Low-bias simulation scheme for the Heston model by Inverse Gaussian approximation

S. T. Tse \textsuperscript{a} & Justin W. L. Wan \textsuperscript{a}

\textsuperscript{a} David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada

Published online: 03 Aug 2012.

To cite this article: S. T. Tse & Justin W. L. Wan (2013) Low-bias simulation scheme for the Heston model by Inverse Gaussian approximation, Quantitative Finance, 13:6, 919-937, DOI: 10.1080/14697688.2012.696678

To link to this article: http://dx.doi.org/10.1080/14697688.2012.696678
Low-bias simulation scheme for the Heston model by Inverse Gaussian approximation

S. T. TSE* and JUSTIN W. L. WAN

David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada

(Received 16 July 2010; in final form 8 May 2012)

Fast and accurate sampling of conditional time-integrated variance in the Heston model is an important and challenging problem. We proved that this very complicated distribution converges to the moment-matched Inverse Gaussian distribution as the time interval goes to infinity. Leveraging on this theoretical result, we develop an efficient and accurate Inverse Gaussian approximation to sample conditional time-integrated variance. Numerical results demonstrate that our scheme compares favourably with state-of-the-art methods in accuracy given the same computational time for moderately path-dependent options.

Keywords: Inverse Gaussian; Asymptotic exactness; Fast moment-matching; Path-dependent options; Heston model

1. Introduction

The Heston stochastic volatility model (Heston 1993) is one of the most popular extensions to the Black–Scholes model in finance. Instead of assuming volatility as a constant, the Heston model assumes that variance, or the square of volatility, follows the square root diffusion process (also known as the CIR process – Cox et al. 1985 – in interest rate modelling), which has the attractive properties of being non-negative and mean-reverting. Under the Heston model, vanilla European options can be computed rapidly by a semi-analytical formula (Heston 1993). Consequently, calibration to market prices can be performed quickly. This degree of analytical tractability of the Heston model partly explains why it has become popular in practice.

Formulae are unavailable, however, for path-dependent derivatives under the Heston model. As a result, practical applications of the Heston model often require the use of Monte Carlo simulation. The Monte Carlo method has two sources of error in calculating derivatives prices: variance and bias. Variance comes from the random nature of Monte Carlo simulation. Bias comes from non-exact time discretization of the underlying stochastic differential equations (SDEs). To reduce variance to acceptable levels, big sample sizes are often needed. To reduce bias, one general approach is to perform finer time discretization, but this can be time-consuming. A more desirable approach is to design a more accurate discretization scheme for the SDEs. On this note, the square root function in Heston dynamics has been shown to be a source of big bias for the Euler and Milstein schemes (Kahl and Jackel 2006, Lord et al. 2010), which are standard discretization methods for SDEs. First, negative values of variance have to be fixed heuristically before taking the square root in the next time step. Second, the square root function violates the Lipschitz condition typically used to establish convergence results. Although many researchers (Deelstra and Delbaen 1998, Higham and Mao 2005, Kahl and Jackel 2006, Berkaoui et al. 2007, Bossy and Diop 2007, Lord et al. 2010) have come up with innovative ways to fix these standard techniques, discretization biases under all these schemes are relatively big unless a large number of time steps is used. We refer readers to Lord et al. (2010) for a comprehensive comparison of such fixes to standard Euler and Milstein schemes.

A breakthrough was made by Broadie and Kaya (2006), who designed an essentially bias-free simulation scheme based on exact sampling from two distributions: (1) the conditional transition distribution of variance, which we will denote by \( (V(t_2)|V(t_1)) \); and (2) the time-integrated variance conditional on the levels of variance at the endpoints, which we will denote by \( I_c \). We note that \( (V(t_1)|V(t_2)) \) is distributed as a non-central chi-square and thus exact simulation is relatively straightforward.

*Corresponding author. Email: sttse@cs.uwaterloo.ca

© 2013 Taylor & Francis
The main result in Broadie and Kaya (2006) is an exact simulation scheme for \( I_c \). Specifically, the paper derives a formula for the characteristic function of \( I_c \). Based on the formula, the paper proposes to sample \( I_c \) by numerically Fourier inverting the characteristic function to obtain the cumulative distribution function (CDF), and then numerically inverting the CDF to obtain the quantile function. The biases introduced by the numerical inversions are negligible (Broadie and Kaya 2006) and hence the method is essentially bias-free. However, numerical results in the literature have shown that the exact scheme is computationally very expensive (Broadie and Kaya 2006, Glasserman and Kim 2011, Lord et al. 2010), and have worse speed–accuracy tradeoff than simpler schemes (Lord et al. 2010). The exact scheme, nevertheless, has encouraged researchers to design discretization schemes which approximate the exact distributions (Andersen 2007, Smith 2007, Glasserman and Kim 2011, Van Haastrecht and Pelsser 2008, Zhu 2008). Notably, two state-of-the-art simulation schemes for the Heston model, the Quadratic Exponential (QE) scheme (Andersen 2007), and the Gamma Expansion (GE) scheme (Glasserman and Kim 2011) employ this approach. QE focuses on accurate approximation of \( (V(t_2)|V(t_1)) \) whereas GE focuses on accurate approximation of \( I_c \). We will summarize the key ideas as well as the speed–accuracy tradeoff of the exact, QE and GE schemes in section 2.

In this paper, we will demonstrate that although the accuracy of QE is arguably sufficient when many time steps are used, the accuracy deteriorates quickly as the number of time steps is reduced. GE, while having an accuracy very close to that of the exact scheme, is relatively expensive and hence its speed–accuracy tradeoff is less favourable than QE when more than very few (two to three) time steps are needed in pricing path-dependent options. Our research tackles the cases in which pricing of the path-dependent options requires a moderate number of time steps (from four to around sixteen).

The contribution of this paper is three-fold. First, we propose the Inverse Gaussian (IG) approximation to \( I_c \) and prove its asymptotic exactness. Leveraging on the convergence result, we develop a simple and accurate method, the IG scheme, for sampling \( I_c \). Second, we propose an efficient and accurate sampling method for \( (V(t_2)|V(t_1)) \), the IPZ scheme, based on a sophisticated balance of speed and accuracy. Using an extensive set of parametric cases in the literature in our numerical tests for fair comparison, we show that the speed–accuracy tradeoff of the combined IPZ–IG scheme compares favourably to QE on pricing European calls and Asian options when a moderate number of time steps (from one to around sixteen) is used in the IPZ–IG scheme. Third, we analyse in detail the efficiency issues in approximate sampling of \( I_c \) and \( (V(t_2)|V(t_1)) \).

The rest of the paper is organized as follows. Section 2 defines the Heston dynamics and summarizes the exact, QE and GE schemes. Section 3 proposes the Inverse Gaussian approximation to \( I_c \), proves its asymptotic exactness, and presents numerical results to demonstrate its accuracy. Section 4 proposes and details the IPZ scheme for sampling \( (V(t_2)|V(t_1)) \) and the IG scheme for sampling \( I_c \). Section 5 presents numerical results. Section 6 concludes the paper, discusses extensions, and outlines directions for future research.

2. Heston model and its simulation

2.1. Definition and basic properties

The Heston model is defined by the coupled two-dimensional stochastic differential equations

\[
\frac{dX(t)}{X(t)} = r dt + \sqrt{\theta(t)\rho dW_Y(t) + \sqrt{1 - \rho^2} dW_X(t)},
\]

\[
dV(t) = \kappa(\theta - V(t)) dt + \sigma \sqrt{V(t)} dW_Y(t),
\]

in which \((W_Y, W_X)\) is a standard two-dimensional Brownian motion in the time variable \( t \), \( \kappa, \theta, \sigma \) are positive constants, \( r \) is a non-negative constant, and the correlation \( \rho \) is a constant in \([-1, 1]\). The initial conditions \( X(0) \) and \( V(0) \) are assumed to be strictly positive. \( X(t) \) represents the price of an underlying asset and \( V(t) \) represents the variance of its instantaneous returns.

The variance process \((2)\) is a square root diffusion process whose conditional transition distribution, \((V(t_2)|V(t_1))\), is well known to be that of a scaled non-central chi-square distribution (Andersen 2007). Assuming \( t_2 > t_1 \), the distribution of \( V(t_2) \) conditional on \( V(t_1) \) is given by

\[
V(t_2) = \frac{e^{-\kappa(t_2-t_1)}}{n(t_1, t_2)} \mathcal{X}^2(\delta(n(t_1, t_2)V(t_1))), \quad \delta = \frac{4\kappa \theta}{\sigma^2},
\]

\[
n(t_1, t_2) = \frac{4\kappa e^{-\kappa(t_2-t_1)}}{\sigma^2(1 - e^{-\kappa(t_2-t_1)})}.
\]

where \( \mathcal{X}^2(\lambda) \) denotes a non-central chi-square random variable with \( \delta \) degrees of freedom and non-centrality parameter \( \lambda \), and \( n(t_1, t_2) \) is defined for notational simplicity. As we will explain in detail in section 4.1, \( \mathcal{X}^2(\lambda) \) can be sampled by first conditioning on a Poisson variate and then generating a sample from a gamma distribution (Van Haastrecht and Pelsser 2008).

