Contents lists available at ScienceDirect

European Journal of Operational Research

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

Interfaces with Other Disciplines

Exact simulation of stochastic volatility models based on conditional Fourier-cosine method

Riccardo Brignone [a], [*], Gero Junike [b]

[a] Department of Economics and Management, University of Pavia, Via San Felice 5, 27100 Pavia, Italy
[b] Institut für Mathematik, Carl von Ossietzky Universität, 26129 Oldenburg, Germany

ARTICLE INFO

JEL classification:
C15
C32
C63
G13

Keywords:
Simulation
Characteristic function inversion
COS method
Error control
Option pricing
Stochastic volatility

ABSTRACT

The traditional methodology used for the exact simulation of stochastic volatility models based on the Gil–Pelaez formula presents implementation problems that are observed by many researchers and practitioners. In particular, although conventionally considered exact, such a method presents a difficult control of the error. The bias of the Monte Carlo simulation estimator can only be computed numerically and is controlled by two parameters, typically determined by running time-consuming simulations under different tuning parameter configurations until an optimal setup is found. In this paper, we propose a new exact simulation scheme based on the Fourier-cosine method, which approximates a probability density given the characteristic function as follows: the density is truncated on a finite interval, and approximated by a classical Fourier-cosine series. The method allows full error control via an effective automatic identification of the tuning parameters given a user-supplied error tolerance. The new approach offers the following advantages: improved control of the error, simplified implementation, and reduction in computing time. The error is controlled by only one parameter instead of two. This parameter has a clear interpretation: it is the maximum tolerable bias. This facilitates the implementation, since the maximum bias becomes an input of the simulation algorithm, instead of an output, and can be set *a priori*, before running simulations. Our analysis shows that the proposed exact simulation scheme is computationally faster than the traditional one, and presents an improved speed-accuracy profile with respect to alternative state-of-the-art fast approximated sampling schemes.

1. Introduction

Exact simulation of stochastic volatility models is a standard and important topic at the frontier of finance and operations research (see Fabozzi et al., 2025, Section 2.4). Broadie and Kaya (2006) first showed that the full transition (variance and log-returns) can be simulated exactly under the Heston model. Since their seminal paper, authors proposed exact simulation schemes for alternative option pricing models like the 3/2 (Baldeaux, 2012), 4/2 (Grasselli, 2017), SABR (Cai et al., 2017), Wishart (Kang et al., 2017), Ornstein–Uhlenbeck (Brignone, 2024; Li & Wu, 2019), Hull and White (Brignone & Gonzato, 2024) stochastic volatility models. In addition, Zeng et al. (2023) proposed a general framework for exact simulation and showed that also time changed Lévy models with stochastic volatility can be simulated exactly.

Developing exact simulation schemes is very important for several reasons summarized next (we refer to Broadie & Kaya, 2006 for more details and additional motivations). When using simulation schemes based on time discretization (e.g., Euler), the bias depends on the number of time steps and cannot be quantified *a priori*. Since the bias is unknown, the standard error may be a poor estimate of the actual error, causing serious problems when computing the prices of derivative securities. Moreover, the convergence rate for exact simulation schemes is $O(s^{-1/2})$, where s is the total computational budget, against the $O(s^{-1/3})$ convergence of the Euler scheme. It is worth noting that some error is introduced in the exact simulation since it requires numerical inversion of characteristic functions. Thus, the term "exact" may sound misleading. However, following standard literature (e.g., Broadie & Kaya, 2006; Kang et al., 2017; Li & Wu, 2019), we still refer to such methodology, as well as our new proposed methodology, as exact.

Despite their theoretical appeal, exact simulation schemes for stochastic volatility models suffer from several drawbacks that have limited their use in practice. The main problem is that for each simulation one has to simulate a random variable (typically, the log-returns conditional on the terminal variance level) given only the knowledge of its characteristic function. This is not trivial and is done by inverting the characteristic function

[*] Corresponding author.
E-mail addresses: riccardo.brignone@unipv.it (R. Brignone), gero.junike@uol.de (G. Junike).

https://doi.org/10.1016/j.ejor.2025.08.061
Received 4 July 2025; Accepted 30 August 2025
Available online 13 September 2025

to the cumulative distribution function based on the Poisson summation formula by Abate and Whitt (1992) and generating a random sample by inverse transform method, implemented via root-finding algorithms. First, the method becomes relatively slow since the numerical inversion must be implemented for each simulation. Moreover, the conditional characteristic function must be evaluated during the root finding algorithm. This dramatically increases computing times, especially in those models where the characteristic function is computationally expensive (e.g., because it is given in terms of special functions). Second, despite conventionally considered "exact", the method leads to two distinct sources of error: (i) discretization error, due to the necessity of computing fast and accurately some integrals with highly oscillating integrands; (ii) truncation error, due to the necessity of truncating the infinite sum involved in the Poisson summation formula. Additionally, it is not possible to establish how the truncation and discretization errors contribute to the bias of the Monte Carlo simulation estimator. Consider, for example, the problem of pricing a European option, how do discretization and truncation errors impact the option price obtained via exact simulation? It is not easy to answer this question. The only way is to compute the option price via exact simulation for different values of the parameters controlling the truncation and discretization errors and compare with a true reference price (which could be potentially unknown). We refer to Kang et al. (2017, Tables 3 and 4) and Li and Wu (2019, Table 1) for concrete examples. The authors compute the approximation error of "exact" sampling schemes under different models and parameter settings and for different values of the parameters controlling the error. They find that (i) the approximation error can be very large; (ii) the optimal choice of the tuning parameters is very sensitive to the change of the model parameters. Note that this procedure to assess the true bias when computing option prices using exact simulation is very computationally expensive because one must run the algorithms many times with a huge number of simulations to find the right setup. Additionally, they can estimate the approximation error because under such a model the price of the European option can be computed via semi-analytical methods at high accuracy. This may not be possible for different models or different kinds of derivative.

The scope of this paper is to develop a new exact simulation scheme for stochastic volatility models that overcomes the limitations of traditional exact simulation schemes, being faster and presenting improved error control. This is done by exploiting results in computational mathematics and statistics that were not available when the original paper by Broadie and Kaya (2006) was written. First, we invert the relevant characteristic function by the Fourier-cosine (COS) method (Fang & Oosterlee, 2009). The idea of the COS method is very simple, it approximates the probability density in three steps: (i) the density is truncated at a finite interval; (ii) The truncated density is approximated by a classical cosine series; (iii) the cosine coefficients are approximated based on the available characteristic function. Note that also Brignone (2024) and Brignone and Gonzato (2024) use the COS method for exact simulation. However, a notorious problem of the COS method is that the error is controlled by the domain truncation range and the truncation of an infinite sum. The truncation range is typically selected through the usage of the cumulants, and can be also considered negligible in practice, while the parameter, denoted by N, that truncates the infinite sums involved must still be determined by trial and error (see Fang & Oosterlee, 2009, Figure 2, Brignone, 2024, Figure 2, Brignone & Gonzato, 2024, Figure 2). Naturally, this is not very efficient because one should run the algorithm several times for various N until finding sufficient accuracy. In this paper, we consider a completely different approach still based on the COS method. We adapt (and significantly extend) the results in Junike and Pankrashkin (2022) and Junike (2024) to the context of exact sampling given the knowledge of conditional characteristic functions (conditional COS). To implement such an approach, it is fundamental to compute high-order moments (usually, the sixth or eighth moment) of the log-returns (conditionally on some other key quantities, typically, the terminal variance, as outlined below). This is far from trivial. To this end, in this paper we develop an efficient methodology for such purpose, extending Kyriakou et al. (2024, Algorithm 1). Finally, we perform random sampling using the inverse transform method as in traditional exact simulation schemes using root-finding algorithms. To speed up the computations, we propose a new methodology for selecting the starting points given to the solver. These are found through the quantile function of a three parameters metalog distribution (Keelin, 2016) whose parameters are found by moment-matching. The idea is to provide to the solver starting points very close to the solution, minimizing the number of iterations and saving computing time.

So, what are the merits of the proposed approach? (i) The error is controlled by only one parameter instead of two as in the traditional exact simulation method. In addition, suppose that one simulates the model with the purpose of pricing a derivative instrument. Through our approach the parameter controlling the error has a clear economic interpretation: it represents the maximum error tolerable when pricing an option. This is an evident advantage of our proposed algorithm over the traditional exact method, where it is not clear how truncation and discretization errors impact the accuracy of the Monte Carlo simulation estimator. Therefore, with our approach, we eliminate the necessity of "playing" with parameters controlling the error, which can be numerically expensive (see Kang et al., 2017, Tables 3 and 4, Li & Wu, 2019, Table 1, for clear examples). When pricing options, the final user can simply input the maximum possible distance between the (potentially unknown) option price and the one computed via our exact simulation scheme. Then, the algorithm automatically selects truncation range and number of terms in such a way that the error is smaller than the one desired by the final user, a great improvement with respect to the methods proposed in Brignone (2024) and Brignone and Gonzato (2024), where the latter is found by trial and error. Additionally, through our approach, due to the interpretability of the parameter controlling the error, by applying the central limit theorem it is possible to compute the minimal number of simulations such that the error is below the selected tolerable bias, this task is unattainable with the traditional exact simulation scheme. In summary, the first merit is an improved control of the error. (ii) The simulation algorithm is expected to be faster than the traditional one. This is for two main reasons. First, in the new algorithm it is possible to compute the characteristic function before implementing the root finding algorithm for random sampling. This is not attainable via the traditional exact scheme (unless introducing additional parameters controlling the error), entailing a huge computational saving. Second, we suggest using the metalog to find starting points for root finding algorithm. In this way, we obtain starting points that are closer to the final solution limiting the number of iterations needed for the root finding algorithms to converge, with an obvious computational saving. (iii) Our approach entails an elegant theoretical error control also for some kind of path-dependent derivatives, such as forward starting options, as discussed below.

