

FIRST AND SECOND ORDER GREEKS IN THE HESTON MODEL

JIUN HONG CHAN AND MARK JOSHI

ABSTRACT. In this paper, we present an efficient approach to compute the first and the second order price sensitivities in the Heston model using the algorithmic differentiation approach. Issues related to the applicability of the pathwise method are discussed in this paper as most existing numerical schemes are not Lipschitz in model inputs. Depending on the model inputs and the discretization step size, our numerical tests show that the sample means of price sensitivities obtained using the Lognormal scheme and the Quadratic-Exponential scheme can be highly skewed and have fat-tailed distribution while price sensitivities obtained using the Integrated Double Gamma scheme and the Double Gamma scheme remain stable.

1. INTRODUCTION

The use of stochastic volatility models to evaluate prices of financial derivatives among market practitioners has increased in the past few years. These models provide a better calibration to market-implied volatility smiles and skews whilst providing realistic dynamics to the underlying stock. The Heston model has become particularly popular because of the availability of closed-form formulas for the Fourier transform of the price of European options. Unlike the Black-Scholes model where the instantaneous 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.

Despite its popularity, there had been little research on 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. This is followed by the introduction of other more efficient short-stepping schemes including the Quadratic-Exponential (QE) scheme (see, Andersen 2008) and Alfonsi's Second Order Scheme (see, Alfonsi 2008). Building on the works of Glasserman and Kim (2008), Chan and Joshi (2010) developed a long-stepping scheme known as the Integrated Double Gamma scheme. This scheme evolves the Heston process accurately over long steps without the need to sample the intervening values and, hence, prices of financial derivatives can be evaluated rapidly. While most authors have demonstrated that their respective schemes are efficient for pricing purposes, only one paper, Broadie and Kaya (2004), has looked at the issue of computing sensitivities and even there the analysis only addresses the delta, the gamma and the rho-derivative. Thus no work has been published on the problem of computing sensitivities with respect to the parameters of the volatility process. This is truly surprising given the importance of Greeks for hedging and risk-management.

Typically, when computing price sensitivities, one will differentiate the underlying numerical scheme and, in doing so, one is effectively differentiating the approximation (to the true process). This has the advantage that differentiating the approximation is generally much easier, particularly for the Heston model. When one has a different approximation to the true process, the derivative of this approximation will also be different. Hence, numerical schemes can give substantially different results for Greeks even when they have similar effects on pricing.

Date: December 1, 2010.

Key words and phrases. Heston stochastic volatility, first and second order Greeks, Algorithmic differentiation, simulation schemes.

The main focus of this paper is to present an efficient approach to compute the first and the second order price sensitivities in the Heston model and compare this approach using different numerical schemes. We first address the issue of discontinuous numerical schemes. When computing price sensitivities using the pathwise method, one is effectively interchanging the order of differentiation and expectation. This requires both the pay-off and the underlying scheme to be Lipschitz continuous. However, as noted by Anderson (2008), the behavior of the Heston process varies significantly depending on the current level of the variance process. Therefore, most efficient numerical schemes such as the QE scheme and Alfonsi’s Second Order Scheme, typically approximate the Heston process by switching between two or more different approximations depending on the level of the variance process. All these schemes can be viewed as regime-switching schemes and so they are generally not Lipschitz continuous in model inputs. Hence, the pathwise method can no longer be applied directly. In order to compute the price sensitivities for these scheme, a combination of the likelihood ratio method (LRM) and the pathwise method can be used (see, Section 4). Roughly, the LRM is used to compute the sensitivity due to the switching between approximations while the pathwise method is used to compute the sensitivity within a smooth approximation.

When implementing the sensitivity computations, we adopt the backward algorithmic differentiation approach introduced by Joshi and Yang (2010). The main advantage of their approach is that, as long as one can decompose a complicated mapping function into multiple simple mapping functions then the price sensitivities (both gradient and Hessian) can be evaluated easily. In this paper, we work with complicated mapping functions directly to reduce the computational time. We also present an efficient algorithm for updating the gradient and the Hessian of the price for such complicated mapping functions.

In this paper, we consider computing price sensitivities using the Lognormal (LN) scheme, the Quadratic-Exponential (QE) scheme, the Double Gamma (DG) scheme and the Integrated Double gamma (IDG) scheme. These four schemes are selected for the following reasons: the LN scheme is a smooth function of inputs; the QE scheme is considered as one of the most efficient short-stepping scheme for pricing purposes; the DG scheme uses the exact representation to sample the variance process; and the IDG scheme is an efficient long-stepping scheme for pricing purposes. We did not consider Alfonsi’s Second Order scheme in this paper as this scheme has multiple discontinuities and it approximates the variance process using discrete random variables. This makes computing unbiased price sensitivities very difficult if not impossible. Our numerical results show that, depending on the model inputs and the discretization step-size, price sensitivities obtained using the QE scheme can potentially blow-up even for first order price sensitivities. Similarly, the sample means of the second order price sensitivities obtained using the LN scheme can be highly skewed and have fat-tailed distributions while the results for the IDG scheme and the DG scheme remain relatively stable.

This paper is organized as follows: in section 2, we present a brief review of the Heston model followed by an overview of the numerical schemes used in this paper in section 3. In section 4, we discuss the applicability of the pathwise method. An efficient approach to compute the price sensitivities is outlined in section 5 with numerical results presented in section 6. We conclude in section 7.

2. HESTON STOCHASTIC VOLATILITY MODEL

2.1. Review of the Heston Model. In this section, we provide a brief review of the Heston model and the contents presented here are well-known. The Heston process is described by the following stochastic

differential equations

$$\frac{dS(t)}{S(t)} = r(t)dt + \sqrt{V(t)}dW^1(t) \quad (2.1)$$

$$dV(t) = \kappa(\theta - V(t))dt + \epsilon\sqrt{V(t)}dW^2(t) \quad (2.2)$$

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 $(W^1(t), W^2(t))$ is a two-dimensional Brownian motion with an instantaneous correlation of ρ (i.e. $dW^1(t)dW^2(t) = \rho dt$), 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, an exact solution for the asset price process at time t conditional on $V(u)$ and $S(u)$ with $u < t$ is given by*

$$S(t) = S(u) \exp \left(\int_u^t r(s)ds + \frac{\rho}{\epsilon}(V(t) - V(u) - \kappa\theta(t - u)) + \left(\frac{\kappa\rho}{\epsilon} - \frac{1}{2} \right) \int_u^t V(s)ds + \sqrt{1 - \rho^2} \int_u^t \sqrt{V(s)}dW^3(s) \right) \quad (2.3)$$

where $W^3(t)$ is a one-dimensional Brownian motion independent of $W^1(t)$ and $W^2(t)$.

Proof. See Broadie and Kaya (2006) □

As noted by Broadie and Kaya, on 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

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

Therefore, the simulation problem is reduced to

- (1) sampling $V(t)$ conditional on $V(u)$,
- (2) sampling $\int_u^t V(s)ds$ conditional on $V(u)$ and $V(t)$,
- (3) sampling $S(t)$ conditional on $V(u)$, $V(t)$ and $\int_u^t V(s)ds$.

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 $\chi_k^2(q)$ represent a noncentral chi-squared random variable with k degrees of freedom and a non-centrality parameter of q where*

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

with $k, q > 0$. Conditioning on $V(u)$, 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 $n(u, t)V(u)$,

$$\text{i.e. } (V(t)|V(u)) \stackrel{d}{=} C(u, t)\chi_{\delta}^2(n(u, t)V(u)), \quad (2.5)$$

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. Conditioning of $V(u)$, the mean, m , and variance, s^2 , of $V(t)$ are given by

$$m = \theta + (V(u) - \theta)e^{-\kappa(t-u)}, \quad (2.6)$$

$$s^2 = \frac{V(u)\epsilon^2 e^{-\kappa(t-u)}}{\kappa} \left(1 - e^{-\kappa(t-u)}\right) + \frac{\theta\epsilon^2}{2\kappa} \left(1 - e^{-\kappa(t-u)}\right)^2. \quad (2.7)$$

Proposition 4. We let

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

We define

- $\Gamma(\alpha, \beta), \Gamma_1(\alpha, \beta), \Gamma_2(\alpha, \beta), \dots$ 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 ,
- $\text{Exp}(\beta), \text{Exp}_1(\beta), \text{Exp}_2(\beta), \dots$ to be independent and identically distributed exponential random variables with a mean of β .

The distribution of $V(t)$ conditional on $V(u)$ admits the representation

$$(V(t)|V(u)) \stackrel{d}{=} Y_1 + Y_2 \quad (2.8)$$

where

$$Y_1 \stackrel{d}{=} \Gamma(\alpha, \beta) \quad Y_2 \stackrel{d}{=} \sum_{i=1}^{N_{\lambda}} \text{Exp}_i(\beta) \quad (2.9)$$

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

The square-root process has been well-studied. The Feller condition 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 $2\kappa\theta \ll \epsilon$. The probability of the variance process, $V(t)$, hitting zero can therefore be quite significant.

2.3. Properties of Integrated Variance Process, $\int_u^t V(s)ds$, conditional on $V(t)$ and $V(u)$. Glasserman and Kim (2008) showed that the exact distribution of $\int_u^t V(s)ds$, conditional on $V(t)$ and $V(u)$ can be represented by infinite sums and mixtures of gamma random variables.

Proposition 5. We let

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

We define

$$X_1 \stackrel{d}{=} \sum_{j=1}^{\infty} \frac{1}{\gamma_j} \sum_{i=1}^{N_{\lambda_j}} \text{Exp}_i(1), \quad X_2 \stackrel{d}{=} \sum_{j=1}^{\infty} \frac{1}{\gamma_j} \Gamma_j(\alpha, 1), \quad Z \stackrel{d}{=} \sum_{j=1}^{\infty} \frac{1}{\gamma_j} \Gamma_j(2, 1),$$

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

$$\left(\int_u^t V(s)ds \middle| V(t), V(u) \right) \stackrel{d}{=} X_1 + X_2 + \sum_{i=1}^{\eta} Z_i \quad (2.10)$$

where $X_1, X_2, \eta, Z_1, Z_2 \dots$ are mutually independent random variable, the Z_i are i.i.d. with the same distribution as Z .

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

Proposition 6. Conditioning on $V(t)$ and $V(u)$, the mean and the variance for the random variable X_1 , X_2 and Z are given by

$$\begin{aligned} \mathbb{E}(X_1) &= (V(t) + V(u))\mu_{X_1}^*, & \text{Var}(X_1) &= (V(t) + V(u))\sigma_{X_1}^{*2}, \\ \mathbb{E}(X_2) &= \delta\mu_{X_2}^*, & \text{Var}(X_2) &= \delta\sigma_{X_2}^{*2}, \\ \mathbb{E}(Z) &= 4\mu_{X_2}^*, & \text{Var}(Z) &= 4\sigma_{X_2}^{*2}, \end{aligned}$$

where

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

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. □

3. EXISTING SIMULATION SCHEMES

In general, there are 2 classes of numerical schemes for simulating the Heston process: the short stepping schemes and the long stepping schemes. Due to the difficulty in sampling the integrated variance process, most short stepping schemes approximate this integral using the trapezoidal rule, i.e.

$$\left(\int_u^t V(s)ds \middle| V(t), V(u) \right) \approx \frac{1}{2}(V(t) + V(u))(t-u). \quad (3.1)$$

As this is a rather crude approximation, a fine partition of the simulation time step is usually needed and this approximation will converge to the true distribution when the step size approaches zero. For long-stepping schemes, more complicated approaches are used to approximate the integrated variance process. While this will increase the sampling time for the integrated variance process, the overall simulation time can be potentially reduced as there is no longer a need to sample the intervening values. Of all the schemes considered in this paper, only the IDG scheme belongs to the class of long stepping schemes.

3.1. Lognormal Scheme. The Lognormal (LN) scheme is constructed by matching the first two moments of the variance process across each step. Given $V(u)$, the variance process at time t , is sampled using the following representation:

$$V(t) = \exp(\mu + \sigma \cdot Z) \quad (3.2)$$

where

$$\sigma^2 = \log(\Psi + 1) \quad \mu = \log(m) - 0.5\sigma^2 \quad \Psi = \frac{s^2}{m^2}$$

and m and s^2 are defined in Proposition 3 and Z is a standard normal random variable. Besides guaranteeing the positivity of the variance process, the LN scheme is infinitely differentiable w.r.t model inputs and monotone in Z . Note that, one could also write equation (3.2) as

$$V(t) = \frac{m}{\sqrt{\Psi + 1}} \exp(\sqrt{\log(\Psi + 1)} \cdot Z) \quad (3.3)$$

and this representation is used section 5.2.1.