As shown in Broadie and Kaya (2006), the independence of \( W_Y \) and \( W_X \) implies that the distribution of log \((X(t_2)|X(t_1))\) is conditionally normal given \( V(t_1), V(t_2) \) and \( I = \int_0^{t_1} V(s) \, ds \):

\[
\log \frac{X(t_2)}{X(t_1)} \sim \mathcal{N}\left[ r(t_2 - t_1) - 0.5 I + \frac{\rho}{\sigma}(V(t_2) - V(t_1)) - \kappa(t_2 - t_1) + \kappa I, (1 - \rho^2) I \right].
\]

It follows immediately that simulating \((X(t_2), V(t_2))\) given \((X(t_1), V(t_1))\) reduces to sampling from the joint distribution of the pair \((V(t_2)|V(t_1), I)\). Moreover,
as \((V(t_2)|V(t_1))\) can be sampled using (3), the problem reduces further to that of sampling

\[ I_c = \left( \int_{t_1}^{t_2} V(s) \, ds \right) |V(t_1), V(t_2)|. \tag{5} \]

### 2.2. The exact scheme

In the exact scheme (Broadie and Kaya 2006), both \((V(t_2)|V(t_1))\) and \(I_c\) are sampled exactly. As discussed in Broadie and Kaya (2006) and Glasserman and Kim (2011), exact sampling of \((V(t_2)|V(t_1))\) is relatively straightforward and inexpensive. Exact sampling of \(I_c\), on the other hand, is very complicated and time-consuming (Smith 2007, Van Haastrecht and Pelsser 2008), as we will explain in detail in the following. For details of the exact scheme please refer to Broadie and Kaya (2006).

#### 2.2.1. Sampling \(I_c\)

The essence of the exact scheme is to use Fourier inversion to obtain the CDF from the characteristic function, and then numerically invert the CDF to obtain the quantile function. The computation is very expensive since the characteristic function\(^\dagger\) of \(I_c\),

\[ \varphi(a; V(t_1), V(t_2)) = \mathbb{E}[e^{iaV}], \]

involves the modified Bessel function of the first kind, which is costly to compute. The order of \(I_c(z)\) is denoted by \(v = \delta/2 - 1\) and the argument \(z\) is an expression which involves \(a\), \(V(t_1)\) and \(V(t_2)\). Let \(F_c(x) = \text{Prob}(I_c \leq x)\) be the CDF. The exact scheme computes the CDF by numerically integrating

\[ F_c(x_0) = \frac{2}{\pi} \int_0^\infty \frac{\sin(x_0 u)}{u} \text{Re}[\varphi(u; V(t_1), V(t_2))] \, du, \tag{6} \]

which typically requires many evaluations of \(\varphi\) and hence \(I_c(z)\). Furthermore, when inverting the CDF numerically, one would need to solve the equation \(F_c(x_U) = U\) for \(x_U\), where \(U\) is a uniform variate. A root-finding algorithm would in turn require evaluating \(F_c\) multiple times until convergence.

Implementation of the exact scheme is not straightforward for two reasons. Accurate numerical integration of (6) hinges on the non-trivial issue of choosing a fine enough grid size and wide enough bounded domain for integration (Broadie and Kaya 2006). Moreover, calculation of \(I_c(z)\) requires care in branch counting the complex number argument \(z\).

Numerical results in Broadie and Kaya (2006) show that the cost of generating one asset price sample using the exact method is roughly equal to that of generating 1600 samples using the Euler scheme. Numerical results in Lord et al. (2010) have shown that the speed--accuracy tradeoff of the exact scheme is less favourable than that of many simpler schemes, particularly for path-dependent options.

\(^\dagger\)We do not show the full expression for the characteristic function here because it is long and complicated; see Broadie and Kaya (2006).

One approach to speed up the exact scheme is by precomputation of \(\varphi\). However, the three-dimensional dependence of \(\varphi(a; V(t_1), V(t_2))\) on \(a\), \(V(t_1)\) and \(V(t_2)\) makes direct precomputation and interpolation of it impractical (Smith 2007), as \(V(t_1)\) and \(V(t_2)\) change at each time step in each simulated path. By using an approximation to \(\varphi\), which has only two-dimensional dependence on \(a\), \(V(t_1)\) and \(V(t_2)\), the almost exact scheme (Smith 2007) is able to speed up the exact scheme by around seven times, while largely maintaining the very low bias of the exact scheme. Despite the speed up, the almost exact scheme is still expensive to the extent that its speed--accuracy tradeoff is also less favourable than many simpler schemes (Van Haastrecht and Pelsser 2008).

#### 2.3. The quadratic exponential scheme

In QE (Andersen 2007), \((V(t_2)|V(t_1))\) is approximated accurately and \(I_c\) is approximated very roughly.

Although we mentioned that the exact sampling of \((V(t_2)|V(t_1))\) is relatively inexpensive compared with that of \(I_c\), the time needed to generate a non-central chi-square variate is still more than ten times of that needed to generate a uniform or Gaussian variate. In QE, two approximations (quadratic and exponential) to the non-central chi-square distribution are proposed to speed up the sampling. Loosely speaking, a quadratic approximation is used when \(V(t_1)\) is big and an exponential approximation is used when \(V(t_1)\) is small. In both the quadratic approximation and the exponential approximation, the first two moments of the approximating distribution are matched to that of the exact distribution. The simple QE approximation to \((V(t_2)|V(t_1))\) turns out to be surprisingly accurate. By plotting the exact non-central chi-square distribution against the QE approximations, one would see that the approximations closely resemble the exact distribution for typical parameter values and variance values encountered in Heston model simulations.

In contrast, QE’s approximation to \(I_c\) is very rough. Although the distribution of \(I_c\) is much more complicated than that of \((V(t_2)|V(t_1))\), \(I_c\) is approximated by a constant random variable taking the value 0.5\((V(t_1) + V(t_2))\). This approximation is equivalent to applying the trapezoidal rule to integrate \(I_c\) numerically, an approximation referred to as Drift Interpolation (DI) in the literature.

In terms of speed--accuracy tradeoff, numerical comparisons in the literature (Andersen 2007, Glasserman and Kim 2011, Van Haastrecht and Pelsser 2008) have shown that QE is one of the best methods when a relatively large number of time steps is used. We note that one main advantage of QE is its speed. More specifically, sampling from the quadratic/exponential approximation and moment-matching can be performed quickly. For details please refer to Andersen (2007).

A drawback of QE is its relatively big bias when the number of time steps is medium or small; see section 5. This is mainly due to a significant deterioration in
accuracy of the rough approximation to \( I_c \) when \( t_2 - t_1 \) is larger. To address this issue, Andersen (2007) suggests using moment-matching to determine the weights in numerical integration. However, we are not aware of any numerical result on moment-matched DI in the literature.

2.4. The gamma expansion scheme

In GE (Glasserman and Kim 2011), \( (V(t_2)|V(t_1)) \) is sampled exactly and \( I_c \) is approximated accurately.

As discussed in section 2.2.1, exact sampling of \( I_c \) is very expensive. To tackle this, the authors derive an exact and explicit expression of \( I_c \) in terms of an infinite sum of mixtures of gamma random variables. Using the expansion, a fast and accurate numerical scheme is developed by truncating the infinite series and approximating the remainder terms. Numerical results in the paper demonstrate that GE has very low bias.

For later reference, we quote the following results from Glasserman and Kim (2011).

Proposition 2.1: Let \( \delta = 4\kappa \theta / \sigma^2 \), \( \nu = \delta / 2 - 1 \), \( \triangle t = t_2 - t_1 \), and \( C_z = 2\kappa / \sigma^2 \sinh(\kappa \triangle t / 2) \). The random variable \( I_c \) representing the time-integrated variance admits the representation

\[
I_c = \left( \int_{t_1}^{t_2} V(s)^2 ds \bigg| V(t_1), V(t_2) \right) = X_1 + X_2 + \sum_{j=1}^{\eta} Z_j, \tag{7}
\]

in which \( X_1, X_2, \eta, Z_1, Z_2, \ldots, Z_\eta \) are mutually independent, the \( Z_j \) are independent copies of a random variable \( Z \). \( \eta \) is a Bessel random variable with parameters \( \nu \) and \( z = C_z V(t_1) V(t_2) \). The Laplace transforms \( \Phi^1, \Phi^2, \Phi^c \) of \( X_1, X_2 \) and \( Z \) are, for \( b \geq 0 \),

\[
\Phi^1(b) = \exp \left( \frac{V(t_1)^2 + V(t_2)^2}{\sigma^2} \left[ \kappa \coth \frac{\kappa \triangle t}{2} - L \coth \frac{L \triangle t}{2} \right] \right), \tag{8}
\]

\[
\Phi^2(b) = \left( \frac{L \sinh \frac{L \triangle t}{2}}{\kappa \sinh \frac{\kappa \triangle t}{2}} \right)^{b/2}, \tag{9}
\]

\[
\Phi^c(b) = \left( \frac{L \sinh \frac{L \triangle t}{2}}{\kappa \sinh \frac{\kappa \triangle t}{2}} \right)^2, \tag{10}
\]

where \( L = \sqrt{2\sigma^2 b + \kappa^2} \).

