Journal of Economic Dynamics & Control 163 (2024) 104861

Contents lists available at ScienceDirect

Journal of Economic Dynamics and Control

journal homepage: www.elsevier.com/locate/jedc

Exact simulation of the Hull and White stochastic volatility model

Riccardo Brignone [a,*], Luca Gonzato [b]

[a] Department of Quantitative Finance, Institute for Economic Research, University of Freiburg, Rempartstr. 16, 79098 Freiburg i. Br., Germany

[b] Department of Statistics and Operations Research, University of Vienna, Kolingasse 14-16, 1090 Vienna, Austria

ARTICLE INFO

JEL classification:
C15
C32
C63
G13

Keywords:
Simulation
Stochastic volatility
Option pricing
Laplace transforms

ABSTRACT

We show how to simulate exactly the asset price and the variance under the Hull and White stochastic volatility model. We derive analytical formulas for the Laplace transform of the time integral of volatility conditional on the variance level at the endpoint of the time interval and the Laplace transform of integrated variance conditional on both integrated volatility and variance. Based on these results, we simulate the model through a nested-conditional factorization approach, where Laplace transforms are inverted through the (conditional) Fourier-cosine (COS) method. Under this model, our approach can be used to generate unbiased estimates for the price of derivatives instruments. We propose some variants of the exact simulation scheme for computing unbiased estimates of option prices and sensitivities, a difficult task in the Hull and White model. These variants also allow for a significant reduction in the Monte Carlo simulation estimator's variance (around 93-98%) and the computing time (around 22%) when pricing options. The performances of the proposed algorithms are compared with various benchmarks. Numerical results demonstrate the faster convergence rate of the error in our method, which achieves an $O(s^{-1/2})$ convergence rate, where s is the total computational budget, largely outperforming the benchmark.

1. Introduction

The exact simulation of stochastic volatility models is a standard topic in financial engineering since the seminal work of Broadie and Kaya (2006), which showed that the transition of the variance and the asset price processes can be simulated exactly (without any time discretization) in the Heston model. In later years, many authors proposed exact simulation schemes for various stochastic volatility models, as we illustrate later on. In this paper, we contribute to this stream of literature by developing an exact simulation scheme for the Hull and White (1987) stochastic volatility model (henceforth, HW-SV).

The possibility to simulate exactly a stochastic volatility model while avoiding time discretization is fundamental for many reasons (Broadie and Kaya, 2006): i) discretization introduces bias into the simulation output, causing serious problems when computing the prices or Greeks of derivative securities; ii) it is not possible to determine *a priori* the number of time steps needed to reduce the discretization bias to an acceptable level; iii) since the bias is unknown, the standard error may be a poor estimate of the actual error, and valid confidence intervals are not available; iv) the convergence rate for exact simulation schemes is $O(s^{-1/2})$, where s is the total computational budget, whereas the error for a first-order method such as Euler discretization has $O(s^{-1/3})$ convergence (Duffie and Glynn, 1995). For these reasons, exact simulation schemes have been proposed in the literature for various stochastic

[*] Corresponding author.
E-mail addresses: riccardo.brignone@finance.uni-freiburg.de (R. Brignone), luca.gonzato@univie.ac.at (L. Gonzato).

https://doi.org/10.1016/j.jedc.2024.104861
Received 19 December 2023; Received in revised form 28 March 2024; Accepted 5 April 2024

Available online 12 April 2024

Table 1
Literature review on exact simulation of stochastic volatility models.

Model	$\sigma_S(S_t, V_t, t)$	$\mu_V(V_t, t)$	$\sigma_V(V_t, t)$	Y	X	Literature
Heston	$S_t\sqrt{V_t}$	$k(\theta - V_t)$	$\sigma\sqrt{V_t}$	V_T	$\int_0^T V_s ds$	Broadie and Kaya (2006)
3/2	$S_t\frac{b}{\sqrt{V_t}}$	$k(\theta - V_t)$	$\sigma\sqrt{V_t}$	V_T	$\int_0^T \frac{1}{V_s} ds$	Baldeaux (2012)
SABR	$S_t^\beta\sqrt{V_t}$	$\sigma^2 V_t$	$2\sigma V_t$	V_T	$\int_0^T V_s ds$	Cai et al. (2017)
4/2	$S_t\left(a\sqrt{V_t} + \frac{b}{\sqrt{V_t}}\right)$	$k(\theta - V_t)$	$\sigma\sqrt{V_t}$	V_T	$\log S_T$	Grasselli (2017)
SV-OU	$S_t\sqrt{V_t}$	$2k\left(\frac{\sigma^2}{2k} + \theta\sqrt{V_t} - V_t\right)$	$2\sigma\sqrt{V_t}$	$\left(V_T, \int_0^T \sqrt{V_s} ds\right)$	$\int_0^T V_s ds$	Li and Wu (2019)
HW-SV	$S_t\sqrt{V_t}$	ηV_t	σV_t	V_T	$\int_0^T \sqrt{V_s} ds$	This paper

volatility models. Despite each proposed exact simulation method is tailored to the specific model of interest, all these schemes are based on a *nested-conditional factorization* approach (Parrish, 1987, 1990): suppose to have two dependent random variables X and Y and one is interested to obtain a joint sample (X, Y), then a possible approach is to simulate Y first and then X conditionally on Y. Various stochastic volatility models can be simulated exactly by choosing appropriately X and Y, which are typically functions of the variance process and its time integral. Then, the asset price process is simulated exploiting the fact that, conditionally on X and Y, the log-asset price distribution is known. Under a suitably defined filtered probability space, a stochastic volatility model is given by the following couple of stochastic differential equations:

$$dS_t = rS_t dt + \sigma_S(S_t, V_t, t)(\rho dB_t + \sqrt{1-\rho^2} dW_t),$$
$$dV_t = \mu_V(V_t, t)dt + \sigma_V(V_t, t)dB_t,$$

where B_t and W_t are independent standard Brownian motions, r is the risk-less rate and $\rho \in [-1, 1]$. The functions μ_V, σ_V, σ_S define a specific stochastic volatility model, as shown in Table 1 where we provide a schematic literature review on the exact simulation of stochastic volatility models. For notational convenience, we excluded from Table 1 the important contribution in Kang et al. (2017), where an exact simulation scheme for the Wishart stochastic volatility model is developed. It is worth noting that the simulation of $(X|Y)$ is performed by numerically inverting the conditional Laplace transform of $(X|Y)$ and then applying inverse transform sampling given the numerical estimate of the cumulative distribution function. As a result, some error is introduced in the simulation during the numerical inversion and the term "exact" may sound misleading. Nevertheless, following the standard literature indicated in Table 1, we will still refer to our methodology as exact.

In this paper, we aim to extend this important stream of literature by developing an exact simulation scheme for the HW-SV model. Let us denote with $(V_t)_{t\geq 0}$ the variance process, we show that, for a fixed maturity $T > 0$, the asset price process S_T depends on the following triplet: variance (V_T), its time integral ($\int_0^T V_s ds$) and the time integral of the volatility ($\int_0^T \sqrt{V_s} ds$). Then, exploiting some theoretical results in Matsumoto and Yor (2005), we derive the conditional Laplace transforms of $\left(1/\int_0^T \sqrt{V_s} ds | V_T\right)$ and $\left(\int_0^T V_s ds | \int_0^T \sqrt{V_s} ds, V_T\right)$ and develop a *nested-conditional factorization* approach for the exact simulation of the transition of asset price and variance processes in the HW-SV model. More specifically, we perform the simulation in three steps: *i*) simulate $(V_T|V_0)$; *ii*) simulate $(\int_0^T \sqrt{V_s} ds | V_T)$; *iii*) simulate $\left(\log S_T | \int_0^T \sqrt{V_s} ds, V_T\right)$. The first step is simple because V is a geometric Brownian motion, while the second and third steps rely on knowledge of the aforementioned Laplace transforms. We use the Fourier-cosine (COS) method by Fang and Oosterlee (2009) to obtain the relevant cumulative distribution functions given the Laplace transforms, and then simulate using inverse transform sampling. The usage of the COS method deserves particular attention. The literature summarized in Table 1 follows a different approach for numerical inversion of the Laplace transforms and applies the algorithm in Abate and Whitt (1992). This choice is convenient because such an algorithm has a theory of error control that also performs well in practice. However, the COS method is a better alternative because it is known to perform exceptionally well in practice and it also presents an error control theory (Fang and Oosterlee, 2009, Section 4). To have a theoretical control of the error for the numerical inversion of the Laplace transforms involved in our simulation scheme using the COS method, as we demonstrate later in this paper, we need to find the so-called truncation range (Fang and Oosterlee, 2009, Section 5.3). To this end, we propose two different methodologies for properly computing the truncation range based on the conditional moments of some relevant distributions. We refer to later sections for more details. However, we anticipate that with our approach the error is controlled by only one parameter and that this error vanishes when the parameter is chosen sufficiently large.

An important difference between this paper and earlier references is that in the case of the HW-SV model the characteristic function of the log-asset price is unknown, precluding the possibility of pricing European options through standard Fourier inversion techniques (Carr and Madan, 1999, Fang and Oosterlee, 2009). Many attempts have been made in the literature for computing efficiently option prices in the HW-SV model. In the original paper, Hull and White (1987) propose to sample discrete approximations. More recently, Heston and Rossi (2017) propose to approximate the risk-neutral density of the log-returns through an orthogonal (logistic) polynomial expansion based on the moments of log-returns. However, these moments are difficult to evaluate numerically and it is not possible to check *a priori* if, for given model parameters, the approximated option price converges (or diverges) to the true option price when using more moments. An alternative approach based on orthogonal polynomial expansions has been proposed by Ackerer and Filipović (2020). However, also in their case, the computation of option prices is numerically unstable due to numerical precision difficulties, as pointed out by the same authors. Finally, Zeng et al. (2023) provide a general framework for the exact simulation of stochastic volatility models. However, they conclude that the HW-SV model can not be simulated exactly through their method, and that only approximations are possible. The simulation approach proposed in the present paper is different (with respect to the one attempted in Zeng et al., 2023) and leads to an exact solution. The difference relies on the fact that Zeng et al. (2023) consider a two steps approach: *i*) simulate $(V_T|V_0)$; *ii*) simulate $(\log S_T|V_T)$. In contrast, our approach is based on three steps: *i*) simulate $(V_T|V_0)$; *ii*) simulate $\left(1/\int_0^T \sqrt{V_s}ds|V_T\right)$; *iii*) simulate $\left(\log S_T|\int_0^T \sqrt{V_s}ds, V_T\right)$. The intermediate step allows deriving all the necessary Laplace transforms leading to an exact simulation scheme. Therefore, the new methodology presented in this paper can be exploited for pricing European options efficiently, extending the stream of literature concerned with the problem of pricing European options in the HW-SV model. In addition, we present some simple modifications of the original algorithm to compute Greeks in the HW-SV model, a problem which is, to the best of our knowledge, unstudied in the literature.

Despite implementation difficulties (such as the lack of the characteristic function of log-returns), the HW-SV model is important for several reasons (Heston and Rossi, 2017): *i*) it has been influential in derivatives valuation; *ii*) performs well in empirical tests (Christoffersen et al., 2010a,b); *iii*) it is the continuous time analog of the classic GARCH(1,1) model, which is one the main tools for the econometric analysis of volatility and has shown excellent performances in empirical studies (Duan, 1995; Hsieh and Ritchken, 2005). Moreover, in Section EC.2 of the e-companion we present an econometric analysis where we compare the performances of the HW-SV model against the popular Heston model on various datasets including equity indices and commodity ETFs. We find that the HW-SV model provides a better fit to market data. This is due to the fact that the HW-SV model is non-affine, leading to expectantly better empirical performances with respect to affine models (Fulop and Li, 2019 and the references therein). This result is remarkable since the HW-SV model is also more parsimonious in terms of the number of model parameters with respect to the Heston model.

