Of all the stochastic volatility models, one particular popular model is the Heston stochastic volatility model. Unlike the Black-Scholes model where the variance process of the asset prices is assumed to be deterministic, the Heston model describes the variance process using a mean-reverting square root process. Such an extension allows the model to capture the market observable volatility smiles and skews. The wide-spread acceptance of the Heston model among market practitioners is mainly due to its tractability, particularly when it comes to computing prices of European options. Efficient model calibrations to market observable prices can be carried out easily as semi-closed form solutions for prices of European options exist under the Heston model.

Despite its popularity, there had been little research on the efficient discretizations of the continuous time dynamics of the Heston process until recently. A breakthrough occurred when Broadie and Kaya (2006) presented an exact simulation scheme for the Heston model. Under this approach, the simulation procedure at each path has 3 parts:

(1) sampling the end-point of the variance process conditional on the initial point,
(2) sampling the integrated variance process conditional on the initial and end-point of the variance process.
(3) sampling the asset price process conditional on the variance process and its integral.

As the variance process implies a scaled non-central chi-squared distribution, Broadie and Kaya used an acceptance-rejection technique to sample the variance process (see, Scott 1996) while samples of the conditional integrated variance process are generated through a numerical transform inversion of the integral's characteristic function. Once the variance process and its integral are sampled, the asset price process can be easily simulated using the exact representation of the share price process. Although this approach is theoretically appealing, the simulation cost of the conditional integrated variance process is too expensive for practical applications.

Several efficient short stepping discretization schemes were introduced recently, notably the Quadratic-Exponential (QE) Scheme (see, Andersen 2008) and Alfonsi's Second Order Scheme (see, Alfonsi 2008). As

Date: June 24, 2010.

Key words and phrases. Heston stochastic volatility, variance process, integrated variace process, long stepping simulation schemes, sampling gamma random variables.

sampling the integrated variance process can be very time consuming, these numerical schemes use a short time-stepping approach to simulate the variance process and the integrated variance process can therefore be approximated using the trapezoidal rule. In particular, the QE scheme approximates the variance process by drawing samples from related continuous distributions which are moment matched to the first two moments of the variance process while Alfonsi's Second Order Scheme uses discrete random variables to approximate the variance process. Several other authors including Van Haastrecht and Pelsser (2008), Kahl and Jäckel (2006), and Lord et al. (2008) have also contributed towards developing efficient discretization schemes for the Heston Model. Kahl and Jäckel suggest discretizing the variance process using an implicit Milstein scheme, however, Anderson showed that this scheme does poorly, with biases that can be are substantially larger the simple Euler scheme with full truncation (see Lord et al. 2008). The Van Haastrecht and Pelsser approach involves caching the inverse of the chi-squared distribution and a brief discussion of this method is presented in section 3.2.2.

However, not much progress has been made in developing efficient medium and long time-stepping discretization schemes for the Heston process. The real challenge involves in developing efficient long time-stepping discretization schemes comes from the need to sample the integrated variance process (conditional on the initial and the end-point of the variance process) accurately and efficiently. The use of the trapezoidal rule to approximate the integral of the variance process in a medium and long time-stepping setup is no longer sufficiently accurate while the computational cost of the Broadie-Kaya approach is too expensive for practical applications. Glasserman and Kim (2008) made an important contribution when they showed that the distribution of integrated variance process can be expressed as a infinite sum of a mixture of gamma distributions. Using this result, they develop an approximation approach to sample the integrated variance process. For readers who are interested in their approach, we refer them to Glasserman and Kim (2008).

Our contribution in this paper is to develop an efficient long-stepping discretization scheme for the Heston model. To do so, we first develop two new numerical schemes for simulating the variance process which we call the *GammaQE scheme* and the *Double Gamma scheme*. Both schemes allow for a fast and accurate simulation the variance process over long steps without the need to sample the intervening values and they requires precisely 3 random numbers per step. While the Double Gamma scheme is an exact simulation scheme for the variance process, the GammaQE scheme is an almost exact simulation scheme for the variance process where the scheme's accuracy improves with simulation step length. This scheme also has an additional virtue of being a smooth numerical scheme of the model inputs.

Conditional on the initial and the end-point of the variance process, we also develop an alternative approach to sample the conditional integrated variance process using the results established by Glasserman and Kim (2008). In particular, we use a different truncation approach to approximate the integrated variance process (which can be expressed as a infinite sum of a mixture of gamma distributions). The main advantage that our new integrated variance scheme has over the Glasserman and Kim approach is that our new scheme has a fixed dimension. That is, the number of random numbers required at each time step can be pre-determined at the beginning of the simulation. This property is important for sensitivities analysis and well as for the use of quasi-random number generators. By combining this new integrated variance scheme with the GammaQE scheme and the Double Gamma scheme, we developed 2 pricing schemes and we call them the *Integrated GammaQE scheme* and the *Integrated Double Gamma scheme* respectively. These schemes allow prices of financial derivatives in the Heston model to be evaluated using a long stepping approach.

This paper is organized as follow: in section 2, we outline the notations used in this paper together with the properties of the variance process and its integral. The GammaQE scheme and the Double Gamma scheme are presented in section 3. In section 4, we proposed a new approach to sample the integrated variance process conditional on the initial and end-point of the variance process. An efficient approach to simulate gamma random variables is presented section 5. The features of our new numerical scheme are discussed in details in section 6. In section 7, a brief discussion of our numerical tests is provided with numerical results presented in section 8. We conclude in section 9.

2. Heston Stochastic Volatility Model

2.1. **Model Setup.** The Heston process is described by the following stochastic differential equations

$$\frac{dS_t}{S_t} = r_t dt + \sqrt{V_t} dW_t^1 \tag{2.1}$$

$$dV_t = \kappa(\theta - V_t)dt + \epsilon\sqrt{V_t} dW_t^2 \tag{2.2}$$

where S_t represents an asset price process with $S_0 > 0$, V_t represents the instantaneous variance of $\frac{dS_t}{S_t}$ with $V_0 > 0$ and (W_t^1, W_t^2) is a two-dimensional Brownian motion with an instantaneous correlation of ρ (i.e. $dW_t^1 dW_t^2 = \rho dt$), while r_t represents a deterministic instantaneous risk-free rate. The parameters κ, θ, ϵ are positive constants with κ representing the rate of reversion of V_t, θ representing the long term mean of V_t and ϵ representing the volatility of V_t.

Proposition 1. *Under the Heston model, the exact solution for the asset price process at time t conditional on V_u and S_u with $u < t$ is given by*

$$S_t = S_u \exp\left(\int_u^t r_s ds + \frac{\rho}{\epsilon}(V_t - V_u - \kappa\theta(t-u)) + \left(\frac{\kappa\rho}{\epsilon} - \frac{1}{2}\right)\int_u^t V_s ds + \sqrt{1-\rho^2}\int_u^t \sqrt{V_s} dW_s^3\right) \tag{2.3}$$

where W_t^3 is a one-dimensional Brownian motion and independent of W_t^1 and W_t^2.

Proof. See Broadie and Kaya (2006) □

As noted by Broadie and Kaya, at each path, once the variance process and its integral have been sampled, the asset price can be easily evolved using the representation above. In particular, the process of $\log(S_t/S_u)$ conditional on both V_u and $\int_u^t V_s ds$ is normally distributed with a mean of

$$\int_u^t r_s ds + \frac{\rho}{\epsilon}(V_t - V_u - \kappa\theta(t-u)) + \left(\frac{\kappa\rho}{\epsilon} - \frac{1}{2}\right)\int_u^t V_s ds$$

and a variance of

$$(1-\rho^2)\int_u^t V_s ds.$$

This result is used by them to evolve the asset price process in the 3rd part of their simulation procedure. Since this problem has been satisfactorily solved, we turn our focus to developing efficient simulation schemes for the first and second parts of the simulation procedure. i.e

1. sampling V_t conditional on V_u,
2. sampling $\int_u^t V_s ds$ conditional on V_u and V_t.

In the following subsections, we briefly discuss the properties of the variance process, V_t as well as the properties of the integrated variance process, $\int_u^t V_s ds$ conditional on V_u and V_t

2.2. **Properties of the Variance Process, V_t.** The variance process, V_t, is also known as the mean-reverting square root process and this has similar dynamics to the celebrated CIR interest rate model. It is well known that the mean-reverting square root process has the following properties (for example, see Cox, Ingersoll and Ross(1985) and Dufresne (2001)):

Proposition 2. *We let $\chi^2_k(q)$ represent a noncentral chi-squared random variable with k degree of freedom and a non-centrality parameter of q where*

$$\mathbb{P}(\chi^2_k(q) < x) = e^{-q/2}\sum_{j=0}^{\infty}\frac{(q/2)^j}{j!2^{k/2+j}\Gamma(k/2+j)}\int_0^x z^{k/2+j-1}e^{-z/2}dz. \tag{2.4}$$

with $k, q > 0$. Conditioning on $V_u = \upsilon_u$, the variance process, V_t, is distributed as $C(u,t)$ times a noncentral chi-squared random variable with δ degrees of freedom and a non-centrality parameter of $n(u,t)\upsilon_u$,

$$i.e. \quad (V_t|V_u = \upsilon_u) \stackrel{d}{=} C(u,t)\chi^2_\delta\left(n(u,t)\upsilon_u\right), \tag{2.5}$$

where

$$C(u,t) = \frac{e^{-\kappa(t-u)}}{n(u,t)}, \quad \delta = \frac{4\kappa\theta}{\epsilon^2}, \quad n(u,t) = \frac{4\kappa e^{-\kappa(t-u)}}{\epsilon^2(1-e^{-\kappa(t-u)})}.$$

Proposition 3. *We let*

$$\alpha = \frac{2\kappa\theta}{\epsilon^2}, \quad \beta = \frac{\epsilon^2}{2\kappa}\left(1 - e^{-\kappa(t-u)}\right), \quad \lambda = \frac{2\kappa}{\epsilon^2(e^{\kappa(t-u)}-1)}\upsilon_u$$

We define

- $\Gamma(\alpha,\beta), \Gamma_1(\alpha,\beta), \Gamma_2(\alpha,\beta)\ldots$ *to be independent and identically distributed gamma random variables with a mean of $\alpha\beta$ and a variance of $\alpha\beta^2$,*
- N_λ *to be a Poisson random variable with a mean of λ,*
- $\mathrm{Exp}(\beta), \mathrm{Exp}_1(\beta), \mathrm{Exp}_2(\beta), \ldots$ *to be independent and identically distributed exponential random variables with a mean of β.*

The distribution of V_t condition on V_u admits the representation

$$(V_t|V_u = \upsilon_u) \stackrel{d}{=} Y_1 + Y_2 \tag{2.6}$$

where

$$Y_1 \stackrel{d}{=} \Gamma(\alpha,\beta) \qquad Y_2 \stackrel{d}{=} \sum_{i=1}^{N_\lambda}\mathrm{Exp}_i(\beta) \tag{2.7}$$

and $\Gamma(\alpha,\beta), N_\lambda$ and $\mathrm{Exp}_i(\beta)$ for all i are mutually independent random variables.

Proof. see Dufresne (2001). Here, we provide an alternative proof using moment generating function (m.g.f.). The m.g.f. of $\chi^2_k(q)$ is given by

$$\mathbb{E}\left[\exp(r\cdot\chi^2_k(q))\right] = (1-2r)^{-\frac{k}{2}}\exp\left(\frac{qr}{1-2r}\right).$$

Therefore, the m.g.f. of V_t conditional on V_u is given by

$$\begin{aligned}\mathbb{E}[\exp(rV_t)|V_u = \upsilon_u] &= (1-2c(u,t)r)^{-\frac{d}{2}}\exp\left(\frac{V_u n(u,t)c(u,t)r}{1-2c(u,t)r}\right), \\ &= \phi(r)^\alpha\exp(\lambda(\phi(r)-1)) \end{aligned}\tag{2.8}$$

where α, β, λ are as defined in Proposition 3 and $\phi(r) = (1-\beta r)^{-1}$ is the m.g.f of an exponential distribution with a mean of β. Since

$$\mathbb{E}\big[\exp(rY_1)\big] = \phi(r)^{\alpha} \quad \Longleftrightarrow \quad Y_1 \overset{d}{=} \Gamma(\alpha,\beta)$$
$$\mathbb{E}[\exp(rY_2)] = \exp(\lambda(\phi(r)-1)) \quad \Longleftrightarrow \quad Y_2 \overset{d}{=} \sum_{i=1}^{N_\lambda} \mathrm{Exp}_i(\beta),$$

the result in Proposition 3 clearly holds. □

We note that the random variable, Y_2, follows a compound Poisson distribution where the summands are exponentially distributed and, gamma random variables and exponential random variables have the following properties:

$$\Gamma(\alpha,\beta) \overset{d}{=} \beta \cdot \Gamma(\alpha,1), \qquad \text{and} \qquad \sum_{i=1}^{n} \mathrm{Exp}_i(\beta) \overset{d}{=} \Gamma(n,\beta). \tag{2.9}$$

Other well-known properties of the variance process include the Feller condition which guarantees strict positivity of the variance process, V_t, if $2\kappa\theta > \epsilon$ while for cases where $2\kappa\theta < \epsilon$, the origin is accessible and strongly reflecting. Typically in financial applications, we have $2\kappa\theta \ll \epsilon$. Hence, the probability of the variance process, V_t, hitting zero can be quite significant.

For simulation purposes, the variance process, V_t, can always be sampled from the noncentral chi-squared distribution using the acceptance and rejection techniques (For example, see Glasserman 2003 and Scott 1996) and this is the same approach used by Broadie and Kaya (2006). However, the main problem with the acceptance and rejection techniques is that the dimensionality of the random numbers required at each time step changes depending on state variables and input parameters. Hence, a small change in parameter inputs can abruptly alter the samples generated and sensitivity estimates obtained under such a simulation approach can have high variances. On top of that, the total number of the random numbers required for each path cannot be determined at the beginning of the simulation and this restricts the use of quasi-random number generators.

2.3. **Properties of Integrated Variance Process, $\int_u^t V_s ds$, conditional on V_t and V_u.** Glasserman and Kim (2008) show that the exact distribution of $\int_0^t V_s ds$, conditional on V_t and V_0 can be represented by infinite sums and mixtures of gamma random variables.

Proposition 4. *We let*

$$\begin{aligned}
\lambda_j^* &= \frac{16\pi^2 j^2}{\epsilon^2(t-u)(\kappa^2(t-u)^2+4\pi^2 j^2)},\\
\lambda_j &= (\upsilon_t+\upsilon_u)\lambda_j^*,\\
\gamma_j &= \frac{\kappa^2(t-u)^2+4\pi^2 j^2}{2\epsilon^2(t-u)^2},\\
z &= \frac{2\kappa/\epsilon^2}{\sinh\left(\kappa(t-u)/2\right)}\sqrt{\upsilon_t\upsilon_u},\\
\nu &= \delta/2-1.
\end{aligned}$$

We define

$$X_1 \overset{d}{=} \sum_{j=1}^{\infty} \frac{1}{\gamma_j} \sum_{i=1}^{N_{\lambda_j}} \mathrm{Exp}_i(1), \qquad X_2 \overset{d}{=} \sum_{j=1}^{\infty} \frac{1}{\gamma_j} \Gamma_j(\delta/2,1), \qquad Z \overset{d}{=} \sum_{j=1}^{\infty} \frac{1}{\gamma_j} \Gamma_j(2,1),$$

and η to be a Bessel random variable with probability mass given by

$$\mathbb{P}(\eta = n) = \frac{(z/2)^{2n+\nu}}{I_\nu(z)n!\Gamma(n+\nu+1)}$$

where $I_\nu(z)$ is the modified Bessel function of the first kind. The distribution of the integrated variance process $\int_u^t V_s ds$, conditional on V_t and V_u admits the representation

$$\left(\int_u^t V_s ds | V_t, V_u\right) \overset{d}{=} X_1 + X_2 + \sum_{i=1}^{\eta} Z_i \tag{2.10}$$

where $X_1, X_2, \eta, Z_1, Z_2 \ldots$ are mutually independent random variable, the Z_i are independent copies of a random variable Z.