2.5. Comparing QE and GE

Overall, GE is more accurate but slower than QE. Numerical results in Glasserman and Kim (2011) compare the speed-accuracy tradeoff of GE and QE on pricing vanilla European options with a relatively short maturity of one year. In one comparison, the bias of GE using one time step is approximately equal to that of QE using 32 time steps, and GE outperforms by a factor of two to three in terms of speed. However, if one prices a path-independent option which requires all 32 values, then ‘generating all 32 values would take approximately 12 times as long using the gamma expansion’ (Glasserman and Kim 2011).

The performance advantage of GE is thus significantly diminished by its significantly higher cost per time step when pricing path-dependent options. Our numerical results indicate that the accuracy of QE using 48 time steps is likely to be considered as accurate enough in practice. Consequently, QE would have better speed-accuracy tradeoff than GE on pricing path-dependent options which require more than 48/12 = 4 time steps.

We note that there is a middle ground between QE and GE. Because of its low cost per time step, QE’s strength is most evident when at least dozens of time steps are required. GE, on the other hand, is accurate and fast enough when very few time steps (fewer than four) are required. When a path-dependent option requires a moderate number of time steps (from four to around sixteen), GE is not cost effective, and QE may not be accurate enough unless many more time steps than required by the path-dependence are used.

3. The Inverse Gaussian approximation

As shown by the GE approach, an accurate approximation to \( I_c \) is key to a low-bias simulation scheme. A very accurate approximation (e.g. GE), however, may be expensive. Our idea is to approximate the sampling of \( I_c \) by a faster, simpler and still very accurate scheme.

We propose to approximate \( I_c \) by the Inverse Gaussian (IG) distribution (Chhikara and Folks 1989). The IG distribution is a family of distributions parameterized by mean parameter \( m \) and shape parameter \( s \), which are determined by moment matching in our approximation.

We will derive explicit formulae for the first two moments of \( I_c \) and prove that the exact distribution converges to the moment-matched IG distribution in a certain sense as the time interval \( t_2 - t_1 \) goes to infinity. We will also illustrate the accuracy of the IG approximation for finite time intervals.

We note that the more general Normal Inverse Gaussian (NIG) distribution, of which the Inverse Gaussian distribution is its mixing density, has been widely used in stochastic volatility modelling; see for example Barndorff-Nielsen et al. (2002). The NIG distribution, however, is too complicated for approximating \( I_c \).

3.1. The Inverse Gaussian distribution

Let IG\((m, s)\) denote the Inverse Gaussian distribution with mean parameter \( m > 0 \) and shape parameter \( s > 0 \). It has support on \((0, \infty)\) and its probability density function \( f(x; m, s) \) and logarithm of Laplace transform \( \log \Phi^{IG(m,s)} \) are given by

\[
f(x; m, s) = \sqrt{\frac{s}{2\pi x^3}} \exp\left(-\frac{(x - m)^2}{2m^2x}\right), \tag{11}
\]
we have that for any positive \( \phi \) and \( \theta\):

\[
\log \Phi^{IG(m, s)}(b) = \frac{s}{m} \left( 1 - \sqrt{1 + 2m^2b/s} \right).
\] (12)

The mean and variance of IG\((m, s)\) are \( m \) and \( m^3/s \), respectively. The ratio \( \phi = s/m \) determines the shape of the distribution. The density is highly skewed for moderate values of \( \phi \). As \( \phi \) tends to infinity, the Inverse Gaussian distribution tends to the normal distribution (Chhikara and Folks 1989). We remark that the Inverse Gaussian distribution is so called because its cumulant generating function (the logarithm of the characteristic function) is the inverse of the cumulant generating function of a Gaussian distribution. More details about the Inverse Gaussian distribution can be found in Folks and Chhikara (1978) and Chhikara and Folks (1989). Figure 1 shows two plots of the probability density functions of the Inverse Gaussian distribution when either the mean or the shape parameter is fixed at unity.

### 3.2. Moments of \( I_c \)

One might be tempted to calculate the first two moments of \( I_c \) by directly differentiating the characteristic function. This brute force approach is not only tedious but also inefficient since the derivatives of the characteristic function may result in large errors. Our approach, based on the gamma expansion (Glasserman and Kim 2011), is a much simpler means of calculating the moments.

**Proposition 3.1:** Let \( C_1 = \coth(\kappa \triangle t/2) \) and \( C_2 = \csch^2(\kappa \triangle t/2) \). The mean \( E[I_c] \) and the variance \( \text{Var}[I_c] \) of \( I_c \) are given by

\[
E[I_c] = E[X_1] + E[X_2] + E[\eta]E[Z],
\] (13)

\[
\text{Var}[I_c] = \sigma_{X_1}^2 + \sigma_{X_2}^2 + E[\eta] \sigma_{Z}^2 + (E[\eta]^2 - E[\eta]^2)E[Z]^2,
\] (14)

where

\[
E[X_1] = (V(t_1) + V(t_2))(C_1/\kappa - \triangle t C_2/2),
\]

\[
\sigma_{X_1}^2 = (V(t_1) + V(t_2))(\sigma_{X_1}^2/\kappa^3 + \sigma_{X_2}^2 \triangle t C_2/(2\kappa^3)) - \sigma_{X_1}^2 \triangle t C_2/C(2\kappa),
\]

\[
E[X_2] = \delta \sigma^2 (-2 + \kappa \triangle t C_1)/(4\kappa^2),
\]

\[
\sigma_{X_2}^2 = \delta \sigma^4 (8 + 2 \kappa \triangle t C_1 + \kappa^2 (\triangle t)^2 C_2)/(8 \kappa^4),
\]

\[
E[Z] = 4E[X_2]/\delta, 
\]

\[
\sigma_{Z}^2 = 4 \delta \sigma_{X_2}/\delta, 
\]

\[
E[\eta] = z I_{\eta-1}(z)/(2I_\eta(z)), 
\]

\[
E[\eta^2] = z^2 I_{\eta+2}(z)/(4I_{\eta+1}(z)) + E[\eta].
\]

**Proof:** The mean and variance of \( X_1, X_2 \) and \( Z \) can be calculated using the Laplace transforms in proposition 2.1. The moments of \( \eta \) can be calculated using the formulae in Yuan and Kalbfleisch (2000). The result then follows from the representation (7) and the mutual independence of \( X_1, X_2, \eta, Z_1, \ldots, Z_\eta \).

### 3.3. Convergence of \( I_c \) to the Inverse Gaussian distribution

**Lemma 3.2:** As \( \triangle t \to \infty \), we have that for any positive integer \( n \):

\[
E \left[ \sum_{j=1}^{n} Z_j \right] = E[\eta]E[Z] = \mathcal{O}(\triangle t^{-n}),
\] (15)

\[
\text{Var} \left[ \sum_{j=1}^{n} Z_j \right] = \mathcal{O}(\eta^2)E[Z]^2 = \mathcal{O}(\triangle t^{-n}),
\] (16)

\[
\frac{E[I_c]}{\triangle t} \to \frac{\delta \sigma^2}{4\kappa} = \theta,
\] (17)

\[
\frac{\text{Var}[I_c]}{\triangle t} \to \frac{\delta \sigma^4}{4\kappa^3} = \frac{\theta \sigma^2}{\kappa^2}.
\] (18)

Figure 1. Probability density functions of the Inverse Gaussian distribution with (a) mean parameter fixed at unity and varying shape parameter, (b) shape parameter fixed at unity and varying mean parameter.
First of all, note that as \( \triangle t \to \infty \), \( C_1 = \text{coth}(\kappa \triangle t/2) \to 1 \), \( C_2 = \text{csch}^2(\kappa \triangle t/2) = o(\triangle t^{-2}) \) and \( z = C_z = \sqrt{V(t_1)V(t_2)} = 2\sqrt{V(t_1)V(t_2)k} \left[ \sigma^2 \sinh(\kappa \triangle t/2) \right]^{-1} = o(\triangle t^{-3}) \).

From the above, \( z \to 0 \) as \( \triangle t \to \infty \). Therefore, \( \lim_{\triangle t \to \infty} I_{z+1}(z)/I_z(z) = \lim_{z \to 0} I_{z+1}(z)/I_z(z) \), which is equal to zero by the results in Yuan and Kalbfleisch (2000). Similarly, \( \lim_{\triangle t \to \infty} I_{z+2}(z)/I_z(z) = 0 \). Hence, \( E[\eta] = E[\eta^2] = o((\triangle t)^{-2}) \) as \( z = o((\triangle t)^{-1}) \).

The lemma follows by applying the asymptotic results on \( C_1, C_2, z, E[\eta] \) and \( E[\eta^2] \) to (13) and (14).