Our proposed exact simulation algorithm is flexible and can be modified in several ways according to specific needs, allowing for efficient variance reduction. First, we propose a conditional COS approach for computing European option prices, which reduces the variance of the Monte Carlo estimator by approximately 93-98%. We also show that, at the cost of increasing the computing time, variance can be further reduced (up to 99%) by employing antithetic sampling. These techniques allow for the generation of unbiased estimates for path-independent option prices with small variance. Second, we propose another variant of the proposed algorithm that allows for unbiased estimates for the Greeks. Following the standard literature listed in Table 1, we compare the performance of the proposed exact simulation method to that of a low-bias scheme (described in Section EC.6 of the e-companion). We find that the exact simulation scheme (in agreement with the literature indicated in Table 1) presents a faster convergence rate of the root mean squared error. To be more specific, our method recovers the $O(s^{-1/2})$ convergence rate of an unbiased Monte Carlo estimator used to simulate derivative prices in the HW-SV model.

In summary, this paper contributes to the literature in four ways. *i*) We extend the literature on the exact simulation of stochastic volatility models, summarized in Table 1, to a new important model, i.e. the Hull-White stochastic volatility model. *ii*) We propose a conditional COS formula for pricing European options under the Hull-White stochastic volatility model as a first variant of our simulation scheme. This approach can be combined with antithetic sampling to further reduce the estimator's variance, resulting in unbiased estimates for European call option prices with extremely tight confidence intervals. *iii*) Through a second variant of our simulation scheme, we derive unbiased estimates for the Greeks of European call options. *iv*) Laplace transform inversions are performed through the well-known COS method. We show how to compute the truncation range (Fang and Oosterlee, 2009, Section 5.3), which is fundamental in order to rely on the theory of error control, extending some results in Kyriakou et al. (2023).

The rest of the paper is organized as follows. In Section 2 we present the HW-SV model, in Section 3 we outline our exact simulation scheme, in Section 4 we propose conditional COS pricing formulas for European options, useful to reduce the variance of our unbiased price estimator and another variant to compute Greeks; in Section 5 we present numerical results, Section 6 concludes. Supplementary results are deferred to the e-companion.

2. Hull and White stochastic volatility model

Let $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\in[0,T]}, \mathbb{Q})$ be a filtered probability space, which supports all the processes we encounter in the sequel and satisfies usual assumptions. In the Hull and White (1987) stochastic volatility model (HW-SV) the asset price and variance processes under the risk-neutral measure $\mathbb{Q}$ are given by the solution of the following stochastic differential equations (SDEs):

$$dS_t = rS_t dt + \sqrt{V_t}S_t\left(\rho dB_t + \sqrt{1-\rho^2}dW_t\right), \tag{1}$$

$$dV_t = \eta V_t dt + \sigma V_t dB_t, \tag{2}$$

where r is the risk-less rate, W_t and B_t are independent standard Brownian motions and the parameter $\rho \in [-1,1]$ controls the correlation between the asset price S_t and its variance V_t. The parameters η and σ are the drift and diffusion coefficients of the geometric Brownian motion driving the variance process.

Given a final date $T > t > 0$, the asset price, variance and volatility at time T are

$$S_T = S_0 \exp\left(rT - \frac{1}{2}\int_0^T V_s ds + \rho\int_0^T \sqrt{V_s}dB_s + \sqrt{1-\rho^2}\int_0^T \sqrt{V_s}dW_s\right), \tag{3}$$

$$V_T = V_0 \exp\left(\left(\eta - \frac{\sigma^2}{2}\right)T + \sigma B_T\right), \tag{4}$$

$$\sqrt{V_T} = \sqrt{V_0} \exp\left(\left(\eta - \frac{\sigma^2}{2} \right) \frac{T}{2} + \frac{\sigma}{2} B_T \right). \tag{5}$$

An application of Itô's lemma to $\sqrt{V_t}$ leads to

$$d(\sqrt{V_t}) = \left(\frac{\eta}{2} - \frac{\upsilon^2}{2} \right) \sqrt{V_t} dt + \upsilon \sqrt{V}_t dB_t,$$

where $\upsilon = \frac{\sigma}{2}$. Integrating both sides we get

$$\int\limits_0^T d\sqrt{V_s} = \left(\frac{\eta}{2} - \frac{\upsilon^2}{2} \right) \int\limits_0^T \sqrt{V_s} ds + \upsilon \int\limits_0^T \sqrt{V_s} dB_s$$

$$\upsilon \int\limits_0^T \sqrt{V_s} dB_s = \sqrt{V_T} - \sqrt{V_0} - \left(\frac{\eta}{2} - \frac{\upsilon^2}{2} \right) \int\limits_0^T \sqrt{V_s} ds.$$

Therefore, the conditional distribution of the log–price at time T is

$$\left(\ln S_T \middle| V_T, \int\limits_0^T V_s ds, \int\limits_0^T \sqrt{V_s} ds, V_0, S_0 \right) \sim \mathcal{N}(m, s^2), \tag{6}$$

where $m = \ln(S_0) + rT - \frac{1}{2} \int_0^T V_s ds + \frac{\rho}{\upsilon} \left(\sqrt{V_T} - \sqrt{V_0} - \frac{1}{2} \left(\eta - \upsilon^2 \right) \int_0^T \sqrt{V_s} ds \right)$, $s^2 = (1 - \rho^2) \int_0^T V_s ds$, $\upsilon = \frac{\sigma}{2}$ and $\mathcal{N}$ denotes the normal distribution. In this paper, we show how to simulate exactly the asset price and variance processes $(S_T, V_T | S_0, V_0)$.

3. Exact simulation scheme

Following standard literature on exact simulation of option pricing models, we rely on a *nested-conditional factorization* approach to sample $(S_T, V_T | S_0, V_0)$: we first simulate V_T, then we show how to sample $\left(\int_0^T \sqrt{V_s} ds | V_T \right)$ and $\left(X_T | \int_0^T \sqrt{V_s} ds, V_T \right)$, where $X_T := \log(S_T / S_0)$. These steps are performed by first numerically inverting the relevant conditional Laplace transforms and then implementing random numbers generation via the inverse transform method. Hence, we start by introducing the conditional Laplace transforms that are relevant to our approach. Then, we describe the COS method to obtain the necessary cumulative distribution functions. Finally, we summarize our proposed algorithm for the exact simulation of the HW-SV model. The usage of the COS method represents a remarkable difference with respect to the literature indicated in Table 1, where the numerical inversion of the Laplace transform is implemented through algorithms proposed in Abate and Whitt (1992, 1995). Nevertheless, more efficient algorithms exist for this purpose, such as convolution methods, FFT and COS. In this paper, we use the COS method, which is expected to perform better than alternatives (Fang and Oosterlee, 2009). In addition, we provide in this section detailed guidelines on how to implement this method in our context, where the characteristic functions depend on the random realization of other random variables. We define methodologies for calculating the domain truncation range, which is critical for theoretical error control (Fang and Oosterlee, 2009; Junike and Pankrashkin, 2022).

3.1. Relevant conditional Laplace transforms

The following proposition provides the Laplace transform of $\left(\frac{1}{\int_0^T \sqrt{V_s} ds} \middle| V_T \right)$.

Proposition 1. *Given $u > 0$ and V_T as in Eq. (4) we have*

$$\mathcal{L}_1(u) := \mathrm{E}\left[\exp\left(-\frac{u}{\int_0^T \sqrt{V_s} ds} \right) \middle| V_T \right] = \exp\left(-\frac{\Xi\left(\frac{1}{4} \log \frac{V_T}{V_0}, \frac{u\sigma^2}{16\sqrt{V_0}} \right)^2 - \left(\frac{1}{4} \log \frac{V_T}{V_0} \right)^2}{T \frac{\sigma^2}{8}} \right), \tag{7}$$

where $\Xi(x, \lambda) = \operatorname{arcosh}(\lambda e^{-x} + \cosh(x))$.

Proof. See Section EC.1.1 of the e-companion. □

In the following proposition we present the conditional Laplace transform of $\int_0^T V_s ds$ given $\int_0^T \sqrt{V_s} ds$ and V_T.

Proposition 2. *Given $u > 0$, V_T and $\sqrt{V_T}$ as in Eqs. (4)–(5), we have*

$$\mathcal{L}_2(u) := \mathrm{E}\left[\exp\left(-u\int_0^T V_s ds\right)\Big|\int_0^T \sqrt{V_s}ds, V_T\right] = \frac{\theta(\phi(v^\star, x^\star, \sqrt{\lambda^\star}), t^\star/4)}{\psi_{t^\star}^{(\mu^\star)}(v^\star, x^\star)} e^{\mu^\star x^\star - (\mu^\star)^2 t^\star/2}$$
$$\times \frac{\sqrt{\lambda^\star}}{4\sinh(\sqrt{\lambda^\star}v^\star/2)} \exp\left(-\sqrt{\lambda^\star}(1+e^{x^\star})\coth(\sqrt{\lambda^\star}v^\star/2)\right), \tag{8}$$

where

$$\phi(v,x,\lambda) = \frac{2\lambda\exp(x/2)}{\sinh(\lambda v/2)}, \quad \psi_t^{(\mu)}(v,x) = \frac{1}{2v}e^{\mu x - \mu^2 t/2}\exp\left(-\frac{2(1+x)}{v}\right)\theta(4e^{x/2}/v, t/4)$$

$$\theta(r,t) = \frac{r}{\sqrt{2\pi^3 t}}\int_0^\infty e^{-\frac{\xi^2}{2t}} e^{-r\cosh(\xi)}\sinh(\xi)\sin\left(\frac{\pi\xi}{t}\right)d\xi$$

and $\lambda^\star = \frac{8uV_0}{\sigma^2}$, $t^\star = T\frac{\sigma^2}{4}$, $x^\star = \frac{1}{2}\log\frac{V_T}{V_0}$ *and* $v^\star = \frac{\int_0^T \sqrt{V_s}ds}{\sqrt{V_0}\frac{4}{\sigma^2}}$, $\mu^\star = \frac{2\eta}{\sigma^2} - 1$.

Proof. See Section EC.1.2 of the e-companion. □

Finally, exploiting Eq. (6), we derive the Laplace transform of X_T given V_T and $\int_0^T \sqrt{V_s}ds$ as a function of $\mathcal{L}_2(u)$.

Proposition 3. *Given $u > 0$ we have*

$$\mathcal{L}_3(u) := \mathrm{E}\left[e^{-uX_T}\Big|V_T, \int_0^T \sqrt{V_s}ds\right] = e^{-u\left(rT + \frac{\rho}{v}\left(\sqrt{V_T} - \sqrt{V_0} - \frac{1}{2}(\eta - v^2)\int_0^T \sqrt{V_s}ds\right)\right)}\mathcal{L}_2\left(-\left(\frac{1}{2} + \frac{1}{2}u(1-\rho^2)\right)\right). \tag{9}$$

Proof. See Section EC.1.3 of the e-companion. □

3.2. Random sampling using conditional COS

Given the Laplace transforms derived above, we assume that the related characteristic functions follow by setting $\varphi_j(u) := \mathcal{L}_j(-iu)$ for $j = \{1, 2, 3\}$ and $i := \sqrt{-1}$. However, since the Laplace transforms derived in Section 3.1 are all derived based on the existing results in the real space, it is not known whether plugging $-iu$ into the results gives the required characteristic function. We discuss this problem in Section EC.3 of the e-companion.