3.2. Quadratic Exponential Scheme. The Quadratic Exponential (QE) scheme was proposed by Anderson (2008). As the distribution of $V(t)$ has a strong dependency on $V(u)$, Anderson suggested two possible distributions to approximate the variance process, $V(t)$, depending on the size of $V(u)$. Specifically, for $\Psi \leq 2.0$, the sample $V(t)$ conditional on $V(u)$ can be generated from

$$V(t) = a(b + Z)^2 \quad (3.4)$$

where Z is a standard normal random variables. The values of a and b are determined by matching the first two moments of the variance process and they are given by

$$\begin{aligned} a &= \frac{m}{1 + b^2}, \\ b^2 &= 2\Psi^{-1} - 1 + \sqrt{2\Psi^{-1}} \sqrt{2\Psi^{-1} - 1}. \end{aligned}$$

We shall call this the *quadratic Gaussian approximation*.

Similarly, for $\Psi \geq 1.0$, the sample $V(t)$ conditional on $V(u)$ can be generated by inverting a zero-modified exponential distribution as follow

$$V(t) = \begin{cases} 0 & 0 \leq U \leq p \\ \beta^{-1} \ln\left(\frac{1-p}{1-U}\right) & p < U \leq 1 \end{cases} \quad (3.5)$$

where U is a uniform random variable. The values of β and p are again determined by moment matching and they are given by

$$\begin{aligned} p &= \frac{\Psi - 1}{\Psi + 1}, \\ \beta &= \frac{1 - p}{m}. \end{aligned}$$

We shall call this the *zero-modified exponential approximation*.

Since the domain of applicability of both distributions overlaps for $\Psi \in [1.0, 2.0]$, users of the QE scheme have the choice to select the switching rule between both distributions. In this paper, we follow the approach taken by Andersen (2008) setting the switching rule at $\Psi = 1.5$.

3.3. Double Gamma Scheme. The Double Gamma (DG) Scheme is constructed using Proposition 4. Specifically, the equation (2.8) can be rewritten as

$$(V(t)|V(u)) \stackrel{d}{=} \frac{1}{\beta} (\Gamma(\alpha, 1) + \Gamma(N_\lambda, 1)) \quad (3.6)$$

where $N_\lambda \in \mathbb{N}$ is a Poisson random variable with mean of λ . The crucial observation made by Chan and Joshi (2010) is that, conditional on $N_\lambda = n$, both $\Gamma(\alpha, 1)$ and $\Gamma(n, 1)$ are state and step-size independent, with the latter also being input independent. They therefore suggest to tabulate the inverse of the gamma distribution for the shape parameter α and for all the integer shape parameters. While the inverse for the shape parameter α has to be tabulated every time α changes, the tabulation of the inverse of the gamma distribution for all the integer shape parameters only has to be done once, since it can be reused to generate $\Gamma(N_\lambda, 1)$ even if the model inputs and state-variables have changed. Hence, the variance process $V(t)$ can be sampled easily once the outcome of the Poisson random variable N_λ is known.

3.4. Integrated Double Gamma Scheme. The Integrated Double Gamma (IDG) Scheme uses the same method to simulate the variance process, $V(t)$, as in the DG scheme. However, a more complicated approach is used to approximate the integrated variance process. This approach is based on the results of Proposition 5. Instead of expressing the integrated variance process as an infinite sum of mixtures of gamma random variables, Chan and Joshi (2010) truncated the series after k terms and approximated the remainder of the series using a Lognormal random variable. Specifically, under the IDG scheme, the integrated variance process is sampled from the following representation:

$$\left(\int_u^t V(s) ds \middle| V(t), V(u) \right) \stackrel{d}{=} \sum_{j=1}^k \frac{1}{\gamma_j} (\Gamma_{j1}(\alpha, 1) + \Gamma_{j2}(2\eta + N_{\lambda_j}, 1)) + \text{Ln}(\mu_{Ln}^k, (\sigma_{Ln}^k)^2) \quad (3.7)$$

where Ln is a Lognormal random variable with parameters μ_{Ln}^k and $(\sigma_{Ln}^k)^2$ determined by moment matching while Γ_{j1} and Γ_{j2} are independent Gamma random variable for $j = 1, 2, \dots, k$. The solutions for μ_{Ln}^k and $(\sigma_{Ln}^k)^2$ are given by

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

where

$$\begin{aligned} \mu_R^k &= (V(t) + V(u)) \left(\mu_{X_1}^* - \sum_{j=1}^k \left(\frac{\lambda_j^*}{\gamma_j} \right) \right) + (4\eta + 2\alpha) \left(\mu_{X_2}^* - \sum_{j=1}^k \left(\frac{1}{2\gamma_j} \right) \right) \\ (\sigma_R^k)^2 &= (V(t) + V(u)) \left((\sigma_{X_1}^*)^2 - \sum_{j=1}^k \left(\frac{2\lambda_j^*}{\gamma_j^2} \right) \right) + (4\eta + 2\alpha) \left((\sigma_{X_2}^*)^2 - \sum_{j=1}^k \left(\frac{1}{2\gamma_j^2} \right) \right) \end{aligned} \quad (3.8)$$

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 Γ_{j1} and Γ_{j2}
- (4) We repeat step 2 and 3 from $j = 1$ up to $j = k$

- (5) Lastly, we sample the Lognormal random variable, $\text{Ln}(\mu_{Ln}^k, \sigma_{Ln}^k)$, and the integrated variance process from time u to t is given by the result in equation 3.7

Sampling the Ln random variable is straight-forward while both the gamma random variables Γ_{j1} and Γ_{j2} can be sampled easily using the caches from the DG scheme. As for the Bessel random variable, η , the probability mass function is given by

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

and, hence, η can be sampled easily using the inverse transformation method. As noted by Chan and Joshi (2010), the most time-consuming step in sampling η is the computation of p_0 . They suggest computing p_0 using the following approximation

$$p_0 \approx \frac{1}{\sum_{n=0}^h C_n \cdot (D(u, t)V(u)V(t))^n} \quad (3.9)$$

where

$$C_0 = 1, \quad C_{n+1} = \frac{C_n}{4(n+1)(n+1+\nu)}, \quad D(u, t) = \left(\frac{2\kappa/\epsilon^2}{\sinh(\kappa(t-u)/2)} \right)^2. \quad (3.10)$$

and h is determined such that

$$\frac{C_h \cdot [D(u, t)V(u)V(t)]^h}{\sum_{n=0}^h C_n \cdot [D(u, t)V(u)V(t)]^n} < 10^{-4}$$

for the first time as h increases. Note that, using this approximation, we have

$$p_m = \frac{C_m [D(u, t)V(u)V(t)]^m}{\sum_{n=0}^h C_n \cdot [D(u, t)V(u)V(t)]^n}$$

4. APPLICATIONS OF THE PATHWISE METHOD

Suppose that g is a payoff function and $K_T(x, z)$ represents the realization of the underlying with an initial input $x \in \mathbf{X} \subseteq \mathbb{R}$ at maturity, T , generated using a vector of uniform random numbers $z \in \mathbf{Z}$ where \mathbf{X} is the set of all possible inputs and \mathbf{Z} is the set of all possible vectors of random numbers. The first order price sensitivity with respect to (w.r.t) x is defined to be

$$\frac{\partial}{\partial x} \mathbb{E}_Z(g(K_T(x, z))) = \lim_{h \rightarrow 0} \mathbb{E}_Z \left[\frac{g(K_T(x+h, z)) - g(K_T(x, z))}{h} \right]. \quad (4.1)$$

where \mathbb{E}_Z denotes the expectation taken w.r.t to the density of z . The applicability of the pathwise method to computing price sensitivities depends on the validity of interchanging the order of differentiation and expectation (integration), i.e.

$$\frac{\partial}{\partial x} \mathbb{E}_Z(g(K_T(x, z))) = \mathbb{E}_Z \left[\frac{\partial}{\partial x} g(K_T(x, z)) \right] \quad (4.2)$$

$$= \mathbb{E}_Z \left[\lim_{h \rightarrow 0} \frac{g(K_T(x+h, z)) - g(K_T(x, z))}{h} \right]. \quad (4.3)$$

In order for the equation above to hold both g and K_T must satisfy the Lipschitz property (for example, see Glasserman (2003) and see Chan and Joshi (2009)). In particular, assuming that the g is a smooth function of K_T , one requires the existence of a constant C_{K_T} such that

$$|K_T(x_2, z) - K_T(x_1, z)| \leq C_{K_T} \|x_2 - x_1\| \quad \forall x_1, x_2 \in \mathbf{X} \quad \text{and} \quad \forall z \in \mathbf{Z}. \quad (4.4)$$

For computing higher order price sensitivities using the pathwise method, similar conditions need to be satisfied. Summarizing, in order to compute the n^{th} order price sensitivity using the pathwise method, it is sufficient that constants $C_{K_T^{(m)}}$ exist such that

$$|K_T^{(m)}(x_2, z) - K_T^{(m)}(x_1, z)| \leq C_{K_T^{(m)}} \|x_2 - x_1\| \quad \forall x_1, x_2 \in \mathbf{X} \quad \text{and} \quad \forall z \in \mathbf{Z} \quad (4.5)$$

for $m = 0, 1, 2, \dots, n - 1$ where $K_T^{(m)}$ is the m^{th} derivative of K_T . As the LN scheme is a smooth function of inputs (i.e. \mathcal{C}^∞), price sensitivities can be evaluated easily by differentiating the LN scheme directly. However, discontinuous numerical schemes (as a function of inputs) such as the QE scheme, DG scheme and IDG scheme do not satisfy the conditions imposed by equation (4.5).

Here, we first consider the DG scheme and the IDG scheme. In order to compute sensitivities for these schemes, one has to employ a hybrid of the likelihood ratio method and the pathwise method. This idea is based on the works of Chan and Joshi (2009a) and Chan and Joshi (2009b). For clarity, here we rewrite the Poisson random variable N_λ as $N_\lambda(x, z)$ to indicate the dependency on x and z and similarly, we do this for η and N_{λ_j} for $j = 1, 2, \dots, k$. Observe that, the discontinuities in the DG scheme arise due to the uncertainty in the outcome of the Poisson random variable N_λ , which in turn is the shape parameter for one of the gamma random variables. Suppose that for a given z , we have $N_\lambda(x_1, z) = n$ and we constrain $N_\lambda(x_2, z) = n$ using importance sampling, then the variance process sampled using the DG scheme will satisfy the constraint in equation (4.5). This approach will come at a cost of a likelihood ratio (Monte-Carlo weight) and price sensitivities will take the following form,

$$\lim_{h \rightarrow 0} \mathbb{E}_Z \left[\frac{w(x+h, z)g(K_T(x+h, z)) - w(x, z)g(K_T(x, z))}{h} \right] \quad (4.6)$$

where $w(x+h, z)$ represents the Monte-Carlo weight. The Monte-Carlo weight is given by

$$w(x+h, z) = \frac{\mathbb{P}(N_\lambda(x+h, z) = n)}{d_n}$$

where d_n is a constant taking the value of $\mathbb{P}(N_\lambda(x, z) = n)$. Since the asset price process and the Monte-Carlo weight are Lipschitz in \mathbf{X} for all $z \in \mathbf{Z}$ and, upon constraining N_λ , the DG scheme is also Lipschitz in \mathbf{X} for all $z \in \mathbf{Z}$, one can therefore interchange the differentiation and expectation operator as the Lipschitz property is preserved under composition. Price sensitivities can now be evaluated by taking the expectation of the pathwise derivative of the weighted payoff, i.e

$$\mathbb{E}_Z \left[\frac{\partial}{\partial x} w(x, z)g(K_T(x, z)) \right] = \mathbb{E}_Z \left[\lim_{h \rightarrow 0} \frac{w(x+h, z)g(K_T(x+h, z)) - w(x, z)g(K_T(x, z))}{h} \right] \quad (4.7)$$

Note that, as the DG scheme usually involves multiple steps, one will have to multiply the payoff with the accumulated Monte-Carlo weight, $W(x, z)$ (i.e. the product of all the individual weights at each time step) instead. Using the same idea, the IDG scheme will also satisfy equation (4.5) by constraining the outcome of $N_\lambda(x, z)$, $\eta(x, z)$ and $N_{\lambda_j}(x, z)$ for $j = 1, 2, \dots, k$. Hence, price sensitivities for the IDG scheme can also be evaluated using the similar approach.

While the discontinuity in QE scheme arises from the switching between the *quadratic Gaussian approximation* and the *zero modified exponential approximation*, one can still interchange the expectation and differentiation operator and proceed to compute the first order derivative using the pathwise method. Recall that, both approximations are applicable when $1.0 \leq \Psi \leq 2.0$ and we typically set the switching rule at $\Psi = 1.5$. Suppose that we wish to compute the price sensitivities w.r.t x for a given $x \in \mathbf{X}$. If we can

construct a set \mathbf{X}^* such that $x \in \mathbf{X}^* \subseteq \mathbf{X}$ and, at every time step, we also have

$$\Psi(x_1, z) < 1.5 \implies \Psi(x_2, z) < 2 \quad \text{and} \quad \Psi(x_1, z) > 1.5 \implies \Psi(x_2, z) > 1 \quad \forall x_1, x_2 \in \mathbf{X}^* \quad \text{and} \quad \forall z \in \mathbf{Z} \quad (4.8)$$

then the variance process simulated using the QE scheme is Lipschitz in \mathbf{X}^* . This is because we could now restrict both variance processes (with an initial input of x_1 and x_2) to be sampled using the same approximation at every time step. Hence, price sensitivities w.r.t to x can be evaluated using the standard pathwise method. We do not attempt to construct the set \mathbf{X}^* explicitly since it does not add anything to the analysis here; however it is clear that such a set exists and that is enough for the computation of first order derivatives.