Proof. See, Glasserman and Kim (2008). □

Proposition 5. *The mean and the variance for the random variable X_1, X_2 and Z are given by*

$$\begin{aligned} \mathbb{E}(X_1) &= (\upsilon_t + \upsilon_u)\mu^*_{X_1}, & \mathrm{Var}(X_1) &= (\upsilon_t + \upsilon_u){\sigma^*_{X_1}}^2, \\ \mathbb{E}(X_2) &= \delta\mu^*_{X_2}, & \mathrm{Var}(X_2) &= \delta{\sigma^*_{X_2}}^2, \\ \mathbb{E}(Z) &= 4\mu^*_{X_2}, & \mathrm{Var}(Z) \ &= 4{\sigma^*_{X_2}}^2, \end{aligned}$$

where

$$\begin{aligned} \mu^*_{X_1} &= \frac{1}{\kappa}\coth\left(\frac{\kappa(t-u)}{2}\right) - \frac{(t-u)}{2}\operatorname{csch}^2\left(\frac{\kappa(t-u)}{2}\right) \\ (\sigma^*_{X_1})^2 &= \frac{\epsilon^2}{\kappa^3}\coth\left(\frac{\kappa(t-u)}{2}\right) + \frac{\epsilon^2(t-u)}{2\kappa^2}\operatorname{csch}^2\left(\frac{\kappa(t-u)}{2}\right) - \frac{\epsilon^2(t-u)^2}{2\kappa}\coth\left(\frac{\kappa(t-u)}{2}\right)\operatorname{csch}^2\left(\frac{\kappa(t-u)}{2}\right) \\ \mu^*_{X_2} &= \frac{\epsilon^2}{4\kappa^2}\left(-2 + \kappa(t-u)\coth\left(\frac{\kappa(t-u)}{2}\right)\right) \\ (\sigma^*_{X_2})^2 &= \frac{\epsilon^4}{8\kappa^4}\left(-8 + 2\kappa(t-u)\coth\left(\frac{\kappa(t-u)}{2}\right) + \kappa^2(t-u)^2\operatorname{csch}^2\left(\frac{\kappa(t-u)}{2}\right)\right) \end{aligned}$$

Proof. See, Glasserman and Kim (2008). They obtained these results by first deriving the Laplace transform of X_1, X_2 and Z. The means and the variances can then be easily calculated. □

Based on the results above Glasserman and Kim (2008) developed a new approach to sample to the conditional integrated variance process. Since both random variables, X_2 and Z, are state-independent, they tabulate the distribution function of X_2 and Z by inverting their respective Laplace transforms. Therefore, the random variable X_2 can be sampled efficiently and similarly, once the outcome of the Bessel random variable is known with $\eta = n_0$, the random variables Z_j for $j = 1, \ldots, n_0$ can also be generated efficiently. As X_1 is state-dependent, tabulating the distribution function is not a feasible option. Instead, they suggested to truncate the series representation of X_1 and approximate the remainder of the truncated series using a gamma random variable.

While it was shown that this simulation scheme is particularly effective for pricing options with long-date maturities, there are still some shortcomings. First of all, this numerical scheme does not have a fixed dimension, in particular, the number of random number requires at each time step changes according to the outcome of the Bessel random variable η. This restricts the possibility of using quasi-random number generators. The computational cost of this approach can be expensive as the modified Bessel function of the first kind has to be evaluated at each time step in order to generate the Bessel random variable. Similarly, a numerical root search is also required at each time step as the remainder of the truncated series is approximated using a gamma random variable.

3. Simulation of the Variance process.

In this section, we present two new approaches to simulate the variance process, V_t which we call the GammaQE scheme and the Double Gamma scheme. These new approaches are built on the representation presented in Proposition 3. Specifically, we divide the simulation of the variance process, V_t, into 2 parts: the gamma random variable, Y_1, and the compound Poisson random variable, Y_2. Both schemes use the same approach to simulate the gamma random variable and their differ in how the compound Poisson random variable is generated.

3.1. **Simulation of the Gamma Random Variable, Y_1.** Using properties of the gamma distribution, the random variable, Y_1, is distributed as

$$Y_1 \overset{d}{=} \beta \cdot \Gamma(\alpha, 1). \tag{3.1}$$

Since the parameter α is state and step-size independent (see Proposition 3), the gamma random variable, $\Gamma(\alpha, 1)$, can be sampled efficiently by creating a one-dimensional cache of the inverse of the gamma distribution function. In particular, we tabulate the inverse of the gamma distribution function at the start of the simulation and then draw samples as needed by sampling from the table. Since the computational overhead for a one-dimensional cache is low, the gamma random variable can therefore be simulated efficiently. A detailed discussion on how such a cache can be created is presented in section 5.

3.2. **Simulation of the Compound Poisson Distribution, Y_2.** While a direct simulation of the compound Poisson random variable is possible, we do not pursue this approach. Instead, under the GammaQE scheme, we approximate the compound Poisson random variable using the sum of two random variables sampled from two different distributions while, under the Double Gamma scheme, the compound Poisson random variable is sampled from a gamma distribution conditional on N_λ.

3.2.1. *Simulating Y_2 using the GammaQE scheme.* Observe that, if λ is small, then

$$P(N_\lambda \leq 1) = (1 + \lambda)e^{-\lambda} \approx 1.$$

and hence, the compound Poisson random variable can be approximated by a zero-modified exponential random variable. For large λ, the compound Poisson random variable can be approximated using a power function applied to a Gaussian variable. Specifically, we write

$$Y_2 \approx a(b + Z)^2$$

where Z is a standard Gaussian random variable and the constant a and b are to be determined. We shall call this the quadratic Gaussian distribution. This has similarities to the QE scheme proposed by Anderson (2006). Under the QE scheme, the variance process is simulated using samples generated from either the zero-modified exponential distribution or the quadratic Gaussian distribution depending on the level of the initial variance process.

However, instead of having to switch from one distribution to another, under the GammaQE scheme, the compound Poisson random variable is approximated using the sum of two random variables, Y_3 and Y_4, sampled from a zero-modified exponential distribution and a zero-modified quadratic Gaussian distribution. Specifically, the cumulative distribution function of the zero-modified exponential distribution, $F_{Y_3}(y)$ is given by

$$F_{Y_3}(y) = \begin{cases} P_{Y_3} & y = 0, \\ (1 - P_{Y_3})(1 - e^{-y/\mu}) & y > 0, \end{cases} \tag{3.2}$$

with $\mu > 0$ and the cumulative distribution function of the zero-modified quadratic Gaussian distribution, $F_{Y_4}(y)$ is given by

$$F_{Y_4}(y) = \begin{cases} P_{Y_4} & y = 0, \\ \left(1 - P_{Y_4}\right) F_{\chi_1^2(b^2)}\left(\frac{y}{a}\right) & y > 0, \end{cases} \tag{3.3}$$

where $F_{\chi_1^2(b^2)}\left(.\right)$ is the cumulative distribution function of the noncentral chi-squared random variable with 1 degree of freedom and a non-centrality parameter of $b^2 > 0$. As the noncentral chi-squared distribution has 1 degree of freedom, the random variable Y_4 can be sampled easily (see Algorithm 1). The parameters in both distributions are determined by moment matching to the first two moments of the compound Poisson distribution. In particular, given the mean and the variance of the compound Poisson distribution, the GammaQE scheme determines the proportion (weight) of the mean and the variance that must be explained (matched) by the zero-modified exponential random variables while the remainder of the mean and the variance are explained (matched) by the zero-modified quadratic Gaussian random variable. The weight is determined based on the value of λ. This approach allows us to have a smooth transition from the zero-modified exponential distribution to the zero-modified Quadratic Gaussian distribution and vice-versa in approximating the compound Poisson distribution.

Algorithm 1. *Under GammaQE scheme, the compound Poisson random variable can be sampled using the following approximation*

$$Y_2 \approx Y_3 + Y_4 \tag{3.4}$$

with

$$Y_3 = \begin{cases} 0 & 0 \leq U_{Y_3} \leq P_{Y_3} \\ \mu \ln\left(\frac{1-P_{Y_3}}{1-U_{Y_3}}\right) & P_{Y_3} < U_{Y_3} \leq 1 \end{cases} \tag{3.5}$$

and

$$Y_4 = \begin{cases} a\left(b + \Phi^{-1}\left(\frac{U_{Y_4}}{1-P_{Y_4}}\right)\right)^2 & 0 \leq U_{Y_4} \leq c_1 \\ 0 & c_1 < U_{Y_4} \leq c_2 \\ a\left(b + \Phi^{-1}\left(\frac{U_{Y_4}-P_{Y_4}}{1-P_{Y_4}}\right)\right)^2 & c_2 < U_{Y_4} < 1 \end{cases} \tag{3.6}$$

where U_{Y_3} and U_{Y_4} are uniformly distributed random variable, P_{Y_3} and P_{Y_4} are the probability of zero for the random variable Y_3 and Y_4 respectively and Φ^{-1} is the inverse cumulative distribution function of the standard normal distribution. The parameters $a > 0$, $b^2 > 0$, $\mu > 0$, $P_{Y_3} \in [0, 1)$ and $P_{Y_4} \in [0, 1)$ are determined by matching the first two moments of the compound Poisson distribution and the probability at zero (i.e. $P(Y_2 = 0) = e^{-\lambda}$) and they are given by

$$P_{Y_3} = \frac{2 - \lambda w}{2 + \lambda w}, \quad \mu = \frac{w\lambda\beta}{1 - P_{Y_3}}, \quad P_{Y_4} = \frac{e^{-\lambda}}{P_{Y_3}}, \quad b^2 = -c - 1 + \sqrt{c(c+1)}, \quad a = \frac{(1-w)\lambda\beta}{(1 - P_{Y_4})(1 + b^2)},$$

where

$$\begin{aligned} w &= (1 + \lambda)e^{-\lambda}, \\ c &= \frac{2\lambda(1-w)}{\lambda(1-w) - (1 - P_{Y_4})(2 + \lambda(1-w))}. \end{aligned}$$

The value of c_1 and c_2 given by $c_1 = (1 - P_{Y_4}) \cdot \Phi(-b)$ and $c_2 = c_1 + P_{Y_4}$.

A detailed proof of the result in Algorithm 1 is presented in Appendix A1. Note that, instead of using the representation below

$$Y_4 = \begin{cases} 0 & 0 \leq U_{Y_4} \leq P_{Y_4} \\ a\left(b + \Phi^{-1}\left(\frac{1-U_{Y_4}}{1-P_{Y_4}}\right)\right)^2 & P_{Y_4} < U_{Y_4} \leq 1 \end{cases} \tag{3.7}$$

to simulate Y_4, the presentation in equation (3.6) is used as that approach ensures that the random variable Y_4 is a continuous function of U_{Y_4}. This is particularly important when it comes to sensitivity analysis.

3.2.2. *Sampling Y_2 using the Double Gamma Scheme.* Observe that, by conditioning on N_λ, the compound Poisson random variable satisfies the following representation

$$\left(\sum_{i=1}^{N_\lambda} \mathrm{Exp}_i(\beta)\middle| N_\lambda = n\right) \overset{d}{=} \beta \cdot \Gamma(n, 1). \tag{3.8}$$

Therefore, once the Poisson random variable, N_λ, is sampled, the compound Poisson random variable can be drawn from a gamma distribution with a shape parameter n. Sampling compound Poisson random variables using this approach can be very time consuming as a direct inversion of the gamma distribution function is usually required as the shape parameter, n, changes on a path by path basis.

Fortunately, the Poisson random variable, N_λ, is a discrete random variable with $N_\lambda \in \mathbb{N}$. Regardless of the model inputs and state-variables, once the Poisson random variable, N_λ, is drawn, the compound Poisson random variable will always be sampled from a gamma distribution with a non-negative integer shape parameter. This makes it worthwhile to tabulate the inverse of the gamma distribution for all the integer shape parameters as this only has to be done once and it can be reused to generate compound Poisson random variables even if the model inputs and state-variables have changed. In particular, we create multiple one-dimensional caches where each cache corresponds to a specific integer shape parameter. From a programming perspective, once such caches are created and stored in a static library, sampling compound Poisson random variables can be done very efficiently with a price of slightly higher storage cost. A detailed explanation on how such caches can be created efficiently is presented in section 5 and the Poisson random variable, N_λ, can be sampled easily using the inverse transformation method (see, pg 128 Glasserman (2003)).

An interesting question is for what value of the integer shape parameter should we cache. In theory, we would like to cache all the integer value of the shape parameter. However, in practice, this is never possible. In general, for a long time step, the outcome the Poisson random variable, N_λ, is often small. We recommend to tabulate the inverse of the gamma distribution function for shape parameters up to 100 and for any shape parameter above 100, the gamma random variable can be approximated using a lognormal random variable moment matched to the first two moments of the gamma random variable. In general, such approximation works well for large shape parameters.

We note that our new Double Gamma scheme has some similarities to the Non-central Chi-squared Inversion (NCI) scheme (Van Haastrecht and Pelsser (2008)). The NCI scheme is built on the following representation

$$(V_t | V_u = v_u, N_\lambda = n) \overset{d}{=} C(u,t)\chi^2_{\delta+2n}, \tag{3.9}$$

where N_λ is the same Poisson random variable as defined in Proposition 3 and $\chi^2_{\delta+2N_\lambda}$ is a chi-squared random variable with $\delta + 2n$ degrees of freedom. Specifically, conditioning on $N_\lambda = n$, the variance process, V_t, can be sampled from a chi-squared distribution with $\delta + 2n$ degrees of freedom. Under the NCI scheme, multiple one-dimensional caches for the inverse of the chi-squared distribution function are created (with each cache corresponds to $\delta + 2n$ degrees of freedom for $n = 0, 1, \ldots, n_{max}$) and sampling the variance process, V_t, can be done easily once N_λ is known.

The crucial difference between the Double Gamma scheme and the NCI scheme is that the multiple one-dimensional caches for the Double Gamma scheme only need to be created once and can be reused even if the model inputs and state-variable change whereas the NCI scheme requires the creation of such caches every time when model inputs differ. Under the Double Gamma scheme, we also studied the asymptotic behaviors of the gamma distribution and, by performing a suitable transformation (instead of direct caching as proposed in NCI scheme), the number of points required to create a reasonably accurate cache can be significantly reduced (see, Section 5).