The intuitive meanings of (17) and (18) are interesting in themselves. The former says that the first moment of \( I_c \) normalized by the time interval tends to \( \theta \) (the mean-reversion level); the latter says that, in the limit, the variance of \( I_c \) normalized by the time interval is proportional to \( \theta \) (the mean-reversion level) and \( \sigma^2 \) (the variance of the of the underlying asset’s instantaneous returns), and inversely proportional to \( \kappa^2 \) (the square of the mean-reversion speed). Another observation is that \( V(t_1) \) and \( V(t_2) \) become irrelevant in the limit.

**Proposition 3.3:** Let \( \Phi^C \) be the Laplace transform of \( I_c \). We have

\[
\lim_{\triangle t \to \infty} \frac{\log \Phi^C(b)}{\triangle t} = \frac{\delta(k - L)}{4}.
\]

**Proof:** Let \( Z' = \sum_{j=1}^n Z_j \) in (7). By proposition 2.1 and the mutual independence of \( X_1, X_2, \eta, Z_1, \ldots, Z_n \), \( \log \Phi^C = \log \Phi^1 + \log \Phi^\eta + \log \Phi^{Z'} \). Using (15) and (16), we have \( \lim_{\triangle t \to \infty} \log \Phi^{Z'}/\triangle t = 0 \). We also have \( \lim_{\triangle t \to \infty} \log \Phi^1/\triangle t = 0 \) by (8) and \( \lim_{\triangle t \to \infty} \text{coth}(\triangle t) = 1 \). To prove \( \lim_{\triangle t \to \infty} \log \Phi^2/\triangle t = \delta(k - L)/4 \), use (9) and observe that

\[
\lim_{\triangle t \to \infty} \left( \frac{\text{coth}(z\triangle t)}{\triangle t} \right) = \kappa - \ln 2.
\]

We are now ready to show that the exact distribution converges to the moment-matched IG distribution.

**Theorem 3.4:** Let \( \log \Phi^{IG(m,s,L)} \) be the logarithm of the Laplace transform of the moment-matched IG distribution. As \( \triangle t \to \infty \), \( I_c \) tends to the IG distribution in the sense that, for any fixed \( b > 0 \),

\[
\lim_{\triangle t \to \infty} \frac{\log \Phi^{IG(m,s,L)}(b)}{\triangle t} = 1.
\]

**Proof:** In the moment-matched IG distribution, \( m = E[I_c] \) and \( s = \sqrt{E[I_c]^2/\text{Var}[I_c]} \) (see section 3.2). By using the formula for \( \log \Phi^{IG(m,s,L)}(b) \), i.e. equation (12),

\[
\lim_{\triangle t \to \infty} \frac{\log \Phi^{IG(m,s,L)}(b)}{\triangle t} = \lim_{\triangle t \to \infty} \frac{1}{\triangle t} \left( 1 - \sqrt{1 + 2\gamma \triangle t/s^2} \right) = \lim_{\triangle t \to \infty} \frac{E[I_c]}{\triangle t \text{Var}[I_c]} \times \left( 1 - \sqrt{1 + 2\gamma \frac{\text{Var}[I_c]}{E[I_c]}} \right).
\]

Using the asymptotic results (17) and (18) in Lemma 3.2, the above is equal to

\[
\frac{\delta \sigma^2 \kappa^2}{4\kappa \sigma^2} \left( 1 - \sqrt{1 + 2\gamma \frac{\text{Var}[I_c]}{E[I_c]}} \right) = \frac{\delta(k - L)}{4}.
\]

The result now follows from proposition 3.3.

**3.4. Convergence of \( I_c \) in the small time limit**

Proposition 3.1 also allows us to determine the convergence behaviour of \( I_c \) as \( \triangle t \to 0^+ \).

**Proposition 3.5:** Given \( V(t_1) \) and \( V(t_2) \), we have

\[
\lim_{\triangle t \to 0^+} E\left[ \frac{I_c}{\triangle t} \right] = \frac{V(t_1) + V(t_2) + \sqrt{V(t_1)V(t_2)}}{3},
\]

\[
\lim_{\triangle t \to 0^+} \text{Var}\left[ \frac{I_c}{\triangle t} \right] = 0.
\]

**Proof:** We have \( z \to +\infty \) as \( \triangle t \to 0^+ \). Therefore, using the results in Yuan and Kalbfleisch (2000) gives \( \lim_{\triangle t \to 0^+} I_{z+1}(z)/I_z(z) = \lim_{z \to +\infty} I_{z+1}(z)/I_z(z) = 1 \) and \( \lim_{\triangle t \to 0^+} I_{z+2}(z)/I_z(z) = 1 \). The small time limit result then follows by direct calculation using the formula in proposition 3.1.

The implication of the above proposition is the intuitively obvious observation that \( I_c \) will converge to a constant as \( \triangle t \to 0^+ \). Consequently, any moment matching scheme of \( I_c \), e.g. the Inverse Gaussian approximation, will also converge to the correct constant in the small time limit.

**3.5. Accuracy of the IG approximation**

**3.5.1. Comparison of probability density functions**. Figure 2 shows that \( I_c \) and the Inverse Gaussian distribution resemble each other very well.† Four notable similarities are: strictly positive support, an acute peak, right-skewness, and a long and fat right tail. We remark that the Log-normal distribution and the Gamma distribution also share these features. Their accuracies in approximating \( I_c \) however, are much worse than that of the Inverse Gaussian distribution, as found in our numerical tests (not shown in this paper). The much better accuracy of the IG approximation is not surprising given our convergence result.

Figure 3 compares graphically the probability density functions of the IG approximation with the exact distribution using the parametric cases listed in table 1 in section 5. For plotting purposes, we set \( V(t_1) = V(t_2) = V(0) \) and \( \triangle t = T \). From figure 3, we see that the IG approximation not only matches the exact distribution very well in the overall shape, but also at the tails.

†Here we plot the probability density function of \( I_c \) to high precision by using the results in Broadie and Kaya (2006).
Figure 2. Probability density functions of $I_c$ with (a) $\Delta t = 5$, $V(t_1) = V(t_2) = 0.09$, $\kappa = 0.2$, $\theta = 0.09$, $\sigma = 1$ (b) $\Delta t = 1$, $V(t_1) = 0.06 = V(t_2) = 0.06$, $\kappa = 0.21$, $\theta = 0.019$, $\sigma = 0.61$.

Figure 3. Probability density functions of $I_c$ and the IG approximation.

Table 1. Heston model parameter cases.

<table>
<thead>
<tr>
<th>Case</th>
<th>$\sigma$</th>
<th>$\kappa$</th>
<th>$\theta$</th>
<th>$V(0)$</th>
<th>$\rho$</th>
<th>$r$</th>
<th>$T$</th>
<th>$X(0)$</th>
<th>$K$</th>
<th>Exact price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>0.04</td>
<td>0.04</td>
<td>$-0.9$</td>
<td>0</td>
<td>10</td>
<td>100</td>
<td>100</td>
<td>13.08467014</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.5</td>
<td>0.04</td>
<td>0.04</td>
<td>$-0.9$</td>
<td>0</td>
<td>10</td>
<td>100</td>
<td>140</td>
<td>0.29577444</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0.5</td>
<td>0.04</td>
<td>0.04</td>
<td>$-0.9$</td>
<td>0</td>
<td>10</td>
<td>100</td>
<td>70</td>
<td>35.84976970</td>
</tr>
<tr>
<td>4</td>
<td>0.61</td>
<td>6.21</td>
<td>0.019</td>
<td>0.010201</td>
<td>$-0.7$</td>
<td>0.0319</td>
<td>1</td>
<td>100</td>
<td>100</td>
<td>6.80611331</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>2</td>
<td>0.09</td>
<td>0.09</td>
<td>$-0.3$</td>
<td>0.05</td>
<td>5</td>
<td>100</td>
<td>100</td>
<td>34.99975835</td>
</tr>
<tr>
<td>6</td>
<td>0.5196</td>
<td>1.0407</td>
<td>0.0886</td>
<td>0.0194</td>
<td>$-0.6747$</td>
<td>0</td>
<td>4</td>
<td>100</td>
<td>100</td>
<td>15.16790670</td>
</tr>
</tbody>
</table>
3.5.2. Comparison of skewness and excess kurtosis. To find out how large $\Delta t$ has to be for the IG approximation of $I_c$ to be accurate, we compare the skewness and the excess kurtosis of the two distributions. The formula for calculating skewness and kurtosis comes from using the Laplace transforms (8), (9) and (10) to calculate the moments of $X_1, X_2$ and $Z$, and using the results in Broadie and Kaya (2006) to calculate the moments of $\eta$.

Note that the time scale of $I_c$ is determined mainly by the mean-reversion rate $\kappa$. All our plots in figures 4 and 5 are therefore scaled by the dimensionless constant $\kappa \Delta t$. More specifically, we compare the accuracy of the IG approximation for values of $\Delta t$ ranging from 0 to 20/$\kappa$, while holding other parameters constant.