For sake of generality, let us consider a generic random variable, Y, and denote its characteristic function according to $\varphi(u) = \mathrm{E}[e^{iuY}]$. The probability density function (pdf in the following) of Y can be computed as

$$f(y) = \frac{1}{2\pi}\int_{\mathcal{D}} e^{-iuy}\varphi(u)du, \tag{10}$$

where $\mathcal{D}$ is the domain of Y. Several algorithms can be used to solve the integral in Eq. (10), we refer to Fang and Oosterlee (2009) for a review. Among them, we adopt in this paper the COS method, where the inverse Fourier integral in Eq. (10) is computed via cosine expansion and the pdf is approximated as

$$f(y) = \sum_{k=1}^{\infty} F_k \cos\left(k\pi\frac{y-a}{b-a}\right) + \frac{1}{b-a} \approx \sum_{k=1}^{N-1} F_k \cos\left(k\pi\frac{y-a}{b-a}\right) + \frac{1}{b-a}, \tag{11}$$

where $F_k = \frac{2}{b-a}\mathrm{Real}\left(\varphi\left(\frac{k\pi}{b-a}\right)\cdot\exp\left(-i\frac{ka\pi}{b-a}\right)\right)$ and $[a,b] \in \mathcal{D}$ is chosen such that

$$\int_a^b e^{iuy}f(y)dy \approx \int_{\mathcal{D}} e^{iuy}f(y)dy. \tag{12}$$

In other words, in order to implement the COS method is necessary to truncate the domain of the pdf through a suitable choice of a and b. We will discuss this issue in Section 3.3. Moreover, the infinite sum in Eq. (11) must be truncated; we will investigate how to choose N efficiently in Section 5.

Next, we aim to perform random numbers generation. We can compute the cumulative distribution function (cdf) from Eq. (11):

$$c(y) \approx \int_a^y f(h)dh = \frac{y-a}{b-a} \sum_{k=1}^{N-1} F_k \frac{(b-a)\sin\left(\frac{\pi k(a-y)}{a-b}\right)}{\pi k}. \tag{13}$$

Then, we generate U uniformly over $[0,1]$ and find y such that $U = c(y)$, using root finding algorithms (e.g. the built–in Matlab® function fzero). We stress that F_k does not depend on y, allowing to compute in advance all the needed values of F_k before running the root finding algorithm. This facilitates implementation and increases the computational performances. The selection of the initial guess y_0 for $U = c(y)$ is discussed in Section 3.3.

3.3. Truncation range computation

We discuss, next, the choice of a and b in Eq. (12). We stress that this point is very important: if the interval $[a,b]$ is too wide, then one needs a larger N to get sufficient accuracy; if the interval is too tight, the approximation in Eq. (12) may be inaccurate (Fang and Oosterlee, 2009). Moreover, in our context we need to compute a different truncation range for any random realization of $\left(\int_0^T \sqrt{V_s}ds, V_T\right)$, hence a stable and efficient procedure is needed to obtain high accuracy with small computing effort. Since the needed quantities are defined on different domains, we propose two different methodologies for $\left(X_T | \int_0^T \sqrt{V_s}ds, V_T\right)$ and $\left(\int_0^T \sqrt{V_s}ds | V_T\right)$. These methodologies are based on the first two integer moments of $\left(\int_0^T V_s ds | V_T, \int_0^T \sqrt{V_s}ds\right)$ or $\left(\frac{1}{\int_0^T \sqrt{V_s}ds} | V_T\right)$. Hence, we first outline an efficient algorithm to compute such moments, then propose procedures for determining a and b.

3.3.1. Moments computation given a Laplace transform

Kyriakou et al. (2023) propose an efficient methodology for computing moments of some unknown distribution given only the knowledge of the Laplace transform. Suppose to be interested in computing the moments of an unknown distribution Y defined on $\mathbb{R}^+$. Given the knowledge of the analytical expression of its Laplace transform $\mathcal{L}(u) = \mathrm{E}[e^{-uY}]$, a first intuitive method is to proceed with analytical differentiation:

$$\mu_m = \mathrm{E}[Y^m] = (-1)^m \frac{\partial^m}{\partial u} \mathcal{L}(u)\Big|_{u=0}. \tag{14}$$

However, this method is not always applicable in practice since the Laplace transform may present very complicated expression (Kyriakou et al., 2023 for more details). To circumvent this problem we calculate the moments using the *numerical inversion of adaptively modified moment generating function* algorithm introduced by Choudhury and Lucantoni (1996). The authors show that μ_m can be computed according to

$$\mu_m = \frac{m!}{2ml r_m^m \alpha_m^m} \cdot \left(\mathcal{L}(\alpha_m r_m) + (-1)^m \mathcal{L}(-\alpha_m r_m) + 2 \sum_{j=1}^{ml-1} \mathrm{Real}\left(\mathcal{L}(\alpha_m r_m e^{\pi i j/nl}) e^{-\pi i j/l}\right) \right) - \hat{e}, \tag{15}$$

where $\hat{e}$ indicates the discretization error (the analytical expression can be found in the original paper) and the function r_m is chosen to bound the discretization error, in particular $r_m = 10^{-\frac{\bar{\gamma}}{2ml}}$ allows to achieve accuracy of order $10^{-\bar{\gamma}}$. We will adopt $\bar{\gamma} = 11$ following Choudhury and Lucantoni (1996) and Kyriakou et al. (2023). In Algorithm 1 we summarize the procedure to obtain the first 2 moments and to compute the variables l and α_m. This algorithm has been also successfully applied in other related studies such as Brignone (2022); Brignone et al. (2023).

Algorithm 1 Numerical inversion of adaptively modified moment generating function.

```
Input: $\bar{\gamma}$, $\mathcal{L}(\cdot)$
Output: $\{\mu_m\}_{m=1}^2$
Set $l = \alpha_1 = 1$ and compute $\mu_1$ according to Eq. (15)
Compute $\alpha_2 = \frac{1}{\mu_1}$ and $\mu_2$
Set $l = 1 \vee 2$ and $\alpha_1 = \alpha_2 = \frac{2\mu_1}{\mu_2}$ and compute new values for $\mu_1$ and $\mu_2$ from Eq. (15)
```

3.3.2. Domain truncation for $\left(X_T | \int_0^T \sqrt{V_s}ds, V_T\right)$

Since X_T is defined on $\mathbb{R}$, we follow Fang and Oosterlee (2009, Section 5.3), which suggest choosing a and b according to

$$a = \tilde{c}_1 - 12\sqrt{\tilde{c}_2}, \quad b = \tilde{c}_1 + 12\sqrt{\tilde{c}_2}, \tag{16}$$

where $\tilde{c}_j$ denotes the j-th cumulant of the risk–neutral distribution of log–returns. We show, next, how to apply Eq. (16) in our context, where we need to determine $\tilde{c}_j$. First, we define the conditional moments of X_T, $\tilde{\mu}_m := \mathrm{E}\left[X_T^m | \int_0^T \sqrt{V_s}ds, V_T\right]$. Then, by iterated expectations and using Eq. (14) we get

$$\tilde{\mu}_m = \mathrm{E}\left[\left((-1)^m \frac{\partial^m}{\partial u^m}\mathrm{E}\left[e^{-uX_T}\middle| V_T, \int_0^T \sqrt{V_s}ds, \int_0^T V_s ds\right]\Bigg|_{u=0}\right)\middle| \int_0^T \sqrt{V_s}ds, V_T\right],$$

where, from Eq. (6), the inner expectation is

$$\mathrm{E}\left[e^{-uX_T}\middle| V_T, \int_0^T \sqrt{V_s}ds, \int_0^T V_s ds\right] = e^{-u\left(rT + \frac{\rho}{v}\left(\sqrt{V_T}-\sqrt{V_0}-\frac{1}{2}(\eta-v^2)\int_0^T \sqrt{V_s}ds\right)\right)+u\left(\frac{1}{2}+\frac{1}{2}u(1-\rho^2)\right)\int_0^T V_s ds}.$$

Solving the derivative, taking expectation and defining $\mu_m^\star := \mathrm{E}\left[\left(\int_0^T V_s ds\right)^m \middle| \int_0^T \sqrt{V_s}ds, V_T\right]$, we get

$$\tilde{\mu}_1 = \gamma - \frac{1}{2}\mu_1^\star, \quad \tilde{\mu}_2 = (1-\rho^2)\mu_1^\star + \left(\gamma^2 - \gamma\mu_1^\star + \frac{1}{4}\mu_2^\star\right), \tag{17}$$

$$\tilde{c}_1 = \tilde{\mu}_1, \quad \tilde{c}_2 = \tilde{\mu}_2 - \tilde{\mu}_1^2, \tag{18}$$

where $\gamma = rT + \frac{\rho}{v}\left(-\frac{1}{2}\left(\eta - v^2\right)\int_0^T \sqrt{V_s}ds - \sqrt{V_0} + \sqrt{V_T}\right)$. Therefore, we have expressed the conditional cumulants of X_T in terms of the conditional moments of $\int_0^T V_s ds$, which can be computed easily given Eq. (8) through Algorithm 1. Finally, we recover the conditional moments and cumulants from Eqs. (17)–(18).

Given conditional moments, we can also find efficient starting points for root finding algorithms necessary to implement random numbers generation and sample $\left(X_T\middle|\int_0^T \sqrt{V_s}ds, V_T\right)$. More specifically, the initial guess for y in Eq. (13) is calculated using the inverse normal distribution function with the mean and standard deviation of the correct distribution.

3.3.3. Domain truncation for $\left(\int_0^T \sqrt{V_s}ds|V_T\right)$

Formula (16) is mainly suited for distributions defined on $\mathbb{R}$. Here we suggest a different approach for $\left(\int_0^T \sqrt{V_s}ds|V_T\right)$, which is obviously defined on $\mathbb{R}^+$. First of all, moments are easier to compute in this case, since we can apply directly Algorithm 1 to the Laplace transform in Eq. (7) and get the first two integer moments of $\left(\frac{1}{\int_0^T \sqrt{V_s}ds}|V_T\right)$, which we denote by $\hat{\mu}_m := \mathrm{E}\left[\left(\frac{1}{\int_0^T \sqrt{V_s}ds}\right)^m |V_T\right]$. Since $\sqrt{V_T}$ is a geometric Brownian motion, then its time integral has an unknown distribution. However, this distribution has received a lot of attention in the literature since it is important in the context of Asian option pricing. In particular, Milevsky and Posner (1998) show that the infinite sum of correlated lognormal random variables is distributed as a reciprocal gamma and, based on this, they suggest to approximate the integrated geometric Brownian motion through a reciprocal gamma random variable, whose shape and scale parameters are found by moment matching. Accordingly, we suggest to approximate $\left(\frac{1}{\int_0^T \sqrt{V_s}ds}\middle|V_T\right)$ with a moment-matched gamma distribution. In this way, we can approximate through simple formulas the extreme event probabilities and compute the truncation range in such a way that the error coming from Eq. (12) is negligible in practice. More specifically, let us define with $\mathcal{G}$ the moment-matched gamma distribution, i.e. $\mathcal{G}$ is a gamma random variable with shape parameter α and scale parameter β. We obtain α and β via moment matching:

$$\begin{cases} \hat{\mu}_1 = \alpha\beta \\ \hat{\mu}_2 - \hat{\mu}_1^2 = \alpha\beta^2 \end{cases} \Rightarrow \begin{cases} \alpha = \frac{\hat{\mu}_1^2}{\hat{\mu}_2 - \hat{\mu}_1^2} \\ \beta = \frac{\hat{\mu}_2 - \hat{\mu}_1^2}{\hat{\mu}_1} \end{cases}. \tag{19}$$

Hence, we choose a and b in Eq. (12) in such a way that the probability of $\mathcal{G} \leq a$ is 10^{-12} and probability that $\mathcal{G} \leq b$ is $1 - 10^{-12}$. As a result, a and b are found quickly from the inverse cdf of the gamma distribution. We investigate the accuracy of the proposed procedure in Section 5.