Whilst this approach works for the first order derivatives, there is a more subtle issue when one considers higher order derivatives since the zero-modified-exponential approximation has a discontinuous first order derivative. We do not attempt to create a solution here as the QE scheme can potentially give a extreme value even for the first-order Greeks. (see, Section 6.1).

5. COMPUTING GREEKS USING ALGORITHMIC DIFFERENTIATION APPROACH

The recent works by Joshi and Yang (2010) have shown that Greeks can be evaluated efficiently using the backward algorithmic differentiation approach. The key to their approach is to study how given the gradient ∇_g and the Hessian \mathbf{H}_g of a function, $g : \mathbb{R}^n \rightarrow \mathbb{R}$, they can compute $\nabla_{g \circ F}$ and $\mathbf{H}_{g \circ F}$ for a sufficiently simple function $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$. The essential observation is that if F is the identity map in all coordinates except one and that particular coordinate depends on only one or two coordinates then the new gradient will agree except in one or two coordinates while the new Hessian will agree except in one or two rows and columns. The new gradient and the new Hessian are then obtained from the old one by overwriting these elements in a simple way. They further showed that if it is possible to decompose a complicated map into a sequence of L elementary operations (i.e. simple functions) with a maximum of M state variables then the computational order of the gradient and Hessian is proportional to L and ML respectively. The strength in their approach is that one only has to specify the mapping functions and then price sensitivities can be evaluated easily using the backward algorithmic differentiation approach. We refer readers to Chao and Joshi (2010) for a better understanding of their approach.

Here, we adopt their approach when computing the first and second order price sensitivities. While their work is appealing, one can always work with a more complicated mapping function, instead of multiple simple functions, to reduce the computational time. Similar to their approach, we let the function $F : \mathbf{x} \rightarrow \mathbf{y}$, where $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, to be the identity map in all coordinates except for the i^{th} coordinate. However, instead of restricting that the i^{th} coordinate to depend only on one or two coordinates, we allow it to depend on more than two coordinates as necessary. We therefore have,

$$y_j = x_j \quad \text{for} \quad j = 1, 2, \dots, i-1, i+1, \dots, n, \quad (5.1)$$

and we assume that

$$y_i = F_i(\mathbf{x}). \quad (5.2)$$

where $F_i : \mathbb{R}^n \rightarrow \mathbb{R}$. While relaxing such a restriction can make updating the gradient $\nabla_{g \circ F}$ and the Hessian $\mathbf{H}_{g \circ F}$ more complicated, one can still update them easily using the algorithm presented below provided one can obtain the gradient ∇_{F_i} and the Hessian \mathbf{H}_{F_i} of F_i . Note that, as the Hessian is a symmetric matrix, we shall represent its elements by a lower triangular matrix with the understanding that the entry (j, k) refers

to $(\max(j, k), \min(j, k))$. When the term ‘‘row j ’’ is used, we shall mean the set of elements (j, k) for some fixed j for $k = 1, 2, \dots, n$.

The new gradient $\nabla_{g \circ F}$ can be obtained as follow:

- (1) Overwrite the gradient ∇_g as follows

$$(\nabla_g)_j = (\nabla_g)_j + (\nabla_{F_i})_j \cdot (\nabla_g)_i \quad (5.3)$$

for all j except for the i^{th} element.

- (2) The i^{th} element is updated as follows

$$(\nabla_g)_i = (\nabla_{F_i})_i \cdot (\nabla_g)_i, \quad (5.4)$$

and the new gradient $\nabla_{g \circ F}$ is then given by the updated ∇_g .

The algorithm to obtain the new Hessian $H_{g \circ F}$ is as follow:

- (1) Overwrite the Hessian H_g on a row by row basis as follows

$$(H_g)_{jk} = (H_g)_{jk} + (\nabla_{F_i})_j \cdot (H_g)_{ji} \quad (5.5)$$

$$(H_g)_{jj} = (H_g)_{jj} + (\nabla_{F_i})_j \cdot (2 \cdot (H_g)_{ji} + (\nabla_{F_i})_j \cdot (H_g)_{ii}) \quad (5.6)$$

except for row i . Within each row, if $j < i$, entries are overwritten according to the increasing sequence of k (i.e. first overwrite $(H_g)_{j1}$ followed by $(H_g)_{j2}$ etc) if $j > i$, entries are overwritten according to the decreasing sequence of k (i.e. first overwrite $(H_g)_{i,n}$ followed by $(H_g)_{i,n-1}$ etc).

- (2) Once all rows (except row i) have been overwritten, entries in row i are overwritten as follow

$$(H_g)_{ik} = (\nabla_{F_i})_i \cdot (H_g)_{ik} \quad (5.7)$$

$$(H_g)_{ii} = (\nabla_{F_i})_i^2 \cdot (H_g)_{ii} \quad (5.8)$$

- (3) Lastly, overwrite the Hessian as follow

$$(H_g)_{jk} = (H_g)_{jk} + (\nabla_g)_i \cdot (H_{F_i})_{jk} \quad (5.9)$$

for all rows and the new Hessian $H_{g \circ F}$ is then given by the updated H_g .

Note that, since we are overwriting the gradient and the Hessian, the order in which the elements in the gradient and the Hessian are overwritten is important and one must overwrite the Hessian before overwriting the gradient. In order to reduce the computational time, there is no need to overwrite the l^{th} element and row l when computing the new gradient and the new Hessian respectively, if y_i has no dependency on x_l (except when $l = i$) for some index l .

5.1. Decomposing the Mappings for the Heston Process - Overview. For simulation purposes, one usually fixes the size of the time step to τ and evolve the Heston process from time t_i to t_{i+1} where $t_{i+1} = t_i + \tau$ for $i = 0, 1, \dots, n - 1$. When computing Greeks in the Heston model, one is typically interested in the first and the second order price sensitivities w.r.t $S(t_0), V(t_0), \kappa, \theta, \epsilon, \rho$ and the relevant cross-sensitivities. While $S(t_0), V(t_0), \kappa, \theta, \epsilon, \rho$ constitute the parameters of interest, only $S(t_0)$ and $V(t_0)$ are considered as state variables since they vary at each time step. For an efficient computation, we perform the following mapping

before the start of the simulation,

$$\begin{bmatrix} S(t_0) \\ V(t_0) \\ \kappa \\ \theta \\ \epsilon \\ \rho \end{bmatrix} \xrightarrow{\mathcal{F}_0} \begin{bmatrix} S(t_0) \\ V(t_0) \\ W(t_0) \\ c_1 \\ c_2 \\ \vdots \\ c_n \\ ax_1 \\ ax_2 \end{bmatrix}$$

where c_1, c_2, \dots, c_n are constants function of $\kappa, \theta, \epsilon, \rho$ and are determined by the numerical scheme used, while ax_1 and ax_2 are auxiliary variables used to update the Heston process. We also define an additional state variable, $W(t_0)$, representing the accumulated Monte-Carlo weight. This is only required when using the DG scheme or the IDG scheme.

In general, the constants c_1, c_2, \dots, c_n are complicated function of κ, θ, ϵ and ρ . Instead of carrying along the parameter κ, θ, ϵ and ρ , and recomputing the constants c_1, c_2, \dots, c_n at every step, it is generally more efficient to compute the constants c_1, c_2, \dots, c_n at the beginning and carry them along as we only have to do this mapping once. Although these parameters do not change from step to step, we include them in the collection of variables to which we are computing the Hessian. Therefore they are state variables in the sense of Joshi–Yang (2010) and their inclusion increases M . Whilst this increases the multiplier on the computational complexity when going from the original computation to the Hessian computation, this is more than compensated by the reduction in complexity of the former.

At each time step, the Heston process can be evolved from t_i to t_{i+1} using the composition of the following 5 mapping functions.

$$\begin{bmatrix} S(t_i) \\ V(t_i) \\ W(t_i) \\ c_1 \\ c_2 \\ \vdots \\ c_n \\ ax_1 \\ ax_2 \end{bmatrix} \xrightarrow{\mathcal{F}_1} \begin{bmatrix} S(t_i) \\ V(t_i) \\ W(t_i) \\ c_1 \\ c_2 \\ \vdots \\ c_n \\ V(t_{i+1}) \\ ax_2 \end{bmatrix} \xrightarrow{\mathcal{F}_2} \begin{bmatrix} S(t_i) \\ V(t_i) \\ W(t_i) \\ c_1 \\ c_2 \\ \vdots \\ c_n \\ V(t_{i+1}) \\ L(t_{i+1}) \end{bmatrix} \xrightarrow{\mathcal{F}_3} \begin{bmatrix} S(t_i) \\ V(t_i) \\ W(t_{i+1}) \\ c_1 \\ c_2 \\ \vdots \\ c_n \\ V(t_{i+1}) \\ L(t_{i+1}) \end{bmatrix} \xrightarrow{\mathcal{F}_4} \begin{bmatrix} S(t_{i+1}) \\ V(t_i) \\ W(t_{i+1}) \\ c_1 \\ c_2 \\ \vdots \\ c_n \\ V(t_{i+1}) \\ L(t_{i+1}) \end{bmatrix} \xrightarrow{\mathcal{F}_5} \begin{bmatrix} S(t_{i+1}) \\ V(t_{i+1}) \\ W(t_{i+1}) \\ c_1 \\ c_2 \\ \vdots \\ c_n \\ ax_1 \\ ax_2 \end{bmatrix}$$

where $L(t_{i+1})$ represents the integrated variance process from time t_i to t_{i+1} . In the next section, we will further decompose the mapping \mathcal{F}_j for $j = 1, 2, \dots, 5$ as necessary. While it is possible to use these complicated mappings directly or breaking them down to some very simple sub-mappings, our intention here is to decompose them into a tractable form. Note that, for a quick simulation, there is no need to sample the asset price at every time step but only at the observation time steps and this is the approach used in our numerical tests.

5.2. Mappings for the Lognormal scheme. For the LN scheme, we let

$$c_1 = \frac{\rho}{\epsilon}, \quad c_2 = \kappa\theta, \quad c_3 = \frac{\kappa\rho}{\epsilon} - \frac{1}{2}, \quad c_4 = \sqrt{1 - \rho^2}, \quad c_5 = \theta(1 - e^{-\kappa\tau}), \quad (5.10)$$

$$c_6 = e^{-\kappa\tau}, \quad c_7 = \frac{\epsilon^2 e^{-\kappa\tau}}{\kappa} (1 - e^{-\kappa\tau}) \quad \text{and} \quad c_8 = \frac{\theta\epsilon^2}{2\kappa} (1 - e^{-\kappa\tau})^2. \quad (5.11)$$

5.2.1. Decomposing \mathcal{F}_1 . Under the LN scheme, the mapping function \mathcal{F}_1 can be further decomposed into sub-mappings as follow

$$\mathcal{F}_1 = \mathcal{F}_{1,3} \circ \mathcal{F}_{1,2} \circ \mathcal{F}_{1,1}$$

$$\begin{bmatrix} \vdots \\ c_8 \\ ax_1 \\ ax_2 \end{bmatrix} \xrightarrow{\mathcal{F}_{1,1}} \begin{bmatrix} \vdots \\ c_8 \\ m \\ ax_2 \end{bmatrix} \xrightarrow{\mathcal{F}_{1,2}} \begin{bmatrix} \vdots \\ c_8 \\ m \\ \Psi \end{bmatrix} \xrightarrow{\mathcal{F}_{1,3}} \begin{bmatrix} \vdots \\ c_8 \\ V(t_{i+1}) \\ ax_2 \end{bmatrix}$$

where

$$m = f_m(V(t_i), c_5, c_6) = c_5 + c_6 \cdot V(t_i) \quad (5.12)$$

$$\Psi = f_\Psi(V(t_i), c_7, c_8, m) = \frac{c_7 \cdot V(t_i) + c_8}{m^2} \quad (5.13)$$

$$V(t_{i+1}) = f_V(m, \Psi) = \frac{m}{\sqrt{\Psi + 1}} \exp(\sqrt{\log \Psi + 1} \cdot Z) \quad (5.14)$$

Since computing the gradient and Hessian of f_m , f_Ψ and f_V is straight forward, one can update the gradient and Hessian of the price easily.

5.2.2. Decomposing \mathcal{F}_2 . Since the trapezoidal rule is used to approximate the integrated variance process there is no need to further decompose \mathcal{F}_2 . Updating the gradient and the Hessian of the price for this step is straight-forward.

5.2.3. Decomposing \mathcal{F}_3 . As the Monte-Carlo weight of the LN scheme is constant and has no sensitivities w.r.t. the model inputs, one can ignore this mapping.