Besides the new simulation algorithm, this paper presents some additional contributions. First, we present the proposed methodology for the model framework developed by Zeng et al. (2023), allowing to simulate the Heston, 3/2, 4/2 models and time changed Lévy versions. However, the methodology can be enhanced in the special cases of the Heston and time changed Lévy models. We show how to adapt the results in Choi and Kwok (2024) to enhance the simulation algorithm exploiting Poisson conditioning. Second, we show how to simulate exactly multifactor stochastic volatility models, such as the Heston $\bigoplus$ 3/2 model (Grasselli, 2017) and the Multi-Heston model (Christoffersen et al., 2009; Recchioni et al., 2021), for which we propose exact simulation schemes. Third, we develop a new effective methodology for pricing forward starting options. With this approach, the user can select a probability q and an error tolerance ϵ. Then, our algorithm provides a forward starting option price that does not differ by more than ϵ from the true option price with probability q. To the best of our knowledge, this elegant and direct control of the error is unattainable through other methods available in the literature.

We compare our new exact simulation scheme against the state-of-the-art fast approximated sampling schemes proposed by Kyriakou et al. (2024) and Zeng et al. (2023). Numerical results show that the new approach largely outperforms the method of Kyriakou et al. (2024) when simulating time changed Lévy, 4/2 and Heston $\bigoplus$ 3/2 models. Besides the numerical performances, the main advantage of our newly proposed approach is that it allows for an improved control of the error and can be more easily extended to multifactor models. With respect to the method of Zeng et al. (2023), our method is expected to be slower for the same number of simulations (especially when it is high), but it presents the following advantages: (i) greatly simplified/improved control of the error and simplified interpretation of the output when pricing forward starting options (see the discussion above); (ii) better extendability to multifactor models. We compare numerical performances of the two competing methods under the 3/2 and 4/2 models under challenging model parameters. We show that our method presents a better speed/accuracy trade-off when a high accuracy is required and a high number of simulations is employed thanks to the fact that the bias is smaller.

The remainder of the paper is organized as follows. In Section 2 we recap the traditional exact simulation scheme and its main limitations. In Section 3 we present our conditional Fourier-cosine approach for Monte Carlo simulation. In Section 4 we extend our framework to the pricing of derivative instruments (e.g., forward starting options). In Section 5 we show how to apply the methodology to simulate exactly stochastic volatility models. In Section 6 we present our numerical studies. Section 7 concludes. Supplementary results are deferred to the e-companion.

2. Traditional exact simulation scheme and related computational issues

We recap the traditional approach for simulating exactly stochastic volatility models based on the Fourier inversion method of Abate and Whitt (1992). In each previous paper (Baldeaux, 2012; Broadie & Kaya, 2006; Kang et al., 2017; Li & Wu, 2019), the methodology is presented with small differences. The method we illustrate in this section can be seen as a generalization of their approaches and it is used as benchmark for our new proposed approach later on. After describing the traditional simulation approach, we briefly recap the main issues associated with exact simulation which have limited its usage in practice. In what follows, we denote by X_T (respectively, V_T) the log-returns (variance) at time T.

2.1. Traditional exact simulation scheme

Analytically solvable (in the sense of Zeng et al., 2023) models can be simulated "exactly" in two steps: (i) Simulate $(V_T|V_0)$; (ii) Simulate $(X_T|V_T, V_0)$. The first step is straightforward since the transition density of the terminal variance is typically known. The second is much more involved, and it is based on the numerical inversion of a conditional characteristic function exploiting classical Fourier inversion, see Abate and Whitt (1992), Gil-Pelaez (1951) and Hughett (1998). We point out that, depending on specific model characteristics, authors present different approaches to simulate $(X_T|V_T)$. In the case of the Heston model, Broadie and Kaya (2006) first simulate $\left(\int_0^T V_s ds|V_T\right)$ and then $\left(X_T|\int_0^T V_s ds, V_T\right)$ exploiting the fact that the latter is normally distributed. Similarly, for the 3/2 model, Baldeaux (2012) simulates $\left(\int_0^T \frac{ds}{V_s}|V_T\right)$ and then $\left(X_T|\int_0^T \frac{ds}{V_s}, V_T\right)$. In the case of the 4/2 model, Grasselli (2017) suggests sampling directly $(X_T|V_T)$. In general, it is more convenient to sample directly $(X_T|V_T)$ if possible, as highlighted in Zeng et al. (2023). In what follows, we briefly outline a unified methodology for sampling $(X_T|V_T)$ based on the Fourier inversion algorithm proposed in Abate and Whitt (1992). This will be used as benchmark for our new proposed methodology. Define $Y := \left(X_T|V_T\right)$, $i := \sqrt{-1}$ and $\tilde{g}(u) := \frac{\mathrm{Real}(\mathcal{F}_Y(u))}{u}(\sin(-ul_e) + \sin(uy)) + \frac{\mathrm{Imag}(\mathcal{F}_Y(u))}{u}(\cos(-ul_e) - \cos(uy))$, where l_e truncates the domain of Y and $\mathcal{F}_Y(u) = \mathbb{E}\left[e^{iuX_T}|V_T\right]$ is the conditional characteristic function of the log-returns. The cumulative distribution function of Y is approximated as (cfr. Abate & Whitt, 1992, Eq. 5.14)

$$F_Y(y) = F_Y(l_e) + \frac{h(y-l_e)}{2\pi} + \frac{h}{\pi}\sum_{j=1}^{N} \tilde{g}(hj) - e_D(h) - e_T(N), \tag{1}$$

where $e_D(h)$ is the discretization error resulting from the usage of the trapezoidal rule when computing the integral and $e_T(N)$ denotes the truncation error caused by truncating the infinite sum at the Nth term. Finally, to simulate the value of Y, it is typical to use the inverse transform method. One generates a uniform random variable u over $[0, 1]$ and then find the value of y for which $F_Y(y)$, approximated from (1), is equal to u using root finding algorithms. Initial points are selected using the quantile function of the two moments matched normal distribution. To implement the methodology, it is thus necessary to specify the step-size h and number of terms N, that impact accuracy and numerical efficiency. This is discussed next.

2.2. Control of the error in the traditional exact simulation scheme

It is not clear *a priori* how the parameters h and N impact the accuracy of the Monte Carlo simulation estimator. According to Broadie and Kaya (2006), one can get h and N that achieve the desired accuracy by trial and error, but this is not optimal. Indeed, through extensive numerical experiments (Kang et al., 2017) found that: (i) many trials are required to find the optimal choice of h and N (this can be very time consuming); (ii) the optimal choice of h and N is very sensitive to changes of the model parameters; (iii) it is not said that, as one would expect, when increasing h and N also the accuracy increases (Kang et al., 2017, Tables 3 and 4). It is therefore evident that is very hard to find the optimal setup and that, despite the methodology is conventionally considered exact, it is very difficult to control the error in practice. An alternative approach to control the error suggested by Broadie and Kaya (2006) and adopted by subsequent researchers is the following. To control the discretization error it is customary to set h as

$$h = \frac{2\pi}{u_e - l_e}, \quad l_e = \mu_1 - m\sqrt{\mu_2 - \mu_1^2}, \quad u_e = \mu_1 + m\sqrt{\mu_2 - \mu_1^2}, \tag{2}$$

where $\mu_n = \mathbb{E}\left[X_T^n|V_T\right]$. Moreover, to control the truncation error, following Abate and Whitt (1992), they suggest selecting the minimum N such that $|\mathcal{F}_{X_T|V_T}(hN)| < \frac{\pi\hat{\xi}}{4}$, where $\hat{\xi}$ is the desired truncation error. Note, however, that such method still requires that the optimal m and $\hat{\xi}$ must be found again by trial error (see Li & Wu, 2019, Table 1 for a clear example). In summary, despite theoretically elegant and bias-free by construction, the traditional exact simulation scheme is slow to run and biased in practice. The error is controlled by two parameters: m and $\hat{\xi}$ (or h and N). It is not possible to know in advance which combination of m and $\hat{\xi}$ (or h and N) allows for a bias in the Monte Carlo simulation estimator smaller than a given tolerance. This can be done by trial and error but it is very time consuming because one must run a huge number of simulations to estimate accurately the approximation error. We refer to Kang et al. (2017, Tables 3 and 4) and Li and Wu (2019, Table 1) for clear numerical illustrations. In addition, the optimal setup can change drastically when considering different model parameter sets. The scope of this paper is to propose an alternative exact simulation scheme that surmounts such limitations.

3. Conditional COS for Monte Carlo simulation

In this section, we present our conditional COS approach for exact simulation. We start by introducing theoretical results. Then, we illustrate the usage of the metalog distribution to reduce computing times. Finally, we discuss the inputs of the algorithm and recap the methodology.