In the same way, we exploit the knowledge of the inverse cumulative distribution function of the moment-matched gamma distribution to select an efficient initial guess for y when sampling $\left(\frac{1}{\int_0^T \sqrt{V_s}ds}\middle|V_T\right)$ by solving numerically $U = c(y)$, with $c(y)$ as in Eq. (13).

3.4. Final algorithm

The proposed exact simulation algorithm can be summarized in three main steps:

- Step 1: Simulate V_T from Eq. (2);
- Step 2: Simulate $\left(\frac{1}{\int_0^T \sqrt{V_s}ds}\middle|V_T\right)$ and recover $\left(\int_0^T \sqrt{V_s}ds\middle|V_T\right)$ by taking the reciprocal;
- Step 3: Simulate $\left(X_T\middle|\int_0^T \sqrt{V_s}ds, V_T\right)$.

Algorithm 2 Exact simulation of the HW-SV model.

Input: $\{S_0, r, \eta, \sigma, \rho\}$ (Model parameters), T (maturity), $\mathcal{M}$ (number of simulations)

Output: $\left\{\left(S_T^{(i)}, V_T^{(i)}\right)\right\}_{i=1}^{\mathcal{M}}$

Notes: $\Phi(\cdot, \underline{\mu}, \underline{\sigma})$ and $G(\cdot, \alpha, \beta)$ denote respectively the cumulative distribution functions of a normal distribution with mean $\underline{\mu}$ and standard deviation $\underline{\sigma}$ and gamma distributions with parameters α and β

```
for $i = 1 : \mathcal{M}$ do
    Draw $\{U_j^{(i)}\}_{j=1}^{3}$ uniformly over $[0,1]$
    Compute $\mathcal{W} = \Phi^{-1}(U_1^{(i)}, 0, 1)\sqrt{T}$ and $V_T^{(i)} = V_0 \exp\left(\left(\eta - \frac{\sigma^2}{2}\right)T + \sigma \mathcal{W}\right)$
    Define $\mathcal{L}_1(\cdot)$ as in Eq. (7) and compute $\tilde{\mu}_1$ and $\tilde{\mu}_2$ using Algorithm 1
    Given $\hat{\mu}_1$ and $\hat{\mu}_2$ compute $\alpha$ and $\beta$ solving Eq. (19)
    Set $\epsilon = 10^{-12}$ and compute $a = G^{-1}(\epsilon, \alpha, \beta)$ and $b = G^{-1}(1-\epsilon, \alpha, \beta)$
    Compute initial guess $y_0 = G^{-1}(U_2^{(i)}, \alpha, \beta)$
    Define $\varphi_1(u) = \mathcal{L}_1(-\sqrt{-1}u)$, compute $\{F_k\}_{k=1}^{N-1}$ as in Eq. (11)
    Find $y$ such that $U_2^{(i)} = c(y)$, with $c$ as in Eq. (13) and set $(\int_0^T \sqrt{V_s}ds)^{(i)} = \frac{1}{y}$
    Given $\left((\int_0^T \sqrt{V_s}ds)^{(i)}, V_T^{(i)}\right)$, define $\mathcal{L}_2(\cdot)$ as in Eq. (7) and compute $\mu_1^\star$ and $\mu_2^\star$ using Algorithm 1
    Compute the conditional cumulants $\tilde{c}_1$ and $\tilde{c}_2$ of $X_T$ from Eqs. (17)–(18) and $[a,b]$ as in Eq. (16)
    Compute initial guess $y_0 = \Phi^{-1}(U^{(3)}, \tilde{c}_1, \sqrt{\tilde{c}_2})$
    Define $\varphi_3(u) := \mathcal{L}_3(-\sqrt{-1}u)$, with $\mathcal{L}_3(u)$ as in Eq. (9), compute $\{F_k\}_{k=1}^{N-1}$ as in Eq. (11)
    Find $y$ such that $U^{(3)} = c(y)$, with $c$ as in Eq. (13), set $X_T^{(i)} = y$ and $S_T^{(i)} = S_0 e^{X_T^{(i)}}$
end
```

Step 1 is very easy to implement, indeed, from Eq. (4) we know that V_T is log-normally distributed with known mean and variance. Hence, Step 1 is performed by means of simple standard normal random numbers generators.

For Step 2 we first compute the moments of $\left(\frac{1}{\int_0^T \sqrt{V_s}ds} | V_T\right)$ using Algorithm 1 and the conditional Laplace transform $\mathcal{L}_1(u)$ in Eq. (7). Given moments, we implement moments matching as in Eq. (19) and then compute a and b in Eq. (12) from the inverse cumulative distribution function of the moment-matched gamma distribution. Then, we compute $\{F_k\}_{k=1}^{N-1}$ using Eq. (7), we draw U as uniform on the interval $[0,1]$ and implement inverse transform sampling, where the equation $U = c(y)$ is solved numerically and the initial guess for y is found through the inverse cumulative distribution function of the moment-matched gamma distribution.

For Step 3 we first compute the moments of $\left(\int_0^T V_s ds \middle| \int_0^T \sqrt{V_s}ds, V_T\right)$ using $\mathcal{L}_2(u)$ and Algorithm 1, then we obtain conditional moments of X_T and related cumulants from Eqs. (17)–(18). We apply formula (16) for domain truncation and implement again inverse transform sampling to generate a random sample for $\left(X_T \middle| \int_0^T \sqrt{V_s}ds, V_T\right)$. More specifically, we compute $\{F_k\}_{k=1}^{N-1}$ using Eq. (9), we draw a new U as uniform on the interval $[0,1]$ and solve $U = c(y)$ numerically. The initial guess for y is found through the quantile function of a moment-matched normal distribution.

By implementing Steps 1–3 one gets a sample from $(S_T, V_T | S_0, V_0)$. The proposed exact simulation scheme is summarized in Algorithm 2.

We remark that, when the truncation range is computed accurately (e.g. as suggested in Formula (16)), we can rely on the theory of error control developed in Fang and Oosterlee (2009, Section 4). In particular, they show that if the truncation range is chosen as in Eq. (16), the truncation error is around 10^{-12} for $0.1 < T < 10$. The error is controlled by the parameter N. We will study the behavior of the error for varying N in Section 5.

3.5. *Computational issues*

We conclude this section by mentioning a possible computational issue with our method: the Laplace transform in Eq. (8) depends on the function $\theta(r,t)$. The problem of numerically computing $\theta(r,t)$ is related to evaluating the Hartman-Watson density:

$$f^{\text{H-W}}(r,t) = \frac{e^{\pi^2/(2t)}}{I_0(r)}\theta(r,t),$$

where $I_0(r)$ is the modified Bessel function of the first kind. As originally observed in Yor (1992), Cai et al., 2017 and the references therein, $f^{\text{H-W}}(r,t)$ is difficult to compute numerically when t tends to 0. Indeed, the component $e^{\pi^2/(2t)}$ outside the integral grows exponentially to infinity and multiplies an integral where the integrand function is highly oscillating due to the term $\sin\left(\frac{\pi\psi}{t}\right)$. To reduce numerical error to an acceptable level, high-precision computations must be employed (Boyle and Potapchik, 2006). However, the computational problem in our context is greatly reduced because the element $e^{\pi^2/(2t)}$ does not appear in our Laplace transform due to simplifications, facilitating the numerical evaluation. Nevertheless, numerical issues are still present due to the presence of $\sin\left(\frac{\pi\xi}{t}\right)$ in the integrand function when $t \ll 0.1$. However, we may wonder whether or not the situation with small t is likely to

Algorithm 3 Conditional COS for European option pricing in the HW-SV model.

Input: $\{S_0, r, \eta, \sigma, \rho\}$ (Model parameters), T (maturity), $\mathcal{M}$ (number of simulations), K (strike price)
Output: C (call option price)

```
for $i = 1 : \mathcal{M}$ do
    Draw $\{U_j^{(i)}\}_{j=1}^2$ uniformly over $[0, 1]$
    Repeat Steps 3–11 in Algorithm 2
    Define $\varphi_3(u) := \mathcal{L}_3(-\sqrt{-1}u)$, with $\mathcal{L}_3(u)$ as in Eq. (9) and compute $C^{(i)}$ from Eq. (20)
end
Compute the European call option price: $C = \frac{1}{\mathcal{M}} \sum_{i=1}^{\mathcal{M}} C^{(i)}$
```

appear in practice. To better understand this aspect, in Section EC.2 of the e-companion we estimate the parameter σ on time series of spot returns of four different assets. Note indeed that, from Eq. (8), θ depends on $t = \frac{\sigma^2 T}{16}$. We find that for the S&P 500 the parameter σ is approximately equal to 4. This means that $\theta(r, t)$ must be evaluated in $t \approx T$. As a result, we do not expect numerical issues unless $T \ll \frac{1}{10}$. Finally, if one is interested in the case where $T \ll 1/10$, alternative, more efficient numerical techniques, such as the Gaver-Stehfest or Bondesson Laplace inversion algorithms, which can increase the accuracy for small levels of t, are available in the literature. We refer to Section EC.4 of the e-companion for more details.

4. Variants of Algorithm 2 for variance reduction, option pricing and computation of the Greeks

We have so far focused on the problem of generating exact samples for the couple $(S_T, V_T | S_0, V_0)$. This is a very general approach that can be applied for many different purposes, for example, generating trajectories observed at discrete dates (useful for pricing path dependent options) or simulating extensions of the HW-SV models with jumps. However, this is not the primary scope of this research. Therefore, we relegate to Section EC.9 of the e-companion the pricing of path dependent derivatives. Adding Lévy jumps is straightforward and can be tackled exactly as in Broadie and Kaya (2006, Section 6). We focus instead on two fundamental applications: efficient pricing of European options and unbiased estimation of the Greeks. These are not trivial problems since in the HW-SV model the characteristic function of log-returns is unknown, making standard Fourier methods (Carr and Madan, 1999) unfeasible.

As discussed in the introductory section, no exact methodologies exist for pricing European options (and compute greeks) in the HW-SV model. We show that a simple variant of Algorithm 2 can be used to produce unbiased estimates of European option prices (and also Greeks). Indeed, if the purpose of simulation is to price European options, instead of simulating $\left(X_T | \int_0^T \sqrt{V_s} ds, V_T \right)$, we can compute directly the conditional option price, i.e. the option price conditional on the random realization of $\left(\int_0^T \sqrt{V_s} ds, V_T \right)$. In what follows, we focus on European call options, the case of put can be obtained via the standard put-call parity. Let us denote with $\Pi(S_T)$ the generic (discounted) payoff of a path-independent derivative instrument. The corresponding price is given by $\mathrm{E}[\Pi(S_T)]$ where expected value is taken with respect to the risk–neutral measure. For an European call option we have $\Pi(S_T) := e^{-rT} \max(0, S_T - K)$. Let us denote by $C := e^{-rT}\mathrm{E}[\max(0, S_T - K)]$ the price of the call option, assuming that $\log\left(\frac{K}{S_0}\right) > a$, from Eq. (11) we get

$$\begin{aligned} C \approx & \frac{e^{-rT}}{b-a} \int\limits_{\log\left(\frac{K}{S_0}\right)}^{b} (S_0 e^y - K) dy + e^{-rT} \sum_{k=1}^{N-1} F_k \int\limits_{\log\left(\frac{K}{S_0}\right)}^{b} (S_0 e^y - K) \cos\left(k\pi \frac{y-a}{b-a}\right) dy \\ = & \frac{e^{-rT}}{b-a} \left(-(b+1)K + e^b S_0 + K \log\left(\frac{K}{S_0}\right) \right) + \sum_{k=1}^{N-1} F_k \frac{e^{-rT} \left(-(b+1)K + e^b S_0 + K \log\left(\frac{K}{S_0}\right) \right)}{b-a} \\ & + (a-b) e^{-rT} \left(\frac{-e^b S_0((b-a)\cos(\pi k) + \pi k \sin(\pi k)) + K(b-a)\cos(\zeta) + \pi k K \sin(\zeta)}{(a-b)^2 + \pi^2 k^2} + \frac{K(\sin(\pi k) - \sin(\zeta))}{\pi k} \right), \end{aligned} \tag{20}$$

where $\zeta = \frac{\pi k \left(a - \log\left(\frac{K}{S_0}\right) \right)}{a-b}$ and F_k is as in Eq. (11) and thus depends on some characteristic function. If we plug φ_3 into Eq. (11) then C denotes the price of the European call option conditional on $\left(\int_0^T \sqrt{V_s} ds, V_T \right)$. Hence, as a first variant of Algorithm 2, we propose the simpler approach to sample first $\left(\int_0^T \sqrt{V_s} ds, V_T \right)$ using Steps 1 and 2 in Section 3.4 and then perform directly option pricing through Eq. (20). This approach (which we label *conditional COS*) presents an important advantage with respect to Algorithm 2: since we do not sample X_T, we remove the additional Monte Carlo variance due to Step 3. Therefore, conditional COS can be used as a variance reduction technique when the purpose is to price European options, allowing to obtain unbiased estimates with tight confidence interval at a small computational cost. In the course of our numerical studies, we will show that the variance of the Monte Carlo estimator is reduced by roughly 93-98%. We summarize this option pricing approach in Algorithm 3.