5.2.4. Decomposing \mathcal{F}_4 . While the mapping of the asset price process can be decomposed into smaller sub-mappings, one can also work with this complicated mapping and we choose to do so. The asset price process can be written as

$$S(t_{i+1}) = S(t_i) \exp \left(c_1(V(t_{i+1}) - V(t_i) - c_2(t_{i+1} - t_i)) + c_3L(t_{i+1}) + c_4\sqrt{L(t_{i+1})}Z_s \right) \quad (5.15)$$

$$= f_s [S(t_i), V(t_{i+1}), V(t_i), L(t_{i+1}), c_1, c_2, c_3, c_4]. \quad (5.16)$$

Computing the gradient and Hessian of f_s is straight forward, and the gradient and Hessian of the price can be updated with ease.

5.3. Mapping for the QE scheme. The mappings for the QE scheme are the same as the mappings for the LN scheme except for \mathcal{F}_1

5.3.1. Decomposing \mathcal{F}_1 . Under the QE scheme, the mapping function \mathcal{F}_1 can be further decomposed into sub-mappings as follow

$$\mathcal{F}_1 = \mathcal{F}_{1,5} \circ \mathcal{F}_{1,4} \circ \mathcal{F}_{1,3} \circ \mathcal{F}_{1,2} \circ \mathcal{F}_{1,1}.$$

While the sub-mappings $\mathcal{F}_{1,1}$ and $\mathcal{F}_{1,2}$ are exact the same as the mappings for the LN scheme, the subsequence mappings i.e $\mathcal{F}_{1,3}$, $\mathcal{F}_{1,4}$ and $\mathcal{F}_{1,5}$ vary depending on the value of Ψ .

For $\Psi < 1.5$, we have

$$\begin{bmatrix} \vdots \\ c_8 \\ m \\ \Psi \end{bmatrix} \xrightarrow{\mathcal{F}_{1,3}} \begin{bmatrix} \vdots \\ c_8 \\ m \\ b^2 \end{bmatrix} \xrightarrow{\mathcal{F}_{1,4}} \begin{bmatrix} \vdots \\ c_8 \\ a \\ b^2 \end{bmatrix} \xrightarrow{\mathcal{F}_{1,5}} \begin{bmatrix} \vdots \\ c_8 \\ V(t_{i+1}) \\ ax_2 \end{bmatrix}$$

while, for $\Psi > 1.5$, we have

$$\begin{bmatrix} \vdots \\ c_8 \\ m \\ \Psi \end{bmatrix} \xrightarrow{\mathcal{F}_{1,3}} \begin{bmatrix} \vdots \\ c_8 \\ m \\ p \end{bmatrix} \xrightarrow{\mathcal{F}_{1,4}} \begin{bmatrix} \vdots \\ c_8 \\ \beta \\ p \end{bmatrix} \xrightarrow{\mathcal{F}_{1,5}} \begin{bmatrix} \vdots \\ c_8 \\ V(t_{i+1}) \\ ax_2 \end{bmatrix}$$

where a, b^2, p , and β are defined in Section 3.2.

5.4. Mappings for the DG scheme. Under the DG scheme, we let

$$c_1 = \frac{\rho}{\epsilon}, \quad c_2 = \kappa\theta, \quad c_3 = \frac{\kappa\rho}{\epsilon} - \frac{1}{2}, \quad c_4 = \sqrt{1 - \rho^2} \quad (5.17)$$

$$c_5 = \alpha, \quad c_6 = \frac{1}{\beta}, \quad \text{and} \quad c_7 = \frac{2\kappa}{\epsilon^2(e^{\kappa\tau} - 1)}. \quad (5.18)$$

where α and β are defined in proposition 4. The mappings for the DG scheme are the same as the mappings for the LN scheme except for \mathcal{F}_1 and \mathcal{F}_3 .

5.4.1. Decomposing \mathcal{F}_1 . Due to the constraint imposed on N_λ , the gamma random variable, $\Gamma(N_\lambda, 1)$, has no sensitivities with respect to the Heston's inputs. The mapping function \mathcal{F}_1 is therefore a simple mapping function. Specifically, suppose that $N_\lambda = n$ at t_i , we have

$$V(t_{i+1}) = f_V(c_5, c_6) = c_6 \cdot (\Gamma(c_5, 1) + \Gamma(n, 1)). \quad (5.19)$$

Provided that the gradient and the Hessian of f_V can be evaluated, the gradient and the Hessian of the price can be updated easily. The real challenge comes from evaluating the first and second order derivative of $\Gamma(\alpha, 1)$ w.r.t α . Here, similar to the approach suggested by Chan and Joshi (2010), we suggest to tabulate the first and second order derivative of the inverse of the gamma distribution for the shape parameter α at the beginning of the simulation. Since this is done outside the Monte-Carlo simulation, there will only be a slight increase in the computational time. The details of our implementation are presented in Appendix A.

5.4.2. Decomposing \mathcal{F}_3 . Suppose that we have $N_\lambda = n$ at t_i , the accumulated Monte-Carlo weight at t_{i+1} is given by

$$W(t_{i+1}) = f_W(W(t_i), V(t_i), c_7) = W(t_i) \cdot \frac{q_n}{d_n} \quad (5.20)$$

where

$$q_n = \mathbb{P}(N_\lambda = n) = \frac{\exp(-c_7 \cdot V(t_i))(c_7 \cdot V(t_i))^n}{n!}$$

and d_n is a constant taking value q_n . Note that, here, by construction, the value of the accumulated Monte-Carlo weight is always equal to 1, however, it has non-zero sensitivities w.r.t inputs. Updating the gradient and Hessian for the price is straight forward once the gradient and Hessian of f_W are determined.

5.5. Mapping for the IDG scheme. Depending on the truncation level, k , the mappings for the IDG scheme are exactly the same as the DG scheme except for \mathcal{F}_2 and \mathcal{F}_3 . In addition, the IDG scheme has

additional constants given by

$$c_8 = \left(\frac{2\kappa/\epsilon^2}{\sinh(\kappa\tau/2)} \right)^2 \quad c_9 = \left(\mu_{X_1}^* - \sum_{j=1}^k \left(\frac{\lambda_j^*}{\gamma_j} \right) \right) \quad c_{10} = \left(\mu_{X_2}^* - \sum_{j=1}^k \left(\frac{1}{2\gamma_j} \right) \right) \quad (5.21)$$

$$c_{11} = \left((\sigma_{X_1}^*)^2 - \sum_{j=1}^k \left(\frac{2\lambda_j^*}{\gamma_j^2} \right) \right) \quad c_{12} = \left((\sigma_{X_2}^*)^2 - \sum_{j=1}^k \left(\frac{1}{2\gamma_j^2} \right) \right) \quad c_{13+j} = \lambda_j^* \quad c_{13+k+j} = \frac{1}{\gamma_j} \quad (5.22)$$

for $j = 1, 2, \dots, k$.

5.5.1. *Decomposing \mathcal{F}_2 .* The mapping function \mathcal{F}_2 can be further decomposed into sub-mappings as follow

$$\mathcal{F}_2 = \mathcal{F}_{2,k} \circ \dots \circ \mathcal{F}_{2,1} \circ \mathcal{F}_{2,0}$$

$$\begin{bmatrix} \vdots \\ c_{13+2k} \\ V(t_{i+1}) \\ ax_2 \end{bmatrix} \xrightarrow{\mathcal{F}_{2,0}} \begin{bmatrix} \vdots \\ c_{13+2k} \\ V(t_{i+1}) \\ l_0 \end{bmatrix} \xrightarrow{\mathcal{F}_{2,1}} \begin{bmatrix} \vdots \\ c_{13+2k} \\ V(t_{i+1}) \\ l_1 \end{bmatrix} \xrightarrow{\mathcal{F}_{2,2}} \dots \xrightarrow{\mathcal{F}_{2,k}} \begin{bmatrix} \vdots \\ c_{13+2k} \\ V(t_{i+1}) \\ l_k = L(t_{i+1}) \end{bmatrix}$$

where

$$l_0 = \text{Ln}(\mu_{Ln}^k, (\sigma_{Ln}^k)^2) \quad (5.23)$$

$$l_j = f_{l_j}(l_{j-1}, c_5, c_{13+k+j}) = l_{j-1} + c_{13+k+j} \cdot (\Gamma_{j1}(c_5, 1) + \Gamma_{j2}(2\eta + N_{\lambda_j}, 1)) \quad (5.24)$$

$$(5.25)$$

for $j = 1, 2, \dots, k$. The first mapping updates the Ln random variable of the IDG scheme. This can be further decomposed using the similar approach presented in Section 5.2.1. The subsequence mappings are only slightly more complicated than the mapping in Section 5.4.1 .

5.5.2. *Decomposing \mathcal{F}_3 .* Under the IDG scheme, there are a total of $k + 2$ individual Monte-Carlo weights at each time step, corresponding to the random variables N_λ , N_{λ_j} and η for $j = 1, 2, \dots, k$. To update the accumulated Monte-Carlo weight from t_i to t_{i+1} , we multiply the accumulated Monte-Carlo weight with the individual Monte-Carlo weights one at a time. In particular, suppose that we let $W_0(t_i) = W(t_i)$ and $w_j(t_i)$ representing the individual Monte-Carlo weight, the mapping \mathcal{F}_3 is decomposed into $k + 2$ sub-mappings

$$\mathcal{F}_3 = \mathcal{F}_{3,k+1} \circ \dots \circ \mathcal{F}_{3,1} \circ \mathcal{F}_{3,0}$$

where the $\mathcal{F}_{3,j}$ sub-mapping updates the interim accumulated Monte-Carlo weight as follow

$$W_{j+1}(t_i) = W_j(t_i) \cdot w_j(t_i) \quad \text{for } j = 0, 1, \dots, k + 1, \quad (5.26)$$

with the accumulated Monte-Carlo weight at t_{i+1} given by

$$W(t_{i+1}) = W_{k+2}(t_i) = W(t_i) \cdot \prod_{j=0}^{k+1} w_j(t_i).$$

Here, we let $w_0(t_i)$, $w_j(t_i)$ and $w_{k+1}(t_i)$ be the individual Monte-Carlo weights for N_λ , N_{λ_j} and η for $j = 1, \dots, k$ respectively. Updating the gradient and Hessian for the price is straight forward as the sub-mapping $\mathcal{F}_{3,0}$ is exactly the same as the mapping function in Section 5.4.2. The sub-mappings $\mathcal{F}_{3,j}$ for $j = 1, \dots, k$ are also very similar to the mapping function in Section 5.4.2. Specifically, suppose that we

have $N_{\lambda_j} = n$ at time t_i , the accumulated Monte-Carlo weight $W_{j+1}(t_i)$ is given by

$$W_{j+1}(t_i) = f_{W_{j+1}}(W_j(t_i), V(t_i), V(t_{i+1}), c_{13+j}) = W_j(t_i) \cdot w_j(t_i) \quad (5.27)$$

where

$$w_j(t_i) = \frac{\mathbb{P}(N_{\lambda_j} = n)}{d_n^j} = \frac{1}{d_n^j} \cdot \frac{\exp[-c_{13+j} \cdot (V(t_i) + V(t_{i+1}))][c_{13+j} \cdot (V(t_i) + V(t_{i+1}))]^n}{n!}$$

and d_n^j is a constant taking the value of $\mathbb{P}(N_{\lambda_j} = n)$ for $j = 1, 2, \dots, k$.

Lastly, suppose that we have $\eta = m$ at t_i , the accumulated Monte-Carlo weight can be updated as follow

$$W(t_{i+1}) = f_{W_{k+2}}(W_{k-1}(t_i), V(t_i), V(t_{i+1}), c_5, c_8) = W_{k-1}(t_i) \cdot w_{k+1}(t_i) \quad (5.28)$$

where

$$w_{k+1}(t_i) = \frac{p_m}{b_m} = \frac{1}{b_m} \cdot \frac{C_m y^m}{\sum_{n=0}^h C_n \cdot y^n}.$$

The constant b_m takes the value of p_m and we have

$$y = c_8 \cdot V(t_i) \cdot V(t_{i+1}) \quad C_{n+1} = \frac{C_n}{4(n+1)(n+c_5)} \quad C_0 = 1. \quad (5.29)$$

However, here, computing the gradient and the Hessian of $f_{W_{k+2}}$ can be rather challenging. We proceed by first evaluating the first and second order derivatives of p_m w.r.t to α and y , the gradient and Hessian of $f_{W_{k+2}}$ can then be obtained easily using chain rules. A direct differentiation of p_m gives

$$\frac{\partial p_m}{\partial \alpha} = p_0 \left(y^m \frac{\partial C_m}{\partial \alpha} - p_m \sum_{n=0}^h \left(y^n \frac{\partial C_n}{\partial \alpha} \right) \right) \quad (5.30)$$

$$\frac{\partial p_m}{\partial y} = \frac{p_m}{y} \left(m - p_0 \sum_{n=0}^h (n y^n C_n) \right) \quad (5.31)$$

$$\frac{\partial^2 p_m}{\partial \alpha^2} = p_0 \left(y^m \frac{\partial^2 C_m}{\partial \alpha^2} - p_m \sum_{n=0}^h \left(y^n \frac{\partial^2 C_n}{\partial \alpha^2} \right) \right) - 2p_0 \frac{\partial p_m}{\partial \alpha} \sum_{n=0}^h \left(y^n \frac{\partial C_n}{\partial \alpha} \right) \quad (5.32)$$

$$\frac{\partial^2 p_m}{\partial y^2} = \frac{p_m}{y^2} \left(m^2 - p_0 \sum_{n=0}^h (n^2 y^n C_n) \right) - \frac{1}{y} \frac{\partial p_m}{\partial y} \left(1 + 2p_0 \sum_{n=0}^h (n y^n C_n) \right) \quad (5.33)$$

$$\frac{\partial^2 p_m}{\partial y \partial \alpha} = \frac{m}{y} \frac{\partial p_m}{\partial y} + \left(\frac{m}{y} - \frac{1}{p_m} \frac{\partial p_m}{\partial y} \right) \left(p_0 y^m \frac{\partial C_m}{\partial \alpha} - 2 \frac{\partial p_m}{\partial \alpha} \right) - \frac{p_0 p_m}{y} \sum_{n=0}^h \left(n y^n \frac{\partial C_n}{\partial \alpha} \right) \quad (5.34)$$

$$(5.35)$$

and these can be evaluated easily when sampling η . Note that, as C_n and its derivatives are state-independent, they can be pre-calculated before the beginning of the simulation.