3.1. Conditional COS

Let $(\Omega, \mathcal{F}, \mathbb{Q})$ be a probability space. Let (E, Σ) be a measurable space. $\mathbb{Q}$ is a probability measure. Since our main application in this paper is the pricing of derivatives, $\mathbb{Q}$ will denote the risk-neutral measure. However, this is not a strict requirement. Let $X : \Omega \to \mathbb{R}$ and $Z : \Omega \to E$ be random variables. Assume that X is continuously distributed and integrable. Let $z \in E$. We aim to develop a new methodology to simulate the couple (X, Z) provided the conditional characteristic function of X given $Z = z$ is given in closed form. By $\mathcal{B}(\mathbb{R})$, we denote the Borel σ-algebra on $\mathbb{R}$. For a function $g : \mathbb{R} \to \mathbb{R}$, we define $\|g\|_\infty := \sup_{x\in\mathbb{R}} |g(x)|$. It is well known that there exists a Markov kernel, i.e., a map $K : \mathcal{B}(\mathbb{R}) \times E \to [0, 1]$ such that (a) $K(A, z) = \mathbb{E}[1_{\{X\in A\}}|Z = z]$ for all $A \in \mathcal{B}(\mathbb{R})$ and all $z \in E$; (b) $A \mapsto K(A, z)$ is a probability measure on the measure space $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$ for all $z \in E$; (c) $z \mapsto K(A, z)$ is measurable for all $A \in \mathcal{B}(\mathbb{R})$. The map $G_{X|Z=z}(y) := K((-\infty, y], z)$, $y \in \mathbb{R}$, $z \in E$, is equal to the conditional cumulative distribution function (CCDF) of X given $Z = z$ for each $y \in \mathbb{R}$ and $z \in E$. We will approximate $G_{X|Z=z}$ numerically by the COS method under the assumption that there exists a *conditional kernel density function*, i.e. a measurable map $g_{X|Z=z}$ with $K(A, z) = \int_A g_{X|Z=z}(x)dx$. We do not assume that $x \mapsto g_{X|Z=z}(x)$ is known precisely, we only need that

$$\hat{g}_{X|Z=z}(u) := \int_{\mathbb{R}} e^{iux} K(dx, z) = \int_{\mathbb{R}} e^{iux} g_{X|Z=z}(x)dx, \quad u \in \mathbb{R}, \quad z \in E \tag{3}$$

called *conditional characteristic function* (CCF) *of* X *given* $Z = z$, is given explicitly. Note that $\hat{g}_{X|Z=z}(u)$ and the CCF $u \mapsto \mathbb{E}[e^{iuX}|Z = z]$ as defined in Yuan and Lei (2016) are equal. By Eq. (3), the map $u \mapsto \hat{g}_{X|Z=z}(u)$ is the Fourier-transform, also called *characteristic function* of the Lebesgue-density $x \mapsto g_{X|Z=z}(x)$ for each fixed z.

We apply the COS method to approximate $g_{X|Z=z}$ by a cosine expansion. Then, we integrate the cosine expansion to estimate $G_{X|Z=z}(y) = \int_{(-\infty,y]} g_{X|Z=z}(x)dx$ given $Z = z$. We summarize the main steps: For $z \in E$, let $\mu := -i\frac{\partial}{\partial u}\, \hat{g}_{X|Z=z}(u)\Big|_{u=0}$. Define $f_z(x) := g_{X|Z=z}(x + \mu)$, which has the Fourier transform $\hat{f}_z(u) = \hat{g}_{X|Z=z}(u)e^{-iu\mu}$ and the first moment of f_z is zero, i.e., $\int_{\mathbb{R}} x f_z(x)dx = 0$ (Junike & Stier, 2023, Proposition 3.2). That is, the main mass of f is centered around zero and we truncate f on the interval $[-L, L]$. We approximate the truncated density by a Fourier-cosine expansion (the symbol $\sum{}'$ denotes that the first addend of the sum is weighted by a half):

$$f_z(x) \approx \sum_{k=0}^{N}{}' a_k^z \cos\left(k\pi\frac{x+L}{2L}\right) 1_{[-L,L]}(x) \approx \sum_{k=0}^{N}{}' \tilde{c}_k^z \cos\left(k\pi\frac{x+L}{2L}\right) 1_{[-L,L]}(x) =: \overline{f}_z(x), \tag{4}$$

where $L > 0$, $N \in \mathbb{N}$ and

$$a_k^z = \frac{1}{L}\int_{-L}^{L} f_z(x)\cos\left(k\pi\frac{x+L}{2L}\right)dx \approx \frac{1}{L}\int_{\mathbb{R}} f_z(x)\cos\left(k\pi\frac{x+L}{2L}\right)dx = \frac{1}{L}\Re\left\{\hat{f}_z\left(\frac{k\pi}{2L}\right)e^{i\frac{k\pi}{2}}\right\} =: \tilde{c}_k^z. \tag{5}$$

The coefficients a_k^z are the cosine coefficients of the truncated density and are approximated by $\tilde{c}_k^z$. We suppress the dependency of L, N and μ on z. The following lemma shows that the probability density function f_z can approximated arbitrarily closely by $\overline{f}_z$.

Lemma 1. *Assume f_z is $J + 1$ times continuously differentiable for some $J \geq 1$ and has semi-heavy tails, i.e., for some constants C_1, C_2, L_0 it holds that*

$$|f_z^{(j)}(x)| \leq C_1 C_2^j e^{-C_2|x|}, \quad j = 0, 1, \ldots, J+1, \quad |x| \geq L_0.$$

Then for every $\eta > 0$ there is a $L > 0$ and a $N \in \mathbb{N}$ such that

$$\sup_{x\in\mathbb{R}} |f_z(x) - \overline{f}_z(x)| < \eta. \tag{6}$$

Proof. See Section EC.1.1. ■

Motivated by Lemma 1, we will assume that $\overline{f}_z$ is a density function. Using the cosine approximation $\overline{f}_z$ of f_z, we get

$$G_{X|Z=z}(y) = \int_{\mathbb{R}} 1_{(-\infty,y]}(x + \mu)f_z(x)dx \approx \sum_{k=0}^{N}{}' \tilde{c}_k^z v_k^z(y) =: \overline{G}_z(y) \tag{7}$$

where $v_k^z(y) := 0$ if $y - \mu < -L$ and otherwise

$$v_k^z(y) := \int_{-L}^{L} 1_{(-\infty,y]}(x + \mu)\cos\left(k\pi\frac{x+L}{2L}\right)dx \ = \begin{cases} \min(y-\mu, L) + L & \text{for } k = 0 \\ \frac{2L}{\pi k}\left(\sin\left(k\pi\frac{\min(y-\mu,L)+L}{2L}\right)\right) & \text{for } k > 0. \end{cases} \tag{8}$$

The COS method converges, see Junike (2024) and Junike and Pankrashkin (2022). The next theorem tells us how to choose the number of terms N and the truncation range $[-L, L]$. Let $\mu_n(z)$ be the n^{th}–moment of f_z, i.e., $\mu_n(z) = \int_{\mathbb{R}} x^n f_z(x)dx = \frac{1}{i^n}\ \frac{\partial^n}{\partial u^n}\hat{f}_z(u)\Big|_{u=0}$. Let $\varepsilon > 0$ be some error tolerance. Define for $n \in \mathbb{N}$ even

$$L(\varepsilon, n, m) := \sqrt[n]{\frac{2\mu_n(z)m}{\varepsilon}}, \quad m > 0 \tag{9}$$

and for some $s \in \mathbb{N}$ odd

$$N(\varepsilon, s, m) := \text{Floor}\left(\left(\frac{2^{s+\frac{5}{2}} B(z)}{s\pi^{s+1}} \frac{L^{s+2} 12m}{\varepsilon} \right)^{\frac{1}{s}} \right) + 1, \quad m > 0, \tag{10}$$

where Floor() is the floor function and $B(z) := \frac{1}{\pi} \int_0^\infty |u|^{s+1} |\hat{f}_z(u)| du$. The next theorem is not only applicable to obtain explicit formulas for L and N to approximate the CCDF $G_{X|Z=z}(y)$ through (7) but is also applied for European option pricing, see Section 4. In the case of a CCDF, set $w_z = 1_{(-\infty,y]}(x)$ so that $\|w_z\|_\infty = 1$ and note that the coefficients v_k^z are given explicitly in (8).

Theorem 2. *Let $z \in E$. Let $w_z : \mathbb{R} \to \mathbb{R}$ be bounded with $\|w_z\|_\infty > 0$. Let $\varepsilon > 0$ be small enough. Assume $J \in \mathbb{N}$ and f_z satisfies the assumptions in Lemma 1. For some even $n \in \mathbb{N}$ let $L = L\left(\varepsilon, n, \|w_z\|_\infty\right)$ and for $s \in \{1, \dots, J-1\}$ odd let $N = N\left(\varepsilon, s, \|w_z\|_\infty\right)$. Then*

$$\left| \int_{\mathbb{R}} w_z(x) g_{X|Z=z}(x) dx - \sum_{k=0}^{N}{}' \tilde{c}_k^z v_k^z \right| \le \varepsilon$$

where $v_k^z := \int_{-L}^{L} w_z(x+\mu) \cos\left(k\pi \frac{x+L}{2L}\right) dx$, $k = 0, 1, \dots$

Proof. We use the fact that $u \mapsto |u|^{s+1} |\hat{f}_z(u)|$ is a symmetric function and we apply Junike (2024, Inequality 3.8 and Theorem 3.8). ■

Note that L and N in Theorem 2 converge to zero if $\|w_z\|_\infty$ converges to zero. Since $\overline{G}_z$ from (7) approximates the CCDF of X given $Z = z$ arbitrarily closely, we invert it in order to simulate samples of X given $Z = z$ by the inversion transform.

3.2. Enhancing convergence of root finding algorithms via metalog distribution

Given $\overline{G}_z(y)$ computed as in (7), we suggest random sampling based on inverse transform method. This means drawing a random sample, u, from a uniform distribution over $[0, 1]$ and then finding y such that $u = \overline{G}_z(y)$. This equation must be solved numerically. Also traditional exact simulation schemes perform random sampling in the same way. Authors typically use the quantile function of two moments-matched normal distribution to find reasonable starting points (see Broadie & Kaya, 2006, p. 221). We suggest an alternative approach based on an alternative statistical distribution with the scope of enhancing the convergence of root finding algorithm (in other words, reduce the number of iterations needed to solve the equation numerically).