Algorithm 4 Option pricing in the HW-SV model through Conditional COS and Antithetic Sampling.

Input: $\{S_0, r, \eta, \sigma, \rho\}$ (Model parameters), T (maturity), $\mathcal{M}$ (number of simulations), K (strike price)
Output: C (call option price)

```
for $i = 1 : \mathcal{M}$ do
    Draw $\{U_j^{(i)}\}_{j=1}^2$ uniformly over $[0, 1]$
    Repeat Steps 3–11 in Algorithm 2
    Define $\varphi_3(u) := \mathcal{L}_3(-\sqrt{-1}u)$, with $\mathcal{L}_3(u)$ as in Eq. (9) and compute $C^{(i,1)}$ from Eq. (20)
    Compute $\bar{U}_1^{(i)} = 1 - U_1^{(i)}$ and $\bar{U}_2^{(i)} = 1 - U_2^{(i)}$
    Repeat Steps 3–11 in Algorithm 2 using $\bar{U}^{(i)}$ instead of $U^{(i)}$
    Define $\varphi_3(u) := \mathcal{L}_3(-\sqrt{-1}u)$, with $\mathcal{L}_3(u)$ as in Eq. (9) and compute $C^{(i,2)}$ from Eq. (20)
end
Compute $C_1 = \frac{1}{\mathcal{M}} \sum_{i=1}^{\mathcal{M}} C^{(i,1)}$ and $C_2 = \frac{1}{\mathcal{M}} \sum_{i=1}^{\mathcal{M}} C^{(i,2)}$
Compute the European call option price: $C = \frac{C_1 + C_2}{2}$
```

Algorithm 5 Greeks estimation in the HW-SV model: the case of $\Delta = \frac{\partial C}{\partial S_0}$.

Input: $\{S_0, r, \eta, \sigma, \rho\}$ (Model parameters), T (maturity), $\mathcal{M}$ (number of simulations), K (strike price)
Output: Δ (Delta of the call option price)

```
for $i = 1 : \mathcal{M}$ do
    Draw $\{U_j^{(i)}\}_{j=1}^2$ uniformly over $[0, 1]$
    Repeat Steps 3–11 in Algorithm 2
    Define $\varphi_3(u) := \mathcal{L}_3(-\sqrt{-1}u)$, with $\mathcal{L}_3(u)$ as in Eq. (9) and compute $\Delta^{(i)}$ from Eq. (21)
end
Compute the $\Delta$ of the call option price: $\Delta = \frac{1}{\mathcal{M}} \sum_{i=1}^{\mathcal{M}} \Delta^{(i)}$
```

Next, we propose another modification to Algorithms 2 and 3 to further reduce the variance of the Monte Carlo estimator. More specifically, we suggest the usage of antithetic sampling. Since this approach is well known (Glasserman, 2004), we simply illustrate how it can be adapted to our framework into Algorithm 4. In the numerical studies, we will show that antithetic sampling can produce extremely small standard errors and tight confidence intervals.

In addition, through our proposed approach, we also obtain unbiased estimates for the Greeks in the HW-SV model, a problem that is, to the best of our knowledge, unstudied in the literature. Exploiting Eq. (20), we can take the derivatives of the European call option price with respect to various parameters. For example, the Δ can be computed as

$$\Delta = \frac{\partial C}{\partial S_0} = \frac{e^{-rT}\left(e^b - \frac{K}{S_0}\right)}{b-a} + \sum_{k=1}^{N-1} F_k(a-b)e^{-rT} \times \left(\frac{\frac{\pi^2 k^2 K\cos(\zeta)}{S_0(a-b)} - \frac{\pi k K(b-a)\sin(\zeta)}{S_0(a-b)} - e^b((b-a)\cos(\pi k) + \pi k\sin(\pi k))}{(a-b)^2 + \pi^2 k^2} - \frac{K\cos(\zeta)}{S_0(a-b)} \right). \tag{21}$$

This is just an example, other Greeks can be also estimated through our approach. Formulas for the elasticity and higher order sensitivities are reported in Section EC.5 of the e-companion. The scheme for computing unbiased estimates for the Δ in the HW-SV model is summarized into Algorithm 5.

5. Numerical studies

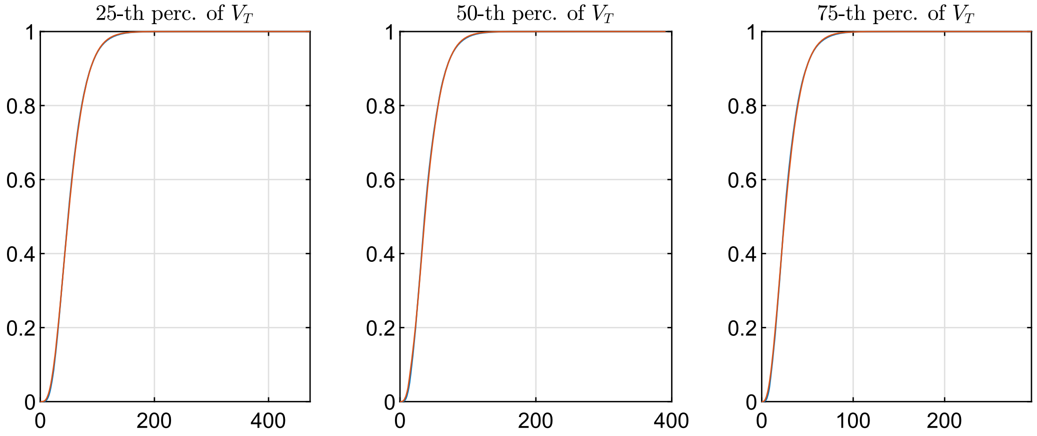
In this section we discuss performances of the various simulation algorithms described above. All the computations are done using Matlab® (Version R2022a) in Microsoft Windows 10® running on a machine equipped with Intel(R) Core(TM) i7-9750HQ CPU @2.60 GHz and 16 GB of RAM. We start by identifying the sources of error implicit in our approach and how to deal with them. Then, we compare our proposed simulation schemes with a benchmark. The typical benchmark in the literature summarized in Table 1 is the classic Euler scheme. However, due to the high values of the parameter σ (see below), it presents very poor performances in the HW-SV model. Indeed, despite the variance process follows a geometric Brownian motion, it can reach negative values when using Euler approximation. Therefore, we consider a different (more competitive) benchmark, which we label "low-bias" scheme (in analogy with Chen et al., 2012). In order to save space, we relegate to Section EC.6 of the e-companion the description of the benchmark methodology, as well as the variants allowing for variance reduction and greeks computation, the numerical study of bias convergence rate and the optimal choice of the number of time discretization steps.

For all the experiments in this section we consider four different parameter sets, reported in Table 2. These parameters are chosen similar to those estimated on real data in Section EC.2 of the e-companion.

Table 2
Parameter settings for testing the accuracy of the exact simulation methods along with the price of the corresponding at the money European call option price.

	V_0	η	σ	ρ	S_0	r	T	true price	Underlying
Set A	0.01	0.1	4	-0.6	100	0.02	1	3.5515	S&P 500
Set B	0.01	0.2	4.5	-0.7	100	0.02	1	3.3464	Eurostoxx 50
Set C	0.04	0.2	2	-0.2	100	0.02	1	8.0361	USO
Set D	0.01	0.15	3	-0.5	100	0.02	1	4.0743	FTSE 100

Notes. USO denotes the "United States Oil" fund. See Section EC.2 of the e-companion for more details.



Notes. Cumulative distribution functions of $\left(\frac{1}{\int_0^T \sqrt{V_s}ds}\middle|V_T\right)$ (blue line) and the moment–matched gamma distribution (red line) for three different values of V_T. x–axis is truncated at a and b computed as in Section 3.3. Parameters as in Set A in Table 2.

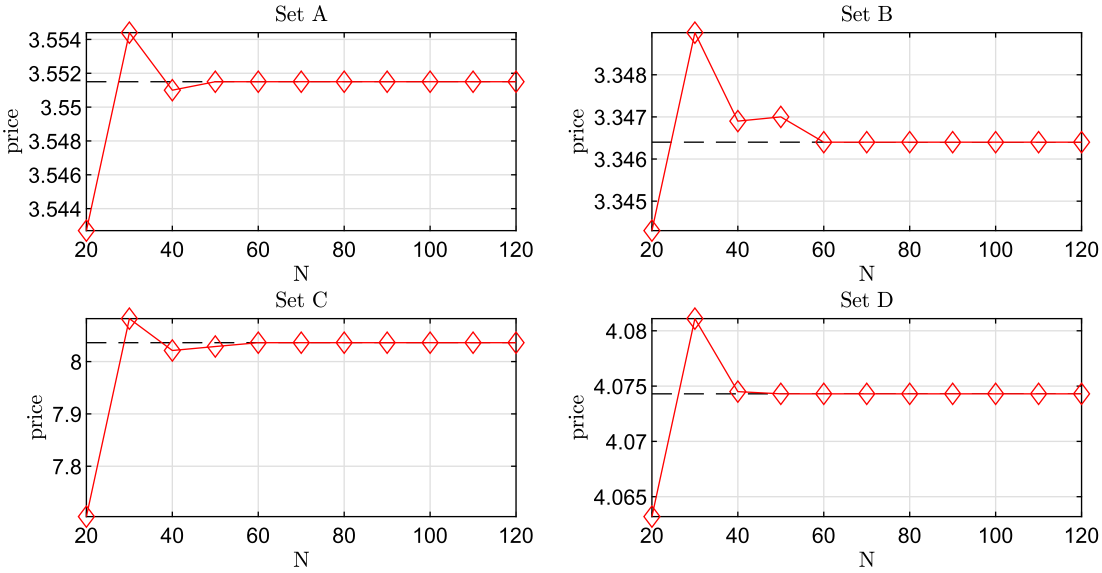
Fig. 1. Truncation range and cumulative distribution function of $\left(\frac{1}{\int_0^T \sqrt{V_s}ds}\middle|V_T\right)$. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

5.1. Error control

We start by discussing the sources of error implicit in our proposed algorithms and how to deal with them. Indeed, exact simulation schemes are unbiased only in theory but not in practice (Broadie and Kaya, 2006, Section 3.2 and Li and Wu, 2019, Section 3.2 and Table 1) due to the necessity of using numerical techniques for inverting the various Laplace transforms involved and performing inverse transform sampling given the numerical estimate of the cdf. This fact is implicit in all the simulation schemes proposed in literature for stochastic volatility models (see Table 1) and is reflected also in our simulation algorithms. However, since we consider a different technique (with respect to existing literature) to invert the Laplace transform, it is of interest to study in more detail the accuracy and the computational efficiency of our proposed methodology. We identify two sources of error in the approach outlined in Section 3: i) truncation of the actual domain as in Eq. (12); ii) truncation of infinite sums in Eq. (11) at the N-th element. We discuss, next, how to control the error implicit in our method.