4. Simulation of the Integrated Variance Process

Using the result in Proposition 4, we develop an alternative approach to simulate the integrated variance process. Observe that, by conditioning on V_t, V_u and η, the integrated variance process can be expressed in the following form

$$\left(\int_u^t V_s ds \middle| V_t = v_t, V_u = v_u, \eta = n_0 \right) \overset{d}{=} \sum_{j=1}^{\infty} \frac{1}{\gamma_j} \left(\sum_{i=1}^{N_{\lambda_j}} \mathrm{Exp}_i(1) + \Gamma_j(2n_0 + \delta/2, 1) \right). \tag{4.1}$$

For a quick simulation, the series in the equation above is truncated after the k-th term and the remainder of the series is approximated using a lognormal random variable moment matched to the first two moments of the remainder of the series i.e.

$$\left(\int_u^t V_s ds \middle| V_t = v_t, V_u = v_u, \eta = n_0 \right) \overset{d}{=} \sum_{j=1}^{k} \frac{1}{\gamma_j} \left(\sum_{i=1}^{N_{\lambda_j}} \mathrm{Exp}_i(1) + \Gamma_j(2n_0 + \delta/2, 1) \right) + \mathrm{Ln}(\mu_{Ln}^k, \sigma_{Ln}^k) \tag{4.2}$$

where parameters μ_{Ln}^k and σ_{Ln}^k are determined by moment matching. Further conditioning on $N_{\lambda_j} = n_j$ for $j = 1, 2, \ldots, k$ gives

$$\sum_{j=1}^{k} \frac{1}{\gamma_j} \left(\sum_{i=1}^{N_{\lambda_j}} \mathrm{Exp}_i(1) + \Gamma_j(2n_0 + \delta/2, 1) \right) \overset{d}{=} \sum_{j=1}^{k} \frac{1}{\gamma_j} \Gamma_j(2n_0 + n_j + \delta/2, 1). \tag{4.3}$$

Once the truncation level, k, is determined, the simulation procedure of the integrated variance process can be summarized as follow

1. At each time step, we first sample the Bessel random variable η.
2. We then sample the Poisson random variable, N_{λ_j}.
3. Once N_{λ_j} is known, we proceed to sample the gamma random variable $\Gamma_j(2n_0 + n_j + \delta/2, 1)$.
4. We repeat step 2 and 3 from $j = 1$ up to $j = k$ and compute the sum of $\frac{1}{\gamma_j}\Gamma_j(2n_0 + n_j + \delta/2, 1)$ for $j = 1, 2, \ldots, k$
5. Lastly, we sample the lognormal random variable, $\mathrm{Ln}(\mu_{Ln}^k, \sigma_{Ln}^k)$, and the integrated variance process from time u to t is given by the sum of $\mathrm{Ln}(\mu_{Ln}^k, \sigma_{Ln}^k)$ and $\frac{1}{\gamma_j}\Gamma_j(2n_0 + n_j + \delta/2, 1)$ for $j = 1, 2, \ldots, k$.

4.1. **Efficient simulation of the Bessel random variable η.** As described by Glasserman and Kim (2008), the simulation procedure of the Bessel random variable, η is as follows: they generate a uniform random number, $U_\eta \in [0,1)$ and calculate the probability mass function recursively,

$$p_{m+1} = \frac{z^2}{4(m+1)(m+1+\nu)} p_m \qquad p_0 = \frac{(z/2)^\nu}{I_\nu(z)\Gamma(\nu+1)}$$

until they reach a value n_0 such that

$$\sum_{m=1}^{n_0-1} p_m \le U_\eta < \sum_{m=1}^{n_0} p_m. \tag{4.4}$$

They set $\eta = n_0$ to be simulated Bessel random variable.

Observe that, the most time consuming step is the computation of p_0 as it involves computing a modified Bessel function of the first kind which in turn is a sum of a infinite series. It is well known that the computational cost can be high for large value of z due to the slow convergence of the series.

Using the definition of the modified Bessel function of the first kind, one can express P_0 as

$$p_0 = \frac{1}{\sum_{n=0}^{\infty} C_n \cdot (\upsilon_u \upsilon_t)^n} \tag{4.5}$$

where

$$C_0 = 1, \qquad C_{n+1} = \frac{D(u,t)}{4(n+1)(n+1+\nu)} C_n, \qquad D(u,t) = \left(\frac{2\kappa/\epsilon^2}{\sinh(\kappa(t-u)/2)} \right)^2. \tag{4.6}$$

As $n \to \infty$, we have $C_n \to 0$. Therefore, it is possible to approximate p_0 accurately by truncating the infinite series in the denominator of equation (4.5). Observe that, C_n increases by a factor of $D(u,t)$ and decreases by a factor of n^2 when n increases and, therefore, our truncation level clearly will depends on $D(u,t)$. The accuracy of P_0 after truncation will also depend on the value of $\upsilon_u \upsilon_t$ as the series converges slowly for large value of $\upsilon_u \upsilon_t$. Given that there is such a complicated dependency, we truncate the infinite series dynamically. In particular, we propose to approximate p_0 using the first h-th terms in the series i.e.

$$p_0 \approx \frac{1}{\sum_{n=0}^{h} C_n \cdot (\upsilon_u \upsilon_t)^n} \tag{4.7}$$

where h is determined such that we have

$$\frac{C_h \cdot (\upsilon_u \upsilon_t)^h}{\sum_{n=0}^{h} C_n \cdot (\upsilon_u \upsilon_t)^n} < 10^{-4}$$

for the first time as h increases. In general, using the inputs given in our numerical tests (which are relevant for financial applications), we rarely need more than 10 terms to approximate P_0 accurately. Therefore, we can reduce the computational time of P_0 substantially. Also, since the value of all C_n's is state-independent, they can therefore be calculated at the beginning of the simulation to further reduce the computational cost.

4.2. **Efficient Simulation of Poisson Random Variable N_{λ_j}.** As explained earlier, the Poisson random variable, N_{λ_j}, can be sampled using the inverse transformation method (see, pg 128 Glasserman (2003)).

4.3. **Efficient Simulation of Gamma Random Variable $\Gamma_j(2n_0 + n_j + \delta/2, 1)$.** Using properties of the gamma distribution, we can decompose the gamma random variable, $\Gamma_j(2n_0 + n_j + \delta/2, 1)$, into the sum of two independent gamma random variables, i.e.

$$\Gamma_j(2n_0 + n_j + \delta/2, 1) \overset{d}{=} \Gamma_{j1}(2n_0 + n_j) + \Gamma_{j2}(\delta/2, 1)$$

where Γ_{j1} and Γ_{j2} are independent gamma random variables. Since we have $\delta/2 = \alpha$, the gamma random variable, $\Gamma_{j2}(\delta/2, 1)$, can be sampled from the cache that we have created when sampling the variance process in section 3.1. Similarly, as $2n_0 + n_j \in \mathbb{N}$, the gamma random variable, $\Gamma_{j1}(2n_0 + n_j)$, can be sampled efficiently from the multiple one-dimensional caches as described in section 3.2.2. Hence, sampling $\Gamma_j(2n_0 + n_j + \delta/2, 1)$ can be done efficiently.

4.4. **Efficient Simulation of Log-normal Random Variable,** $\mathrm{Ln}(\mu^k_{Ln}, \sigma^k_{Ln})$**.** The remainder of the truncated series is approximated using a log-normal random variable, $\mathrm{Ln}(\mu^k_{Ln}, \sigma^k_{Ln})$ with a mean, M_{Ln}, and a variance, $(S_{Ln})^2$ of

$$\begin{aligned} M_{Ln} &= \exp\left(\mu^k_{Ln} + 0.5(\sigma^k_{Ln})^2\right), \\ (S_{Ln})^2 &= ((\sigma^k_{Ln})^2 - 1)\exp\left(2\mu^k_{Ln} + (\sigma^k_{Ln})^2\right). \end{aligned}$$

The parameters μ^k_{Ln} and σ^k_{Ln} are determined by matching the first two moments of the remainder of the truncated series.

Proposition 6. *we define μ^k_R and $(\sigma^k_R)^2$ to be the mean and the variance for the remainder of the truncated series conditioning on $V_u = \upsilon_u$, $V_t = \upsilon_t$ and $\eta = n_0$ and they are given by*

$$\begin{aligned} \mu^k_R &= (\upsilon_t + \upsilon_u)\left(\mu^*_{X_1} - \sum_{j=1}^{k}\left(\frac{\lambda^*_j}{\gamma_j}\right)\right) + (4n_0 + \delta)\left(\mu^*_{X_2} - \sum_{j=1}^{k}\left(\frac{1}{2\gamma_j}\right)\right) \\ (\sigma^k_R)^2 &= (\upsilon_t + \upsilon_u)\left((\sigma^*_{X_1})^2 - \sum_{j=1}^{k}\left(\frac{2\lambda^*_j}{\gamma^2_j}\right)\right) + (4n_0 + \delta)\left((\sigma^*_{X_2})^2 - \sum_{j=1}^{k}\left(\frac{1}{2\gamma^2_j}\right)\right) \end{aligned} \tag{4.8}$$

Proof. By conditioning on V_u, V_t and η, the mean and the variance of the integrated variance process are

$$\begin{aligned} \mathbb{E}\left(\int_u^t V_s ds \middle| V_t = \upsilon_t, V_u = \upsilon_u, \eta = n_0\right) &= (\upsilon_t + \upsilon_u)\mu^*_{X_1} + (4n_0 + \delta)\mu^*_{X_2} \\ \mathrm{Var}\left(\int_u^t V_s ds \middle| V_t = \upsilon_t, V_u = \upsilon_u, \eta = n_0\right) &= (\upsilon_t + \upsilon_u){\sigma^*_{X_1}}^2 + (4n_0 + \delta){\sigma^*_{X_2}}^2 \end{aligned}$$

Similarly, we have

$$\begin{aligned} \mathbb{E}\left(\sum_{j=1}^{k}\frac{1}{\gamma_j}\left(\sum_{i=1}^{N_{\lambda_j}}\mathrm{Exp}_i(1) + \Gamma_j(2n_0 + \delta/2, 1)\right)\right) &= \sum_{j=1}^{k}\frac{1}{\gamma_j}(\lambda_j + 2n_0 + \delta/2) \\ &= (\upsilon_t + \upsilon_u)\sum_{j=1}^{k}\left(\frac{\lambda^*_j}{\gamma_j}\right) + (4n_0 + \delta)\sum_{j=1}^{k}\left(\frac{1}{2\gamma_j}\right) \\ \mathrm{Var}\left(\sum_{j=1}^{k}\frac{1}{\gamma_j}\left(\sum_{i=1}^{N_{\lambda_j}}\mathrm{Exp}_i(1) + \Gamma_j(2n_0 + \delta/2, 1)\right)\right) &= \sum_{j=1}^{k}\frac{1}{\gamma^2_j}(2\lambda_j + 2n_0 + \delta/2) \\ &= (\upsilon_t + \upsilon_u)\sum_{j=1}^{k}\left(\frac{2\lambda^*_j}{\gamma^2_j}\right) + (4n_0 + \delta)\sum_{j=1}^{k}\left(\frac{1}{2\gamma^2_j}\right) \end{aligned} \tag{4.9}$$

using the fact that the mean and the variance of the compound Poisson distribution is given by λ_j and $2\lambda_j$ respectively. Therefore, results in the Proposition 6 clearly holds as all the random variables in the infinite series are mutually independent. □

Once the value of μ_R^k and $(\sigma_R^k)^2$ are determined, a simple moment matching exercise gives

$$\begin{aligned}\left(\sigma_{Ln}^k\right)^2 &= \log\left(\left(\frac{\sigma_R^k}{\mu_R^k}\right)^2 + 1\right), \\ \mu_{Ln}^k &= \log\left(\mu_R^k\right) - 0.5\left(\sigma_{Ln}^k\right)^2.\end{aligned}$$

In order to sample from a log-normal distribution with the above parameters, we first generate a standard normal random variable Z_{Ln}. The log-normal random variable can then be generated using

$$\mathrm{Ln}(\mu_{Ln}^k, \sigma_{Ln}^k) = \exp\left(\mu_{Ln}^k + \sigma_{Ln}^k \cdot Z_{Ln}\right).$$

5. Caching and Sampling Gamma Random Variables

We devote this section to discuss an efficient approach to cache the inverse of gamma distribution functions and to sample gamma random variables from the cache. We define $F_\alpha(x)$ to be the cumulative distribution function for $\Gamma(\alpha, 1)$ and it is given by

$$F_\alpha(x) = \mathrm{P}(\Gamma(\alpha, 1) \leq x) = \int_0^x \frac{y^{\alpha-1}\exp(-y)}{\Gamma(\alpha)} dy. \tag{5.1}$$

One possible naive approach to create a one-dimensional cache is to tabulate the value of $F_\alpha^{-1}(u_i)$ at each

$$u_i = \frac{i-1}{N} \quad \text{for} \quad i = 1, 2 \ldots, N.$$

where $F_\alpha^{-1} : [0, 1) \longrightarrow [0, \infty)$ represents the inverse of the gamma distribution function and N represents the total number of points in the cache.

Instead, we suggest that the cache to be created by tabulating $\left[g_\alpha(u_i), F_\alpha^{-1}(u_i)\right]$ for all i where

$$g_\alpha(u) = \left((\alpha\Gamma(\alpha))^{1/\alpha} - \log(1-u)\right) u^{1/\alpha} \tag{5.2}$$

This is because, the lead term of the asymptotic expansion of the inverse of the gamma distribution function is given by

$$F_\alpha^{-1}(u) \sim (\alpha\Gamma(\alpha)u)^{1/\alpha} \tag{5.3}$$

as $u \rightarrow 0$. Similarly, the lead term of the asymptotic expansion of the inverse of the gamma distribution function is given by

$$F_\alpha^{-1}(u) \sim -\log(1-u) \tag{5.4}$$

as $u \rightarrow 1$. The result in equation 5.3 can be derived by expressing the exponential term in equation 5.1 as a power series and eliminating the the higher order terms while the result in equation 5.4 can be derived by evaluating the equation 5.1 using integration by part and eliminating the lower order terms. The function $g(u)$ can be viewed as a weighted average of both lead terms such that we have a smooth transition from one lead term to another as u changes. By transforming u to $g(u)$, the number of points required to create an reasonably accurate cache can be significantly reduced as the plot of $F_\alpha^{-1}(u)$ against $g(u)$ is a better-behaved function compared to the plot of $F_\alpha^{-1}(u)$ against u (see figure 5.1 and 5.2). In particular, we recommend using $N = 100$ for $\alpha \geq 1$ and $N = 1000$ for $\alpha < 1$. In order to increase the accuracy of the cache at the tail of the inverse distribution function, an additional point $\left[g_\alpha(u_{max}), F_\alpha^{-1}(u_{max})\right]$, where $u_{max} = 0.99999$, is also tabulated.

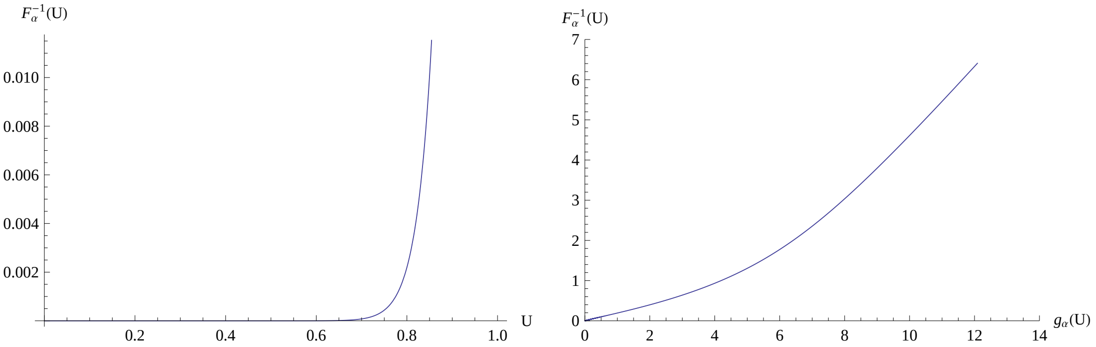

FIGURE 5.1. The figure shows the plot $F_\alpha^{-1}(u)$ against u and the plot $F_\alpha^{-1}(u)$ against $g_\alpha(u)$ for $\alpha = 0.04$.