6. NUMERICAL TESTS AND RESULTS

For numerical tests, we consider computing the price sensitivities for a 1-year European call option and a 1-year Asian option with monthly fixing. All products have a strike of 100 and they are evaluated using the test inputs given in table 6.1.

The European call option will form a standard test case as the semi-analytical solution for the price sensitivities exist while the Asian option is used as the test case for a more path-dependent option. As the payoff functions considered in our numerical test are not second-order smooth, we use the smoothing technique presented in Joshi and Yang (2010) with a smoothing window of 0.5.

	Case 1	Case 2	Case 3
ϵ	1.0	0.9	1.0
κ	0.5	0.3	1.0
$V(0), \theta$	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 6.1. Inputs for numerical tests.

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. Uniform random numbers were generated using randomised quasi-Monte Carlo with random shift (see, Glasserman 2003). In particular, we first generate a vector of random shifts for each simulation using the Mersenne Twister random number generator where the size of the vector is the dimensionality of simulation. Within each simulation, the uniform random numbers on each path are generated by applying the same vector of random shifts to the vector of Sobol numbers generated using the Sobol-Levitani-Lemieux initialization as implemented in Quantlib (2008).

6.1. Problems with the QE scheme. In this section, we highlight the problems with the QE scheme. For our discussion purposes, we only present part of the numerical results in Table 6.2.

	Case 1				
	Actual	mean	stdev	skewness	kurtosis
Price	4.403	4.395	0.006	0.03	-0.04
∂S	0.796	0.794	0.001	-0.07	0.07
∂V	60.213	60.003	6.157	24.54	1459.44
$\partial \kappa$	1.313	1.313	0.096	-4.35	141.94
$\partial \theta$	24.747	24.833	1.324	-4.75	127.28
$\partial \epsilon$	-2.708	-2.711	0.535	-19.12	1219.24
$\partial \rho$	2.608	2.668	0.048	0.02	0.09

	Case 2				
	Actual	mean	std. dev	skewness	kurtosis
Price	5.100	5.128	0.012	0.04	0.08
∂S	0.663	0.653	0.001	0.06	-0.03
∂V	74.957	74.380	4.753	-0.26	180.26
$\partial \kappa$	1.316	1.261	0.149	-24.91	1,267.39
$\partial \theta$	16.388	16.003	1.185	-22.72	1,117.54
$\partial \epsilon$	-2.716	-2.599	0.465	0.77	176.52
$\partial \rho$	1.176	1.273	0.037	0.06	0.00

	Case 3				
	Actual	mean	std. dev	skewness	kurtosis
Price	9.774	9.782	0.015	0.08	0.14
∂S	0.608	0.607	0.001	0.02	0.03
∂V	39.495	39.517	0.206	0.07	0.08
$\partial \kappa$	1.005	0.990	0.014	0.06	0.20
$\partial \theta$	27.573	27.409	0.162	0.09	0.20
$\partial \epsilon$	-2.638	-2.604	0.060	-0.07	0.28
$\partial \rho$	1.148	1.165	0.053	0.04	-0.09

TABLE 6.2. Tables above show the estimated mean, standard deviation, skewness and excess kurtosis of prices and first order price sensitivities of call options evaluated using QE scheme with 16 steps per year for all 3 cases. The results were calculated using 5,000 independent simulations with each simulation consisting 65,535 paths.

Table 6.2 demonstrates the sample mean can take much longer to converge to a normal distribution depending on the Heston's inputs used. In particular, for Case 1 and Case 2, the distributions of the first order price sensitivities w.r.t $V, \kappa, \theta, \epsilon$ are so highly skewed and fat-tailed that even after 65,535 paths the sample means are not even close to being normally distributed. This can be a significant problem, since it makes confidence intervals obtained using standard error estimates effectively meaningless.

i	$V(t_i)$	m	s_2	Ψ	P	β	U	$V(t_{i+1})$	$\frac{\partial V(t_{i+1})}{\partial V(t_i)}$	$\frac{\partial V(t_{i+1})}{\partial V(t_0)}$
0	0.0100	0.0109	0.0006	5.316	0.683	28.990	0.703	0.0023	2.5	2.464E+00
1	0.0023	0.0034	0.0002	14.782	0.873	37.141	0.893	0.0046	5.9	1.464E+01
2	0.0046	0.0057	0.0003	9.611	0.812	32.993	0.832	0.0034	4.2	6.168E+01
3	0.0034	0.0045	0.0002	11.747	0.843	34.666	0.863	0.0039	5.0	3.075E+02
4	0.0039	0.0050	0.0003	10.712	0.829	33.859	0.849	0.0037	4.6	1.421E+03
5	0.0037	0.0048	0.0003	11.183	0.836	34.225	0.856	0.0038	4.8	6.803E+03
6	0.0038	0.0049	0.0003	10.962	0.833	34.053	0.853	0.0037	4.7	3.204E+04
7	0.0037	0.0049	0.0003	11.064	0.834	34.133	0.854	0.0038	4.7	1.521E+05
8	0.0038	0.0049	0.0003	11.017	0.834	34.096	0.854	0.0038	4.7	7.192E+05
9	0.0038	0.0049	0.0003	11.039	0.834	34.113	0.854	0.0038	4.7	3.407E+06
10	0.0038	0.0049	0.0003	11.029	0.834	34.105	0.854	0.0038	4.7	1.613E+07
11	0.0038	0.0049	0.0003	11.033	0.834	34.109	0.854	0.0038	4.7	7.637E+07
12	0.0038	0.0049	0.0003	11.031	0.834	34.107	0.854	0.0038	4.7	3.616E+08
13	0.0038	0.0049	0.0003	11.032	0.834	34.108	0.854	0.0038	4.7	1.712E+09
14	0.0038	0.0049	0.0003	11.032	0.834	34.108	0.854	0.0038	4.7	8.107E+09
15	0.0038	0.0049	0.0003	11.032	0.834	34.108	0.854	0.0038	4.7	3.838E+10

TABLE 6.3. Table above shows the realization of the variance process for Case 1 that leads to sensitivities blow-up.

The main reason that Case 1 and Case 2 have high skewness and kurtosis is that the sensitivities of the variance process can blow up for some specific paths. In particular, this occurs when the variance process at each step is sampled repeatedly using the zero-modified exponential approximation with the uniform random variable, U , marginally greater than P . In order to illustrate this, we construct a path with 16 steps per year using the inputs for Case 1 with $V(t_0) = 0.01$ and we set the pathwise uniform random variable, U , at each time step to be $U = P + 0.02$. From the table 6.3, we can see that the sensitivity of the terminal variance process w.r.t to the initial variance process is

$$\frac{\partial V(t_{16})}{\partial V(t_0)} = \prod_{i=0}^{15} \frac{\partial V(t_{i+1})}{\partial V(t_i)} = 3.838 \times 10^{10}. \quad (6.1)$$

We can also deduce that the sensitivity w.r.t. κ will blow-up as

$$\frac{\partial V(t_{16})}{\partial \kappa} = \sum_{i=0}^{15} \frac{\partial V(t_{16})}{\partial V(t_i)} \frac{\partial V(t_i)}{\partial \kappa} \quad (6.2)$$

and similar conclusion can be made for the sensitivities w.r.t. θ and ϵ .

Recall that, under the QE scheme, the zero-modified exponential approximation is used to sample $V(t_{i+1})$ when $V(t_i)$ is close to zero. The main issue with Case 1 and Case 2 is that, when $V(t_i)$ is close to zero, the zero-modified exponential approximation will generate a very small $V(t_{i+1})$ with the derivative of $\frac{\partial V(t_{i+1})}{\partial V(t_i)}$ significantly greater than 1 whenever a random number, U , marginally greater than P , is drawn. If this is repeated (i.e. drawing U which is marginally greater than P) at every step, the sensitivities of the variance process w.r.t. initial inputs can grow very quickly and potentially blow up. However, under the same scenario, the sensitivities for Case 3 do not blow-up as the derivative of $\frac{\partial V(t_{i+1})}{\partial V(t_i)}$ is close to 1.

6.2. Numerical Results. In this section, we present the numerical results for the European call option and the Asian call option for all 3 cases. The Table 6.4, 6.5 and 6.6 show the results for the European call option while the results for the Asian call option are presented in Table 6.7, 6.8 and 6.9. The mean, standard deviation, skewness and excess kurtosis are evaluated using 5,000 simulations with 65,535 paths each. For the European call option, we will only present the first and the second order price sensitivities w.r.t $S(0), V(0), \kappa, \theta, \epsilon, \rho$ but not the cross-sensitivities.

6.2.1. *Call Option Results.* A general observation is that the standard error of price sensitivities increases as the number of steps per year increases. For any given fixed number of steps per year, the LN scheme generally has the lowest standard deviation for price sensitivities followed by the DG scheme and the IDG scheme. However, similar to pricing, the LN scheme typically has a higher discretization bias for the price sensitivities. One can see that from Table 6.4, 6.5 and 6.6, the discretization bias for both prices and sensitivities are relatively high even for the LN scheme with 64 steps per year. While, smaller step sizes (i.e. increasing number of steps per year) can be used to reduce the discretization bias, this can potentially increase the standard error and the simulation time significantly.

While all three schemes give stable results for the first order price sensitivities, the standard deviation for the second order derivatives can be high. Based on our numerical results, the LN scheme gives a mixed result for the second order price sensitivities. While the second order price sensitivities for Case 3 are very stable, the results for case 1 and case 2 has a very high standard deviation and they are highly skewed and fat-tailed that even after 65,535 paths the sample means are not even close to being normally distributed. As explained earlier, this can be a significant problem, since it makes confidence intervals obtained using standard error estimates effectively meaningless and occasionally one can obtain some extreme value of the second order price sensitivities. Therefore the LN scheme is generally not suitable for practical use when it comes to computing higher order price sensitivities as practitioners prefer schemes that work well for a large range of inputs. Unlike the LN scheme, the second order price sensitivities of the IDG scheme and the DG scheme remain stable for all 3 cases where both the skewness and the excess kurtosis are close zero. Overall, the RMS of the standard deviation of the second order price sensitivities of the IDG scheme and the DG scheme increases approximately proportional to the number of steps per year.

As for the timing consideration, the computational time for the DG scheme and the LN scheme are very similar while the IDG scheme's computational time is longer than both the DG scheme and the LN scheme for any given fixed step size. However, as the IDG scheme can evolve the Heston process accurately over long steps without the need to sample the intervening values, one typically finds that the IDG scheme performs more efficiently for medium to long dated options. As we can see from the tables, even with just 1 steps a year, the IDG scheme give negligible discretization bias for prices and price sensitivities with reasonable standard deviation.

6.2.2. *Asian Option Results.* The numerical results of the Asian option are similar to the European call option results. Given that we are now considering a 1-year Asian option with monthly fixing, it is generally more efficient to use short-stepping schemes. As we could see from table 6.7 and 6.8 and 6.9, the DG scheme (with 24 steps per year) gives a similar discretization bias as the IDG scheme (with 12 steps per year) but has a lower standard deviation and a computational time. Similar to the European call results, the second order sensitivities of the LN scheme for Case 1 and 2 can be highly skewed and fat-tailed depending the number of steps per year. While for Case 3, the LN scheme gives very stable results (regardless of step sizes) and outperforms the IDG scheme and the DG scheme.