The key observation is that the IG approximation becomes accurate when $\kappa \Delta t \geq 6$. In figure 4, we can see that the IG curve becomes close to the exact curve when $\Delta t \geq 6/\kappa$. In figure 5, we see that the percentage error in skewness becomes less than 5% and the percentage error in excess kurtosis becomes less than 10%. More importantly, these observations are robust with respect to large changes to other parameters ($\sigma, \theta, V(t_1)$ and $V(t_2)$), i.e. the plots in figures 4 and 5 remain largely the same as long as we keep the scale of $\kappa \Delta t$ constant, no matter how we change the other parameters!

4. Implementation

In this section, we will propose a scheme for fast sampling of $(V(t_2)|V(t_1))$, the IPZ scheme, and develop the IG scheme for fast sampling of the moment-matched IG approximation to $I_c$. The combined IPZ-IG scheme is the scheme we propose for simulating the Heston model.

4.1. Fast sampling of $(V(t_2)|V(t_1))$ – the IPZ scheme

Note that (3) can be rewritten ((Van Haastrecht and Pelsser 2008) as

$$V(t_2) = \frac{2e^{-s(t_2-t_1)}}{m(t_1, t_2)} \text{Gamma} \left( \frac{V(t_1)m(t_1, t_2)}{2} + \delta/2 \right),$$

where $\text{Gamma}(s)$ is a unit-scale gamma variate with shape parameter $s$ and $\text{Poisson}(m_p)$ is a Poisson variate with mean parameter $m_p$. We note that generating $\text{Gamma}(s)$ for $s < 1$ is significantly more costly than for $s \geq 1$. For typical Heston parameters, $\delta/2 < 1$, and hence the case $s < 1$ occurs if and only if $\text{Poisson}(V(t_1)m(t_1, t_2)/2) = 0$ since $\text{Poisson}(m_p)$ takes on integral values. Noting that the case $\text{Poisson}(V(t_1)m(t_1, t_2)/2) = 0$ can occur very often for realistic Heston parameters, it is particularly advantageous to use precomputation and interpolation to speed up the computation of the single special case $s = \delta/2 < 1$, i.e. $\text{Gamma}(\delta/2)$. Since $\text{Gamma}(\delta/2)$ and $V(t_2)$ differ by only a constant, in the following we will discuss primarily the sampling of $V(t_2)$.

We replace the direct sampling of $(V(t_2)|V(t_1))$ by nearest-neighbour interpolation of its quantile function, $Q$, as follows.

**Algorithm 1:**

- Define an equally spaced grid $\tilde{u} = \{0, \ldots, 1\}$ with $N_u$ nodes.

![Figure 4. Skewness and excess kurtosis of $I_c$ and the IG approximation with $V(t_1) = V(t_2) = 0.0194$, $\theta = 0.0586$, $\sigma = 0.5196$ and $\Delta t$ ranging from 0 to 20/$\kappa$.](image)
Precompute a vector \( \tilde{q} = (q_i) \) whose components \( q_i \) are quantile function values on \( u_i \), i.e. \( q_i = Q(u_i) \).

To sample \( (V(t_2)|V(t_1)) \), do the following.

- Sample \( m_p \sim \text{Poisson}(V(t_1)m(t_1,t_2)/2) \). If \( V(t_1) = 0 \), simply set \( m_p = 0 \) since \( \text{Poisson}(0) \) is always zero.
- If \( m_p = 0 \), sample as below. Otherwise, sample from \( \Gamma(m_p + \delta/2) \), i.e. use (21).
- Draw a uniform variate \( U \).
- Find the index \( i \) such that \( u_i \) is closest to \( U \).
- Use \( q_i \) as the sample.

Since \( \tilde{u} \) is an equally-spaced grid, nearest interpolation has a constant cost independent of the number of grid points \( N_u \). In the above algorithm, we did not explicitly mention how the quantile function \( Q \), which is unavailable in closed-form, is computed. Numerically inverting the CDF is one possibility, but not an efficient approach. We approximate \( Q \) using nearest-neighbour interpolation as follows.

**Algorithm 2:**

- Define another equally spaced grid \( \tilde{v} = \{v_{\text{min}}, \ldots, v_{\text{max}}\} \) with the same number of nodes as \( \tilde{u} \), i.e. \( N_v \) nodes.
- Compute a vector \( \tilde{p} = (p_i) \) whose components \( p_i \) are cumulative probabilities on \( v_i \), i.e. \( p_i = \Pr\{V(t_2) \leq v_i\} \).
  - To approximate \( q_i \), do the following.
  - For each index \( i \), if \( u_i < p_0 \), we set \( q_i = 0 \).
  - Otherwise, use a binary search to look for the index \( j \) such that \( p_j \) is closest to \( u_i \), and set \( q_i = v_j \).

Our approach of approximating very small quantile function values \( q_i \) by zero is similar in spirit to what is done in the exponential approximation in QE. Note also that, in our algorithm, if an index \( i \) such that \( q_i = 0 \) is selected, \( V(t_2) \) would be set to zero. This has the benefit of allowing us to skip the Poisson variate generation in the next time step, as described in Algorithm 1.

We do not use precomputation and interpolation for the case \( \text{Poisson}(V(t_1)m(t_1,t_2)/2) > 0 \). This is because the gamma variate simulation is much faster when the shape parameter is greater than one (Marsaglia and Tsang 2000b). Indeed, our experience (not shown in this paper) indicates that using precomputation and interpolation for the case \( \text{Poisson}(V(t_1)m(t_1,t_2)/2) > 0 \) does not yield significant speedup. We denote our scheme IPZ since we are Interpolating for the case when the Poisson variate is equal to Zero.

**4.2. Sampling \( I_c \) by the Inverse Gaussian approximation**

As discussed in section 3, we approximate \( I_c \) by the moment-matched IG distribution. IG variates can be generated efficiently by an acceptance-rejection type algorithm (Michael et al. 1976). A recent survey about generating IG variates can be found in Lai (2009).

**Algorithm 3:** To generate a random variate from \( \text{IG}(m,s) \), do the following.

- Generate a standard normal variate \( N \) and a uniform variate \( U \).
Compute $x = 1 + N^2/(2s/m) - \sqrt{(2s/m + 2s/m)^2 + (N^2)^2}/(2s/m)$.

- If $U(1 + x) > 1$, return $m/x$. Otherwise, return $mx$.

This algorithm is usually presented in the literature in a way slightly different from ours. Observing that $IG(m, s) = mIG(1, s/m)$, we generate $IG(m, s)$ by first generating $IG(1, s/m)$ and then multiplying $m$ back, which is slightly faster than generating $IG(m, s)$ directly.

### 4.3. Fast moment calculation – the IG scheme

If $E[L]$ and $\text{Var}[L]$ are calculated by direct evaluations of the formulae in proposition 3.1, the modified Bessel functions would have to be evaluated many times, which is very expensive. Here we design an interpolation scheme to perform fast and accurate moment calculations.

For notational convenience, let us define

$$E[L|_{V(t_1)|V(t_2)}] = E[X_2] + E[\eta|E[Z]],$$
$$\text{Var}[L|_{V(t_1)|V(t_2)}] = \sigma_X^2 + E[\eta^2|\sigma_Z^2 + (E[\eta^2] - E[\eta]^2)]E[Z]^2,$$

which depend only on the product $V(t_1)V(t_2)$ (for fixed Heston parameters and $\lambda$). Under this notation, $E[L] = E[X_1] + E[L|_{V(t_1)|V(t_2)}]$ and $\text{Var}[L] = \sigma_X^2 + \text{Var}[L|_{V(t_1)|V(t_2)}]$, where $E[X_1]$ and $\sigma_X^2$ depend only on $(V(t_1) + V(t_2))$. The rationale behind this notation is that the argument $z$ to the modified Bessel functions depends only on $V(t_1)V(t_2)$.

For fast moment calculation, do the following.

**Algorithm 4:**

- Precompute $E[L|_{V(t_1)|V(t_2)}]$ and $\text{Var}[L|_{V(t_1)|V(t_2)}]$ on an equally spaced grid $\sqrt{V(t_1)V(t_2)} = \tilde{v}$, where $\tilde{v}$ is as defined in algorithm 2.
  
  During simulation, compute $E[L]$ and $\text{Var}[L]$ as follows.
  
  - Compute $E[X_1]$ and $\sigma_X^2$.
  - If $V(t_1) = 0$ or $V(t_2) = 0$, $E[\eta] = E[\eta^2] = 0$, and hence $E[L] = E[X_1] + E[X_2]$ and $\text{Var}[L] = \sigma_X^2 + \sigma_X^2$, i.e. only two additions are required in this step as $E[X_2]$ and $\sigma_X^2$ are constants.
  
  - Otherwise, use nearest neighbour interpolation to approximate $E[L|_{V(t_1)|V(t_2)}]$ and $\text{Var}[L|_{V(t_1)|V(t_2)}]$. Add them to $E[X_1]$ and $\sigma_X^2$ to obtain $E[L]$ and $\text{Var}[L]$, respectively.

The reason to interpolate on $\sqrt{V(t_1)V(t_2)}$ is that when $E[L|_{V(t_1)|V(t_2)}]$ and $\text{Var}[L|_{V(t_1)|V(t_2)}]$ are regarded as functions of $\sqrt{V(t_1)V(t_2)}$, their graphs appear very similar to piecewise linear curves on a log–log scale. Again, since $\tilde{v}$ is an equally spaced grid, nearest interpolation has a constant cost independent of the number of grid points.