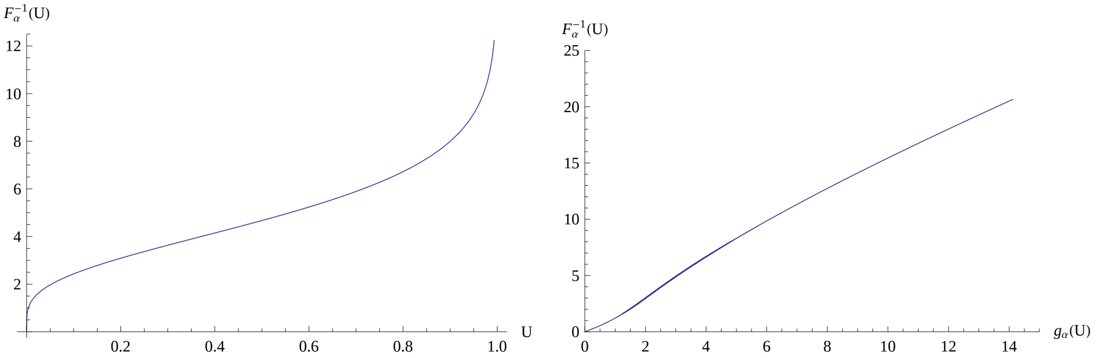

FIGURE 5.2. The figure shows the plot $F_\alpha^{-1}(u)$ against u and the plot $F_\alpha^{-1}(u)$ against $g_\alpha(u)$ for $\alpha = 5$.

For a smooth interpolation between the adjacent points in the cache, the first order derivative of $F_\alpha^{-1}(u_i)$ with respect to $g(u_i)$ given by

$$\frac{dF_\alpha^{-1}(u_i)}{dg_\alpha(u_i)} = \frac{dF_\alpha^{-1}(u_i)}{du_i}\frac{du_i}{dg_\alpha(u_i)} \tag{5.5}$$

where

$$\begin{aligned}\frac{dF_\alpha^{-1}(u)}{du} &= \Gamma(\alpha)\exp(F_\alpha^{-1}(u))(F_\alpha^{-1}(u))^{1-\alpha} \\ \frac{du}{dg_\alpha(u)} &= \frac{\alpha(1-u)(u)^{1-1/\alpha}}{\alpha u + (1-u)((\alpha\Gamma(\alpha))^{1/\alpha} - \log(1-u))}\end{aligned}$$

for all i are also tabulated at the beginning of the simulation. Note that, at $u = 0$, we have

$$\frac{dF_\alpha^{-1}(0)}{dg_\alpha(0)} = \lim_{u\to 0}\frac{F_\alpha^{-1}(u) - F_\alpha^{-1}(0)}{g_\alpha(u) - g_\alpha(0)} = 1 \tag{5.6}$$

using the result in equation (5.3).

When sampling gamma random variables, we propose to use a cubic interpolation between the adjacent points. At each interval, the coefficients of the cubic polynomial are determined by matching the cached value and derivative at the starting point and the ending point of the interval. Specifically, for the interval

N	RMS Absolute Error	Maximum Absolute Error	RMS Relative Error	Maximum Relative Error	Caching Time (second)
10	0.0393	0.3540	6.653%	27.596%	0.001
50	0.0100	0.1398	0.752%	7.407%	0.003
100	0.0030	0.0543	0.209%	2.498%	0.005
500	0.0003	0.0715	0.015%	0.845%	0.027
1,000	0.0002	0.0612	0.006%	0.722%	0.054
2,500	0.0001	0.0404	0.002%	0.477%	0.135
5,000	0.0000	0.0268	0.001%	0.316%	0.270

Table 5.1. The table above shows the accuracy of caches as N increases for $\alpha = 0.04$.

N	RMS Absolute Error	Maximum Absolute Error	RMS Relative Error	Maximum Relative Error	Caching Time (second)
10	0.0046	0.0379	0.230%	1.832%	0.001
50	0.0007	0.0232	0.010%	0.121%	0.003
100	0.0003	0.0186	0.002%	0.079%	0.006
500	0.0000	0.0103	0.001%	0.058%	0.029
1,000	0.0000	0.0076	0.000%	0.043%	0.058
2,500	0.0000	0.0049	0.000%	0.026%	0.147
5,000	0.0000	0.0033	0.000%	0.016%	0.293

Table 5.2. The table above shows the accuracy of caches as N increases for $\alpha = 5$.

$[g(u_i), g(u_{i+1}))$, the inverse of the gamma distribution functions is approximated using

$$F_\alpha^{-1}(u) = a_i g(u)^3 + b_i g(u)^2 + c_i g(u) + d_i, \tag{5.7}$$

where

$$\begin{aligned}
a_i &= \frac{-(2y_i - 2y_{i+1} - x_i y_i' + x_{i+1} y_i' - x_i y_{i+1}' + x_{i+1} y_{i+1}')}{(x_i - x_{i+1})^3} \\
b_i &= \frac{(y_i' - y_{i+1}') - 3a \cdot \left((x_i)^2 - (x_{i+1})^2\right)}{2(x_i - x_{i+1})}, \\
c_i &= y_i' - 3a \cdot (x_i)^2 - 2b \cdot x_i, \\
d_i &= y_i - a \cdot (x_i)^3 - b \cdot (x_i)^2 - c \cdot x_i.
\end{aligned}$$

with

$$y_i \equiv F_\alpha^{-1}(u_i), \quad y_i' \equiv \frac{dF_\alpha^{-1}(u_i)}{dg_\alpha(u_i)}, \quad x_i \equiv g(u_i), \tag{5.8}$$

for all i except at the final interval $[g(u_N), \infty)$ where the solution for the coefficient are determined by matching the cached value and derivative at the point $g(u_N)$ and $g(u_{max})$. Once the cache has been constructed, sampling gamma random variables are straight forward. We first generate a uniform random number U. Suppose that $u_i \leq U < u_{i+1}$, the gamma random variable (i.e $F_\alpha^{-1}(U)$) is given by equation 5.7.

We present a simple test to illustrate the accuracy of our cache as we increase the number of point, N. Specifically, we compute the root-mean-squared (RMS) relative error and the RMS absolute error of gamma random variables generated using the cache at $u = \{0.000001, 0.000002, \ldots, 0.999999\}$. The results are presented in table 5.1 and 5.2. As we see from the tables, for a large alpha, using approximately 100 points produces a very accurate cache while, for a small alpha, more points are required. Additionally, the time taken to create the cache is insignificant and therefore, gamma random variables can be sampled efficiently using the approach presented in this section.

6. Features of the New Schemes

6.1. **Features of the GammaQE Scheme and the Double Gamma Scheme.** Based on the result in Proposition 3, one can see that the proportion of the mean and variance of the variance process explained by the gamma random variable increases as the step size increases. In other words, for a long simulation step size, the variance process is driven mainly by the gamma random variable. When constructing the GammaQE scheme, we sample the gamma random variable precisely from the inverse of the gamma distribution while the compound Poisson distribution is approximated using a smooth vary mixture of zero-modified exponential distribution and zero-modified quadratic Gaussian distribution. Therefore, the GammaQE scheme is naturally well adapted to simulating the variance process for a medium or long simulation step size and, at a same time, giving a reasonable approximation to the variance process for short time steps. Whilst for the Double Gamma scheme, the variance process is sampled exactly. Hence, regardless of the simulation step size, the accuracy of the Double Gamma scheme solely depends on the accuracy of the cache.

While both schemes have a fixed dimension (i.e. the total number of random numbers required at each time step are fixed) with 3 random numbers required at each simulation time step, the GammaQE scheme has an additional virtue of being a smooth numerical scheme of model inputs. Unlike the Double Gamma scheme where a small change in the model inputs can change the outcome of the Poison discrete random variable and therefore, potentially causing an abrupt change to the outcome of the variance process, under the GammaQE scheme, the outcome of the variance process changes smoothly with respect to the model inputs. We wish to stress that while sensitivities evaluated using a discontinuous scheme can have high variance, such approach is still better than the acceptance and rejection method where the dimensionality of the scheme is unknown in advance.

6.2. **Features of the Integrated Variance Scheme.** By construction, the accuracy of the integrated variance scheme can be improved by using more terms in the series (i.e increasing k). However, this comes with a price of higher computational time. For every increase in k by one, 3 additional random variables - 2 gamma random variables and 1 Poisson random variable - are required. Our numerical results suggest that, even by truncating after the first term, our new approaches perform significantly better than the existing methods.

The simulation time of the integrated variance process for a step varies according to the step size. In short, the sampling time reduces as the size step increases. At each step, the Bessel random variable, η, and all the Poisson random variables, N_{λ_j}, are sampled through computing the probability mass function recursively starting with the probability of zero. For cases with a long step size, the Bessel random variable and the Poisson random variables have a light-tailed probability distribution with a large portion of the probability mass concentrated at around zero. Hence, sampling the Bessel random variable and the Poisson random variables is relatively fast for such cases as only a few iterations are required. Also, when approximating the probability of zero for the Bessel random variable, the truncation level is selected dynamically. In general, for a long simulation step, fewer terms are required to approximate the probability at zero accurately, as the constant $D(u,t)$ (see, equation 4.6) is inversely proportional to the size of the time-step.

Another interesting properties of the scheme for the integrated variance process is that it has a fixed dimension as the random number required at each time step can be determined before the start of the simulation. Once we have decided on the truncation level, k, only $2+3k$ random numbers are required at each time step.

7. Numerical Test Specifications

There are 2 parts to our numerical tests. We first test the accuracy and the efficiency of the GammaQE scheme and the Double Gamma scheme in simulating the variance process. As the true distribution of the variance process, V_t, is known, we compare the simulated results against the true distribution to gauge the magnitude of the discretization bias. In order to do so, we generate a very large sample of V_1 starting from V_0 using the inputs outlined in Table 7.1. We then construct a distribution function from the generated samples and compare it against the true distribution function. The $\mathcal{L}^2$-norm of the difference between the true distribution function and the simulated distribution function is used as a measure of the discretization error. Specifically, the $\mathcal{L}^2$-norm error is defined to be

$$\sqrt{\int_0^\infty \Big(F_{\hat{V}_1}(v) - F_{V_1}(v)\Big)^2 dv}. \tag{7.1}$$

where $F_{\hat{V}_1}(v)$ and $F_{V_1}(v)$ are the simulated and true distribution function of V_1 respectively. As a direct computation of the $\mathcal{L}^2$-norm error is rather complicated, we approximate the integral using numerical integration. In particular, we limit the domain of our integration to $[0, 2]$ (as the probability of V_1 to be greater than 2 is close to zero for the test inputs in table 7.1) and we subdivide the integration domain into $20,000$ subintervals when computing the integral in (7.1). We use the well known QE scheme introduced by Andersen (2008) as a benchmark for our test. To ensure that the simulated distribution function fully converges, a large sample size of 2^{25} Sobol paths is used. In our numerical test, we also consider using different discretization step sizes ranging from 1 step per year to 32 steps per year.

In the second part of our numerical tests, we consider pricing European options using the Integrated GammaQE scheme (i.e GammaQE scheme and the Integrated Variance scheme) and the Integrated Double Gamma scheme (i.e Double Gamma scheme and the Integrated Variance scheme) with truncation level, $k = 1$ and $k = 3$. The test inputs are given by Table 7.1. For each set of inputs, we compute prices of European call options with one year maturity for strike prices of $90, 100$ and 110. We also compute prices of double digital options where the payoff is defined as follow

$$\text{Payoff} = \begin{cases} 1 & K_1 \leq S_T < K_2, \\ 0 & \text{otherwise.} \end{cases} \tag{7.2}$$

Since the interest rate, r_t, is zero, the price of a double digital option also represents the probability of S_T finishing between K_1 and K_2, i.e. $\mathbb{P}(K_1 \leq S_T < K_2)$. In our numerical test, we price ten double digital options with one-year maturity. The strikes for the double digital options are given by Table 7.2. The strikes are selected such that all the double digital options have a price of 0.1 and the strikes do not overlap. As semi-closed form solutions for prices of double digital options and European call options exist, the true prices of these options can be calculated directly (without the need of Monte-Carlo simulation) and they are used to determine the accuracy of the simulated prices. Again, we use the QE scheme as a benchmark for our numerical tests. To ensure that the difference between true prices and simulated prices is purely due to discretization errors, the simulated prices are evaluated using 2^{26} Sobol paths to minimize the Monte-Carlo noise.

8. Numerical Results

In this section, we present the numerical results. All tests were carried out using a 3.16 Ghz Intel Core 2 Duo PC with 4 Gb of RAM, with single threaded C++ code. We first consider the discretization errors of the GammaQE scheme and the Double Gamma scheme in simulating the variance process. Specifically, we

	Case 1	Case 2	Case 3
ϵ	1.0	0.9	1.0
κ	0.5	0.3	1.0
V_0, θ	0.04	0.04	0.09
S_0	100	100	100
ρ	-0.9	-0.5	-0.3
r_t	0.0	0.0	0.0

Table 7.1. Inputs for numerical tests.

	Case 1		Case 2		Case 3	
	K_1	K_2	K_1	K_2	K_1	K_2
DD1	0.00	87.03	0.00	67.24	0.00	83.9
DD2	87.03	97.98	67.24	81.57	83.9	94.04
DD3	97.98	101.42	81.57	89.77	94.04	97.95
DD4	101.42	103.14	89.77	95.63	97.95	100.23
DD5	103.14	104.26	95.63	100.48	100.23	101.95
DD6	104.26	105.16	100.48	105.07	101.95	103.52
DD7	105.16	106.05	105.07	110.08	103.52	105.29
DD8	106.05	107.13	110.08	116.62	105.29	107.77
DD9	107.13	109.00	116.62	128.36	107.77	112.92
DD10	109.00	∞	128.36	∞	112.92	∞

Table 7.2. Strikes for Double Digital Options.

look at the $\mathcal{L}^2$-norm of the difference between the true distribution function and the simulated distribution function for each scheme. Next, we study the pricing bias of the Integrated GammaQE scheme and the Integrated Double Gamma scheme. The speed-accuracy tradeoff between these schemes are also investigated.