The metalog distributions are a family of continuous probability distributions introduced by Keelin (2016). Among the various positive features of this family of distributions, we are interested in the following: (i) they present a simple quantile function; (ii) parameter estimation is performed efficiently through moment-matching. Suppose we are interested in obtaining an approximate sample from a random variable $Y : \Omega \to \mathbb{R}$. If it is possible to compute the first three cumulants of Y, then the metalog distributions offer a viable tool. Define c_j as the jth cumulant of Y, $s = c_3/c_2^{1.5}$ is the skewness. If $|s| < 2.07093$ then the quantile function of a three parameter metalog is (Keelin, 2016)

$$Q(y; a_1, a_2, a_3) = a_1 + a_2 \ln\left(\frac{y}{1-y}\right) + a_3(y - 0.5) \ln\left(\frac{y}{1-y}\right), \tag{11}$$

where the parameters a_1, a_2 and a_3 are computed by matching cumulants: $a_1 = c_1 - \frac{a_3}{2}$,

$$a_2 = \frac{1}{\pi} \sqrt{3\left(c_2 - \left(\frac{1}{12} + \frac{\pi^2}{36}\right) a_3^2\right)}, \; a_3 = 4\sqrt{\frac{6c_2}{6+\pi^2}} \cos\left(\frac{1}{3}\left(\arccos\left(-\frac{s}{4}\left(1 + \frac{\pi^2}{6}\right)^{\frac{1}{2}}\right) + 4\pi\right)\right). \tag{12}$$

Given the first three cumulants of $(X|Z)$, which can be easily obtained from the first tree moments, we compute a_1, a_2 and a_3 from (12) and the starting points $y_0 = Q(u, a_1, a_2, a_3)$ from (11). The three parameters metalog distribution is potentially a better choice than the normal, since it presents a heavier left tail and (more importantly) matches three moments instead of two. Naturally, the more accurate the approximation, the closer the initial points to the solution, meaning a smaller amount of iterations and smaller computing times.[1] The role of the metalog is illustrated in Section EC.3.2.

3.3. Inputs of the algorithm

Before implementing our suggested simulation scheme, the final user must first input four tuning parameters: (i) ε, which represents the maximum difference between the true cumulative distribution function and the one computed via the COS method; (ii) s, that is an odd number, which should be chosen such that $B(z)$ (and hence N) in (10) is minimal; (iii) $\tilde{q}$, which denotes the number of quadrature points needed to compute $B(z)$; (iv) n, denoting that one computes the nth central moment of $(X|Z)$ to compute L as in (9).

Some comments are in order. First, the only parameter controlling the error is ε. The other parameters control the computing time but not the error.[2] Second, the number of quadrature points, $\tilde{q}$, is just used to compute $B(z)$, which is then used to obtain the number of terms N. Since the minimal number of terms to ensure that the approximation by COS method stays below a given error tolerance is usually overestimated by (10), one does not need that $B(z)$ is computed with high accuracy. Therefore, $\tilde{q}$ can be set very small to speed up the computation of $B(z)$. Third, L is not monotone in the number of moments, n. We observed that the higher n the smaller L but if n is too large, L will increase again. We study numerically how the various tuning parameters impact the computing time in Section EC.3.1.

[1] The methodology works only if $|s| < 2.07093$. In some rare cases, it may happen that this condition is not respected and we use the quantile function of the moment matched normal distribution in such cases.

[2] $\tilde{q}$ should be set reasonably large to avoid any impact on the accuracy, as we illustrate in Section EC.3.1.

Besides tuning parameters, to implement the suggested approach the final user must also supply the characteristic function of $(X|Z = z)$, the first three cumulants (used to find the starting points through the metalog distribution) and the nth central moment of $(X|Z = z)$. In Section 5 we consider concrete examples where the proposed conditional COS approach can be used for simulation and show how to compute conditional characteristic functions and moments of $(X|Z = z)$ for a class of stochastic volatility models. In reality, to implement our algorithm we need to compute the nth central moments of $(X|Z = z)$, μ_n. Given cumulants, central moments can be computed analytically using Faá di Bruno's formula for high derivatives of composite functions: $\mathbb{E}[(X-\mu)^n|Z] = \sum_{j=1}^{n} \mathrm{B}_{n,j}(0, c_2, \dots, c_{n-j+1})$, where $\mathrm{B}_{n,j}$ are the incomplete Bell polynomials, μ is defined in Section 3.1, and c_j is the jth cumulant of $(X|Z)$.

3.4. Final algorithm

We can now depict our improved exact simulation scheme. The final user must first input four tuning parameters: ε, s, $\tilde{q}$, and n. Then, we compute $\tilde{q}$ Gauss–Laguerre quadrature nodes and points $\{(w_j, u_j)\}_{j=1}^{\tilde{q}}$. We can now start our simulation. We suppose that it is possible to sample from the random variable Z and denote the random realization by z. We compute $B(z)$:

$$B(z) := \int_0^\infty \Psi(u, z) e^{-u} du \approx \sum_{j=1}^{\tilde{q}} w_j \Psi(u_j, z), \tag{13}$$

where $\Psi(u, z) := \frac{|u|^{s+1}}{\pi} |\hat{f}_{X|Z=z}(u)| e^u$. Given $B(z)$ we can compute L and N from (9) and (10). At this stage, given the conditional characteristic function of $(X|Z)$ we compute $\tilde{c}_j^z$ as in (5) for $j = 1, \dots, N$ and then $\overline{G}_z(y)$ as in (7). We sample from a uniform $u \sim U(0, 1)$ and find y such that $u = \overline{G}_z(y)$. This is solved using root finding algorithms (Matlab function `fzero`). The initial guess, y_0, is obtained from the metalog: based on the cumulants of $(X|Z)$ we compute $\{a_j\}_{j=1}^3$ from (12) and set $y_0 = Q(u, a_1, a_2, a_3)$ where Q is as in (11). The methodology is summarized in Algorithm 1.

4. Error propagation when pricing options

In Section 3 we discussed the general problem of simulating a random variable X conditionally on the realization of another random variable Z given the knowledge of the characteristic function of $(X|Z)$ and the moments. In this section, we study how the error implicit in our simulation approach propagates to the price of derivatives when the couple (X, Z) defines a stochastic volatility option pricing model (it will be the scope of Section 5 to provide concrete examples of stochastic volatility models that fit into this framework). We consider the case of path-independent and forward starting options. All the expectations are taken under the risk-neutral probability.

4.1. Path-independent options

For $z \in E$, let $w_z : \mathbb{R} \to \mathbb{R}$ be a measurable function. Let $X_T = \log \frac{S_T}{S_0}$, where S is the asset price, and Z_T be a random variable measurable with respect to the information available at time $T > 0$. We aim to compute $\mathbb{E}[w_{Z_T}(X_T)]$ by Monte Carlo simulation and the COS method provided that the conditional characteristic function of X_T given $Z_T = z$ is given in closed form. In stochastic volatility models, Z_T could be for instance the variance at time T. For example, a European put option can be described by the payoff function $w_z(x) = e^{-rT} \max(K - S_0 e^x, 0)$ where K is the strike price, S_0 is the initial price of the underlying, r is the risk-less rate and T is the maturity. Since w_z does not depend on z, we will write w instead of w_z in the next corollary, which explains how the error on $\overline{G}_z$ as computed in (7) and the uncertainty introduced by the Monte Carlo estimation propagates to the approximation of the price of w. In analogy to Section 3, we define $f_z(x) := g_{X_T|Z_T=z}(x + \mu)$, where $\mu = \mathbb{E}[X_T|Z_T = z]$ and $g_{X_T|Z_T=z}$ is the conditional kernel density function of X_T given $Z_T = z$.

Corollary 3. *Let $w : \mathbb{R} \to \mathbb{R}$ describe some payoff function of a path-independent option such that $\|w\|_\infty \in (0, \infty)$. Assume f_z satisfies the assumption in Theorem 2 and let n and s as in Theorem 2. Let $\varepsilon > 0$. Define $\overline{G}_z$ as in Eq. (7) using $L = L\left(\frac{\varepsilon}{2\|w\|_\infty}, n, 1\right)$ and $N = N\left(\frac{\varepsilon}{2\|w\|_\infty}, s, 1\right)$. Set $\tilde{P} := \frac{1}{\mathcal{M}} \sum_{j=1}^{\mathcal{M}} w\left(\overline{G}_{Z_T^j}^{-1}(U_j)\right)$ where U_j and Z_T^j are independent copies of a uniform random variable U and Z_T, respectively. Let $q \in (0, 1)$ and σ be the standard deviation of $w\big(\overline{G}_{Z_T}^{-1}(U)\big)$. If $\mathcal{M} \geq \left(\Phi^{-1}\left(\frac{q+1}{2}\right) \frac{2\sigma}{\varepsilon}\right)^2$ it holds with probability q (asymptotically under the risk neutral measure) that $|\mathbb{E}[w(X_T)] - \tilde{P}| < \varepsilon$.*

Proof. See Section EC.1.2 ■

Note the useful interpretation of the parameter ε: it is the maximum possible distance between the true price of the option and the one obtained through the Monte Carlo simulation with probability q. This is a noticeable advantage with respect to alternative approaches where the bias must be computed via extensive Monte Carlo simulations. If the overall error tolerance is $\varepsilon > 0$, the error tolerance for the Monte Carlo simulation is $\frac{\varepsilon}{2}$ and the error tolerance for the approximation $\overline{G}_z$ is $\frac{\varepsilon}{2\|w\|_\infty}$. It is therefore clear that it is very easy to control the error under our approach. Indeed, it is sufficient to select the maximum error tolerable when pricing the option. Then, our algorithm automatically selects the tuning parameters and the number of simulations such that the true bias will be below ε (with probability q). This is unattainable with all the alternative simulation schemes available in the literature (see Section 2). The proposed simulation scheme for path-independent options is summarized in Algorithm 2.