We note that the approximations to $E[L|_{V(t_1)|V(t_2)}]$ and $\text{Var}[L|_{V(t_1)|V(t_2)}]$ need not be very accurate since the terms $E[X_1]$ and $\sigma_X^2$ often have much bigger magnitudes. In light of the expensive cost in computing the modified Bessel functions, we do full precomputation only at one-fourth of the nodes. Values at other nodes are calculated by linear interpolation. This approximation gives negligible errors in our numerical tests.

We note that the precomputation for sampling $(V(t_2)|V(t_1))$ is less expensive and therefore we do not use interpolation there.

### 4.4. Summary of the IPZ–IG scheme

Before simulation, precompute $Q$, $E[L|_{V(t_1)|V(t_2)}]$ and $\text{Var}[L|_{V(t_1)|V(t_2)}]$ as specified in algorithm 2 and 3. During simulation at time $t_2$, when $(X(t_1), V(t_1))$ is known, the IPZ–IG scheme samples $(X(t_2), V(t_2))$ as follows.

- Sample $V(t_2)$ by algorithm 1.
- Calculate $E[L]$ and $\text{Var}[L]$ using algorithm 4.
- Sample $L_c$ using the moment-matched IG distribution by algorithm 3.
- Conditional on $V(t_1)$, $V(t_2)$ and $L_c$, sample $X(t_2)$ using equation (4).

Note that this last step is actually irrelevant to the IPZ–IG scheme, but nevertheless necessary for getting a sample of $X(t_2)$.

We remark that the implementation of the IPZ–IG scheme may seem more complicated than it actually is. Most of the complexity in fact comes from using precomputation for better efficiency. Without precomputation, the sampling of $(V(t_2)|V(t_1))$ is as simple as equation (21), and the sampling of $L_c$ involves only direct calculation of $E[L]$ and $\text{Var}[L]$ in equations (13) and (14), and implementing algorithm 3 to generate an Inverse Gaussian variate. In this perspective, the implementation difficulty of the IPZ–IG scheme (without precomputation techniques) is therefore similar to that of the QE scheme. Interested readers may implement the IPZ–IG scheme without the precomputation techniques at first, and then proceed to full-efficiency implementation.

### 5. Numerical results

In our numerical tests, we compare the IPZ–IG scheme and the QE scheme. To emphasize that the QE scheme comprises the quadratic/exponential approximation and the drift–interpolation approximation, we denote QE by QE–DI in this section.

Although we do not compare the IPZ–IG scheme with the GE scheme directly, indirect comparisons can be made based on the efficiency comparison between QE–DI and GE. As discussed in section 2.5, GE has no practical advantage over QE–DI except when very few time steps (one to three) are required. Since our numerical results will show that IPZ–IG has better speed–accuracy trade off than QE on pricing path-dependent options which require a moderate number of time steps (four to around sixteen), IPZ–IG would also outperform GE in such cases. Our numerical results will also show that IPZ–IG has better speed–accuracy trade off than QE on pricing path-dependent options which require very few time steps.
(one to three). The speed–accuracy trade off of IPZ–IG, however, will be less favourable than GE in such cases.

In the first part of our numerical results, we compare the schemes on pricing European call options. Comparison using European call options is convenient and standard practice in the literature since semi-analytical prices are available (Heston 1993). We emphasize here that such comparisons based on European calls are intended to be regarded as approximations to comparisons based on genuinely path-dependent options. In our comparison of speed–accuracy trade off, we follow the literature and pretend that a European call has path-dependency that requires time stepping. (Otherwise, the semi-analytical method in Heston (1993) always wins.)

In the second part of our numerical results, we compare the schemes on pricing Asian call options. In the third part, we analyse in detail the relative computational time in approximated sampling of $\{V(t_j) V(t_i)\}$ and $L_i$.

Our timing results are obtained by running our C program (compiled using Visual C++ express 2008) on an HP laptop with a 2.0GHz Intel Core 2 Quad Q9000 processor and 4GB RAM. For fair comparisons, all expressions in the two schemes that depend only on the Heston parameters and thus do not change across simulations are computed only once at the initialization of the simulations.

We compare the performance of the schemes using an extensive collection of realistic and challenging parametric cases in the literature, as shown in table 1. Cases 1 to 3 are from Andersen (2007), cases 4 and 5 are from Broadie and Kaya (2006), and case 6 is from Smith (2007). Cases 1 to 3 have a big $\sigma = 1$, a big $T = 10$, a very negative $\rho = -0.9$, and cover at-the-money, out-of-the-money and in-the-money options. Case 4 has a relatively short maturity $T = 1$ and large $\kappa = 6.21$. Case 5 has a big $\sigma$ and a mildly negative $\rho = -0.3$. Case 6, which is used in Smith (2007) for pricing Asian options, has moderate parameter values. This variety of parametric cases allow us to perform a fair and comprehensive comparison of the two schemes.

Unless otherwise stated, all numerical results shown are obtained using a sample size of $M = 2^{23}$ in standard Monte Carlo simulation, i.e. no variance reduction technique is applied. For convenience, we use constant time steps to compare the two schemes. We note that either scheme can be adapted to variable time stepping without additional computational cost.

In all numerical tests, the interpolation parameters for the IPZ–IG scheme are set as follows: $v_{min} = 0.0001$, $v_{max} = 8\sigma$ and $N_g = 2^{15 + \text{ceil} \log_2(N)} + 1$, where $N$ is the total number of time steps and $\text{ceil}$ is the ceiling function. The dependence of $N_g$ on $N$ is designed to keep the precomputation time proportional to the total simulation time as $N$ varies. We note that a similar adaptation should be applied if $M$ is to vary. Numerical results for the IPZ–IG scheme is insensitive to reasonable changes to these interpolation parameters.

The generation of uniform and normal variates is provided by the routines in Marsaglia and Tsang (2000a). Generation of Poisson variates (algorithm 1) is performed as in Van Haastrecht and Pelsser (2008). These are the fastest generators of which the authors are aware. The modified Bessel function (algorithm 4), the gamma function, and the incomplete gamma function (both in algorithm 2), are calculated by the routines provided in Press et al. (1992). The inverse cumulative normal function (quadratic approximation in QE) is implemented as in Acklam (2000). In our experience, the gamma variate generator in Best (1983) is slower than our choice of Marsaglia and Tsang (2000b). The Beasley-Springer–Moro algorithm (Glasserman 2004) for the inverse cumulative normal function is slower than our choice of (Acklam 2000).

5.1. Bias comparison for European option

Let $\alpha$ be the exact price of a European call option and $\alpha'$ the estimator returned by a simulation scheme. Then the bias of the estimator is given by $(E[\alpha'] - \alpha)$ and the standard deviation is given by $\sqrt{E[(\alpha' - E[\alpha'])^2]}$. These performance metrics can be estimated by using Monte Carlo simulations to estimate $E[\alpha']$ and calculating the sample standard deviation. The sample standard deviation divided by the square root of the sample size (simply called ‘stdv’ henceforth) sheds light on how statistically significant the bias estimate is. In our numerical results we will report the absolute value of the percentage bias, i.e. $(|E[\alpha'] - \alpha|/\alpha) \times 100\%$, as well as the percentage stdv, i.e. $\%\text{stdv} = (\text{stdv}/\alpha) \times 100\%$.

For each parametric case, we run the IPZ–IG scheme using $N=1$ to $N=16$ and run the QE–DI scheme using $N=1$ to $N=50$. Bigger values of $N$ are used in QE–DI to match the computational times of IPZ–IG since QE–DI requires less computational time per time step.

In figure 6 we plot the absolute value of the percentage bias of IPZ–IG (line with marker ‘x’) and that of QE–DI (line with marker ‘o’) on a log–log scale. To illustrate the statistical significance of the bias estimations, we draw horizontal lines to show the levels of one, three and five $\%\text{stdv}$’s, where $\%\text{stdv}$ refers to that in the IPZ–IG scheme when $N=1$. Within each parametric case, the $\%\text{stdv}$’s vary very little between the two schemes and the choice of $N$.

From figure 6, we observe the following. Overall, IPZ–IG has lower bias than QE–DI for the same computational time. The difference is significant when smaller values of $N$ are used. The accuracy of IPZ–IG is quite good even for very small values of $N$, whereas QE–DI can have big biases in such cases. When we compare the two schemes for larger values of $N$, the comparison becomes more tricky as the true biases become so low that Monte Carlo variance defies precise bias estimation. Graphically, this phenomenon happens when the curves of IPZ–IG or QE–DI become close to or go below the horizontal lines, which represent various levels of statistical significance. The curves look very oscillatory as they get near the horizontal lines because Monte Carlo variance has become dominant. Fortunately, these cases represent bias levels so low (less than 0.1%) that they are likely to
be considered as accurate enough in practice. For larger values of \( N \), IPZ–IG clearly has lower bias for the same computational time in cases 1 and 2. In the other cases, it is more difficult to say which scheme is better. For both schemes, accuracies are the worst in case 2, in which \( \sigma \) and \( T \) are big and the option is out-of-the-money.