8.1. **Discretization errors in simulating the Variance Process.** The table 8.1 shows the $\mathcal{L}^2$-norm of the difference between the true distribution function and the simulated distribution function, while table 8.2, table 8.3 and table 8.4 tabulate the value of the simulated distribution function and the true distribution function at some specific points for all relevant step sizes. As expected, the $\mathcal{L}^2$-norm error for QE scheme decreases as we increase the number of simulation steps per year. For the Double Gamma scheme and the GammaQE scheme, the $\mathcal{L}^2$-norm error is lowest for cases with one simulation step per year and increases slightly for cases with two or more steps per year. The slight increase in the Double Gamma's $\mathcal{L}^2$-norm errors can be attributed to simulation noises as intermediate values of the variance process are sampled in obtaining the terminal value of the process. While for the GammaQE scheme, the higher $\mathcal{L}^2$-norm errors are attributed to simulation noises as well as discretization errors of the scheme. As explained earlier, the GammaQE scheme is naturally well adapted for a medium or long simulation step size and hence, the discretization error can increase as the step size decreases.

As we see from table 8.1, for any given step size, the Double Gamma scheme has the lowest $\mathcal{L}^2$-norm error followed closely by the GammaQE scheme while the QE scheme has a significantly higher $\mathcal{L}^2$-norm error. The main reason that the QE scheme[1] has a higher $\mathcal{L}^2$-norm error for a given step size is that it fails to match the true distribution function satisfactorily for small V_1 as we can see from table 8.2, table 8.3 and table 8.4.

While the Double Gamma scheme and the GammaQE scheme outperform the QE scheme for a given step size, this comes with a price of higher computational cost. This is because the QE scheme only uses one random variable per step to approximate the variance process while the Double Gamma scheme and the GammaQE scheme require three random variables per step. In order to study the speed-accuracy tradeoff

[1] Note that, a refined QE scheme was also proposed by Anderson (2008). However, based on our test similar conclusion is obtained that is the Double Gamma scheme and the GammaQE scheme have lower $\mathcal{L}^2$-norm errors than the refined QE scheme.

	Case 1			Case 2			Case 3		
Steps	GammaQE	Double Gamma	QE	GammaQE	Double Gamma	QE	GammaQE	Double Gamma	QE
1	0.0002%	0.0002%	1.1133%	0.0003%	0.0001%	0.7977%	0.0004%	0.0002%	2.8834%
2	0.0130%	0.0003%	0.6322%	0.0170%	0.0004%	0.4180%	0.0190%	0.0006%	1.7311%
4	0.0212%	0.0007%	0.4376%	0.0240%	0.0011%	0.2936%	0.0412%	0.0016%	1.1240%
8	0.0171%	0.0024%	0.2999%	0.0160%	0.0022%	0.2025%	0.0393%	0.0026%	0.7109%
16	0.0106%	0.0025%	0.2002%	0.0088%	0.0024%	0.1347%	0.0296%	0.0019%	0.4383%
32	0.0076%	0.0023%	0.1294%	0.0061%	0.0022%	0.0867%	0.0196%	0.0026%	0.2630%

Table 8.1. The tables shows the $\mathcal{L}^2$-norm of the difference between the true density function and the simulated density function for all three cases.

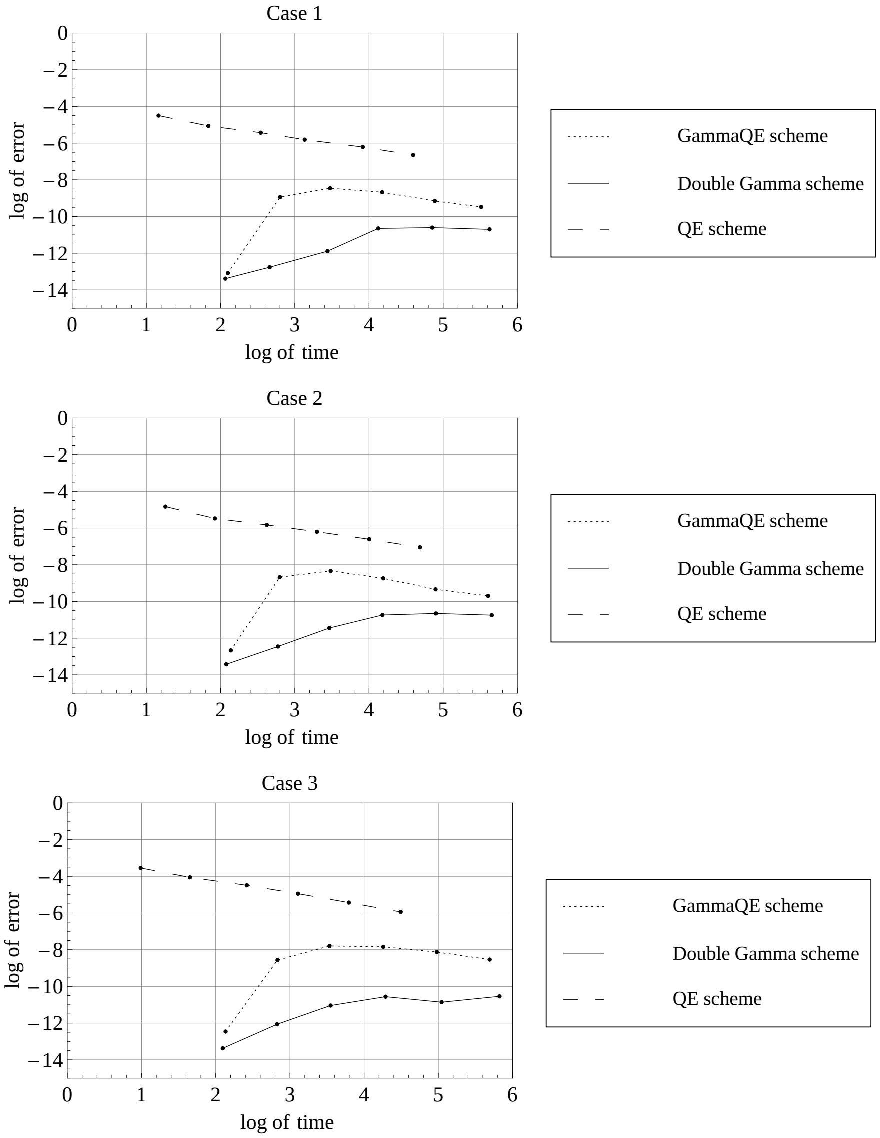

Figure 8.1. The graphs above show the plot for the log of the error against the log of the computational time for all 3 test cases.

between these schemes, we plot the log of the $\mathcal{L}^2$-norm error against the log of the computational time. From figure 8.1, we see that the plot for all 3 cases are very similar and we can conclude that the Double Gamma scheme outperforms the GammaQE scheme and the QE scheme.

	actual	GammaQE Scheme						Double Gamma Scheme						QE Scheme					
v	$P(V_1 < v)$	1 step	2 steps	4 steps	8 steps	16 steps	32 steps	1 step	2 steps	4 steps	8 steps	16 steps	32 steps	1 step	2 steps	4 steps	8 steps	16 steps	32 steps
0.0001	0.6901	0.6901	0.6901	0.6903	0.6905	0.6904	0.6904	0.6901	0.6901	0.6901	0.6901	0.6901	0.6900	0.8810	0.8443	0.8209	0.7998	0.7793	0.7592
0.0005	0.7360	0.7360	0.7361	0.7364	0.7365	0.7365	0.7365	0.7360	0.7360	0.7360	0.7360	0.7360	0.7359	0.8812	0.8449	0.8220	0.8015	0.7820	0.7639
0.0010	0.7567	0.7567	0.7568	0.7571	0.7573	0.7573	0.7572	0.7567	0.7567	0.7567	0.7567	0.7567	0.7567	0.8813	0.8455	0.8230	0.8032	0.7850	0.7690
0.0050	0.8072	0.8072	0.8074	0.8078	0.8080	0.8079	0.8077	0.8072	0.8072	0.8072	0.8071	0.8072	0.8071	0.8827	0.8495	0.8298	0.8146	0.8041	0.8000
0.0100	0.8301	0.8301	0.8304	0.8308	0.8309	0.8307	0.8303	0.8301	0.8301	0.8301	0.8301	0.8301	0.8301	0.8845	0.8538	0.8371	0.8266	0.8226	0.8250
0.0500	0.8869	0.8869	0.8872	0.8875	0.8872	0.8868	0.8866	0.8869	0.8869	0.8869	0.8869	0.8869	0.8869	0.8974	0.8816	0.8793	0.8832	0.8871	0.8873
0.1000	0.9135	0.9135	0.9137	0.9137	0.9133	0.9133	0.9134	0.9135	0.9135	0.9135	0.9135	0.9135	0.9135	0.9116	0.9068	0.9102	0.9138	0.9140	0.9135
0.5000	0.9761	0.9761	0.9760	0.9760	0.9762	0.9761	0.9761	0.9761	0.9761	0.9761	0.9761	0.9761	0.9761	0.9731	0.9768	0.9765	0.9762	0.9761	0.9761
1.0000	0.9936	0.9936	0.9936	0.9937	0.9936	0.9936	0.9936	0.9936	0.9936	0.9936	0.9936	0.9936	0.9936	0.9939	0.9937	0.9936	0.9936	0.9936	0.9936
1.5000	0.9982	0.9982	0.9982	0.9982	0.9982	0.9982	0.9982	0.9982	0.9982	0.9982	0.9982	0.9982	0.9982	0.9986	0.9982	0.9982	0.9982	0.9982	0.9982

TABLE 8.2. The table compares the simulated distribution function and the true distribution function of Case 1 for all relevant step sizes.

	actual	GammaQE Scheme						Double Gamma Scheme						QE Scheme					
v	$P(V_1 < v)$	1 step	2 steps	4 steps	8 steps	16 steps	32 steps	1 step	2 steps	4 steps	8 steps	16 steps	32 steps	1 step	2 steps	4 steps	8 steps	16 steps	32 steps
0.0001	0.7334	0.7334	0.7335	0.7338	0.7338	0.7338	0.7337	0.7334	0.7334	0.7334	0.7334	0.7334	0.7334	0.8768	0.8442	0.8264	0.8115	0.7967	0.7819
0.0005	0.7693	0.7693	0.7694	0.7698	0.7698	0.7698	0.7697	0.7693	0.7693	0.7692	0.7693	0.7693	0.7692	0.8769	0.8450	0.8275	0.8131	0.7993	0.7863
0.0010	0.7853	0.7853	0.7855	0.7859	0.7859	0.7859	0.7858	0.7853	0.7853	0.7853	0.7853	0.7853	0.7853	0.8771	0.8456	0.8286	0.8148	0.8020	0.7910
0.0050	0.8241	0.8241	0.8245	0.8250	0.8249	0.8248	0.8245	0.8241	0.8241	0.8241	0.8241	0.8242	0.8241	0.8786	0.8499	0.8354	0.8256	0.8196	0.8184
0.0100	0.8419	0.8419	0.8423	0.8428	0.8426	0.8424	0.8420	0.8419	0.8419	0.8418	0.8419	0.8419	0.8419	0.8805	0.8544	0.8426	0.8366	0.8358	0.8390
0.0500	0.8878	0.8878	0.8883	0.8884	0.8879	0.8877	0.8877	0.8878	0.8878	0.8878	0.8879	0.8879	0.8878	0.8944	0.8823	0.8821	0.8860	0.8882	0.8881
0.1000	0.9115	0.9115	0.9118	0.9116	0.9113	0.9114	0.9115	0.9115	0.9115	0.9115	0.9115	0.9116	0.9115	0.9094	0.9069	0.9101	0.9121	0.9118	0.9115
0.5000	0.9754	0.9754	0.9753	0.9753	0.9754	0.9754	0.9754	0.9754	0.9754	0.9754	0.9754	0.9754	0.9754	0.9736	0.9761	0.9757	0.9755	0.9754	0.9754
1.0000	0.9940	0.9940	0.9940	0.9941	0.9940	0.9940	0.9940	0.9940	0.9940	0.9940	0.9940	0.9940	0.9941	0.9943	0.9940	0.9940	0.9940	0.9940	0.9940
1.5000	0.9985	0.9985	0.9985	0.9985	0.9985	0.9985	0.9985	0.9985	0.9985	0.9985	0.9985	0.9985	0.9985	0.9988	0.9985	0.9985	0.9985	0.9985	0.9985

TABLE 8.3. The table compares the simulated distribution function and the true distribution function of Case 2 for all relevant step sizes.

	actual	GammaQE Scheme						Double Gamma Scheme						QE Scheme					
v	$P(V_1 < v)$	1 step	2 steps	4 steps	8 steps	16 steps	32 steps	1 step	2 steps	4 steps	8 steps	16 steps	32 steps	1 step	2 steps	4 steps	8 steps	16 steps	32 steps
0.0001	0.2286	0.2286	0.2286	0.2288	0.2289	0.2290	0.2291	0.2286	0.2286	0.2286	0.2285	0.2285	0.2285	0.6555	0.5704	0.5074	0.4507	0.4002	0.3554
0.0005	0.3053	0.3053	0.3054	0.3057	0.3060	0.3061	0.3063	0.3053	0.3053	0.3053	0.3053	0.3053	0.3053	0.6561	0.5722	0.5105	0.4556	0.4075	0.3667
0.0010	0.3459	0.3459	0.3460	0.3464	0.3468	0.3469	0.3471	0.3459	0.3459	0.3459	0.3459	0.3459	0.3458	0.6567	0.5740	0.5136	0.4604	0.4150	0.3785
0.0050	0.4617	0.4617	0.4620	0.4627	0.4632	0.4633	0.4632	0.4617	0.4617	0.4617	0.4617	0.4617	0.4617	0.6619	0.5858	0.5330	0.4910	0.4627	0.4500
0.0100	0.5225	0.5225	0.5229	0.5238	0.5242	0.5241	0.5235	0.5225	0.5225	0.5225	0.5225	0.5225	0.5225	0.6683	0.5988	0.5538	0.5230	0.5093	0.5115
0.0500	0.6924	0.6924	0.6929	0.6936	0.6932	0.6920	0.6918	0.6924	0.6924	0.6924	0.6924	0.6925	0.6924	0.7154	0.6826	0.6748	0.6825	0.6919	0.6935
0.1000	0.7763	0.7763	0.7766	0.7767	0.7759	0.7756	0.7762	0.7763	0.7763	0.7763	0.7763	0.7763	0.7762	0.7650	0.7581	0.7664	0.7762	0.7775	0.7765
0.5000	0.9543	0.9543	0.9541	0.9541	0.9544	0.9543	0.9543	0.9543	0.9543	0.9543	0.9543	0.9543	0.9543	0.9492	0.9558	0.9550	0.9543	0.9543	0.9543
1.0000	0.9909	0.9910	0.9910	0.9910	0.9910	0.9909	0.9909	0.9909	0.9909	0.9909	0.9909	0.9910	0.9910	0.9925	0.9912	0.9909	0.9909	0.9909	0.9909
1.5000	0.9981	0.9981	0.9981	0.9981	0.9981	0.9981	0.9981	0.9981	0.9981	0.9981	0.9981	0.9981	0.9981	0.9989	0.9981	0.9980	0.9981	0.9981	0.9981

TABLE 8.4. The table compares the simulated distribution function and the true distribution function of Case 3 for all relevant step sizes.

8.2. **Pricing Bias.** The table 8.5, 8.6 and 8.6 show the estimated bias in option prices evaluated using the Integrated GammaQE scheme, the Integrated Double gamma scheme and the QE scheme. As we see from the tables, the estimated bias of the QE scheme decreases as the number of steps per year increases. While the QE scheme with 16 steps per year produces reasonably accurate prices for the European call options, the relative error for prices of double digital options can still be huge. As we see from table 8.5, the relative error for DD7 is approximated 6.5%.