4.2. Forward starting options

Let $0 < T_1 < T_2$ be two dates. We consider a forward starting put option with payoff $(S_{T_1} - S_{T_2})^+$, where S_{T_1} is the price of an asset at time T_1. Here, we deal with forward starting options, which are at-the-money at time T_1. The approach can easily be generalized to in-the-money or out-of-the money options by considering $(kS_{T_1} - S_{T_2})^+$ for some $k > 0$. We apply Theorem 2 to price options with such payoff. The price of the corresponding forward starting call option with payoff, $(S_{T_2} - S_{T_1})^+$, can be recovered applying the put–call parity relation. In addition, in analogy with Section 3, let $X_{T_1} = \log \frac{S_{T_1}}{S_0}$, $X_{T_2} = \log \frac{S_{T_2}}{S_{T_1}}$ and let Z_{T_1} and Z_{T_2} be random variables measurable with respect to the information available at time T_1 and T_2 respectively. Let $\ell \in [0, \infty]$ and

$$w_z(x, \ell) := e^{-r(T_2-T_1)}\mathbb{E}\left[\left(S_{T_1} \wedge \ell - S_{T_2}\right)^+ | S_{T_1} = S_0 e^x, Z_{T_1} = z\right], \quad x \in \mathbb{R}.$$

At time T_1, $w_z(x, \ell)$ is just the price of a European put option with starting underlying price $S_0 e^x$ and strike equal to $S_0 e^x \wedge \ell$. To price the forward starting option, we first simulate (X_{T_1}, Z_{T_1}) and then approximate $w_z(x, \ell)$, where $z = Z_{T_1}$ and $x = X_{T_1}$ as in Junike (2024). Such approximation is denoted by $\tilde{w}_z(x, \ell)$. The time-0 price of the option is given by $e^{-rT_1}\mathbb{E}\left[w_{Z_{T_1}}(X_{T_1}, \infty)\right]$. We introduce the technical bound ℓ to be able to apply Theorem 2. In analogy with Section 3, we define $f_{T_1,z}(x) := g_{X_{T_1}|Z_{T_1}=z}(x + \mu)$, where $\mu = \mathbb{E}[X_{T_1}|Z_{T_1} = z]$ and $g_{X_{T_1}|Z_{T_1}=z}$ is the conditional kernel density function of X_{T_1} given $Z_{T_1} = z$.

Corollary 4. *Let $\varepsilon > 0$ be small enough. Let $\ell \in (0, \infty)$ such that $e^{-rT_2}\mathbb{E}\left[S_{T_1} 1_{\{S_{T_1} > \ell\}}\right] < \frac{\varepsilon}{3}$. Let $\tilde{w}_z(x, \ell)$ be an approximation of $w_z(x, \ell)$ such that*

$$\left|\tilde{w}_{Z_{T_1}}(X_{T_1}, \ell) - w_{Z_{T_1}}(X_{T_1}, \ell)\right| < \frac{\varepsilon}{6} e^{rT_1}, \quad a.s. \tag{14}$$

Assume $f_{T_1,z}$ satisfies the assumption in Theorem 2 and let n and s as in Theorem 2. Let $m := e^{-r(T_2-T_1)}\ell$, then $\|w_z(.,\ell)\|_\infty \leq m$. Assume $\|\tilde{w}_z(.,\ell)\|_\infty \leq m$ as well. Define $\overline{G}_z$ as in Eq. (7) using $L = L(\frac{\varepsilon e^{rT_1}}{6m}, n, 1)$ and $N = N(\frac{\varepsilon e^{rT_1}}{6m}, s, 1)$. Let $\tilde{P} := \frac{e^{-rT_1}}{\mathcal{M}} \sum_{j=1}^{\mathcal{M}} \tilde{w}_{Z^j_{T_1}}\left(\overline{G}^{-1}_{Z^j_{T_1}}(U_j), \ell\right)$ where U_j and $Z^j_{T_1}$ are independent copies of a uniform random variable U and Z_{T_1}, respectively. Let $q \in (0, 1)$ and σ be the standard deviation of $e^{-rT_1}\tilde{w}_{Z_{T_1}}\left(\overline{G}^{-1}_{Z_{T_1}}(U), \ell\right)$. If $\mathcal{M} \geq \left(\Phi^{-1}\left(\frac{q+1}{2}\right)\frac{3\sigma}{\varepsilon}\right)^2$ it holds with probability q (asymptotically under the risk neutral measure) that $|e^{-rT_1}\mathbb{E}\left[w_{Z_{T_1}}(X_{T_1}, \infty)\right] - \tilde{P}| < \varepsilon$.

Proof. See Section EC.1.3. ■

Remark 1. How can we find ℓ? We suggest to approximate S_{T_1} by a simple model such as the Black–Scholes model $S^{BS}_{T_1} = S_0 e^{\eta + \sigma\sqrt{T_1}\Xi}$, where Ξ is a standard normal random variable, $\sigma > 0$ is chosen such that $\sigma^2 T_1$ is equal to the variance of S_{T_1} and $\eta = \log(S_0) + \left(r - \frac{1}{2}\sigma^2\right)T_1$. Let $\Phi_{\sigma\sqrt{T_1},1}$ denote the CDF of a normally distributed random variable with mean $\sigma\sqrt{T_1}$ and standard deviation 1. A simple calculation shows that

$$\ell \geq \exp\left(\eta + \sigma\sqrt{T_1}\Phi^{-1}_{\sigma\sqrt{T_1},1}\left(1 - \frac{\varepsilon}{3}e^{rT_2 - \eta - \frac{\sigma^2 T_1}{2}}\right)\right) \tag{15}$$

implies $e^{-rT_2}\mathbb{E}[S^{BS}_{T_1} 1_{\{S^{BS}_{T_1} > \ell\}}] \leq \frac{\varepsilon}{3}$.

We summarize Corollary 4 in Algorithm 3. Note that to implement this algorithm one needs also the characteristic function (and related moments) of $X_{T_2} - \mathbb{E}[X_{T_2}]$ given $Z_{T_1} = z$ and $X_{T_1} = x$, that is denoted by $\hat{f}_{T_2,z,x}$, in order to obtain $\tilde{w}_z(x, \ell)$. We postpone to Section 6.3 a discussion on how to obtain such characteristic function and moments for the option pricing models considered in Section 5. We remark that the final price of the forward starting put option is given by the value $\tilde{P}$ and differs by no more than ε from the true price with probability q if $\mathcal{M}$ is as in Corollary 4.

5. Exact simulation of analytically solvable stochastic volatility models

In the previous section we have seen that to implement the conditional COS approach we need: (i) the conditional characteristic function of the log-returns, $\hat{g}_{X|Z=z}$; (ii) the first three cumulants of $(X|Z)$, $\{c_j\}_{j=1}^3$; (iii) the nth central moment of $(X|Z)$, μ_n. In Table 1 we illustrate how to bridge the notation of this section with that of Section 3. Therefore, the scope of this section is twofold. First, we identify analytically solvable (in the sense of Zeng et al., 2023) stochastic volatility models, i.e., models for which it is possible to find the relevant conditional characteristic function (and provide its explicit expression). Second, we propose a new methodology to compute efficiently high order cumulants (or, equivalently, moments) of $(X|Z)$. Given conditional characteristic functions and moments, one can simulate exactly the model using Algorithm 1. Pseudo-codes for the simulation of the various models are given in Section EC.2. Throughout the rest of this paper, 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. We denote by V_T (respectively, S_T) the variance (asset price) at time $T > 0$.

5.1. 4/2 stochastic volatility model

The following model is known as the 4/2 model and was introduced by (Grasselli, 2017)[3]

$$\frac{dS_t}{S_t} = r dt + \frac{aV_t + b}{\sqrt{V_t}}\left(\rho dW_t + \sqrt{1-\rho^2}dB_t\right), \tag{16}$$

$$dV_t = k(\theta - V_t)dt + \sigma\sqrt{V_t}dW_t, \tag{17}$$

[3] See also Callegaro et al., 2019 and Gnoatto et al., 2022 for empirical applications in the equity and FX markets.

Table 1
Recovering the notation of Section 3.