To illustrate the accuracy in the computation of the truncation range we present the following experiment. We consider three different realizations of V_T corresponding to the 25-th, 50-th and 75-th percentile of its distribution. Then, we compute the true cumulative distribution function of $\left(\frac{1}{\int_0^T \sqrt{V_s}ds}\middle|V_T\right)$ and the moment–matched gamma approximation. The two cdfs are reported in Fig. 1 for Set A. We can see that the difference is very small, especially in the tails of the distributions. Moreover, x–axis in the figure are truncated exactly at a and b computed as in Section 3.3. As a result, through the inverse cumulative distribution function of the moment–matched gamma we obtain a good approximation for small tail probability of $\left(\frac{1}{\int_0^T \sqrt{V_s}ds}\middle|V_T\right)$, entailing high accuracy when implementing the approximation in Eq. (16). Furthermore, the closeness of the two densities ensures that the initial guess for implementing root finding algorithms needed for inverse transform sampling is quite close to the final solution, implying that only few iterations are necessary to solve the equation. The robustness of accuracy with respect to parameter choice, as well as the closeness of the initial guesses to the final solution, are illustrated in Section EC.7 of the e-companion. Similar results are available for the truncation range of the domain of $\left(X_T\middle|\int_0^T \sqrt{V_s}ds, V_T\right)$, however, since we are applying the same formula suggested in Fang and Oosterlee (2009), we do not provide them here in the interest of saving space.

Regarding the choice of N in Eq. (11), we perform the following experiment: we simulate 10^9 random realizations of $\left(\int_0^T \sqrt{V_s}ds, V_T\right)$ and, for each couple, we compute the conditional option price using Eq. (20) for various N. Then, we take the



Notes. The price is computed using 10^9 simulations, black horizontal lines represent the true option price as reported in Table 2.

Fig. 2. European call option price computed through Eq. (20) for various N.

average across simulations obtaining an unbiased estimate of the European call option price. We plot in Fig. 2 the option prices computed in this way for various N and for various sets of parameters. We see that the option price converges to the true value quickly. The results of this experiment agree with Fang and Oosterlee (2009, Figure 2) showing that the sum in Eq. (11) presents exponential convergence. In general, we have found across different parameter sets that by setting $N = 70$ the error is negligible and smaller than the fourth significant decimal place when pricing options.

5.2. *Performances of Algorithms 2–5*

First of all, we need to identify a benchmark and quantify its accuracy. We use a “low-bias” simulation scheme which is described in Section EC.6 of the e-companion. To evaluate its accuracy we follow standard literature (e.g. Broadie and Kaya, 2006; Glasserman and Kim, 2011; Cai et al., 2017; Kang et al., 2017; Li and Wu, 2019; Kyriakou et al., 2023). If C is the simulation estimator used for the European call option price and C_{true} is the true price, then the bias and the variance of the estimator are

$$\text{bias} = (\mathrm{E}[C] - C_{\text{true}}), \quad \text{variance} = \mathrm{E}[(C_{\text{true}} - \mathrm{E}[C])^2].$$

The Root Mean Squared Error (RMSE) is then defined as

$$\text{RMSE} = \sqrt{\text{bias}^2 + \frac{\text{variance}}{\mathcal{M}}},$$

where $\mathcal{M}$ denotes the number of simulations. The bias for the benchmark discretization scheme with a specific number of time steps can be estimated using a very large number of simulations to estimate $\mathrm{E}[C]$ and then taking difference with the true price (similar reasoning applies for the computation of the variance). We compute the bias for various numbers of time discretization steps, i.e. $n = \{138, 241, 420, 731, 1273\}$ steps using 10^9 simulations. These are chosen following the optimal allocation rule suggested in Duffie and Glynn (1995) (see Section EC.6 of the e-companion for more details). Results are reported in Table 3. The true option prices of at the money European call options (which are computed through Algorithm 4 with 5×10^8 simulations) are reported for each parameter sets in Table 2. We conventionally set the bias of our exact scheme to 0 as in Broadie and Kaya (2006); Cai et al. (2017); Kang et al. (2017); Li and Wu (2019); Bernis et al. (2021).

We start by examining the performance of Algorithm 2 and compare with the low-bias scheme in terms of RMSE and running time in seconds. Results are reported in Table 3. The bias of the low-bias scheme obviously decreases with the number of time discretization steps. In addition to Table 3, we also report the same results in log–log scale in Fig. 3. It is clear that our exact simulation outperforms the benchmark, presenting a steeper downward slope than the low-bias scheme and, as a consequence, a faster convergence rate of the RMSE.

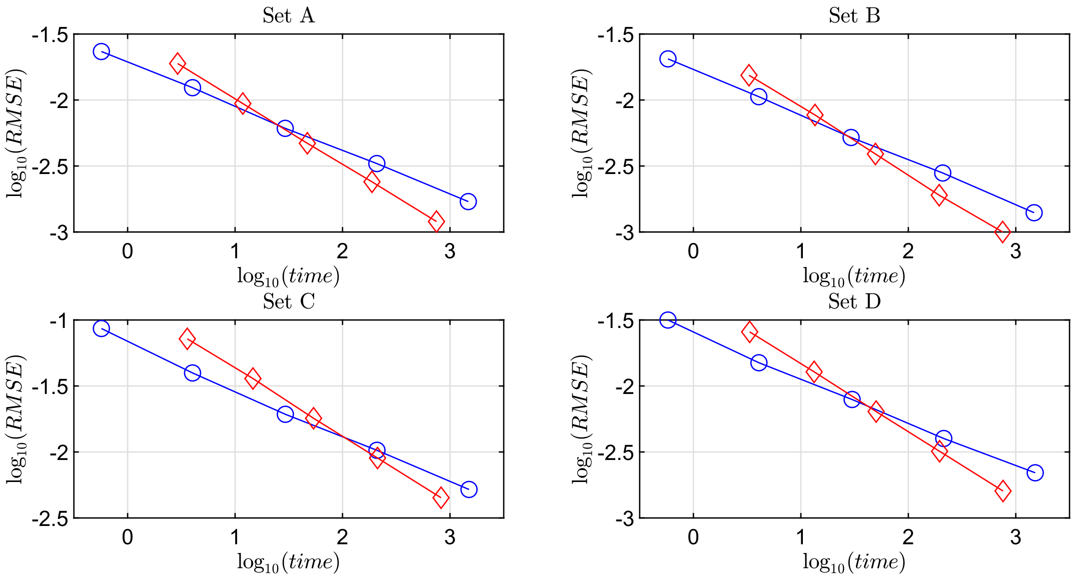
Having tested the simulation scheme, we now focus on the usage of variance reduction techniques for pricing European options. We start with Algorithm 3. In order to have a comparable benchmark, we follow Willard (1997) and compute the option price by simulating paths only for the variance process. Then, given the discrete path of variance and volatility, we compute the integrals with the trapezoidal rule and compute option price through the conditional Black–Scholes formula (see Section EC.6 of the e-companion). This benchmark is more suitable for comparisons than the crude low-bias scheme because, exactly as Algorithm 3, it eliminates the

Table 3
Speed-accuracy profiles of Algorithm 2 (plain exact simulation scheme) and the low-bias scheme for parameter sets in Table 2: the case of European plain vanilla option.

	Set A						Set B					
	low-bias				Algorithm 2		low-bias				Algorithm 2	
$\mathcal{M}$	n	bias	RMSE	time	RMSE	time	n	bias	RMSE	time	RMSE	time
4×10^4	138	0.0140	0.0233	0.57	0.0189	2.92	138	0.0135	0.0205	0.58	0.0154	3.29
16×10^4	241	0.0077	0.0124	4.02	0.0094	11.80	241	0.0072	0.0106	4.06	0.0077	13.54
64×10^4	420	0.0039	0.0061	29.28	0.0047	47.05	420	0.0036	0.0052	29.33	0.0039	49.42
256×10^4	731	0.0023	0.0033	208.48	0.0024	187.89	731	0.0020	0.0028	209.04	0.0019	193.81
1024×10^4	1273	0.0012	0.0017	1480.37	0.0012	748.47	1273	0.0010	0.0014	1481.58	0.0010	754.76

	Set C						Set D					
	low-bias				Algorithm 2		low-bias				Algorithm 2	
$\mathcal{M}$	n	bias	RMSE	time	RMSE	time	n	bias	RMSE	time	RMSE	time
4×10^4	138	0.0297	0.0862	0.57	0.0720	3.59	138	0.0162	0.0316	0.58	0.0257	3.34
16×10^4	241	0.0158	0.0397	4.02	0.0360	14.67	241	0.0087	0.0150	4.07	0.0128	13.29
64×10^4	420	0.0076	0.0193	29.37	0.0180	53.76	420	0.0047	0.0079	29.99	0.0064	50.17
256×10^4	731	0.0050	0.0103	209.11	0.0090	211.65	731	0.0026	0.0040	213.67	0.0032	195.44
1024×10^4	1273	0.0025	0.0052	1503.21	0.0045	825.72	1273	0.0014	0.0022	1516.83	0.0016	761.09

Notes. All computing times are expressed in seconds. The choice of n for the low-bias scheme is discussed in Section EC.6 of the e-companion.



Notes. Algorithm 2: plots with red diamond markers; benchmarks: plots with blue circle markers. All computing times are expressed in seconds.

Fig. 3. Speed-accuracy comparisons of Algorithm 2 (plain exact simulation scheme) and competent benchmark (low-bias scheme) for different parameter sets: the case of European plain vanilla option.

variance associated to the simulation of the asset price. Numerical results are reported in Table 4 and (in log–log scale) in Fig. 4. The superiority of Algorithm 3 with respect to the benchmark is evident with a faster convergence rate of the RMSE. Monte Carlo variance is reduced by approximately 93-98% (see also the discussion below). Further, we observe a reduction of the running time because we can directly price the option without simulating $\left(X_T|\int_0^T \sqrt{V_s}ds, V_T\right)$. For example, for 1024×10^4 simulations the reduction is approximately 24%.

Next, we aim to assess the performances of Algorithm 4. Since it employs antithetic sampling, as benchmark we consider in this case the low-bias scheme with antithetic sampling and conditional Black–Scholes formula. Numerical results are reported in Table 5 and Fig. 5. Also in this case, Algorithm 4 presents a faster convergence rate of the RMSE and outperforms the benchmark. Algorithm 4 is slower than Algorithm 3 since we have to repeat some steps twice. However, the variance is drastically reduced, as we discuss later on.