In order to compare more precisely the performance of the two schemes, we also tabulate absolute values of the percentage bias and computational time of the two schemes for selected values of \( N \). For the IPZ–IG scheme, we always choose \( N = 1, 2, 4, 8, 16 \). For the QE–DI scheme, we choose values of \( N \) to match the computational times needed in IPZ–IG. In table 2, ‘\%stdv’ stands for the absolute value of percentage bias. ‘Time’ stands for the combined time required for precomputation (needed only in IPZ–IG), sampling \((V(t_2)|V(t_1))\) and sampling \( I_c \). Note that the time required for generating \( X(t_2) \) is excluded as it is irrelevant to the comparison.

From table 2, we observe the following. The better accuracy of IPZ–IG is evident when smaller values of \( N \) are used. IPZ–IG maintains a better speed–accuracy tradeoff at least up to \( N = 16 \). Even though QE–DI uses
an $N$ which is double to triple that used by IPZ–IG, QE–DI still has bigger bias than IPZ–IG overall.$^\dagger$

### 5.2. Asian option prices

The motivation for simulating the Heston dynamics is to price and hedge derivatives for which closed-form solutions are unavailable. Numerical comparisons in the literature, however, have largely focused on European vanilla options. In this paper, we calculate high-precision Asian option prices for our parametric cases and use these as reference prices to compare IPZ–IG and QE–DI.

Specifically, we compute fixed-strike Asian call option prices whose average value is taken over the points $t_i = iT/N_A$, $1 \leq i \leq N_A$, where $T$ is the maturity and $N_A \in \{2, 4, 8, 16\}$. Our reference Asian option prices are calculated using the IPZ–IG and the QE–DI schemes with $N = 128$ and $M = 2^{30}$ to obtain low bias and low stdv. The reference price differences between the two schemes are within 0.3% of each other and thus it makes no practical difference to use either as reference prices. In our numerical comparison, we will use the reference prices from the QE–DI scheme.

We list the reference Asian option prices in table 3, in which IPZ–IG–128 represents the IPZ–IG scheme with $N = 128$ time steps and QE–DI–128 represents the QE–DI scheme with $N = 128$ time steps.

#### 5.3. Bias comparison for Asian options

When pricing vanilla European options, we are free to choose a small $N$ as long as the bias is low enough. This flexibility, however, vanishes when pricing path-dependent options. For example, when pricing an Asia option with maturity $T$ that depends on the asset prices $\{X(t_i); i = 1, 2, 3, \ldots, 16\}$, one must simulate all the asset prices. Consequently, if two schemes have the same bias and require the same amount of time to compute, the scheme which uses a bigger $N$ would be more flexible in pricing path-dependent options.

For each parametric case and each value of $N_A \in \{1, 2, 4, 8, 16\}$, we price the Asian option using the IPZ–IG scheme with $N = N_A$ and using the QE–DI scheme with $N = N_A$, $2N_A$, $3N_A$ and $4N_A$. Using various values of $N$ in the QE–DI scheme allows us to compare the

---

$n$ 1, 2, 3, $n$ 4, 8, 16

**Table 2. Comparison of the absolute value of percentage bias and computational time on pricing European options.**

<table>
<thead>
<tr>
<th>Case</th>
<th>N</th>
<th>%bias</th>
<th>Time</th>
<th>N</th>
<th>%bias</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1.2320</td>
<td>1.014</td>
<td>2</td>
<td>1.4227</td>
<td>1.012</td>
</tr>
<tr>
<td>2</td>
<td>0.4520</td>
<td>2.169</td>
<td>4</td>
<td>12.7518</td>
<td>2.107</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1.0358</td>
<td>4.65</td>
<td>10</td>
<td>7.6547</td>
<td>4.835</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.4907</td>
<td>9.406</td>
<td>20</td>
<td>2.5452</td>
<td>9.529</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>0.0811</td>
<td>19.224</td>
<td>40</td>
<td>0.4901</td>
<td>20.269</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4.0352</td>
<td>1.311</td>
<td>3</td>
<td>6.8368</td>
<td>1.419</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.3914</td>
<td>2.386</td>
<td>5</td>
<td>30.2549</td>
<td>2.542</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3.1003</td>
<td>4.352</td>
<td>10</td>
<td>26.0417</td>
<td>4.867</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>2.3760</td>
<td>9.188</td>
<td>19</td>
<td>8.1873</td>
<td>9.236</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>0.0706</td>
<td>19.766</td>
<td>41</td>
<td>0.7034</td>
<td>19.892</td>
<td></td>
</tr>
</tbody>
</table>

**Table 3. Reference prices for Asian calls computed using 128 time steps and $2^n$ samples.**

<table>
<thead>
<tr>
<th>Case</th>
<th>$N_A$</th>
<th>IPZ–IG–128</th>
<th>stdv</th>
<th>QE–DI–128</th>
<th>stdv</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>10.297189</td>
<td>3.18E–04</td>
<td>10.288824</td>
<td>3.18E–04</td>
</tr>
<tr>
<td>4</td>
<td>8.963870</td>
<td>2.74E–04</td>
<td>8.955930</td>
<td>2.74E–04</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>8.322643</td>
<td>2.53E–04</td>
<td>8.314791</td>
<td>2.53E–04</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>8.000178</td>
<td>2.42E–04</td>
<td>7.992200</td>
<td>2.42E–04</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.083577</td>
<td>4.06E–05</td>
<td>0.083820</td>
<td>4.06E–05</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.040602</td>
<td>2.80E–05</td>
<td>0.040638</td>
<td>2.80E–05</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.026770</td>
<td>2.24E–05</td>
<td>0.026762</td>
<td>2.24E–05</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>0.020761</td>
<td>1.95E–05</td>
<td>0.020740</td>
<td>1.95E–05</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3.238738</td>
<td>8.82E–04</td>
<td>3.238747</td>
<td>8.83E–04</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3.067611</td>
<td>5.26E–04</td>
<td>3.059649</td>
<td>5.26E–04</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>3.269475</td>
<td>5.00E–04</td>
<td>3.267829</td>
<td>5.00E–04</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>3.250039</td>
<td>4.87E–04</td>
<td>3.249306</td>
<td>4.86E–04</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5.187462</td>
<td>1.71E–04</td>
<td>5.184144</td>
<td>1.71E–04</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>4.389704</td>
<td>1.44E–04</td>
<td>4.389694</td>
<td>1.44E–04</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>3.996457</td>
<td>1.31E–04</td>
<td>3.994062</td>
<td>1.31E–04</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>26.378587</td>
<td>1.26E–03</td>
<td>26.379998</td>
<td>1.26E–03</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>22.457799</td>
<td>1.04E–03</td>
<td>22.47118</td>
<td>1.04E–03</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>19.203531</td>
<td>8.87E–04</td>
<td>19.20468</td>
<td>8.87E–04</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>11.469086</td>
<td>5.00E–04</td>
<td>11.468053</td>
<td>5.00E–04</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>8.857739</td>
<td>3.76E–04</td>
<td>8.857091</td>
<td>3.76E–04</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>8.441458</td>
<td>3.57E–04</td>
<td>8.440885</td>
<td>3.57E–04</td>
<td></td>
</tr>
</tbody>
</table>

$^\dagger$One may notice that the bias behaves rather erratically at very few time steps for the QE–DI scheme, particularly in cases 1 and 2, as opposed to the smoother convergence behaviour tabulated in Andersen (2007). This is due to the fact that the coarsest time step size considered in Andersen (2007) is one year, which translates to $N = 10$ (ten time steps) in our convergence analysis. Erratic behaviour is observed for $N < 10$. 

---

931

Downloaded by [University of Waterloo] at 07:22 19 November 2014
speed–accuracy tradeoff between the two schemes. As in section 5.1, we plot the absolute value of percentage bias against the computational time on a log–log scale in figures 7–12. In the figures, the speed–accuracy tradeoff of IPZ–IG for $N=N_A$ is represented by a diamond; that of QE–DI for $N=N_A$, $2N_A$, $3N_A$ and $4N_A$ is represented by a solid line. We also draw horizontal lines to represent levels of statistical significance, as previously done in figure 6. We also tabulate our results in tables 4–9.

From the figures and the tables, we have similar observations as for European options. Overall, IPZ–IG has lower bias than QE–DI for the same computational time. The difference is significant when smaller values of $N$ are used. When we compare the two schemes for $N_A=16$, the biases of both schemes have become so low that they are statistically insignificant, and we can only say that the two schemes are equally good.

5.4. Efficiency analysis

To analyse the efficiencies of IPZ–IG and QE–DI, we break down the computational times into precomputation (required only in IPZ–IG), sampling $(V(t_2)|V(t_1))$ and sampling $L_c$. Again, the time required in sampling $X(t_2)$ from the lognormal distribution is excluded from our analysis. We choose to analyse the timing results for cases 1 and 4 because they are, respectively, the least expensive and the most expensive to compute (per time step), for both IPZ–IG and QE–DI. In table 10, ‘Pre.’ refers to the time required for precomputation, ‘Sum’ refers to the sum of the components, and ‘|%bias|’ stands for absolute value of percentage bias.