Model	E	Z	$\hat{g}_{X\|Z}(u)$
Heston	$(0,\infty)\times\mathbb{N}_0$	(V_T,ζ)	(30)
3/2 and 4/2	$(0,\infty)$	V_T	(21)
CGMYCIR	$(0,\infty)\times\mathbb{N}_0$	(V_T,ζ)	(34)
Heston $\bigoplus$ 3/2	$(0,\infty)\times\mathbb{N}_0\times(0,\infty)$	$(V_T^{(1)},\zeta^{(1)},V_T^{(2)})$	(40)
Double Heston	$(0,\infty)\times\mathbb{N}_0\times(0,\infty)\times\mathbb{N}_0$	$(V_T^{(1)},\zeta^{(1)},V_T^{(2)},\zeta^{(2)})$	(48)

Notes. In all cases $X = X_T = \log(S_T/S_0)$. $\mathbb{N}_0$ is the set of natural numbers including 0.

where W_t and B_t are independent standard Brownian motions. The 4/2 model nests as special case two important models in finance. (i) The Heston model ($a = 1$, $b = 0$); (ii) The 3/2 stochastic volatility model ($a = 0$). To simulate this model exactly, we recover the setting of Section 3 according to $Z = V_T$, $X := X_T = \log(S_T/S_0)$ and $E = \mathbb{R}_+$. A very nice feature of this model is that it is analytically solvable, as shown in Grasselli (2017) and Zeng et al. (2023), and thus it can be simulated exactly. The terminal variance is given as follows

$$V_T \overset{\text{law}}{=} \frac{\sigma^2(1-e^{-kT})}{4k}\chi_\delta'^2(\lambda), \tag{18}$$

where $\chi_\delta'^2(\lambda)$ denotes the non-central chi-squared distribution with $\delta := 4\theta k/\sigma^2$ degrees of freedom and non-centrality parameter $\lambda := 4ke^{-kT}V_0/\sigma^2(1-e^{-kT})$. Hence, $(V_T|V_0)$ is simulated exactly using standard random numbers generators from the non-central chi-squared distribution (e.g., the built–in Matlab® function `ncx2rnd`). Then, from (Grasselli, 2017, p. 1020)

$$\left(X_T\middle|\int_0^T V_s ds, \int_0^T \frac{ds}{V_s}, V_T\right) \sim \mathcal{N}(\underline{m}, \underline{s}^2), \tag{19}$$

where $\underline{m} = \xi_1 + \xi_2 V_T + \xi_3\log V_T + \xi_4\int_0^T V_s ds + \xi_5\int_0^T \frac{ds}{V_s}$, $\underline{s}^2 = \xi_6 + \xi_7\int_0^T V_s ds + \xi_8\int_0^T\frac{ds}{V_s}$ and

$$\xi_1 = \left(r - ab - \frac{a\rho k\theta}{\sigma} + \frac{b\rho k}{\sigma}\right)T - \frac{a\rho V_0}{\sigma} - \frac{b\rho}{\sigma}\log(V_0),\ \xi_2 = \frac{a\rho}{\sigma}, \xi_3 = \frac{b\rho}{\sigma},\ \xi_4 = \frac{a\rho k}{\sigma} - 0.5a^2,$$
$$\xi_5 = \frac{b\rho}{\sigma}\left(\frac{\sigma^2}{2} - k\theta\right) - \frac{b^2}{2}, \xi_6 = 2(1-\rho^2)abT,\ \xi_7 = (1-\rho^2)a^2, \xi_8 = (1-\rho^2)b^2. \tag{20}$$

The characteristic function of $(X_T|V_T)$ can be obtained according to

$$\hat{g}_{X_T|V_T}(u) = \mathbb{E}\left[e^{iuX_T}\middle|V_T\right] = \mathbb{E}\left[\mathbb{E}\left[e^{iuX_T}\middle|\int_0^T V_s ds, \int_0^T\frac{ds}{V_s}, V_T\right]|V_T\right] = \mathbb{E}\left[e^{iu\underline{m}-\frac{u^2}{2}\underline{s}^2}|V_T\right]$$
$$= e^{iu(\xi_1+\xi_2V_T+\xi_3\log V_T)-\frac{u^2}{2}\xi_6}\mathcal{L}\left(-iu\xi_4 + \frac{u^2}{2}\xi_7, -iu\xi_5 + \frac{u^2}{2}\xi_8\right), \tag{21}$$

where, from (Grasselli, 2017, Proposition A.4),

$$\mathcal{L}(u_1,u_2) := \mathbb{E}\left[e^{-u_1\int_0^T V_s ds - u_2\int_0^T\frac{ds}{V_s}}|V_T\right] = \exp\left(\frac{V_0+V_T}{\sigma^2}\left(k\coth\left(\frac{kT}{2}\right) - \sqrt{A_1}\coth\left(\sqrt{A_1}\frac{T}{2}\right)\right)\right)$$
$$\times\frac{\sqrt{A_1}\sinh\left(\frac{kT}{2}\right)}{k\sinh\left(\sqrt{A_1}\frac{T}{2}\right)}\frac{I_{\frac{\sqrt{(2k\theta-\sigma^2)^2+8\sigma^2u_2}}{\sigma^2}}\left(\frac{2\sqrt{A_1V_0V_T}}{\sigma^2\sinh\left(\sqrt{A_1}\frac{T}{2}\right)}\right)}{I_{\frac{2k\theta}{\sigma^2}-1}\left(\frac{2k\sqrt{V_0V_T}}{\sigma^2\sinh\left(\frac{kT}{2}\right)}\right)}, \tag{22}$$

with $A_1 = k^2 + 2u_1\sigma^2$ and $I_\nu(\cdot)$ being the modified Bessel function of the first kind. We have therefore an analytic expression for the characteristic function of $(X_T|V_T)$. Next, we need to compute the moments (or, equivalently, the cumulants) of $(X_T|V_T)$. It is tantamount to use the following well-known formula $\mu_n = \mathbb{E}\left[X_T^n|V_T\right] = \frac{1}{i^n}\frac{\partial^n}{\partial u^n}\hat{g}_{X_T|V_T}(u)\Big|_{u=0}$. However, this methodology would not be very efficient in our case due the fact that it requires computing many times the modified Bessel functions and their derivatives. To better illustrate this point we report in Section EC.4 the analytical expressions for the first two moments of $(X_T|V_T)$. Of course, when computing higher order moments (e.g., eight moment), the approach based on analytical derivation becomes impossible and also endangers numerical malfunctions and non negligible approximation errors when computing derivatives. We suggest therefore a different approach. The nth cumulant of $(X_T|V_T)$ can be computed in the following way:

$$c_n = \frac{\partial^n}{\partial u^n}\log\mathbb{E}\left[e^{uX_T}|V_T\right]\Big|_{u=0} = \frac{\partial^n}{\partial u^n}\log\mathbb{E}\left[\mathbb{E}\left[e^{uX_T}\middle|\int_0^T V_s ds, \int_0^T\frac{ds}{V_s}, V_T\right]\middle|V_T\right]\Big|_{u=0}$$
$$= \underbrace{\frac{\partial^n}{\partial u^n}\left(\xi_1+\xi_2V_T+\xi_3\log V_T\right)u + \frac{u^2\xi_6}{2}\Big|_{u=0}}_{=:C_1(n)} + \underbrace{\frac{\partial^n}{\partial u^n}\log\mathbb{E}\left[e^{\left(u\xi_4+\frac{u^2}{2}\xi_7\right)\int_0^T V_s ds+\left(u\xi_5+\frac{u^2}{2}\xi_8\right)\int_0^T\frac{ds}{V_s}}\middle|V_T\right]\Big|_{u=0}}_{=:C_2(n)},$$

where

$$C_1(1) = \xi_1 + \xi_2V_T + \xi_3\log V_T,\ C_1(2) = \xi_6,\ C_1(n) = 0 \text{ for } n > 2. \tag{23}$$

It remains to compute $C_2(n)$. First, note that

$$C_2(n) = \frac{\partial^n}{\partial u^n}\log\mathcal{L}\left(-u\xi_4 - \frac{u^2}{2}\xi_7, -u\xi_5 - \frac{u^2}{2}\xi_8\right)\Big|_{u=0}.$$

Following Kyriakou et al. (2024), since both $\int_0^T V_s ds$ and $\int_0^T \frac{ds}{V_s}$ are defined on $\mathbb{R}_+$, we suggest computing the second derivative using the *numerical inversion of adaptively modified moment generating function* algorithm introduced by Choudhury and Lucantoni (1996). Define

$$\bar{\mathcal{L}}(u) := \mathcal{L}\left(-u\xi_4 - \frac{u^2}{2}\xi_7, -u\xi_5 - \frac{u^2}{2}\xi_8\right). \tag{24}$$

Let $l = 1$ and choose α_j adaptively, i.e., for the jth cumulant we set $\alpha_j = (j-1)\frac{c_{j-2}}{c_{j-1}}$ for $j > 2$. We then have that

$$C_2(n) = \frac{n!}{2nlr_n^n\alpha_n^n} \cdot \left(\log \bar{\mathcal{L}}(\alpha_n r_n) + (-1)^n \log \bar{\mathcal{L}}(-\alpha_n r_n) + 2\sum_{j=1}^{nl-1} \text{Real}\left(\log \bar{\mathcal{L}}(\alpha_n r_n e^{\pi i j/nl})e^{-\pi i j/l}\right)\right) - \hat{e}, \tag{25}$$

where $\hat{e}$ indicates the discretization error and the function r_n is chosen to bound the discretization error, in particular $r_n = 10^{-\frac{\gamma}{2nl}}$ allows to achieve accuracy of order $10^{-\gamma}$. In Algorithm 4 we summarize the procedure to obtain the first n cumulants and to compute adaptively α_n. Given the nth cumulant, we can recover the nth central moment, μ_n. Given the first three cumulants, the nth central moment and the characteristic function of $(X_T|V_T)$, we can simulate the 4/2 model using our conditional COS approach. The resulting exact simulation scheme is summarized in Algorithm 5. Note that through this algorithm it is also possible to simulate the 3/2 stochastic volatility model and the Heston model.