Unlike the QE scheme, there are two variables - the number of steps per year and the truncation level - that determine the accuracy of the Integrated GammaQE scheme and the Integrated Double Gamma Scheme. In general, for a fixed step size, an increase in the truncation level reduces the discretization bias for the Integrated GammaQE scheme and Integrated Double Gamma scheme. While the discretization bias of the Integrated Double Gamma scheme reduces as the number of steps per year increases, the Integrated GammaQE scheme shows a mix result. In particular, as we see from table 8.5, the estimated bias for the price of the call option with a strike of 90 increases as we increase the number of steps per year (i.e. reducing the step size). This is not surprising given that the GammaQE scheme, by construction, works well for a medium and long simulation step size and hence, the pricing bias will inevitably increase as the step size is reduced.

To allow for a comprehensive speed-accuracy tradeoff comparison, we plot the log of the root mean squared relative errors (RMSE) for the European options against the log of relative computational time required in figure 8.2. Overall, all three plots are very similar. The first interesting observation is that, for the Integrated GammaQE scheme (regardless of the truncation level), the log of RMSE can increase as the number of steps per year increases (i.e. increase simulation time), and this observation has been discussed above. For the Integrated Double Gamma scheme (with $k = 1$ and $k = 3$), the log of RMSE decreases and then flattens as the log of simulation time increases. The RMSE at the flat section of the graphs can be viewed as errors due to Monte-Carlo noises and it is similar to the RMSE of the QE scheme with 512 steps per year.

In general, for any given simulation time, the Integrated Double Gamma scheme with $k = 3$ has the lowest RMSE followed by the Integrated Double Gamma scheme with $k = 1$ and the QE scheme. As for the Integrated GammaQE scheme, it has a similar speed-accuracy tradeoff as the Integrated Double Gamma scheme for large step sizes (i.e 1 to 4 steps per year) and it becomes less efficient as the step size decreases (i.e more than 4 steps per year). Our numerical results suggest that the most efficient way in reducing the discretization bias for the Integrated GammaQE scheme and the Integrated Double Gamma scheme is to increase the truncation level, k rather than increasing the number of steps per year. Therefore, when it comes to pricing a path-dependent financial product, the product itself will determine the number of time steps per year while users of these schemes will select a suitable truncation level.

9. Conclusion

Three new schemes for the Heston Stochastic volatility model are presented in this paper. Two schemes for simulating the variance process - known as the GammaQE scheme and the Double Gamma - and one scheme for simulating the integrated variance process conditional on the initial and the end-point of the variance process. All these scheme have a fixed dimension with the total random number required at each step can be determined before the start of the simulation. While both the GammaQE scheme and the Double Gamma scheme can evolve the variance process accurately over long steps without the need to sample the intervening values, the GammaQE scheme has the additional virtues of being a smooth numerical scheme of the model

			DD1	DD2	DD3	DD4	DD5	DD6	DD7	DD8	DD9	DD10	Call 90	Call 100	Call 110
True Price			0.100	0.100	0.100	0.100	0.100	0.100	0.100	0.100	0.100	0.100	12.7585	4.4032	0.2892

Integrated GammaQE K=1

steps per year	time (relative)	RMSE	DD1	DD2	DD3	DD4	DD5	DD6	DD7	DD8	DD9	DD10	Call 90	Call 100	Call 110
1	4.9	0.0239	- 0.0001	- 0.0007	0.0045	0.0028	- 0.0013	- 0.0033	- 0.0032	- 0.0030	0.0009	0.0036	- 0.0003	- 0.0078	- 0.0002
2	9.3	0.0106	- 0.0000	- 0.0005	0.0006	0.0019	0.0003	- 0.0015	- 0.0019	- 0.0011	0.0015	0.0007	0.0014	- 0.0010	- 0.0009
4	18.2	0.0031	- 0.0001	- 0.0001	- 0.0001	0.0004	0.0004	- 0.0004	- 0.0006	- 0.0002	0.0006	- 0.0000	0.0029	0.0013	- 0.0000
8	37.1	0.0011	- 0.0000	- 0.0003	0.0000	0.0000	0.0001	- 0.0000	0.0001	- 0.0001	0.0002	- 0.0000	0.0032	0.0012	- 0.0005
16	77.3	0.0021	0.0001	- 0.0003	- 0.0002	- 0.0002	0.0001	0.0002	0.0004	0.0001	0.0001	- 0.0003	0.0020	0.0012	- 0.0007

Integrated GammaQE K=3

steps per year	time (relative)	RMSE	DD1	DD2	DD3	DD4	DD5	DD6	DD7	DD8	DD9	DD10	Call 90	Call 100	Call 110
1	7.8	0.0046	0.0000	- 0.0003	0.0003	0.0009	0.0000	- 0.0007	- 0.0006	- 0.0006	0.0006	0.0003	0.0002	- 0.0006	- 0.0007
2	15.4	0.0014	- 0.0000	- 0.0000	- 0.0000	0.0001	0.0002	- 0.0002	- 0.0001	- 0.0002	0.0003	- 0.0000	0.0013	0.0008	0.0000
4	31.0	0.0008	- 0.0001	- 0.0000	0.0001	- 0.0000	0.0001	- 0.0001	0.0001	- 0.0001	0.0001	0.0000	0.0033	0.0016	0.0003
8	63.8	0.0010	- 0.0000	- 0.0003	0.0001	- 0.0001	0.0000	0.0000	0.0002	- 0.0001	0.0001	- 0.0000	0.0034	0.0014	- 0.0004
16	133.9	0.0020	0.0001	- 0.0003	- 0.0003	- 0.0001	0.0001	0.0002	0.0004	0.0001	0.0001	- 0.0003	0.0018	0.0012	- 0.0006

Integrated Double Gamma K=1

steps per year	time (relative)	RMSE	DD1	DD2	DD3	DD4	DD5	DD6	DD7	DD8	DD9	DD10	Call 90	Call 100	Call 110
1	4.8	0.0239	- 0.0001	- 0.0007	0.0045	0.0028	- 0.0013	- 0.0033	- 0.0032	- 0.0030	0.0009	0.0036	- 0.0003	- 0.0078	- 0.0002
2	9.1	0.0106	0.0001	- 0.0005	0.0007	0.0019	0.0004	- 0.0015	- 0.0019	- 0.0011	0.0015	0.0006	0.0002	- 0.0017	- 0.0013
4	18.2	0.0031	0.0000	- 0.0001	- 0.0001	0.0004	0.0004	- 0.0004	- 0.0006	- 0.0003	0.0006	- 0.0001	0.0002	0.0002	- 0.0003
8	37.1	0.0007	0.0000	- 0.0000	0.0000	- 0.0000	0.0001	- 0.0001	0.0000	- 0.0001	0.0002	- 0.0000	0.0003	0.0004	- 0.0000
16	77.9	0.0007	0.0000	- 0.0001	0.0001	- 0.0001	0.0000	- 0.0001	0.0001	- 0.0001	0.0001	- 0.0000	0.0000	0.0002	- 0.0001

Integrated Double Gamma K=3

steps per year	time (relative)	RMSE	DD1	DD2	DD3	DD4	DD5	DD6	DD7	DD8	DD9	DD10	Call 90	Call 100	Call 110
1	7.8	0.0046	0.0000	- 0.0003	0.0003	0.0009	0.0000	- 0.0007	- 0.0006	- 0.0006	0.0006	0.0003	0.0002	- 0.0006	- 0.0007
2	15.3	0.0014	0.0000	- 0.0000	- 0.0000	0.0001	0.0001	- 0.0002	- 0.0001	- 0.0002	0.0003	- 0.0000	0.0001	0.0002	- 0.0002
4	31.0	0.0007	0.0000	- 0.0000	0.0001	- 0.0000	0.0000	- 0.0001	0.0001	- 0.0001	0.0001	- 0.0000	- 0.0001	0.0001	- 0.0002
8	63.8	0.0005	0.0000	- 0.0000	0.0000	- 0.0001	- 0.0000	- 0.0000	0.0001	- 0.0001	0.0001	- 0.0000	0.0003	0.0004	- 0.0001
16	133.9	0.0007	0.0001	- 0.0001	0.0001	- 0.0000	- 0.0000	- 0.0001	0.0001	- 0.0001	0.0001	- 0.0000	- 0.0001	0.0001	- 0.0002

QE Scheme

steps per year	time (relative)	RMSE	DD1	DD2	DD3	DD4	DD5	DD6	DD7	DD8	DD9	DD10	Call 90	Call 100	Call 110
1	1.0	0.5258	- 0.0187	0.0808	0.0673	- 0.0040	- 0.0368	- 0.0505	- 0.0536	- 0.0473	- 0.0204	0.0831	- 0.2668	- 0.0529	0.2614
2	1.4	0.2838	- 0.0014	0.0080	0.0553	0.0131	- 0.0195	- 0.0351	- 0.0388	- 0.0313	- 0.0020	0.0517	- 0.0480	- 0.0573	0.0581
4	2.3	0.1429	0.0007	- 0.0030	0.0222	0.0170	- 0.0059	- 0.0204	- 0.0242	- 0.0167	0.0084	0.0218	- 0.0277	- 0.0305	0.0002
8	4.3	0.0626	- 0.0002	0.0011	0.0043	0.0096	0.0003	- 0.0097	- 0.0126	- 0.0068	0.0080	0.0059	- 0.0183	- 0.0123	- 0.0012
16	8.0	0.0302	- 0.0002	0.0006	0.0028	0.0040	0.0008	- 0.0047	- 0.0065	- 0.0031	0.0038	0.0025	- 0.0087	- 0.0084	0.0009
32	15.6	0.0115	- 0.0001	- 0.0000	0.0010	0.0015	0.0006	- 0.0018	- 0.0025	- 0.0011	0.0015	0.0008	- 0.0023	- 0.0031	- 0.0000
64	30.8	0.0038	0.0000	- 0.0001	0.0003	0.0005	0.0003	- 0.0006	- 0.0007	- 0.0005	0.0006	0.0002	- 0.0003	- 0.0008	- 0.0001
128	61.7	0.0013	0.0000	- 0.0000	0.0001	0.0001	0.0001	- 0.0002	- 0.0002	- 0.0002	0.0002	0.0001	0.0001	0.0001	- 0.0000
256	125.7	0.0007	0.0000	- 0.0000	0.0001	0.0000	0.0001	- 0.0001	0.0000	- 0.0002	0.0001	0.0000	0.0002	0.0002	- 0.0002
512	262.4	0.0006	0.0000	- 0.0001	0.0000	- 0.0000	0.0001	- 0.0001	0.0001	- 0.0001	0.0001	0.0000	0.0004	0.0005	- 0.0001

TABLE 8.5. The table above shows the estimated bias for Case 1

	DD1	DD2	DD3	DD4	DD5	DD6	DD7	DD8	DD9	DD10	Call 90	Call 100	Call 110
True Price	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	12.7578	5.0998	1.7003

Integrated GammaQE K=1

steps per year	time (relative)	RMSE	DD1	DD2	DD3	DD4	DD5	DD6	DD7	DD8	DD9	DD10	Call 90	Call 100	Call 110
1	4.8	0.0247	- 0.0008	0.0024	0.0041	- 0.0002	- 0.0032	- 0.0038	- 0.0034	- 0.0007	0.0042	0.0013	- 0.0092	0.0193	- 0.0006
2	9.3	0.0106	- 0.0001	- 0.0002	0.0017	0.0009	- 0.0011	- 0.0020	- 0.0014	0.0006	0.0018	- 0.0003	- 0.0008	0.0031	- 0.0027
4	18.2	0.0026	- 0.0000	- 0.0001	0.0002	0.0005	- 0.0002	- 0.0006	- 0.0003	0.0004	0.0002	- 0.0001	- 0.0010	- 0.0013	- 0.0019
8	37.1	0.0009	- 0.0001	- 0.0001	- 0.0000	0.0001	0.0001	- 0.0000	0.0000	0.0002	- 0.0000	- 0.0001	- 0.0007	- 0.0026	- 0.0021
16	77.4	0.0028	0.0000	- 0.0004	- 0.0004	0.0001	0.0004	0.0005	0.0004	0.0001	- 0.0003	- 0.0003	0.0009	- 0.0039	- 0.0018

Integrated GammaQE K=3

steps per year	time (relative)	RMSE	DD1	DD2	DD3	DD4	DD5	DD6	DD7	DD8	DD9	DD10	Call 90	Call 100	Call 110
1	7.8	0.0051	- 0.0000	- 0.0002	0.0009	0.0004	- 0.0006	- 0.0009	- 0.0007	0.0002	0.0009	- 0.0002	- 0.0005	0.0014	- 0.0013
2	15.5	0.0013	- 0.0000	- 0.0001	0.0001	0.0002	- 0.0001	- 0.0003	- 0.0001	0.0002	0.0001	- 0.0000	0.0004	0.0002	- 0.0002
4	31.0	0.0003	- 0.0001	0.0000	- 0.0000	0.0000	0.0000	- 0.0000	0.0000	0.0000	0.0000	- 0.0000	- 0.0000	- 0.0008	- 0.0009
8	63.9	0.0008	- 0.0001	- 0.0001	0.0000	0.0000	0.0001	0.0001	0.0001	0.0000	- 0.0000	- 0.0001	- 0.0005	- 0.0025	- 0.0021
16	135.2	0.0028	- 0.0000	- 0.0004	- 0.0003	0.0000	0.0004	0.0005	0.0004	0.0001	- 0.0003	- 0.0003	0.0013	- 0.0039	- 0.0017

Integrated Double Gamma K=1

steps per year	time (relative)	RMSE	DD1	DD2	DD3	DD4	DD5	DD6	DD7	DD8	DD9	DD10	Call 90	Call 100	Call 110
1	4.8	0.0247	- 0.0008	0.0025	0.0042	- 0.0002	- 0.0032	- 0.0038	- 0.0034	- 0.0008	0.0042	0.0013	- 0.0092	0.0193	- 0.0006
2	9.3	0.0106	- 0.0001	- 0.0002	0.0017	0.0009	- 0.0011	- 0.0020	- 0.0014	0.0006	0.0018	- 0.0003	- 0.0011	0.0029	- 0.0027
4	18.2	0.0027	0.0000	- 0.0001	0.0002	0.0005	- 0.0002	- 0.0006	- 0.0003	0.0004	0.0002	- 0.0001	0.0001	0.0001	- 0.0004
8	37.3	0.0004	0.0000	- 0.0000	0.0000	0.0000	0.0000	- 0.0001	- 0.0000	0.0001	- 0.0000	0.0000	- 0.0002	- 0.0002	- 0.0002
16	78.2	0.0001	0.0000	- 0.0000	- 0.0000	0.0000	- 0.0000	0.0000	- 0.0000	0.0000	- 0.0000	0.0000	- 0.0006	- 0.0004	- 0.0004