7. CONCLUSION

In this paper, we have presented an efficient approach to computing the first and the second order price sensitivities in the Heston model. We have discussed the applicability of the pathwise method given that most existing numerical schemes are not Lipschitz continuous in model inputs. Depending on the model inputs and the discretization step size, our numerical tests show that the sample means of price sensitivities obtained using the Lognormal scheme and the Quadratic-Exponential scheme can be highly skewed and

have fat-tailed distribution, while price sensitivities obtained using the Integrated Double Gamma scheme and the Double Gamma scheme remain stable. The ideas and techniques presented in this paper could be easily applied to other numerical schemes.

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. (2004). Exact Simulation Of Option Greeks Under Stochastic Volatility And Jump Diffusion Models. R. G. Ingalls, M. D. Rossetti, J. S. Smith, B. A. Peters, eds. *Proc. 2004 Winter Simulation Conf.* Institute of Electrical and Electronics Engineers, Piscataway, NJ, 1607-1615.
- [5] Broadie, M., and Kaya, Ö. (2006). Exact simulation of stochastic volatility and other affine jump diffusion processes. *Operations Research* **54**(2), 217-231.
- [6] Chan, J. H. and Joshi, M. S. (2009a). Minimal Partial Proxy Simulation Schemes for Generic and Robust Monte-Carlo Greeks. To appear in *Journal of Computational Finance*.
- [7] Chan, J. H. and Joshi, M. S. (2009b). Fast Monte-Carlo Greeks for Financial Products with Discontinuous Pay-Offs. <http://ssrn.com/abstract=1500343>.
- [8] Chan, J. H. and Joshi, M. S. (2010). Fast and Accurate Long Stepping Simulation of the Heston Stochastic Volatility Model. <http://ssrn.com/abstract=1617187>
- [9] Cox, J., Ingersoll, J., and Ross, S. A. (1985). A theory of term structure of interest rates. *Econometrica* **53**(2), 185-407.
- [10] Dufresne, D. (2001). The integrated square-root process. Working Paper, University of Montreal.
- [11] Glasserman, P. (2003). *Monte Carlo Methods in Financial Engineering (Stochastic Modelling and Applied Probability)*. Springer, New York.
- [12] 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
- [13] Jäckel, P. (2002). *Monte Carlo Methods in Finance*. Wiley, New York.
- [14] Joshi, M. S. (2003). *The Concepts and Practice of Mathematical Finance*. Cambridge University Press, Cambridge.
- [15] Joshi, M. S. and Yang, C. (2009). Efficient Greek Estimation in Generic Market Models. <http://ssrn.com/abstract=1437847>
- [16] Joshi, M. S. and Yang, C. (2010). Algorithmic Hessians and the Fast Computation of Cross-Gamma Risk. <http://ssrn.com/abstract=1626547>
- [17] QuantLib, a free/open-source library for quantitative finance, Release 0.9.6 6th August 2008, <http://quantlib.org/>
- [18] 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
- [19] Van Haastrecht, A., Pelsser, A. (2008) Efficient, almost exact simulation of the Heston stochastic volatility model. Working paper, University of Amsterdam. <http://papers.ssrn.com>.

APPENDIX A

Sensitivities of Gamma Random Variables. In this section, we present an efficient method to obtain the first and second order sensitivities of the inverse of the gamma distribution. This method is an extension of the approach presented by Chan and Joshi (2010).

We let $F_\alpha(x)$ to be the cumulative distribution function for $\Gamma(\alpha, 1)$ i.e.

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

and $F_\alpha^{-1}(u_i) : [0, 1) \rightarrow [0, \infty)$ to represent the inverse of the gamma distribution function. Chan and Joshi (2010) suggested to cache the inverse of the gamma distribution by tabulating $[g(u_i), F_\alpha^{-1}(u_i)]$ for all u_i

where

$$\begin{aligned} g(u) &= \left((\alpha_c \Gamma(\alpha_c))^{1/\alpha_c} - \log(1-u) \right) u^{1/\alpha_c} \\ u_i &= \frac{i-1}{N} \quad \text{for } i = 1, 2, \dots, N \end{aligned}$$

with α_c being a constant taking the value of α and N representing the total number of points in the cache. They also tabulated the first order derivatives of $F_\alpha^{-1}(u_i)$ w.r.t $g(u_i)$ for all i i.e.

$$\frac{dF_\alpha^{-1}(u_i)}{dg(u_i)} = \frac{dF_\alpha^{-1}(u_i)}{du_i} \frac{du_i}{dg(u_i)}$$

where

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

When sampling gamma random variables, Chan and Joshi (2010) proposed to use a cubic interpolation between the adjacent points in the cache. Specifically, given a random number U such that $u_i \leq U < u_{i+1}$, the gamma random variable (i.e $F_\alpha^{-1}(U)$) is given by

$$F_\alpha^{-1}(U) = a_i g(U)^3 + b_i g(U)^2 + c_i g(U) + d_i, \quad (\text{A-1})$$

where

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

with

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

Note that, for a quick simulation, one usually pre-computes and stores the value of a_i , b_i , c_i and d_i for all i at the beginning of the simulation.

Using their approach, we can compute the the first and second order sensitivities of $F_\alpha^{-1}(U)$ w.r.t to α quickly by differentiating the equation (A-1) directly. In particular, we pre-compute and store the first and second order derivatives of a_i , b_i , c_i and d_i w.r.t to α for all i at the beginning of the simulation. In order to do so, we first need to compute

$$\frac{\partial F_\alpha^{-1}(u_i)}{\partial \alpha}, \quad \frac{\partial^2 F_\alpha^{-1}(u_i)}{\partial \alpha^2}, \quad \frac{\partial}{\partial \alpha} \left(\frac{dF_\alpha^{-1}(u_i)}{dg(u_i)} \right), \quad \text{and} \quad \frac{\partial^2}{\partial \alpha^2} \left(\frac{dF_\alpha^{-1}(u_i)}{dg(u_i)} \right) \quad (\text{A-2})$$

for all i . We know that

$$u_i = F_\alpha(F_\alpha^{-1}(u_i)) = \int_0^{F_\alpha^{-1}(u_i)} \frac{y^{\alpha-1} \exp(-y)}{\Gamma(\alpha)} dy \quad (\text{A-3})$$

from the definition of gamma distribution function. As the closed-form solution for the inverse of the gamma distribution does not exist, we will work with the representation in equation (A-3) when computing

the sensitivities. Note that, for clarity of notation, we let

$$F_{\alpha}^{-1}(\cdot) \equiv F^{-1}(\cdot, \alpha) \quad \text{and} \quad F_{\alpha}(\cdot) \equiv F(\cdot, \alpha)$$

and hence, equation (A-3) becomes

$$u_i = F(F^{-1}(u_i, \alpha), \alpha). \quad (\text{A-4})$$

By differentiating both sides of equation (A-4) w.r.t α , we have

$$0 = \frac{\partial F}{\partial \alpha} + \frac{\partial F}{\partial F^{-1}} \frac{\partial F^{-1}}{\partial \alpha}$$

and a simple rearrangement yields

$$\frac{\partial F^{-1}}{\partial \alpha} = -\frac{\partial F}{\partial \alpha} \Big/ \frac{\partial F}{\partial F^{-1}} \quad (\text{A-5})$$

Using a similar approach, we have the second order derivative of F^{-1} w.r.t α given by

$$\frac{\partial^2 F^{-1}}{\partial \alpha^2} = -\left(\frac{\partial^2 F}{\partial \alpha^2} + 2 \frac{\partial^2 F}{\partial F^{-1} \partial \alpha} \frac{\partial F^{-1}}{\partial \alpha} + \frac{\partial^2 F}{\partial (F^{-1})^2} \left(\frac{\partial F}{\partial F^{-1}} \right)^2 \right) \Big/ \frac{\partial F}{\partial F^{-1}} \quad (\text{A-6})$$

Since the closed-form solutions for $\frac{\partial F}{\partial \alpha}$ and $\frac{\partial^2 F}{\partial \alpha^2}$ do not exist, we approximate them using finite difference method while the remaining partial derivatives on the RHS of the equations (A-5) and (A-6) are evaluated analytically. The partial derivatives $\frac{\partial F^{-1}}{\partial \alpha}$ and $\frac{\partial^2 F^{-1}}{\partial \alpha^2}$ can therefore be obtained easily for all i . Using the results for $\frac{\partial F^{-1}}{\partial \alpha}$ and $\frac{\partial^2 F^{-1}}{\partial \alpha^2}$, the partial derivatives

$$\frac{\partial}{\partial \alpha} \left(\frac{dF^{-1}(u_i, \alpha)}{dg(u_i)} \right), \quad \text{and} \quad \frac{\partial^2}{\partial \alpha^2} \left(\frac{dF^{-1}(u_i, \alpha)}{dg(u_i)} \right)$$

can also be evaluated easily for all i as we have

$$\frac{dF^{-1}(u_i, \alpha)}{dg(u_i)} = \Gamma(\alpha) \exp(F^{-1}(u_i, \alpha)) (F^{-1}(u_i, \alpha))^{1-\alpha} \cdot \frac{\partial u_i}{\partial g(u_i)} \quad (\text{A-7})$$

with $\frac{\partial u_i}{\partial g(u_i)}$ independent of α . Differentiating equation (A-7) w.r.t α is straight forward.

Once we have evaluated the partial derivatives listed in (A-2), computing and storing the first and the second order derivatives of a_i , b_i , c_i and d_i w.r.t to α for all i are straight-forward. The first order sensitivities of $F_{\alpha}^{-1}(U)$ w.r.t to α is given by

$$\frac{\partial F_{\alpha}^{-1}(U)}{\partial \alpha} = \frac{\partial a_i}{\partial \alpha} g(U)^3 + \frac{\partial b_i}{\partial \alpha} g(U)^2 + \frac{\partial c_i}{\partial \alpha} g(U) + \frac{\partial d_i}{\partial \alpha}. \quad (\text{A-8})$$

and, similarly, the second order sensitivities of $F_{\alpha}^{-1}(U)$ w.r.t to α is given by

$$\frac{\partial^2 F_{\alpha}^{-1}(U)}{\partial \alpha^2} = \frac{\partial^2 a_i}{\partial \alpha^2} g(U)^3 + \frac{\partial^2 b_i}{\partial \alpha^2} g(U)^2 + \frac{\partial^2 c_i}{\partial \alpha^2} g(U) + \frac{\partial^2 d_i}{\partial \alpha^2}. \quad (\text{A-9})$$

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.joshi@gmail.com

IDG scheme - Call Option - Case 1													
	Actual	1 Step per year				2 Steps per year				4 Steps per year			
		mean	std dev	skewness	kurtosis	mean	std dev	skewness	kurtosis	mean	std dev	skewness	kurtosis
Price	4.403	4.402	0.00	0.01	-0.07	4.403	0.00	-0.04	0.17	4.403	0.01	0.03	-0.05
∂S	0.80	0.80	0.00	-0.06	0.11	0.80	0.00	0.03	0.10	0.80	0.00	-0.02	-0.39
∂V	60.21	60.25	0.13	0.03	0.06	60.23	0.22	-0.01	0.06	60.22	0.30	-0.02	-0.04
$\partial \kappa$	1.31	1.31	0.01	0.09	0.05	1.31	0.03	0.04	0.08	1.31	0.05	0.01	-0.01
$\partial \theta$	24.75	24.77	0.12	0.11	0.13	24.76	0.34	0.03	0.06	24.77	0.65	0.01	0.01
$\partial \epsilon$	-2.71	-2.71	0.01	-0.14	0.03	-2.71	0.03	0.03	0.13	-2.71	0.06	-0.01	-0.01
$\partial \rho$	2.61	2.60	0.03	0.06	-0.05	2.60	0.04	-0.01	0.07	2.61	0.05	0.04	0.07
$\partial S \partial S$	0.03	0.03	0.00	0.00	0.01	0.03	0.00	0.06	0.01	0.03	0.00	0.01	-0.03
$\partial V \partial V$	-278.00	-278.47	10.09	0.05	-0.04	-278.84	15.46	0.06	0.11	-278.34	27.60	0.05	0.07
$\partial \kappa \partial \kappa$	-0.72	-0.71	0.20	0.21	0.29	-0.72	0.21	0.12	0.00	-0.73	0.44	0.09	-0.00
$\partial \theta \partial \theta$	-78.24	-78.17	21.55	0.07	0.18	-78.17	34.96	0.11	-0.02	-80.13	74.07	0.10	0.03
$\partial \epsilon \partial \epsilon$	3.28	3.27	0.17	0.11	0.03	3.28	0.38	0.06	0.08	3.27	0.70	0.06	0.02
$\partial \rho \partial \rho$	-8.48	-8.26	4.88	0.57	1.00	-8.45	4.95	0.59	0.99	-8.32	4.97	0.65	1.26
Time		0.56 seconds				0.90 seconds				1.53 seconds			