From table 10, we observe the following. For both IPZ and QE, case 4 takes significantly more time than case 1. For IPZ, it is because $\text{Poisson}(V(t_1)n(t_1,t_2)/2)=0$ occurs much less often in case 4. For QE, it is because the slower quadratic approximation is used much more often than the faster exponential approximation. For IG, case 4 takes more time than case 1. This is because $V(t_1)V(t_2)\neq0$ happens more often and hence the computations of $E[X(t_1)|V(t_2)]$ and $\text{Var}[X(t_1)|V(t_2)]$ cannot be skipped. The difference is much less dramatic than that between IPZ and QE as the time required for IG variate generation, which stays roughly constant, dominates the time required for moment calculation. For DI, cases 1 and 4 make little difference, as expected. Precomputation in IPZ–IG takes between 3 and 6% of the total time. We remark that one main reason why QE is faster than IPZ is that QE avoids the Poisson variate generation altogether. The performance difference between IPZ and QE is bigger since $V(t_1)n(t_1,t_2)/2$ is often bigger in case 4 and thus it takes more time to generate the Poisson variate.

6. Conclusion

In this paper, we proposed the Inverse Gaussian approximation to $L_c$ and proved that the moment-matched IG approximation is asymptotically exact. Numerical results verified that the IG approximation is very accurate.
Figure 8. Absolute value of percentage bias against computational time on pricing Asian options in case 2.

Figure 9. Absolute value of percentage bias against computational time on pricing Asian options in case 3.
Figure 10. Absolute value of percentage bias against computational time on pricing Asian options in case 4.

Figure 11. Absolute value of percentage bias against computational time on pricing Asian options in case 5.
Table 4. Comparison of the absolute value of percentage bias and computational time on pricing Asian options in case 1.

| Method   | \( N_A \) | \(|\text{bias}|\) | Time  | \( N_A \) | \(|\text{bias}|\) | Time  |
|----------|----------|----------------|------|----------|----------------|------|
| IPZ–IG   | 2 2      | 0.357          | 2.449| 8 8      | 0.362          | 7.489|
| QE–DI    | 2 2      | 3.920          | 1.061| 8 8      | 4.790          | 2.746|
|          | 4 2      | 7.594          | 1.856| 16 8     | 2.386          | 5.74  |
|          | 6 2      | 8.281          | 3.042| 24 8     | 1.166          | 8.922|
|          | 8 2      | 7.133          | 3.667| 32 8     | 0.683          | 12.325|
| IPZ–IG   | 4 4      | 0.844          | 4.244| 16 16    | 0.057          | 18.886|
| QE–DI    | 4 4      | 5.148          | 1.809| 16 16    | 2.274          | 8.379|
|          | 8 4      | 5.626          | 3.697| 32 16    | 0.639          | 15.084|
|          | 12 4     | 3.883          | 5.634| 48 16    | 0.189          | 24.31 |
|          | 16 4     | 2.653          | 7.705| 64 16    | 0.103          | 27.988|

Table 5. Comparison of the absolute value of percentage bias and computational time on pricing Asian options in case 2.

| Method   | \( N_A \) | \(|\text{bias}|\) | Time  | \( N_A \) | \(|\text{bias}|\) | Time  |
|----------|----------|----------------|------|----------|----------------|------|
| IPZ–IG   | 2 2      | 1.146          | 2.293| 8 8      | 1.251          | 7.019|
| QE–DI    | 2 2      | 146.829        | 0.998| 8 8      | 36.515         | 2.793|
|          | 4 2      | 11.283         | 1.966| 16 8     | 6.921          | 5.677|
|          | 6 2      | 34.845         | 3.042| 24 8     | 0.764          | 8.736|
|          | 8 2      | 29.681         | 3.571| 32 8     | 1.800          | 12.489|
|          | 8 4      | 31.019         | 3.667| 32 16    | 0.444          | 17.659|
|          | 12 4     | 13.244         | 6.319| 48 16    | 0.287          | 23.669|
|          | 16 4     | 5.865          | 7.801| 64 16    | 0.350          | 27.802|

Table 6. Comparison of the absolute value of percentage bias and computational time on pricing Asian options in case 3.

| Method   | \( N_A \) | \(|\text{bias}|\) | Time  | \( N_A \) | \(|\text{bias}|\) | Time  |
|----------|----------|----------------|------|----------|----------------|------|
| IPZ–IG   | 2 2      | 0.444          | 2.309| 8 8      | 0.020          | 7.083|
| QE–DI    | 2 2      | 1.784          | 1.015| 8 8      | 1.483          | 2.778|
|          | 4 2      | 4.884          | 1.841| 16 8     | 0.354          | 6.335|
|          | 6 2      | 3.516          | 2.887| 24 8     | 0.067          | 8.846|
|          | 8 2      | 2.364          | 3.587| 32 8     | 0.027          | 12.417|
| IPZ–IG   | 4 4      | 0.452          | 4.179| 16 16    | 0.034          | 20.155|
| QE–DI    | 4 4      | 3.349          | 1.841| 16 16    | 0.344          | 7.94  |
|          | 8 4      | 1.801          | 4.198| 32 16    | 0.012          | 17.506|
|          | 12 4     | 0.859          | 6.07 | 48 16    | 0.024          | 21.106|
|          | 16 4     | 0.407          | 7.47 | 64 16    | 0.018          | 27.953|

Table 7. Comparison of the absolute value of percentage bias and computational time on pricing Asian options in case 4.

| Method   | \( N_A \) | \(|\text{bias}|\) | Time  | \( N_A \) | \(|\text{bias}|\) | Time  |
|----------|----------|----------------|------|----------|----------------|------|
| IPZ–IG   | 2 2      | 0.125          | 3.01 | 8 8      | 0.026          | 12.059|
| QE–DI    | 2 2      | 16.277         | 0.998| 8 8      | 3.920          | 2.793|
|          | 4 2      | 5.859          | 3.542| 16 8     | 0.458          | 11.791|
|          | 6 2      | 2.814          | 5.568| 24 8     | 0.237          | 18.484|
|          | 8 2      | 1.681          | 7.3  | 32 8     | 0.162          | 23.31 |
| IPZ–IG   | 4 4      | 0.039          | 6.194| 16 16    | 0.003          | 18.886|
| QE–DI    | 4 4      | 5.998          | 3.385| 16 16    | 0.227          | 18.484|
|          | 8 4      | 1.780          | 7.16 | 32 16    | 0.138          | 23.31 |
|          | 12 4     | 0.782          | 10.296| 48 16    | 0.033         | 36.707|
|          | 16 4     | 0.453          | 14.397| 64 16    | 0.036         | 48.467|
To facilitate moment-matching, we derived simple formulae for the moments of \( I_c \) and designed a very fast and accurate algorithm for calculating the moments. We also developed the IPZ scheme, which is simple, efficient and accurate, for sampling \( (V(t_2)) / (V(t_1)) \). Our numerical results showed that the combined IPZ-IG scheme has lower bias for the same computational time compared with QE and GE on pricing both European and Asian options when a moderate number of time steps is used. Throughout the paper, we discussed in detail the speed—accuracy tradeoff of different approaches to approximate the sampling of \( I_c \) and \( (V(t_2)) / (V(t_1)) \). In particular, we presented a detailed efficiency analysis for IPZ-IG and QE-DI under different parameter settings.

Although our discussion focused primarily on the Heston model, we note that our schemes can also be used to simulate more general affine jump diffusion processes. More specifically, the extensions considered in Broadie and Kaya (2006) and Andersen (2007) are directly applicable to the IPZ-IG scheme. We also note that proposition 3.3 can be translated into a convergence result for a squared Ornstein—Uhlenbeck (OU) bridge with endpoints equal to zero. Readers are referred to Glasserman and Kim (2011) for a detailed discussion on the connection between \( I_c \) and the OU bridge. Another potential application of our convergence result would be on large-maturity asymptotics of implied volatility in the Heston model (Tehranchi 2009, Forde et al. 2011, 2010).

Better schemes may be developed based on the ideas in this paper. One direction would be in finding an approximation to \( I_c \) that is accurate, has moments that can be calculated quickly, and has a fast sampling algorithm. Another direction is to develop faster algorithms to generate IG variates. Modifications to the IPZ scheme may also be considered. For example, one may compute the quantile function of the gamma distribution by numerical inversion or interpolating for the cases where the Poisson variate is small but non-zero. It would also be interesting to combine the IG scheme with the very recent work of Halley et al. (2008) on the efficient and exact sampling of \((V(t_2)) / (V(t_1))\).

### References


Best, D., A note on gamma variate generators with shape parameter less than unity. Computing, 1983, 30, 185–188.

Bossy, M. and Diop, A., An efficient discretisation scheme for one dimensional SDEs with a diffusion coefficient function of the form \(|x|^a\), \(a \in [1/2, 1)\). RR-5396, INRIA, Décembre 2007.