5.2. Heston model

The Heston model is a special case of the 4/2 stochastic volatility model and can be simulated using Algorithm 5. However, we propose next a different approach exploiting innovations from Choi and Kwok (2024). The variance process, V_T, is related to a δ-dimensional squared Bessel process, where $\delta := 4\theta k/\sigma^2$. Hence, following Yuan and Kalbfleisch (2000) it can be simulated from a randomized Gamma distribution:

$$\zeta \sim \text{Poisson}\left(\frac{2V_0 k}{\sigma^2(e^{kT}-1)}\right), \quad V_T \sim \text{Gamma}\left(\frac{\delta}{2} - 1 + \zeta, \frac{\sigma^2(1-e^{kT})}{2k}\right). \tag{26}$$

Conditionally on (ζ, V_T), from Choi and Kwok (2024) we have

$$\mathcal{L}_P(u) = \mathbb{E}\left[e^{-u\int_0^T V_s ds}|\zeta, V_T\right] = \mathcal{L}_{X_1}(u)\mathcal{L}_{X_2}(u)\mathcal{L}_Z(u)^{\zeta}, \tag{27}$$

$$\mathcal{L}_{X_1}(u) = \exp\left(\frac{V_0 + V_T}{\sigma^2}\left(k\coth\left(\frac{kT}{2}\right) - \sqrt{2\sigma^2 u + k^2}\coth\left(\frac{\sqrt{2\sigma^2 u + k^2}T}{2}\right)\right)\right), \tag{28}$$

$$\mathcal{L}_{X_2}(u) = \left(\frac{\sqrt{2\sigma^2 u + k^2}\sinh(kT/2)}{k\sinh(\sqrt{2\sigma^2 u + k^2}T/2)}\right)^{\frac{\delta}{2}}, \quad \mathcal{L}_Z(u) = \left(\frac{\sqrt{2\sigma^2 u + k^2}\sinh(kT/2)}{k\sinh(\sqrt{2\sigma^2 u + k^2}T/2)}\right). \tag{29}$$

Note the advantage of the alternative representation in (27) with respect to (24): it does not depend anymore on the ratio of modified Bessel functions of the first kind. Thus, it can be computed more efficiently. The characteristic function of $(X_T|\zeta, V_T)$ is given as follows:

$$\hat{g}_{X_T|\zeta,V_T}(u) = e^{iu(\xi_1+\xi_2 V_T)}\mathcal{L}_P\left(-iu\xi_4 + \frac{u^2}{2}\xi_7\right). \tag{30}$$

The nth cumulant of $(X_T|\zeta, V_T)$ can be computed in the following way:

$$\begin{aligned} c_n &= \frac{\partial^n}{\partial u^n}\log\mathbb{E}\left[e^{uX_T}|\zeta, V_T\right]\Big|_{u=0} = \underbrace{\frac{\partial^n}{\partial u^n}\left(\xi_1 + \xi_2 V_T\right)u\Big|_{u=0}}_{=:C_1(n)} + \frac{\partial^n}{\partial u^n}\log\mathbb{E}\left[e^{\left(u\xi_4 + \frac{u^2}{2}\xi_7\right)\int_0^T V_s ds}\Big|\zeta, V_T\right]\Big|_{u=0} \\ &= C_1(n) + \underbrace{\frac{\partial^n}{\partial u^n}\log\left(\mathcal{L}_P(-u\xi_4 - \frac{u^2}{2}\xi_7)\right)\Big|_{u=0}}_{=:C_2(n)} = \begin{cases} \xi_1 + \xi_2 V_T & \text{if } n = 1 \\ 0 & \text{if } n > 1 \end{cases} + C_2(n). \end{aligned} \tag{31}$$

$C_2(n)$ can be computed using Algorithm 4 by setting

$$\bar{\mathcal{L}}(u) := \mathcal{L}_P\left(-u\xi_4 - \frac{u^2}{2}\xi_7\right). \tag{32}$$

The resulting exact simulation scheme for the Heston model is summarized into Algorithm 6. It is clear that this approach is expected to present superior performances than Algorithm 5 (in the special case of the Heston model) since the evaluation of the modified Bessel function is sidestepped.

5.3. Time-changed Lévy models

Time-changed Lévy models are built by taking, e.g., a homogeneous Lévy process and subordinating it by the integrated variance. Given an initial value S_0, the asset price at time T is

$$S_T = S_0\exp\left(rT + \Xi_{\left(2abT + a^2\int_0^T V_s ds + b^2\int_0^T \frac{ds}{V_s}\right)}\right), \tag{33}$$

where Ξ is an independent Lévy process and V is as in (17). Also this class of models is analytically solvable, as shown in Zeng et al. (2023, Corollary 2.13) and therefore can be simulated exactly by suitably modifying Algorithm 5. However, in what follows, let us consider the special case where $a = 1$ and $b = 0$ and where Ξ follows a CGMY model (Carr et al., 2003). This corresponds to the CGMYCIR model.[4] The conditional characteristic function of the log-returns is

$$\hat{g}_{X_T|\zeta,V_T}(u) = \mathbb{E}\left[\mathbb{E}\left[e^{iuX_T}\Big|\int_0^T V_s ds, \zeta, V_T\right]\Big|\zeta, V_T\right] = e^{iurT}\mathcal{L}_P(-\psi(iu)), \tag{34}$$

where $\psi(u)$ is logarithm of the moment generating function of the compensated driving Lévy process, Ξ. For the CGMY model we have $\psi(u) = u(C\Gamma(-Y)(M^Y - (M-1)^Y + G^Y - (G+1)^Y)) + C\Gamma(-Y)((M-u)^Y - M^Y + (G+u)^Y - G^Y)$. Next, we need the cumulants of $(X_T|\zeta, V_T)$:

$$\begin{aligned} c_n &= \frac{\partial^n}{\partial u^n}\log\mathbb{E}\left[e^{uX_T}|\zeta, V_T\right]\Big|_{u=0} = \frac{\partial^n}{\partial u^n}rTu\Big|_{u=0} + \frac{\partial^n}{\partial u^n}\log\mathbb{E}\left[e^{\left(u\xi_4 + \frac{u^2}{2}\xi_7\right)\int_0^T V_s ds}\Big|\zeta, V_T\right]\Big|_{u=0} \\ &= \underbrace{\frac{\partial^n}{\partial u^n}rTu\Big|_{u=0}}_{=:C_1(n)} + \underbrace{\frac{\partial^n}{\partial u^n}\log\left(\mathcal{L}_P(-\psi(u))\right)\Big|_{u=0}}_{=:C_2(n)} = \begin{cases} rT & \text{if } n = 1 \\ 0 & \text{if } n > 1 \end{cases} + C_2(n). \end{aligned} \tag{35}$$

$C_2(n)$ can be computed using Algorithm 4 by setting

$$\bar{\mathcal{L}}(u) := \mathcal{L}_P\left(-\psi(u)\right). \tag{36}$$

The resulting exact simulation scheme for the CGMYCIR model is given into Algorithm 7.

5.4. Multifactor models

In this section, we show that multifactor stochastic volatility models are naturally included into our framework and can simulated exactly and efficiently through the conditional COS method developed in this paper. This is important because multifactor models are more realistic than single factor models.[5] The possibility to perform exact simulation in a multifactor model is more important than the single factor case. Indeed, model sophistication is likely to reduce analytical tractability, potentially preventing the development of exact closed formulas for derivatives prices. Derivatives pricing is thus typically done through simulation. Moreover, when using approximations, the error control becomes harder with the number of parameters controlling the error which increases proportionally to the number of factors (see the discussion in Section 6.2.2).

5.4.1. Heston $\bigoplus$ 3/2 model

Grasselli (2017) introduces the following model (henceforth, Heston $\bigoplus$ 3/2 model)

$$\frac{dS_t}{S_t} = rdt + a\sqrt{V_t^{(1)}}\left(\rho_1 dB_t^{(1)} + \sqrt{1-\rho_1^2}dW_t^{(1)}\right) + \frac{b}{\sqrt{V_t^{(2)}}}\left(\rho_2 dB_t^{(2)} + \sqrt{1-\rho_2^2}dW_t^{(2)}\right) \tag{37}$$

$$dV_t^{(1)} = k_1(\theta_1 - V_t^{(1)})dt + \sigma_1\sqrt{V_t^{(1)}}dB_t^{(1)},\ dV_t^{(2)} = k_2(\theta_2 - V_t^{(2)})dt + \sigma_2\sqrt{V_t^{(2)}}dB_t^{(2)}, \tag{38}$$

where $B_t^{(i)}$ and $W_t^{(i)}$ for $i = 1, 2$ are independent standard Brownian motions. As special cases, we recover the 3/2 model ($a = 0$) and the Heston model ($b = 0$). Grasselli (2017) presents the model in a more general form with potentially more than two variance factors.[6]

We show that this model can be simulated exactly. This is interesting because this model does not fall directly in the class of models considered by Zeng et al. (2023). Moreover, under this model we expect our method will perform much better than possible benchmark methodologies such as Kyriakou et al. (2024), and Zeng et al. (2023). Additionally, we show that it is possible to speed up the exact simulation of the Heston $\bigoplus$ 3/2 model by exploiting the Poisson conditioning method of Choi and Kwok (2024). Under this model, it is therefore possible to reduce the amount of modified Bessel functions evaluations, but not to eliminate it completely.

To simulate the model through our proposed approach, we need to derive the conditional characteristic function of the log-returns. We thus need to extend the results in Zeng et al. (2023) to the case with more than one variance factor. First, setting $\zeta^{(1)}$ is as in (26), note that

$$\left(X_T\Big|\int_0^T V_s^{(1)}ds, \int_0^T \frac{ds}{V_s^{(2)}}, \zeta^{(1)}, V_T^{(1)}, V_T^{(2)}\right) \sim \mathcal{N}(\underline{m}, \underline{s}^2) \tag{39}$$

where

$$\begin{aligned} &\underline{m} = Y_1 + Y_2V_T^{(1)} + Y_3\log V_T^{(2)} + Y_4\int_0^T V_s^{(1)}ds + Y_5\int_0^T \frac{ds}{V_s^{(2)}},\ \underline{s}^2 = Y_6\int_0^T V_s^{(1)}ds + Y_7\int_0^T \frac{ds}{V_s^{(2)}} \\ &Y_1 = rT - \frac{a\rho_1}{\sigma_1}V_0^{(1)} - k_1\theta_1T\frac{a\rho_1}{\sigma_1} - \frac{b\rho_2}{\sigma_2}\log V_0^{(2)} + \frac{b\rho_2}{\sigma_2}k_2T, \quad Y_2 = \frac{a\rho_1}{\sigma_1}, \quad Y_3 = \frac{b\rho_2}{\sigma_2} \\ &Y_4 = -\frac{a^2}{2} + \frac{ak_1\rho_1}{\sigma_1}, \quad Y_5 = -\frac{b^2}{2} + \frac{b\rho_2}{\sigma_2}\left(\frac{\sigma_2^2}{2} - k_2\sigma_2\right), \quad Y_6 = a^2(1-\rho_1^2), \quad Y_7 = b^2(1-\rho_2^2). \end{aligned}$$

[4] We make this choice for two reasons. (i) It is a common choice in the literature (e.g., Corsaro et al., 2019). (ii) The CGMY model is more computationally challenging than alternative Lévy models.