DG scheme - Call Option - Case 1													
	Actual	8 Steps per year				16 Steps per year				32 Steps per year			
		mean	std dev	skewness	kurtosis	mean	std dev	skewness	kurtosis	mean	std dev	skewness	kurtosis
Price	4.403	4.385	0.01	0.01	0.03	4.398	0.01	0.02	-0.00	4.401	0.01	-0.06	0.08
∂S	0.80	0.80	0.00	0.02	0.05	0.80	0.00	-0.01	0.14	0.80	0.00	0.00	-0.15
∂V	60.21	59.97	0.20	0.03	0.07	60.16	0.27	0.01	-0.09	60.21	0.40	-0.03	-0.04
$\partial \kappa$	1.31	1.32	0.02	0.08	-0.01	1.32	0.03	0.06	-0.12	1.31	0.05	0.03	-0.02
$\partial \theta$	24.75	25.05	0.26	0.10	0.04	24.84	0.36	0.07	-0.07	24.77	0.57	0.03	-0.01
$\partial \epsilon$	-2.71	-2.74	0.03	-0.09	0.07	-2.72	0.03	-0.04	-0.01	-2.71	0.05	-0.02	-0.17
$\partial \rho$	2.61	2.60	0.05	0.02	-0.08	2.59	0.06	0.11	0.16	2.60	0.06	0.06	0.10
$\partial S \partial S$	0.03	0.03	0.00	-0.03	-0.05	0.03	0.00	-0.01	-0.05	0.03	0.00	-0.02	-0.11
$\partial V \partial V$	-278.00	-271.62	12.04	0.14	0.13	-276.67	16.76	0.05	-0.04	-278.13	31.05	0.03	0.06
$\partial \kappa \partial \kappa$	-0.72	-0.72	0.11	0.20	0.14	-0.72	0.15	0.13	0.01	-0.72	0.26	0.04	0.08
$\partial \theta \partial \theta$	-78.24	-81.69	15.18	0.21	0.10	-79.31	22.70	0.12	0.11	-77.80	40.16	0.07	0.06
$\partial \epsilon \partial \epsilon$	3.28	3.28	0.22	0.10	0.06	3.27	0.31	0.05	0.06	3.28	0.55	0.02	-0.01
$\partial \rho \partial \rho$	-8.48	-6.15	5.01	0.63	1.02	-8.20	4.93	0.62	1.44	-8.36	4.94	0.56	1.08
Time		0.51 seconds				0.96 seconds				1.78 seconds			

LN scheme - Call Option - Case 1													
	Actual	16 Steps per year				32 Steps per year				64 Steps per year			
		mean	std dev	skewness	kurtosis	mean	std dev	skewness	kurtosis	mean	std dev	skewness	kurtosis
Price	4.403	4.254	0.01	-0.01	0.06	4.324	0.01	0.01	0.02	4.364	0.01	-0.01	0.07
∂S	0.80	0.77	0.00	-0.05	-0.02	0.78	0.00	0.00	0.03	0.79	0.00	0.02	0.01
∂V	60.21	57.93	0.17	0.04	0.04	59.29	0.25	-0.09	-0.01	59.86	0.39	-0.20	0.23
$\partial \kappa$	1.31	1.11	0.01	-0.03	0.00	1.21	0.01	-0.03	0.04	1.27	0.01	-0.04	-0.03
$\partial \theta$	24.75	22.84	0.10	-0.07	0.02	23.81	0.14	-0.06	0.05	24.32	0.20	-0.08	0.08
$\partial \epsilon$	-2.71	-2.57	0.02	-0.02	0.04	-2.66	0.02	0.10	0.08	-2.70	0.04	0.23	0.31
$\partial \rho$	2.61	3.91	0.05	0.00	-0.04	3.19	0.05	-0.04	-0.05	2.85	0.06	0.07	0.03
$\partial S \partial S$	0.03	0.04	0.00	-0.03	0.13	0.04	0.00	-0.02	0.10	0.03	0.00	0.04	0.06
$\partial V \partial V$	-278.00	-263.05	61.82	0.21	0.21	-268.57	224.95	0.50	0.76	-295.92	2,798.51	8.30	149.12
$\partial \kappa \partial \kappa$	-0.72	-0.55	0.11	0.16	0.19	-0.63	0.31	0.53	0.92	-0.67	1.89	3.45	56.82
$\partial \theta \partial \theta$	-78.24	-58.57	21.95	0.15	0.16	-66.13	61.48	0.55	1.07	-71.89	370.66	3.26	47.06
$\partial \epsilon \partial \epsilon$	3.28	3.45	0.60	0.21	0.18	3.42	2.42	0.58	1.20	3.18	27.40	7.65	132.82
$\partial \rho \partial \rho$	-8.48	-8.46	4.84	0.56	1.46	-7.49	4.97	0.57	1.37	-7.62	5.01	0.66	1.73
Time		1.07 seconds				2.06 seconds				3.98 seconds			

TABLE 6.4. Tables above shows the estimated mean, standard deviation, skewness and excess kurtosis of the first and second order price sensitivities of call option obtained using LN scheme for Case 1. The results were calculated using 5,000 independent simulations with each simulation consisting 65,535 paths.

IDG scheme - Call Option - Case 2													
	Actual	1 Step per year				2 Steps per year				4 Steps per year			
		mean	std dev	skewness	kurtosis	mean	std dev	skewness	kurtosis	mean	std dev	skewness	kurtosis
Price	5.100	5.100	0.01	0.09	0.18	5.099	0.01	0.03	0.19	5.099	0.01	0.07	0.02
∂S	0.66	0.66	0.00	- 0.01	- 0.00	0.66	0.00	0.07	- 0.07	0.66	0.00	0.02	- 0.09
∂V	74.96	74.91	0.46	0.07	0.02	74.97	0.56	0.01	0.05	74.99	0.71	0.00	0.15
$\partial \kappa$	1.32	1.31	0.06	0.08	0.05	1.32	0.10	0.06	0.11	1.32	0.14	0.02	0.03
$\partial \theta$	16.39	16.36	0.48	0.11	0.00	16.39	0.75	0.08	0.20	16.39	1.04	0.02	0.03
$\partial \epsilon$	- 2.72	- 2.71	0.06	- 0.07	0.09	- 2.72	0.08	- 0.09	0.11	- 2.72	0.11	0.00	0.05
$\partial \rho$	1.18	1.19	0.02	0.14	0.24	1.18	0.03	0.09	0.15	1.18	0.04	0.03	0.04
$\partial S \partial S$	0.05	0.05	0.00	0.02	0.01	0.05	0.00	0.00	- 0.04	0.05	0.00	- 0.03	0.03
$\partial V \partial V$	- 519.39	- 517.45	23.83	0.24	0.32	- 519.54	31.56	0.10	0.12	- 519.85	58.25	0.03	0.01
$\partial \kappa \partial \kappa$	- 0.91	- 0.91	0.50	0.22	0.29	- 0.92	1.12	0.23	0.07	- 0.93	1.72	0.13	- 0.04
$\partial \theta \partial \theta$	- 38.26	- 38.77	32.01	0.27	0.36	- 39.38	71.95	0.22	0.09	- 40.10	103.11	0.13	- 0.02
$\partial \epsilon \partial \epsilon$	2.86	2.90	0.56	0.27	0.42	2.87	0.93	0.15	- 0.00	2.84	1.30	0.12	0.05
$\partial \rho \partial \rho$	- 1.86	- 1.85	1.82	1.14	4.37	- 1.84	1.82	0.75	1.69	- 1.83	1.80	0.88	2.64
Time		0.57 seconds				0.88 seconds				1.52 seconds			

DG scheme - Call Option - Case 2													
	Actual	8 Steps per year				16 Steps per year				32 Steps per year			
		mean	std dev	skewness	kurtosis	mean	std dev	skewness	kurtosis	mean	std dev	skewness	kurtosis
Price	5.100	5.154	0.01	0.09	0.10	5.112	0.02	0.02	0.12	5.102	0.02	0.06	0.20
∂S	0.66	0.64	0.00	0.07	- 0.10	0.66	0.00	- 0.03	- 0.12	0.66	0.00	- 0.02	0.10
∂V	74.96	74.50	0.56	0.10	0.18	74.81	0.67	0.12	0.11	74.92	1.06	0.10	- 0.11
$\partial \kappa$	1.32	1.23	0.06	0.08	0.16	1.29	0.07	0.01	0.17	1.31	0.10	- 0.06	0.11
$\partial \theta$	16.39	15.70	0.45	0.08	0.17	16.19	0.53	0.04	0.22	16.37	0.76	- 0.08	0.09
$\partial \epsilon$	- 2.72	- 2.55	0.05	- 0.04	0.09	- 2.67	0.07	- 0.05	- 0.04	- 2.71	0.11	- 0.03	- 0.06
$\partial \rho$	1.18	1.37	0.04	0.02	0.07	1.24	0.05	0.06	- 0.03	1.19	0.05	0.02	0.02
$\partial S \partial S$	0.05	0.05	0.00	- 0.00	- 0.02	0.05	0.00	0.07	0.08	0.05	0.00	0.07	0.13
$\partial V \partial V$	- 519.39	- 526.21	34.34	0.32	0.98	- 518.82	44.58	0.48	0.81	- 520.46	90.86	0.22	0.17
$\partial \kappa \partial \kappa$	- 0.91	- 0.74	0.42	0.36	0.28	- 0.84	0.53	0.21	0.04	- 0.88	0.80	0.13	0.01
$\partial \theta \partial \theta$	- 38.26	- 28.68	24.51	0.38	0.37	- 34.48	30.07	0.24	0.07	- 36.39	45.63	0.13	0.00
$\partial \epsilon \partial \epsilon$	2.86	3.07	0.43	0.15	0.02	2.98	0.57	0.21	0.02	2.90	1.08	0.04	0.17
$\partial \rho \partial \rho$	- 1.86	- 2.16	1.82	0.80	2.29	- 1.86	1.82	0.82	2.46	- 1.89	1.79	0.73	2.49
Time		0.53 seconds				0.96 seconds				1.77 seconds			

LN scheme - Call Option - Case 2													
	Actual	16 Steps per year				32 Steps per year				64 Steps per year			
		mean	std dev	skewness	kurtosis	mean	std dev	skewness	kurtosis	mean	std dev	skewness	kurtosis
Price	5.100	5.328	0.01	0.02	0.06	5.216	0.01	- 0.02	0.07	5.155	0.02	0.00	- 0.08
∂S	0.66	0.62	0.00	- 0.02	0.04	0.64	0.00	0.03	- 0.15	0.65	0.00	0.00	0.01
∂V	74.96	73.32	0.24	- 0.01	0.17	73.91	0.36	0.00	0.02	74.40	0.54	0.05	- 0.02
$\partial \kappa$	1.32	1.12	0.01	0.03	- 0.00	1.22	0.01	0.01	- 0.04	1.27	0.02	- 0.03	0.09
$\partial \theta$	16.39	15.47	0.07	0.05	- 0.02	15.95	0.10	0.04	0.12	16.18	0.16	- 0.01	0.01
$\partial \epsilon$	- 2.72	- 2.23	0.02	0.05	0.08	- 2.45	0.03	- 0.04	0.06	- 2.58	0.05	- 0.04	- 0.02
$\partial \rho$	1.18	1.72	0.04	- 0.00	- 0.17	1.49	0.04	0.06	- 0.07	1.34	0.05	0.06	- 0.00
$\partial S \partial S$	0.05	0.04	0.00	- 0.02	0.08	0.05	0.00	- 0.01	- 0.08	0.05	0.00	0.03	- 0.04
$\partial V \partial V$	- 519.39	- 520.44	36.33	0.24	0.11	- 509.51	125.89	0.60	1.28	- 472.47	1,703.43	7.33	134.40
$\partial \kappa \partial \kappa$	- 0.91	- 0.81	0.09	0.23	0.10	- 0.88	0.25	0.67	2.29	- 0.95	2.43	- 16.17	729.88
$\partial \theta \partial \theta$	- 38.26	- 36.09	6.30	0.25	0.12	- 38.30	16.67	0.65	2.05	- 40.84	154.18	- 13.82	593.87
$\partial \epsilon \partial \epsilon$	2.86	2.78	0.34	0.30	0.19	2.89	1.40	0.71	1.69	3.25	18.15	6.76	119.91
$\partial \rho \partial \rho$	- 1.86	- 3.34	1.79	0.85	2.64	- 2.61	1.80	1.04	3.88	- 2.22	1.79	1.05	3.55
Time		1.14 seconds				2.11 seconds				3.99 seconds			

TABLE 6.5. Tables above show the estimated mean, standard deviation, skewness and excess kurtosis of the first and second order price sensitivities of call option obtained using the IDG scheme, the DG scheme and the LN scheme for Case 2. The results were calculated using 5,000 independent simulations with each simulation consisting 65,535 paths.