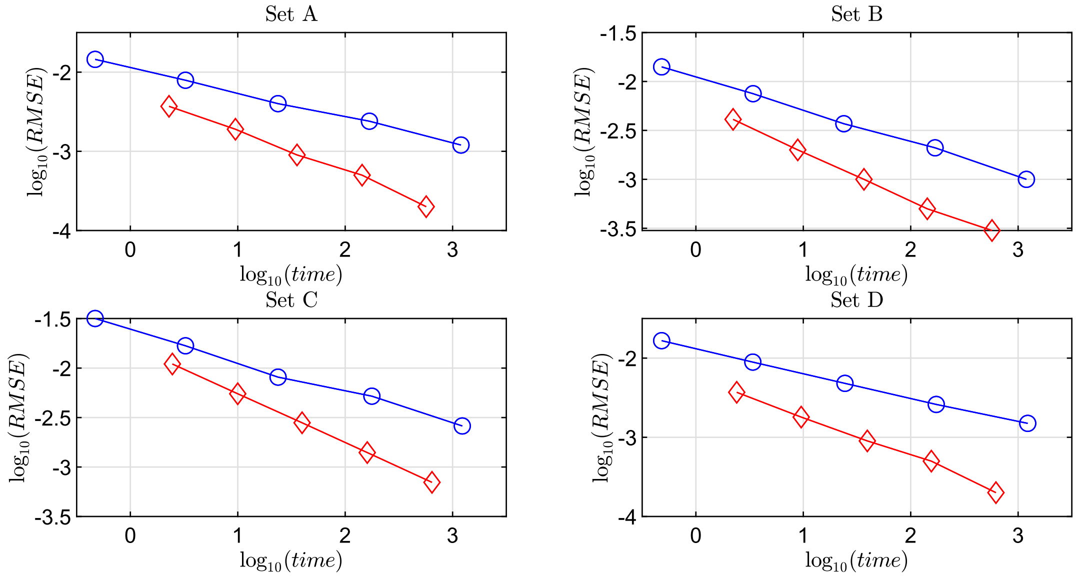
To better illustrate the variance reduction when implementing Algorithms 3 and 4, we perform the following experiment: we run the various algorithms for $\mathcal{M} = 1024\times10^4$ simulations, then we compute the variance of the Monte Carlo estimator for at the money

Table 4
Speed-accuracy profiles of Algorithm 3 (conditional COS) and the low-bias scheme (with conditional Black-Scholes formula) for parameter sets in Table 2: the case of European plain vanilla option.

	Set A						Set B					
	low-bias				Algorithm 3		low-bias				Algorithm 3	
$\mathcal{M}$	n	bias	RMSE	time	RMSE	time	n	bias	RMSE	time	RMSE	time
4×10^4	138	0.0140	0.0145	0.47	0.0037	2.29	138	0.0135	0.0141	0.48	0.0041	2.22
16×10^4	241	0.0077	0.0079	3.26	0.0019	9.51	241	0.0072	0.0075	3.41	0.0020	8.88
64×10^4	420	0.0039	0.0040	23.69	0.0009	35.62	420	0.0036	0.0037	23.86	0.0010	36.71
256×10^4	731	0.0023	0.0024	168.17	0.0005	143.74	731	0.0020	0.0021	168.63	0.0005	142.74
1024×10^4	1273	0.0012	0.0012	1197.60	0.0002	567.88	1273	0.0010	0.0010	1195.31	0.0003	571.44

	Set C						Set D					
	low-bias				Algorithm 2		low-bias				Algorithm 2	
$\mathcal{M}$	n	bias	RMSE	time	RMSE	time	n	bias	RMSE	time	RMSE	time
4×10^4	138	0.0297	0.0317	0.47	0.0110	2.46	138	0.0162	0.0166	0.48	0.0037	2.40
16×10^4	241	0.0158	0.0168	3.26	0.0055	9.99	241	0.0087	0.0089	3.39	0.0018	9.56
64×10^4	420	0.0076	0.0081	23.72	0.0028	39.70	420	0.0047	0.0048	24.43	0.0009	39.46
256×10^4	731	0.0050	0.0052	177.29	0.0014	160.51	731	0.0026	0.0026	173.02	0.0005	155.66
1024×10^4	1273	0.0025	0.0026	1231.46	0.0007	643.04	1273	0.0014	0.0015	1229.25	0.0002	621.72

Notes. All computing times are expressed in seconds. The choice of n for the low-bias scheme is discussed in Section EC.6 of the e-companion.



Notes. Algorithm 3: plots with red diamond markers; benchmarks: plots with blue circle markers. All computing times are expressed in seconds.

Fig. 4. Speed-accuracy comparison of Algorithm 3 (conditional COS) and competent benchmark (low-bias scheme with conditional Black-Scholes formula) for different parameter sets: the case of European plain vanilla option.

European call option price. In Table 6 we report the Monte Carlo variance obtained in this way and the reduction in % with respect to Algorithm 2. Results show that Algorithm 3 allows for a reduction of the variance around 93 – 98% overall, while Algorithm 4 by more than 98-99%. As a result, both algorithms provide viable ways for obtaining unbiased estimates of European call option prices under the HW-SV model with tight confidence intervals.

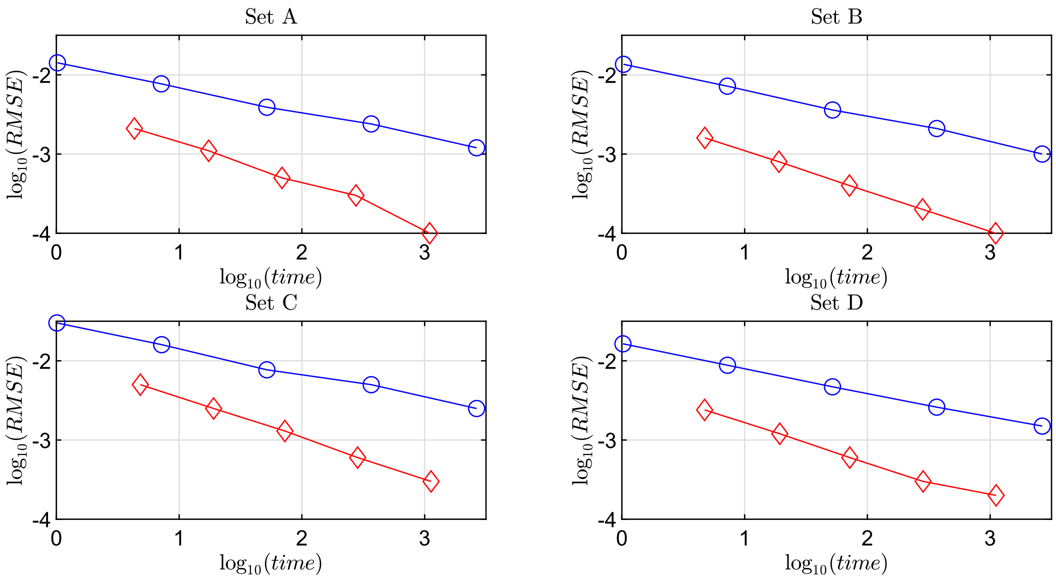
We conclude by illustrating the performances of Algorithm 5 for the unbiased estimation of the Δ. Following Willard (1997) and Broadie and Kaya (2006), as benchmark we implement the low-bias discretization, we compute integrated variance and volatility using the trapezoidal rule and then we obtain the Δ through a conditional Black-Scholes formula (see Section EC.6 of the e-companion). Results are reported in Table 7 and Fig. 6 and show that, as expected, Algorithm 5 presents a faster convergence rate and outperforms the benchmark.

Table 5
Speed-accuracy profiles of Algorithm 4 (conditional COS with antithetic sampling) and the low-bias scheme (with conditional Black-Scholes formula and antithetic sampling) for parameter sets in Table 2: the case of European plain vanilla option.

	Set A						Set B					
	low-bias				Algorithm 4		low-bias				Algorithm 4	
$\mathcal{M}$	n	bias	RMSE	time	RMSE	time	n	bias	RMSE	time	RMSE	time
4×10^4	138	0.0140	0.0142	1.02	0.0021	4.32	138	0.0135	0.0136	1.03	0.0016	4.74
16×10^4	241	0.0077	0.0077	7.16	0.0011	17.41	241	0.0072	0.0072	7.23	0.0008	19.02
64×10^4	420	0.0039	0.0039	51.84	0.0005	68.88	420	0.0036	0.0036	52.03	0.0004	71.40
256×10^4	731	0.0023	0.0024	367.09	0.0003	275.90	731	0.0020	0.0021	367.32	0.0002	281.67
1024×10^4	1273	0.0012	0.0012	2652.36	0.0001	1098.85	1273	0.0010	0.0010	2655.38	0.0001	1108.84

	Set C						Set D					
	low-bias				Algorithm 5		low-bias				Algorithm 5	
$\mathcal{M}$	n	bias	RMSE	time	RMSE	time	n	bias	RMSE	time	RMSE	time
4×10^4	138	0.0297	0.0301	1.01	0.0050	4.83	138	0.0162	0.0164	1.02	0.0024	4.72
16×10^4	241	0.0158	0.0160	7.21	0.0025	19.10	241	0.0087	0.0088	7.25	0.0012	19.33
64×10^4	420	0.0076	0.0077	51.89	0.0013	72.72	420	0.0047	0.0047	51.94	0.0006	71.88
256×10^4	731	0.0050	0.0050	367.30	0.0006	284.60	731	0.0026	0.0026	367.31	0.0003	284.56
1024×10^4	1273	0.0025	0.0025	2653.58	0.0003	1127.98	1273	0.0014	0.0015	2656.59	0.0002	1118.70

Notes. All computing times are expressed in seconds. The choice of n for the low-bias scheme is discussed in Section EC.6 of the e-companion.



Notes. Algorithm 4: plots with red diamond markers; benchmarks: plots with blue circle markers. All computing times are expressed in seconds.

Fig. 5. Speed-accuracy comparisons of Algorithm 4 (conditional COS with antithetic sampling) and competent benchmark (low-bias scheme with conditional Black-Scholes formula and antithetic sampling) for different parameter sets: the case of European plain vanilla option.

Table 6
Variance (and reduction in %) of the Monte Carlo estimators for at the money European call option price using Algorithms 2– 4 in the various parameter sets.

	Set A	Set B	Set C	Set D
Algorithm 2	14.2279	9.5146	207.2956	26.3971
Algorithm 3	0.5556 (-96.09%)	0.6585 (-93.08%)	4.8603 (-97.66%)	0.5466 (-97.93%)
Algorithm 4	0.2200 (-98.45%)	0.1768 (-98.14%)	1.0079 (-99.51%)	0.2336 (-99.12%)

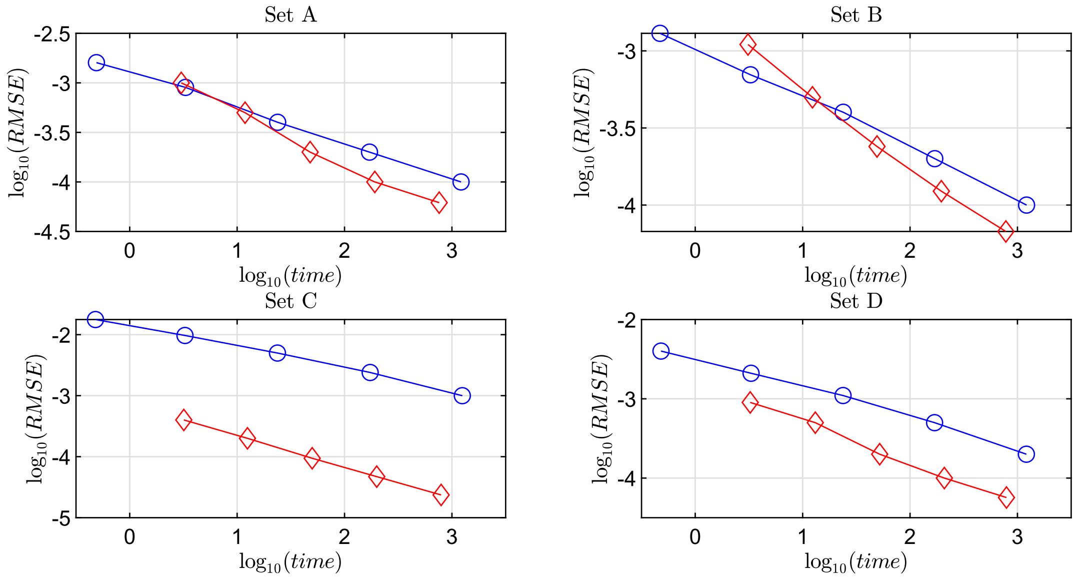
Notes. Elements between parenthesis denote the variance reduction with respect to Algorithm 2.