Integrated Double Gamma K=5

steps per year	time (relative)	RMSE	DD1	DD2	DD3	DD4	DD5	DD6	DD7	DD8	DD9	DD10	Call 90	Call 100	Call 110
1	7.7	0.0051	- 0.0000	- 0.0002	0.0009	0.0004	- 0.0006	- 0.0009	- 0.0007	0.0002	0.0009	- 0.0002	- 0.0005	0.0014	- 0.0013
2	15.5	0.0012	0.0000	- 0.0001	0.0001	0.0002	- 0.0001	- 0.0003	- 0.0001	0.0002	0.0001	- 0.0000	- 0.0000	- 0.0000	- 0.0002
4	31.1	0.0002	- 0.0000	0.0000	- 0.0000	0.0000	0.0000	- 0.0000	- 0.0000	0.0000	- 0.0000	0.0000	0.0001	- 0.0000	- 0.0001
8	64.1	0.0002	- 0.0000	0.0000	- 0.0000	0.0000	- 0.0000	- 0.0000	0.0000	0.0000	0.0000	- 0.0000	0.0000	- 0.0001	- 0.0002
16	135.9	0.0002	- 0.0000	- 0.0000	0.0000	- 0.0000	- 0.0000	0.0000	- 0.0000	0.0000	0.0000	- 0.0000	- 0.0005	- 0.0006	- 0.0004

QE Scheme

steps per year	time (relative)	RMSE	DD1	DD2	DD3	DD4	DD5	DD6	DD7	DD8	DD9	DD10	Call 90	Call 100	Call 110
1	1.0	0.4801	0.0137	0.0972	0.0103	- 0.0322	- 0.0490	- 0.0541	- 0.0507	- 0.0352	0.0114	0.0886	- 0.0143	1.0496	0.6888
2	1.6	0.2943	- 0.0057	0.0574	0.0250	- 0.0153	- 0.0342	- 0.0402	- 0.0362	- 0.0186	0.0261	0.0418	- 0.0992	0.4459	0.1921
4	2.6	0.1577	- 0.0022	0.0167	0.0244	- 0.0023	- 0.0199	- 0.0260	- 0.0218	- 0.0051	0.0253	0.0109	- 0.0454	0.1649	0.0296
8	4.4	0.0771	- 0.0002	0.0031	0.0127	0.0029	- 0.0093	- 0.0144	- 0.0109	0.0012	0.0132	0.0018	- 0.0131	0.0607	0.0053
16	8.5	0.0398	- 0.0002	0.0017	0.0058	0.0026	- 0.0046	- 0.0082	- 0.0058	0.0017	0.0061	0.0010	- 0.0073	0.0285	0.0026
32	17.1	0.0127	- 0.0001	0.0004	0.0016	0.0012	- 0.0013	- 0.0029	- 0.0018	0.0009	0.0017	0.0002	- 0.0022	0.0075	0.0004
64	32.7	0.0040	- 0.0000	0.0001	0.0004	0.0004	- 0.0003	- 0.0010	- 0.0005	0.0003	0.0005	0.0001	- 0.0004	0.0023	0.0003
128	68.1	0.0012	- 0.0000	0.0001	0.0001	0.0002	- 0.0001	- 0.0003	- 0.0002	0.0001	0.0001	0.0000	- 0.0001	0.0006	- 0.0001
256	132.2	0.0004	- 0.0000	0.0000	0.0000	0.0001	- 0.0001	- 0.0001	- 0.0001	- 0.0000	0.0001	0.0000	0.0000	0.0004	0.0002
512	277.8	0.0003	0.0000	0.0000	- 0.0000	0.0000	- 0.0000	- 0.0001	- 0.0000	- 0.0000	0.0000	0.0000	0.0001	0.0004	0.0005

Table 8.6. The table above shows the estimated bias for Case 2

			DD1	DD2	DD3	DD4	DD5	DD6	DD7	DD8	DD9	DD10	Call 90	Call 100	Call 110
True Price			0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	15.8641	9.7738	5.7434

Integrated GammaQE K=1

steps per year	time (relative)	RMSE	DD1	DD2	DD3	DD4	DD5	DD6	DD7	DD8	DD9	DD10	Call 90	Call 100	Call 110
1	4.8	0.0027	- 0.0001	0.0000	0.0005	0.0002	- 0.0002	- 0.0004	- 0.0004	- 0.0002	0.0004	0.0001	- 0.0013	0.0047	0.0052
2	9.6	0.0007	- 0.0000	- 0.0000	0.0000	0.0001	0.0000	- 0.0001	- 0.0001	0.0000	0.0001	- 0.0000	- 0.0003	0.0007	0.0008
4	19.3	0.0003	- 0.0000	- 0.0000	0.0000	- 0.0000	0.0001	0.0000	- 0.0000	0.0000	0.0000	- 0.0000	- 0.0015	- 0.0021	- 0.0023
8	40.3	0.0013	- 0.0000	- 0.0002	- 0.0001	- 0.0001	0.0001	0.0002	0.0002	0.0001	- 0.0001	- 0.0002	- 0.0012	- 0.0049	- 0.0057
16	86.5	0.0016	0.0000	- 0.0001	- 0.0002	- 0.0001	0.0001	0.0003	0.0003	0.0001	- 0.0002	- 0.0001	0.0005	- 0.0036	- 0.0042

Integrated GammaQE K=3

steps per year	time (relative)	RMSE	DD1	DD2	DD3	DD4	DD5	DD6	DD7	DD8	DD9	DD10	Call 90	Call 100	Call 110
1	8.0	0.0002	- 0.0000	- 0.0000	0.0000	0.0000	- 0.0000	- 0.0000	- 0.0000	- 0.0000	0.0000	- 0.0000	- 0.0002	0.0001	0.0001
2	16.3	0.0002	- 0.0000	0.0000	0.0000	- 0.0000	0.0000	- 0.0000	- 0.0000	0.0000	0.0000	0.0000	0.0001	0.0000	0.0000
4	33.6	0.0003	- 0.0001	- 0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	- 0.0000	- 0.0006	- 0.0014	- 0.0016
8	70.5	0.0013	- 0.0000	- 0.0002	- 0.0001	- 0.0000	0.0001	0.0002	0.0001	0.0001	- 0.0000	- 0.0002	- 0.0015	- 0.0051	- 0.0059
16	153.8	0.0015	0.0000	- 0.0001	- 0.0002	- 0.0001	0.0001	0.0003	0.0003	0.0001	- 0.0002	- 0.0002	- 0.0004	- 0.0042	- 0.0049

Integrated Double Gamma K=1

steps per year	time (relative)	RMSE	DD1	DD2	DD3	DD4	DD5	DD6	DD7	DD8	DD9	DD10	Call 90	Call 100	Call 110
1	4.8	0.0027	- 0.0001	0.0000	0.0005	0.0003	- 0.0002	- 0.0004	- 0.0004	- 0.0002	0.0004	0.0001	- 0.0014	0.0046	0.0051
2	9.6	0.0007	- 0.0000	- 0.0000	0.0000	0.0001	0.0000	- 0.0001	- 0.0001	0.0000	0.0001	- 0.0000	- 0.0004	0.0006	0.0008
4	19.4	0.0002	- 0.0000	0.0000	- 0.0000	0.0000	0.0000	- 0.0000	- 0.0000	- 0.0000	0.0000	0.0000	0.0000	- 0.0002	- 0.0001
8	40.9	0.0001	- 0.0000	0.0000	0.0000	- 0.0000	0.0000	- 0.0000	0.0000	0.0000	- 0.0000	0.0000	0.0001	- 0.0000	- 0.0001
16	88.9	0.0002	0.0000	0.0001	- 0.0000	- 0.0000	- 0.0000	- 0.0000	0.0000	0.0000	0.0000	0.0000	- 0.0001	- 0.0000	- 0.0002

Integrated Double Gamma K=3

steps per year	time (relative)	RMSE	DD1	DD2	DD3	DD4	DD5	DD6	DD7	DD8	DD9	DD10	Call 90	Call 100	Call 110
1	8.0	0.0002	- 0.0000	- 0.0000	0.0000	0.0000	- 0.0000	- 0.0000	- 0.0000	- 0.0000	0.0000	- 0.0000	- 0.0003	0.0000	0.0001
2	16.3	0.0002	- 0.0000	0.0000	0.0000	- 0.0000	0.0000	- 0.0000	0.0000	0.0000	- 0.0000	0.0000	0.0001	0.0001	0.0001
4	33.8	0.0002	- 0.0000	- 0.0000	0.0000	- 0.0000	- 0.0000	0.0000	- 0.0000	- 0.0000	0.0000	- 0.0000	- 0.0002	- 0.0002	- 0.0002
8	71.0	0.0002	- 0.0000	0.0000	- 0.0000	- 0.0000	- 0.0000	0.0000	0.0000	- 0.0000	0.0000	- 0.0000	- 0.0003	- 0.0003	- 0.0004
16	156.0	0.0002	- 0.0000	0.0000	0.0000	0.0000	- 0.0001	0.0000	0.0000	- 0.0000	0.0000	- 0.0000	0.0001	0.0002	0.0001

QE Scheme

steps per year	time (relative)	RMSE	DD1	DD2	DD3	DD4	DD5	DD6	DD7	DD8	DD9	DD10	Call 90	Call 100	Call 110
1	1.0	0.1903	- 0.0028	0.0323	0.0205	- 0.0027	- 0.0182	- 0.0257	- 0.0256	- 0.0166	0.0075	0.0313	0.3164	0.7760	0.8175
2	1.6	0.0722	- 0.0010	0.0049	0.0112	0.0034	- 0.0063	- 0.0120	- 0.0120	- 0.0049	0.0091	0.0076	0.0519	0.2240	0.2293
4	2.7	0.0234	0.0001	0.0011	0.0026	0.0020	- 0.0015	- 0.0044	- 0.0043	- 0.0007	0.0038	0.0013	0.0146	0.0670	0.0673
8	5.0	0.0115	- 0.0002	0.0010	0.0014	0.0008	- 0.0007	- 0.0022	- 0.0021	- 0.0004	0.0015	0.0009	0.0050	0.0327	0.0347
16	9.7	0.0035	- 0.0001	0.0002	0.0005	0.0003	- 0.0003	- 0.0006	- 0.0006	- 0.0001	0.0004	0.0003	0.0010	0.0095	0.0102
32	18.5	0.0011	- 0.0000	0.0001	0.0002	0.0001	- 0.0001	- 0.0002	- 0.0002	- 0.0000	0.0001	0.0001	0.0004	0.0031	0.0033
64	38.0	0.0005	- 0.0000	0.0000	0.0001	0.0000	0.0000	- 0.0001	- 0.0001	- 0.0000	0.0000	0.0000	0.0004	0.0013	0.0014
128	74.6	0.0001	- 0.0000	0.0000	- 0.0000	0.0000	0.0000	- 0.0000	- 0.0000	0.0000	0.0000	0.0000	0.0001	0.0002	0.0002
256	151.9	0.0002	0.0000	0.0000	- 0.0000	0.0000	0.0000	- 0.0000	- 0.0000	0.0000	- 0.0000	- 0.0000	0.0001	0.0002	0.0004
512	307.7	0.0003	0.0000	- 0.0000	0.0001	- 0.0000	- 0.0000	- 0.0000	- 0.0000	- 0.0000	0.0000	- 0.0000	- 0.0002	0.0002	0.0002

TABLE 8.7. The table above shows the estimated bias for Case 3

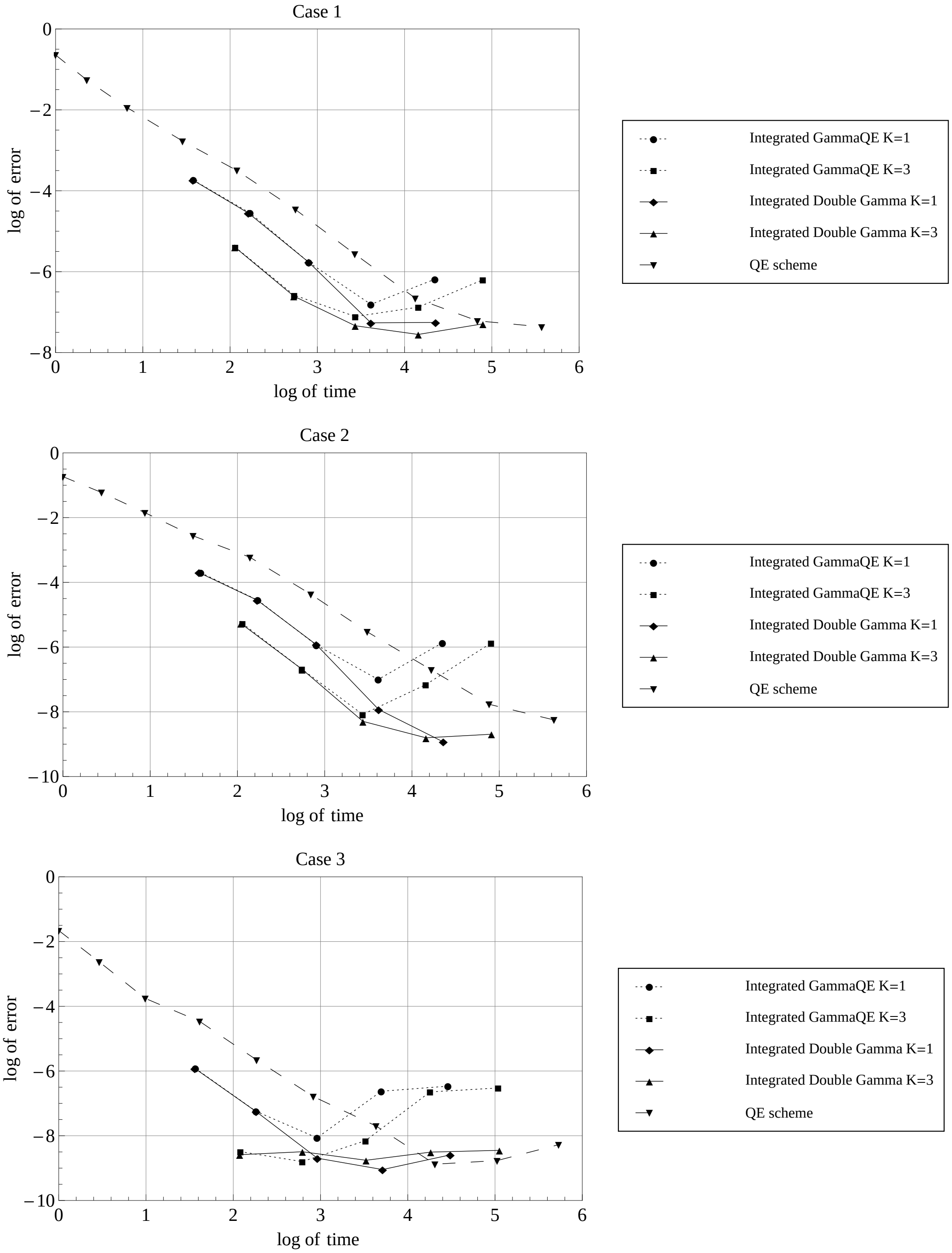

FIGURE 8.2. The graphs above show the plot for the log of the pricing error against the log of the computational time for all 3 test cases.

inputs and it requires less caching. Numerical results suggest that both the GammaQE scheme and the Double Gamma scheme perform significantly better than the QE scheme of Anderson (2008) when it comes to simulating the variance process. Unlike most existing schemes where short stepping approach is used, under the Integrated GammaQE scheme and the Integrated Double Gamma scheme (combining the new integrated variance scheme with the GammaQE scheme and the Double Gamma scheme respectively), prices of financial derivatives in the Heston model can be evaluated accurately using a long stepping approach. Our numerical results suggest that the Integrated GammaQE scheme and the Integrated Double Gamma scheme are more efficient than the well-established QE scheme when it comes to the pricing of financial derivatives. We therefore conclude that our new schemes can be used simulated the Heston process efficiently in a practical setting particularly when the intermediate values of the process is not required.