IDG scheme - Call Option - Case 3													
	Actual	1 Step per year				2 Steps per year				4 Steps per year			
		mean	std dev	skewness	kurtosis	mean	std dev	skewness	kurtosis	mean	std dev	skewness	kurtosis
Price	9.774	9.773	0.01	0.17	0.09	9.773	0.02	0.06	0.15	9.773	0.02	0.04	- 0.14
∂S	0.61	0.61	0.00	- 0.01	- 0.05	0.61	0.00	- 0.03	0.02	0.61	0.00	- 0.02	- 0.09
∂V	39.49	39.48	0.41	0.14	0.20	39.49	0.55	0.06	0.28	39.50	0.70	0.07	0.04
$\partial \kappa$	1.00	1.00	0.04	- 0.11	0.46	1.00	0.08	0.00	0.03	1.01	0.13	0.02	- 0.00
$\partial \theta$	27.57	27.56	0.37	0.10	0.15	27.57	0.80	0.07	0.10	27.59	1.38	0.00	0.07
$\partial \epsilon$	- 2.64	- 2.63	0.10	- 0.10	0.17	- 2.64	0.17	- 0.04	0.03	- 2.64	0.28	0.02	0.01
$\partial \rho$	1.15	1.15	0.04	0.18	0.00	1.15	0.05	0.19	0.33	1.15	0.07	0.04	0.22
$\partial S \partial S$	0.02	0.02	0.00	0.00	- 0.04	0.02	0.00	0.02	0.08	0.02	0.00	0.01	- 0.04
$\partial V \partial V$	- 109.28	- 109.52	11.07	0.19	0.12	- 109.69	17.90	0.14	0.15	- 108.35	37.14	0.11	0.09
$\partial \kappa \partial \kappa$	- 0.68	- 0.68	0.12	0.55	1.65	- 0.68	0.27	0.51	5.29	- 0.68	0.60	0.07	0.17
$\partial \theta \partial \theta$	- 73.56	- 73.47	11.04	0.32	0.39	- 73.49	41.52	0.06	- 0.01	- 73.84	88.76	0.05	0.31
$\partial \epsilon \partial \epsilon$	0.83	0.82	0.99	0.28	0.36	0.81	2.32	0.04	0.03	0.82	4.38	0.04	0.13
$\partial \rho \partial \rho$	- 1.30	- 1.30	2.30	0.67	1.30	- 1.27	2.30	0.59	1.26	- 1.31	2.29	0.76	1.82
Time		0.56 seconds				0.92 seconds				1.61 seconds			

DG scheme - Call Option - Case 3													
	Actual	8 Steps per year				16 Steps per year				32 Steps per year			
		mean	std dev	skewness	kurtosis	mean	std dev	skewness	kurtosis	mean	std dev	skewness	kurtosis
Price	9.774	9.798	0.02	0.12	0.01	9.780	0.02	0.03	- 0.05	9.775	0.03	0.11	0.17
∂S	0.61	0.61	0.00	0.00	- 0.05	0.61	0.00	- 0.02	0.06	0.61	0.00	- 0.05	0.21
∂V	39.49	39.66	0.48	0.28	1.21	39.54	0.60	0.08	0.17	39.51	1.01	0.10	- 0.05
$\partial \kappa$	1.00	0.97	0.06	0.05	0.01	1.00	0.08	- 0.02	- 0.05	1.00	0.10	0.07	0.02
$\partial \theta$	27.57	27.19	0.66	0.10	0.01	27.49	0.71	- 0.00	0.13	27.56	1.04	0.05	0.04
$\partial \epsilon$	- 2.64	- 2.57	0.13	- 0.10	0.01	- 2.62	0.16	- 0.02	0.09	- 2.64	0.26	- 0.05	- 0.12
$\partial \rho$	1.15	1.20	0.07	0.05	0.15	1.16	0.08	- 0.00	0.21	1.15	0.09	- 0.01	- 0.01
$\partial S \partial S$	0.02	0.02	0.00	0.04	0.07	0.02	0.00	0.04	- 0.07	0.02	0.00	- 0.03	0.04
$\partial V \partial V$	- 109.28	- 114.15	17.27	0.92	8.19	- 110.37	22.43	0.51	1.14	- 109.43	47.12	0.24	0.20
$\partial \kappa \partial \kappa$	- 0.68	- 0.64	0.14	0.14	0.09	- 0.67	0.19	0.06	0.09	- 0.68	0.33	0.11	0.05
$\partial \theta \partial \theta$	- 73.56	- 67.91	17.32	0.12	- 0.09	- 71.54	24.12	0.03	0.14	- 72.68	41.63	0.10	0.04
$\partial \epsilon \partial \epsilon$	0.83	0.92	1.02	0.14	0.14	0.91	1.60	0.09	0.10	0.88	3.03	0.04	- 0.13
$\partial \rho \partial \rho$	- 1.30	- 1.48	2.25	0.61	1.20	- 1.35	2.29	0.58	1.04	- 1.37	2.30	0.63	1.42
Time		0.53 seconds				0.97 seconds				1.89 seconds			

LN scheme - Call Option - Case 3													
	Actual	16 Steps per year				32 Steps per year				64 Steps per year			
		mean	std dev	skewness	kurtosis	mean	std dev	skewness	kurtosis	mean	std dev	skewness	kurtosis
Price	9.774	9.925	0.01	0.10	0.12	9.853	0.02	0.03	0.08	9.812	0.03	0.07	- 0.01
∂S	0.61	0.60	0.00	0.04	- 0.11	0.60	0.00	- 0.04	0.07	0.60	0.00	- 0.01	- 0.02
∂V	39.49	39.06	0.08	0.02	0.02	39.22	0.11	0.05	0.01	39.35	0.14	0.03	0.02
$\partial \kappa$	1.00	0.87	0.01	- 0.09	0.14	0.94	0.01	- 0.10	0.07	0.97	0.01	- 0.08	- 0.09
$\partial \theta$	27.57	26.67	0.06	- 0.03	- 0.04	27.11	0.09	0.00	0.06	27.35	0.11	0.05	- 0.01
$\partial \epsilon$	- 2.64	- 2.24	0.02	0.13	0.12	- 2.42	0.03	0.12	0.34	- 2.53	0.04	0.05	0.03
$\partial \rho$	1.15	1.49	0.05	0.06	- 0.02	1.34	0.07	0.07	0.05	1.25	0.08	0.07	0.12
$\partial S \partial S$	0.02	0.02	0.00	0.04	- 0.08	0.02	0.00	- 0.00	0.02	0.02	0.00	0.08	- 0.01
$\partial V \partial V$	- 109.28	- 109.47	2.23	0.05	- 0.02	- 108.27	2.89	0.04	0.00	- 108.46	4.38	0.07	- 0.04
$\partial \kappa \partial \kappa$	- 0.68	- 0.57	0.01	0.03	- 0.01	- 0.62	0.01	0.07	- 0.05	- 0.65	0.02	0.05	0.04
$\partial \theta \partial \theta$	- 73.56	- 66.18	1.63	0.02	0.01	- 69.83	2.39	0.05	- 0.05	- 71.72	3.93	0.12	0.08
$\partial \epsilon \partial \epsilon$	0.83	1.15	0.09	0.30	1.49	1.05	0.16	0.04	0.01	0.95	0.32	0.10	0.06
$\partial \rho \partial \rho$	- 1.30	- 2.80	2.35	0.71	1.61	- 2.06	2.37	0.56	1.58	- 1.63	2.32	0.65	1.66
Time		1.07 seconds				2.07 seconds				4.01 seconds			

TABLE 6.6. Tables above show the estimated mean, standard deviation, skewness and excess kurtosis of the first and second order price sensitivities of call option obtained using the IDG scheme, the DG scheme and the LN scheme for Case 3. The results were calculated using 5,000 independent simulations with each simulation consisting 65,535 paths.

$\partial S \partial S$	0.03	0.00	0.1	0.2	0.03	0.00	0.0	0.1	0.03	0.00	0.1	-0.0	0.03	0.00	-0.0	-0.1	0.03	0.00	0.0	0.0	0.1
$\partial S \partial V$	0.04	0.06	-0.0	0.1	0.05	0.05	0.1	0.0	0.04	0.06	0.1	0.1	0.07	0.05	-0.0	-0.0	0.06	0.05	0.0	-0.0	0.1
$\partial S \partial \kappa$	-0.01	0.01	-0.0	0.1	-0.01	0.00	0.1	-0.0	-0.01	0.01	-0.1	0.0	-0.01	0.00	-0.0	0.0	-0.01	0.00	-0.0	-0.1	0.1
$\partial S \partial \theta$	-0.06	0.08	-0.0	0.1	-0.06	0.06	0.1	-0.0	-0.06	0.08	-0.0	0.1	-0.04	0.04	-0.0	-0.0	-0.05	0.05	-0.1	-0.1	0.1
$\partial S \partial \epsilon$	0.03	0.02	-0.0	0.0	0.03	0.02	-0.1	0.0	0.03	0.02	-0.0	0.0	0.02	0.01	-0.0	-0.1	0.02	0.01	0.0	-0.1	0.1
$\partial S \partial \rho$	-0.18	0.02	0.0	0.1	-0.17	0.02	0.0	-0.1	-0.17	0.02	0.0	0.1	-0.16	0.02	0.1	-0.0	-0.17	0.02	0.0	0.0	0.0
$\partial V \partial V$	-127.82	36.98	0.0	-0.0	-128.40	24.87	0.2	0.2	-128.22	46.38	0.0	0.1	-126.63	3.39	-0.0	-0.0	-126.82	4.66	0.1	-0.1	0.1
$\partial V \partial \kappa$	-7.85	3.09	0.0	-0.0	-7.81	1.56	0.0	-0.0	-7.80	3.06	-0.0	-0.1	-7.68	0.16	0.0	0.1	-7.74	0.22	0.1	0.1	0.1
$\partial V \partial \theta$	-46.19	34.94	-0.0	0.0	-46.17	17.30	0.1	-0.0	-45.98	33.61	-0.0	-0.1	-44.94	2.09	0.0	0.1	-45.48	2.87	0.1	0.1	0.1
$\partial V \partial \epsilon$	-1.81	9.40	-0.0	-0.0	-1.70	5.74	-0.2	0.1	-1.79	10.61	0.0	0.1	-2.03	0.72	0.0	-0.0	-1.98	1.11	-0.1	-0.0	0.0
$\partial V \partial \rho$	1.22	1.48	0.0	-0.0	1.20	1.40	0.0	-0.0	1.27	1.86	0.1	0.0	1.32	0.89	-0.2	0.2	1.27	0.92	-0.2	0.2	0.2
$\partial \kappa \partial \kappa$	-0.15	0.54	0.1	0.0	-0.15	0.19	0.1	0.1	-0.15	0.34	0.1	-0.1	-0.13	0.02	0.1	0.0	-0.14	0.02	0.1	0.0	0.0
$\partial \kappa \partial \theta$	7.57	6.48	0.1	0.0	7.55	2.30	0.1	0.0	7.53	3.91	0.0	-0.1	7.53	0.21	0.1	-0.1	7.55	0.29	0.1	0.1	0.1
$\partial \kappa \partial \epsilon$	0.35	1.28	-0.1	0.0	0.35	0.49	-0.1	0.1	0.35	0.88	-0.0	0.0	0.31	0.04	-0.1	-0.1	0.33	0.06	-0.1	0.1	0.1
$\partial \kappa \partial \rho$	-0.05	0.14	-0.1	0.1	-0.06	0.11	-0.2	0.2	-0.05	0.12	-0.1	0.2	-0.07	0.05	-0.1	0.4	-0.06	0.05	-0.0	0.1	0.1
$\partial \theta \partial \theta$	-16.99	73.24	0.1	0.0	-16.70	24.56	0.1	0.1	-17.65	42.53	0.1	-0.1	-16.09	2.94	0.1	-0.0	-16.37	3.77	0.1	0.1	0.1
$\partial \theta \partial \epsilon$	1.08	14.98	-0.1	0.1	1.08	5.51	-0.1	0.1	1.15	9.84	-0.0	0.0	0.84	0.66	-0.1	-0.0	0.94	0.94	-0.1	0.2	0.2
$\partial \theta \partial \rho$	0.16	1.14	0.0	0.0	0.04	0.71	0.0	0.0	0.12	0.89	-0.0	-0.0	0.03	0.46	-0.1	0.1	0.07	0.50	-0.1	0.0	0.0
$\partial \epsilon \partial \epsilon$	0.06	3.58	0.1	0.1	0.05	1.64	0.1	0.0	0.05	3.04	0.0	0.1	0.15	0.22	0.1	-0.1	0.12	0.34	0.1	0.1	0.1
$\partial \epsilon \partial \rho$	0.50	0.40	-0.0	0.0	0.54	0.35	-0.1	-0.1	0.50	0.43	0.0	-0.1	0.62	0.14	-0.0	0.1	0.58	0.15	-0.0	0.0	0.0
$\partial \rho \partial \rho$	-0.65	1.05	0.5	1.0	-0.68	1.04	0.5	0.9	-0.66	1.05	0.3	0.4	-1.06	1.05	0.4	0.7	-0.91	1.03	0.4	0.5	0.5
Time	3.71 seconds				1.97 seconds				3.51 seconds				3.64 seconds				4.82 seconds				

TABLE 6.9. Table above shows the estimated mean, standard deviation, skewness and excess kurtosis of the first and second order price sensitivities of the Asian call option evaluated using the IDG scheme, the DG scheme and the LN scheme for Case 3. The results were calculated using 5,000 independent simulations with each simulation consisting of 65,535 paths.