Table 7
Speed-accuracy profiles of Algorithm 5 and the low-bias scheme for parameter sets in Table 2: the case of the Δ of the European plain vanilla option.

	Set A						Set B					
	low-bias				Algorithm 5		low-bias				Algorithm 5	
$\mathcal{M}$	n	bias	RMSE	time	RMSE	time	n	bias	RMSE	time	RMSE	time
4×10^4	138	0.0013	0.0016	0.49	0.0010	3.02	138	0.0008	0.0013	0.47	0.0011	3.10
16×10^4	241	0.0007	0.0009	3.31	0.0005	11.86	241	0.0004	0.0007	3.27	0.0005	12.34
64×10^4	420	0.0003	0.0004	23.85	0.0002	47.85	420	0.0002	0.0004	23.88	0.0003	49.25
256×10^4	731	0.0002	0.0002	170.85	0.0001	191.43	731	0.0002	0.0002	169.87	0.0001	195.56
1024×10^4	1273	0.0001	0.0001	1216.00	6.18E-05	761.04	1273	0.0001	0.0001	1215.80	6.74E-05	779.82

	Set C						Set D					
	low-bias				Algorithm 5		low-bias				Algorithm 5	
$\mathcal{M}$	n	bias	RMSE	time	RMSE	time	n	bias	RMSE	time	RMSE	time
4×10^4	138	0.0177	0.0177	0.48	0.0004	3.19	138	0.0039	0.0040	0.48	0.0009	3.25
16×10^4	241	0.0097	0.0097	3.27	0.0002	12.48	241	0.0021	0.0021	3.31	0.0005	13.12
64×10^4	420	0.0050	0.0050	23.75	9.42E-05	49.99	420	0.0011	0.0011	23.77	0.0002	52.16
256×10^4	731	0.0024	0.0024	173.29	4.71E-05	199.44	731	0.0005	0.0005	169.25	0.0001	208.58
1024×10^4	1273	0.0010	0.0010	1249.36	2.36E-05	793.12	1273	0.0002	0.0002	1211.00	5.68E-05	790.10

Notes. Elements between parenthesis denote the variance reduction with respect to Algorithm 2. The choice of n for the low-bias scheme is discussed in Section EC.6 of the e-companion.



Notes. Algorithm 5: plots with red diamond markers; benchmarks: plots with blue circle markers. All computing times are expressed in seconds.

Fig. 6. Speed-accuracy comparisons of Algorithm 5 and competent benchmark for different parameter sets: the case of the Δ of plain vanilla option.

6. Conclusions

In this paper, we develop an exact simulation scheme for the Hull and White (1987) stochastic volatility model. This contribution extends a large stream of literature concerned with the exact simulation of stochastic volatility models. More specifically, we first derive expressions for some relevant Laplace transforms, and then we perform random sampling by the inverse transform method, where the numerical estimate of the cumulative distribution function is obtained via the Fourier-cosine (COS) method. We propose efficient methodologies for computing the truncation range for the domain of the various random variables involved based on their conditional moments. This is important for the practical implementation of the suggested exact simulation scheme since it allows for a theoretical control of the error based on the results in Fang and Oosterlee (2009, Sections 4 and 5.1). We propose some variants of the exact simulation scheme that allow for the computation of unbiased estimates of European option prices and their sensitivities. In addition, these variants can be used to reduce the computational time (in the case of Algorithm 3) and the variance of the Monte Carlo estimator (for example, our Algorithm 4 allows for a variance reduction of around 99% with respect to the standard exact simulation scheme when pricing European options). Therefore, our methodologies can be used to obtain unbiased estimates of the European call option price and sensitivities with tight confidence intervals. In this sense, we significantly extend the stream of literature concerned with European option pricing under the HW-SV model, for which only approximations have been proposed so far (Heston and Rossi, 2017; Ackerer and Filipović, 2020; Zeng et al., 2023). We test the proposed algorithms throughout various realistic parameter sets calibrated on real data and compare their performances with the low-bias (and variants) discretization scheme. The results are striking: the proposed approaches present a faster convergence rate of the root mean squared error and outperform the benchmarks.

Appendix A. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.jedc.2024.104861.

References

Abate, J., Whitt, W., 1992. The Fourier-series method for inverting transforms of probability distributions. Queueing Syst. 10, 5–88.

Abate, J., Whitt, W., 1995. Numerical inversion of Laplace transforms of probability distributions. ORSA J. Comput. 7, 36–43.

Ackerer, D., Filipović, D., 2020. Option pricing with orthogonal polynomial expansions. Math. Finance 30, 47–84.

Baldeaux, J., 2012. Exact simulation of the 3/2 model. Int. J. Theor. Appl. Finance 15, 1250032.

Bernis, G., Brignone, R., Scotti, S., Sgarra, C., 2021. A Gamma Ornstein-Uhlenbeck model driven by a Hawkes process. Math. Financ. Econ. 15, 747–773.

Boyle, P., Potapchik, A., 2006. Application of high-precision computing for pricing arithmetic Asian options. In: Proceedings of the 2006 International Symposium on Symbolic and Algebraic Computation, pp. 39–46.

Brignone, R., 2022. Moments of integrated exponential Lévy processes and applications to Asian options pricing. Quant. Finance 22, 1717–1729.

Brignone, R., Gonzato, L., Sgarra, C., 2023. Commodity Asian option pricing and simulation in a 4-factor model with jump clusters. Ann. Oper. Res.

Broadie, M., Kaya, O., 2006. Exact simulation of stochastic volatility and other affine jump diffusion processes. Oper. Res. 54, 217–231.

Cai, N., Song, Y., Chen, N., 2017. Exact simulation of the SABR model. Oper. Res. 65, 931–951.

Carr, P., Madan, D., 1999. Option valuation using the fast Fourier transform. J. Comput. Finance 2, 61–73.

Chen, B., Oosterlee, C.W., van der Heide, H., 2012. A low-bias simulation scheme for the SABR stochastic volatility model. Int. J. Theor. Appl. Finance 15, 125–161.

Choudhury, G.L., Lucantoni, D.M., 1996. Numerical computation of the moments of a probability distribution from its transform. Oper. Res. 44, 368–381.

Christoffersen, P., Dorion, C., Jacobs, K., Wang, Y., 2010a. Volatility components, affine restrictions, and nonnormal innovations. J. Bus. Econ. Stat. 28, 483–502.

Christoffersen, P., Jacobs, K., Mimouni, K., 2010b. Volatility dynamics for the S&P500: evidence from realized volatility, daily returns, and option prices. Rev. Financ. Stud. 23, 3141–3189.

Duan, J., 1995. The GARCH option pricing model. Math. Finance 5, 13–32.

Duffie, D., Glynn, P., 1995. Efficient Monte Carlo estimation of security prices. Ann. Appl. Probab. 4, 897–9058.

Fang, F., Oosterlee, C., 2009. A novel pricing method for European options based on Fourier-cosine series expansions. SIAM J. Sci. Comput. 31, 826–848.

Fulop, A., Li, J., 2019. Bayesian estimation of dynamic asset pricing models with informative observations. J. Econom. 209, 114–138.

Glasserman, P., 2004. Monte Carlo Methods in Financial Engineering. Stochastic Modelling and Applied Probability. Springer, New York.

Glasserman, P., Kim, K.K., 2011. Gamma expansion of the Heston stochastic volatility model. Finance Stoch. 15, 267–296.

Grasselli, M., 2017. The 4/2 stochastic volatility model: a unified approach for the Heston and the 3/2 model. Math. Finance 27, 1013–1034.

Heston, S.L., Rossi, A.G., 2017. A spanning series approach to options. Rev. Asset Pricing Stud. 7, 2–42.

Hsieh, K., Ritchken, P., 2005. An empirical comparison of GARCH option pricing models. Rev. Deriv. Res. 8, 129–150.

Hull, J., White, A., 1987. The pricing of options on assets with stochastic volatilities. J. Finance 42, 281–300.

Junike, G., Pankrashkin, K., 2022. Precise option pricing by the COS method - how to choose the truncation range. Appl. Math. Comput. 421, 126935.

Kang, C., Kang, W., Lee, J., 2017. Exact simulation of the Wishart multidimensional stochastic volatility model. Oper. Res. 65, 1190–1206.

Kyriakou, I., Brignone, R., Fusai, G., 2023. Unified moment–based modelling of integrated stochastic processes. Oper. Res.

Li, C., Wu, L., 2019. Exact simulation of the Ornstein-Uhlenbeck driven stochastic volatility model. Eur. J. Oper. Res. 275, 768–779.

Matsumoto, H., Yor, M., 2005. Exponential functionals of Brownian motion 1: probability laws at fixed time. Probab. Surv. 2, 312–347.

Milevsky, M.A., Posner, S.E., 1998. Asian options, the sum of lognormals and the reciprocal gamma distribution. J. Financ. Quant. Anal. 33, 409–422.

Parrish, R.S., 1987. Evaluation and approximation of multivariate cumulative joint probabilities. J. Stat. Comput. Simul. 27, 1–33.

Parrish, R.S., 1990. Generating random deviates from multivariate Pearson distributions. Comput. Stat. Data Anal. 9, 283–295.

Willard, G.A., 1997. Calculating prices and sensitivities for path-independent derivative securities in multifactor models. J. Deriv. 5, 54–61.

Yor, M., 1992. On some exponential functionals of Brownian motion. Adv. Appl. Probab. 24, 509–531.

Zeng, P., Xu, Z., Jiang, P., Kwok, Y.K., 2023. Analytical solvability and exact simulation in models with affine stochastic volatility and Lévy jumps. Math. Finance 33, 842–890.

References from supplementary material

Abate, J., Whitt, W., 2006. A unified framework for numerically inverting Laplace transforms. INFORMS J. Comput. 18, 408–421.

Andersen, L., 2008. Simple and efficient simulation of the Heston stochastic volatility model. J. Comput. Finance 11, 1–42.

Bernhart, P., Mai, J.-F., 2015. A note on the numerical evaluation of the Hartman-Watson density and distribution function. In: Glau, K., Scherer, M., Zagst, R. (Eds.), Innovations in Quantitative Risk Management. Springer International Publishing, pp. 337–345.

Doetsch, G., 1974. Introduction to the Theory and Application of the Laplace Transformation. Springer, New York.

Doucet, A., Johansen, A.M., 2008. A tutorial on particle filtering and smoothing: fifteen years later. In: Handbook of Nonlinear Filtering, vol. 12, pp. 656–704.

Dufays, A., Jacobs, K., Liu, Y., Rombouts, J., 2023. Fast filtering with large option panels: implications for asset pricing. J. Financ. Quant. Anal., 1–56.

Ignatieva, K., Wong, P., 2022. Modelling high frequency crude oil dynamics using affine and non-affine jump–diffusion models. Energy Econ. 108, 105873.

Johannes, M.S., Polson, N.G., Stroud, J.R., 2009. Optimal filtering of jump diffusions: extracting latent states from asset prices. Rev. Financ. Stud. 22, 2559–2599.

Kienitz, J., Wetterau, D., 2012. Financial Modelling - Theory, 2 edn. Implementation and Practice with Matlab. The Wiley Finance Series. John Wiley & Sons, Chichester, West Sussex.

Malik, S., Pitt, M.K., 2011. Particle filters for continuous likelihood evaluation and maximisation. J. Econom. 165, 190–209.

Ponnusamy, S., Silverman, H., 2006. Complex Variables with Applications. Birkhäuser, Boston.

Rouah, F.D., 2013. The Heston Model and Its Extensions in Matlab and C#, 2 edn. The Wiley Finance Series. John Wiley & Sons, Hoboken, New Jersey.

Ruan, X., Zhang, J., 2018. Risk-neutral moments in the crude oil market. Energy Econ. 72, 583–600.

Widder, D.V., 1941. The Laplace Transform. Princeton University Press, Princeton, N.J.