APPENDIX A

A1 Derivation of Algorithm 1. In deriving the results in Algorithm 1, we first compute the mean, M_{Y_2}, and variance, $(S_{Y_2})^2$, of the compound Poisson distribution and they are given by

$$\begin{aligned} M_{Y_2} &= \lambda\beta, \\ (S_{Y_2})^2 &= 2\lambda\beta^2. \end{aligned}$$

We then define

$$w = P(N_\lambda \leq 1) = (1+\lambda)e^{-\lambda}$$

to be the proportion of the mean and the variance of the compound Poisson distribution that must be explained (i.e. matched) by the zero-modified exponential random variable, Y_3, i.e.

$$\mathbb{E}(Y_3) = w \cdot M_{Y_2}, \qquad \mathrm{Var}(Y_3) \;=\; w \cdot (S_{Y_2})^2. \tag{A-1}$$

Since the mean and the variance of the zero-modified exponential random variable, Y_3, are given by

$$\mathbb{E}(Y_3) = (1-P_{Y_3})\mu, \qquad \mathrm{Var}(Y_3) = (1-(P_{Y_3})^2)\mu^2 \tag{A-2}$$

we therefore have

$$P_{Y_3} = \frac{2-\lambda w}{2+\lambda w}, \qquad \mu = \frac{w\lambda\beta}{1-P_{Y_3}}. \tag{A-3}$$

The residual mean and variance of the compound Poisson distribution are then explained (i.e. matched) by the zero-modified Quadratic Gaussian random variable, Y_4, i.e.

$$\mathbb{E}(Y_4) = (1-w)\cdot M_{Y_2}, \qquad \mathrm{Var}(Y_4) = (1-w)\cdot(S_{Y_2})^2. \tag{A-4}$$

As the zero-modified Quadratic Gaussian distribution has three parameters (i.e. a,b and P_{Y_4}) and we have only two constraints to satisfy, we are left with one free parameter. This extra degree of freedom is used to match the probability at zero i.e.

$$P_{Y_3} \cdot P_{Y_4} = P(Y_2 = 0) = e^{-\lambda}. \tag{A-5}$$

Since the mean and variance of a zero-modified Quadratic Gaussian distribution can be calculated easily using a direct integration with

$$\begin{aligned} \mathbb{E}(Y_4) &= a(1+b^2)(1-P_{Y_4}) \\ \mathrm{Var}(Y_4) &= a^2(3+6b^2+b^4)(1-P_{Y_4}) - (a(1+b^2)(1-P_{Y_4}))^2, \end{aligned}$$

one can therefore solve for a and b^2, and they are given by

$$b^2 = -c-1+\sqrt{c(c+1)}, \quad a = \frac{(1-w)\lambda\beta}{(1-P_{Y_4})(1+b^2)},$$

where

$$c = \frac{2\lambda(1-w)}{\lambda(1-w)-(1-P_{Y_4})(2+\lambda(1-w))}$$

A2 Existence of solutions. In this section, we show that the solutions for a, b^2, μ, P_{Y_3} and P_{Y_4} always exist. In particular, we prove that $a >$, $b^2 > 0$, $\mu > 0$, $P_{Y_3} \in [0,1)$ and $P_{Y_4} \in [0,1)$. From equation (A-3), one can see that solution of μ and P_{Y_3} always exists as w, λ and β are always greater than zero. Similarly, the solution for a exists as long as we have $P_{Y_4} \in [0,1)$ and $b^2 > 0$. In the following subsections, we first show that $P_{Y_4} \in [0,1)$ followed by $b^2 > 0$.

A2-1 Existence of P_{Y_4}. In order to show that $P_{Y_4} \in [0,1)$, we first recall that

$$P_{Y_4} = \frac{e^{-\lambda}}{P_{Y_3}}.$$

We can easily deduce that $P_{Y_4} \geq 0$ as $e^{-\lambda}, P_{Y_3} \geq 0$. Proving that $P_{Y_4} < 1$ is equivalent to proving that

$$P_{Y_3} > e^{-\lambda}$$

Since

$$\begin{aligned}
P_{Y_3} - e^{-\lambda} &= \frac{2-\lambda(1+\lambda)e^{-\lambda}}{2+\lambda(1+\lambda)e^{-\lambda}} - e^{-\lambda} \\
&= \frac{e^{-2\lambda}}{2+\lambda(1+\lambda)e^{-\lambda}}\left(2e^{2\lambda} - 2e^{\lambda} - \lambda(1+\lambda)e^{\lambda} - \lambda(1+\lambda)\right) \\
&= \frac{e^{-2\lambda}}{2+\lambda(1+\lambda)e^{-\lambda}}\left(\left[2e^{\lambda} - \lambda(1+\lambda)\right]\left[e^{\lambda} - 1\right] - 2\lambda(1+\lambda)\right) \\
&= \frac{e^{-2\lambda}}{2+\lambda(1+\lambda)e^{-\lambda}}\left(\left[2+\lambda+\sum_{n=3}^{\infty}\frac{\lambda^n}{n!}\right]\left[\lambda+\frac{\lambda^2}{2}+\sum_{n=3}^{\infty}\frac{\lambda^n}{n!}\right] - 2\lambda(1+\lambda)\right) \\
&= \frac{e^{-2\lambda}}{2+\lambda(1+\lambda)e^{-\lambda}}\left(\frac{\lambda^3}{2} + (e^{\lambda}+1+\lambda)\sum_{n=3}^{\infty}\frac{\lambda^n}{n!}\right) \\
&> 0 \qquad \text{for} \quad \lambda > 0,
\end{aligned} \tag{A-6}$$

we therefore have $P_{Y_3} > e^{-\lambda}$ and we conclude that $P_{Y_4} \in [0,1)$.

A2-2 Existence of b^2. In order to show that

$$b^2 = -c - 1 + \sqrt{c(c+1)} > 0,$$

we first identify the valid range of c such that $b^2 > 0$. In fact, one can easily verify that

$$c < -1 \quad \Longrightarrow \quad b^2 > 0.$$

As c is a function of λ, the solution of b^2 will always exist provided that $c < -1$ for all $\lambda > 0$. For clarity, we divide our proof into several parts. We first show that the denominator of c, i.e

$$\lambda(1-w) - (1-P_{Y_4})(2+\lambda(1-w))$$

is always less than zero all $\lambda > 0$. Once this have been shown, proving that

$$c = \frac{2\lambda(1-w)}{\lambda(1-w)-(1-P_{Y_4})(2+\lambda(1-w))} < -1$$

for all positive λ, is therefore equivalent to showing

$$3\lambda(1-w)-(1-P_{Y_4})(2+\lambda(1-w))>0$$

for all positive λ.

Proposition 7. *The expression*

$$\lambda(1-w)-(1-P_{Y_4})(2+\lambda(1-w)) \tag{A-7}$$

is less than zero provided that

$$x_n = 4\cdot 3^n - 2^n\left(4+2n+\frac{n^2-n}{2}\right) - n^3+2n^2-n>0 \quad \textit{for} \quad n\geq 5$$

and $\lambda > 0$

Proof. Using the fact that

$$e^{\lambda}=\sum_{n=0}^{\infty}\frac{\lambda^n}{n!},$$

the expression in equation (A-7) can be written as

$$\begin{aligned} & \lambda(1-w)-(1-P_{Y_4})(2+\lambda(1-w)) \\ = & \frac{-e^{-3\lambda}}{2-\lambda(1+\lambda)e^{-\lambda}}\left(4e^{3\lambda}-2(2+2\lambda+\lambda^2)e^{2\lambda}-\lambda^2(1+\lambda)e^{\lambda}+\lambda^2(1+\lambda)^2\right) \\ = & \frac{-e^{-2\lambda}}{2e^{\lambda}-\lambda(1+\lambda)}\left(\frac{2}{3}\lambda^3+\lambda^4+\sum_{n=5}^{\infty}\frac{x_n\lambda^n}{n!}\right) \end{aligned}$$

where

$$x_n = 4\cdot 3^n - 2^n\left(4+2n+\frac{n^2-n}{2}\right) - n^3+2n^2-n$$

Provided that $\lambda > 0$ and $x_n > 0$ for $n \geq 5$ then the result in proposition 7 will hold as

$$2e^{\lambda}-\lambda(1+\lambda)=2+\lambda+\sum_{n=3}^{\infty}\frac{\lambda^n}{n!}>0.$$

□

Proposition 8. *The series,*

$$x_n = 4\cdot 3^n - 2^n\left(4+2n+\frac{n^2-n}{2}\right) - n^3+2n^2-n \tag{A-8}$$

is always greater than zero for $n \geq 5$.

Proof. The result in Proposition 8 can be derived easily using mathematical induction. One can easily show that

$$x_{n+1}=2x_n+4\left(3^n-2^n-\frac{n\cdot 2^n}{2}\right)+n^3-5n^2+2n.$$

By using Taylors theorem, we have

$$3^n = 2^n + n\cdot 2^{n-1}+\frac{1}{2}n(n-1)2^{n-2}+\ldots+\frac{1}{(n-1)!}n(n-1)\ldots 2\cdot 1\cdot 2^1,$$

therefore, we can conclude that

$$3^n-2^n-\frac{n\cdot 2^n}{2}>0 \qquad \text{for} \quad n\geq 3$$

as the higher order terms are positive. Also, for $n \geq 5$, we have

$$n(n^2 - 5n + 2) > 0$$

As $x_5 > 0$, we conclude that $x_n > 0$ for $n \geq 5$. □

Proposition 9. *The expression*

$$3\lambda(1-w) - (1 - P_{Y_4})(2 + \lambda(1-w)) \tag{A-9}$$

is greater than zero provided that

$$y_n = 4(2^n - 3^n) + \frac{n}{6}\left(8 \cdot 3^n + 3 \cdot 2^n - 30\right) - \frac{n^2}{4}\left(2^n - 40\right) - \frac{n^3}{4}\left(2^n + 28\right) + 2n^4 > 0$$

and $\lambda > 0$.

Proof. Similar to the proof for Proposition 7, one can easily show that

$$\begin{aligned} & 3\lambda(1-w) - (1 - P_{Y_4})(2 + \lambda(1-w)) \\ = \; & \frac{e^{-2\lambda}}{2e^{\lambda} - \lambda(1+\lambda)} \left(\frac{4}{3}\lambda^3 + \frac{8}{3}\lambda^4 + \sum_{n=5}^{\infty} \frac{y_n \lambda^n}{n!} \right) > 0 \end{aligned}$$

provided that $\lambda > 0$ and $y_n > 0$ for $n \geq 5$ where

$$y_n = 4(2^n - 3^n) + \frac{n}{6}\left(8 \cdot 3^n + 3 \cdot 2^n - 30\right) - \frac{n^2}{4}\left(2^n - 40\right) - \frac{n^3}{4}\left(2^n + 28\right) + 2n^4$$

□

Proposition 10. *The series*

$$y_n = 4(2^n - 3^n) + \frac{n}{6}\left(8 \cdot 3^n + 3 \cdot 2^n - 30\right) - \frac{n^2}{4}\left(2^n - 40\right) - \frac{n^3}{4}\left(2^n + 28\right) + 2n^4, \tag{A-10}$$

is always greater than zero for $n \geq 5$.

Proof. Similar to the proof for Proposition 8, we can easily derive the result above using mathematical induction. We have

$$y_5 = 256, \qquad y_6 = 742 \qquad \text{and} \qquad y_7 = 2936,$$

and a simple manipulation gives

$$y_{n+1} = 2y_n + \frac{s_n}{6},$$

where

$$s_n = 8 \cdot 3^n + 90n^2 + 72 - 15 \cdot 2^n - 9n \cdot 2^n - 12n^3 - 114n.$$

Using the fact that,

$$3^n > 2^n + \frac{n}{1!}2^{n-1} + \frac{n(n-1)}{2!}2^{n-2} + \frac{n(n-1)(n-2)}{3!}2^{n-3} \qquad \text{for} \qquad n \geq 7,$$

we therefore have,

$$s_n > \left[90n^2 - 114n + 72\right] + \frac{2^n}{16}\left[3n^2 - 112\right] + \frac{n \cdot 2^n}{96}\left[7n^2 + 30n - 544\right] + \frac{3n^3}{32}\left[2^n - 128\right].$$

for $n \geq 7$. One can easily check that all the second order polynomials in the inequality above is greater than 0 for $n \geq 7$, and similarly,

$$2^n - 128 \geq 0 \qquad \text{for} \quad n \geq 7.$$

Therefore, we conclude that $s_n > 0$ for $n \geq 7$ and hence,

$$y_n > 0 \qquad \text{for} \quad n \geq 5$$

$\square$

Using results in Proposition 7 and Proposition 8, we can conclude that the denominator of c, i.e

$$\lambda(1-w) - (1 - P_{Y_4})(2 + \lambda(1-w))$$

is always less than zero for all $\lambda > 0$. We can therefore conclude that

$$3\lambda(1-w) - (1 - P_{Y_4})(2 + \lambda(1-w)) > 0$$

for all positive λ using the results from Proposition 9 and Proposition 10. Hence, the solution for $b^2 > 0$ always exists as $c < -1$ for all $\lambda > 0$.

References

[1] Alfonsi, A. (2005). On the discretization schemes for the CIR (and Bessel squared) processes. *Monte Carlo Methods and Applications* **11**, 355-384.

[2] Alfonsi, A. (2008). High order discretization schemes for the CIR process: application to affine term structure and Heston models. *Mathematics of Computation, forthcoming.*

[3] Andersen, L. (2008). Simple and efficient simulation of the Heston stochastic volatility model. *The Journal of Computational Finance* **11**(3), 1-42.

[4] Broadie, M., and Kaya, Ö. (2006). Exact simulation of stochastic volatility and other affine jump diffusion processes. *Operations Research* **54**(2), 217-231.

[5] Cox, J., Ingersoll, J., and Ross, S. A. (1985). A theory of term structure of interest rates. *Econometrica***53**(2), 185-407.

[6] Dufresne, D. (2001). The integrated square-root process. Working Paper, University of Montreal.

[7] Glasserman, P. (2003). *Monte Carlo Methods in Financial Engineering (Stochactic Modelling and Applied Probability).* Springer, New York.

[8] 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

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

[10] Lord, R., Koekkoek, R., van Dijk, D. (2008) A comparison of biased simulation schemes for stochastic volatility models. *Quantitative Finance, forthcoming.* http://www.rogerlord.com.

[11] Jäckel, P. (2002). *Monte Carlo Methods in Finance.* Wiley, New York.

[12] Joshi, M. S. (2003). *The Concepts and Practice of Mathematical Finance.* Cambrigde University Press, Cambridge.

[13] Scott, L.O. (1996). *Simulating a multi-factor term structure model over relatively long discrete time periods, in Proceedings of the IAFE First Annual Computational Finance Conference.* Graduate School of Business, Stanford University

[14] Van Haastrecht, A., Pelsser, A. (2008) Effcient, almost exact simulation of the Heston stochastic volatility model. Working paper, University of Amsterdam. http://papers.ssrn.com.

Centre for Actuarial Studies, Department of Economics, University of Melbourne, Victoria 3010, Australia

E-mail address: `j.chan23@pgrad.unimelb.edu.au`

E-mail address: `mark@markjoshi.com`