[5] Without sake of exhaustiveness, we refer to Ballotta and Rayée (2022), Brignone et al. (2023), Christoffersen et al. (2009) and Recchioni et al. (2021) for empirical studies on time series of option prices supporting multiple variance factors in the equity market and to Cortazar et al., 2017 for commodity markets.

[6] For sake of clarity and ease of exposition let us consider, following Grasselli (2017), the special case presented in (37)–(38). The approach we are going to propose would work also for the more general case which includes for example the FX Heston-based model of De Col et al. (2013) and the multi 3/2 model (Baldeaux et al., 2015).

The conditional characteristic function of the log-returns is

$$\hat{g}_{X_T|\zeta^{(1)},V_T^{(1)},V_T^{(2)}}(u) = \mathbb{E}\left[e^{iuX_T}|\zeta^{(1)},V_T^{(1)},V_T^{(2)}\right] = \mathbb{E}\left[e^{iu\left(\underline{m}+\frac{iu}{2}\underline{s}^2\right)}|\zeta^{(1)},V_T^{(1)},V_T^{(2)}\right]$$

$$= e^{iu(Y_1+Y_2V_T^{(1)}+Y_3\log V_T^{(2)})}\mathcal{L}_P\left(-iu\left(Y_4+\frac{iu}{2}Y_6\right)\right)\mathcal{L}\left(0,-iu\left(Y_5+\frac{iu}{2}Y_7\right)\right), \tag{40}$$

where $\mathcal{L}_P$ is as in (27) and $\mathcal{L}$ is as in (24). Given the characteristic function of $(X_T|\zeta^{(1)},V_T^{(1)},V_T^{(2)})$, to implement our method is necessary to compute high order moments. The nth cumulant of $(X_T|\zeta^{(1)},V_T^{(1)},V_T^{(2)})$ can be computed in the following way:

$$c_n = \frac{\partial^n}{\partial u^n}\log\mathbb{E}\left[e^{uX_T}|\zeta^{(1)},V_T^{(1)},V_T^{(2)}\right]\Big|_{u=0} = \underbrace{\frac{\partial^n}{\partial u^n}\left(Y_1+Y_2V_T^{(1)}+Y_3\log V_T^{(2)}\right)u\Big|_{u=0}}_{=:C_1(n)} +$$

$$\frac{\partial^n}{\partial u^n}\log\mathbb{E}\left[e^{\left(uY_4+\frac{u^2}{2}Y_6\right)\int_0^T V_s^{(1)}ds}\Big|\zeta^{(1)},V_T^{(1)}\right]\Big|_{u=0} + \frac{\partial^n}{\partial u^n}\log\mathbb{E}\left[e^{\left(uY_5+\frac{u^2}{2}Y_7\right)\int_0^T\frac{ds}{V_s^{(2)}}}\Big|V_T^{(2)}\right]\Big|_{u=0}$$

$$= C_1(n) + \underbrace{\frac{\partial^n}{\partial u^n}\log\left(\mathcal{L}_P(-uY_4-\frac{u^2}{2}Y_6)\right)\Big|_{u=0}}_{=:C_2(n)} + \underbrace{\frac{\partial^n}{\partial u^n}\log\left(\mathcal{L}(0,-uY_5-\frac{u^2}{2}Y_7)\right)\Big|_{u=0}}_{=:C_3(n)},$$

where

$$C_1(1) = Y_1+Y_2V_T^{(1)}+Y_3\log V_T^{(2)},\ C_1(n)=0 \text{ for } n>1. \tag{41}$$

$C_2(n)$ and $C_3(n)$ can be computed using Algorithm 4 by setting, respectively, $\bar{\mathcal{L}}(u) = \mathcal{L}_P\left(-uY_4-\frac{u^2}{2}Y_6\right)$ and $\bar{\mathcal{L}}(u) = \mathcal{L}\left(-uY_5-\frac{u^2}{2}Y_7\right)$. The proposed exact simulation scheme is given in Algorithm 8.

5.4.2. Multi-Heston model

Under the risk–neutral measure, the D-dimensional Heston model is given by[7]

$$\frac{dS_t}{S_t} = rdt + \sum_{d=1}^{D}\sqrt{V_t^{(d)}}\left(\rho_d dW_t^{(2d)} + \sqrt{1-\rho_d^2}dW_t^{(2d-1)}\right) \tag{42}$$

$$dV_t^{(d)} = k_d(\theta_d - V_t^{(d)})dt + \sigma_d\sqrt{V_t^{(d)}}dW_t^{(2d)}, \tag{43}$$

where all Brownian motions are mutually independent and $V^{(d)}$ are independent variance factors. Christoffersen et al. (2009) considered the case $D=2$, while, more recently, Recchioni et al. (2021) consider the cases $D=2$ and $D=3$. We propose, next, an exact simulation method for the model (42)–(43) for a general $D \in \mathbb{N}^+$. The content of this subsection is related to Brignone (2024), proposing an exact simulation scheme for the multifactor Ornstein–Uhlenbeck stochastic volatility model. However, we are considering a different model, a different simulation method, and a different approach for the computation of the conditional moments. As in (20), we define

$$\xi_1^{(d)} := \left(\frac{r}{D} - \frac{\rho_d k_d\theta_d}{\sigma_d}\right)T - \frac{\rho_d V_0^{(d)}}{\sigma_d},\quad \xi_2^{(d)} := \frac{\rho_d}{\sigma_d},\quad \xi_4^{(d)} := \frac{\rho_d k_d}{\sigma_d} - \frac{1}{2},\quad \xi_7^{(d)} := (1-\rho_d^2).$$

The conditional distribution of the log–returns is

$$\left(X_T\Big|\left\{V_T^{(d)},\int_0^T V_s^{(d)}ds\right\}_{d=1}^{D}\right) \sim \mathcal{N}(\underline{m},\underline{s}^2) \tag{44}$$

with

$$\underline{m} = \sum_{d=1}^{D}\xi_1^{(d)} + \xi_2^{(d)}V_T^{(d)} + \xi_4^{(d)}\int_0^T V_s^{(d)}ds,\ \underline{s}^2 = \sum_{d=1}^{D}\xi_7^{(d)}\int_0^T V_s^{(d)}ds.$$

Further define

$$\mathcal{L}_P^{(d)}(u) := \mathbb{E}\left[e^{-u\int_0^T V_s^{(d)}ds}|\zeta^{(d)},V_T^{(d)}\right] = \mathcal{L}_{X_1}^{(d)}(u)\mathcal{L}_{X_2}^{(d)}(u)\mathcal{L}_Z^{(d)}(u)^{\zeta^{(d)}} \tag{45}$$

where

$$\mathcal{L}_{X_1}^{(d)}(u) = \exp\left(\frac{V_0^{(d)}+V_T^{(d)}}{\sigma_d^2}\left(k_d\coth\left(\frac{k_dT}{2}\right) - \sqrt{2\sigma_d^2u+k_d^2}\coth\left(\frac{\sqrt{2\sigma_d^2u+k_d^2}T}{2}\right)\right)\right) \tag{46}$$

$$\mathcal{L}_{X_2}^{(d)}(u) = \left(\frac{\sqrt{2\sigma_d^2u+k_d^2}\sinh(k_dT/2)}{k_d\sinh(\sqrt{2\sigma_d^2u+k_d^2}T/2)}\right)^{\frac{d}{2}},\quad \mathcal{L}_Z^{(d)}(u) = \left(\frac{\sqrt{2\sigma_d^2u+k_d^2}\sinh(k_dT/2)}{k_d\sinh(\sqrt{2\sigma_d^2u+k_d^2}T/2)}\right). \tag{47}$$

[7] Note that with D-dimensional Heston model we mean a model with D independent variance factors and a single asset price process.

Table 2
Model parameter sets.

Heston										
V_0	k	θ	σ	ρ	a	b	r			
0.010201	6.21	0.019	0.61	−0.7	1	0	3.19%			
3/2 model										
V_0	k	θ	σ	ρ	a	b	r			
0.0432	4.8689	0.0912	0.6363	−0.9641	0	−0.0510	2%			
4/2 model										
V_0	k	θ	σ	ρ	a	b	r			
0.0201	5.1993	0.0371	0.4723	−0.8621	0.4600	−0.0390	2%			
CGMY-CIR										
V_0	k	θ	σ	a	b	C	G	M	Y	r
0.008836	3.99	0.014	0.27	1	0	15.684	10.2115	43.151	0.8	4%
Heston $\bigoplus$ 3/2 model										
d	$V_0^{(d)}$	k_d	θ_d	σ_d	ρ_d	a	b	r		
1	0.1	2.91	0.24	3.5	−0.7	0.74	−0.015	2%		
2	0.07	2.78	0.27	0.28	−0.4					
Double Heston										
d	$V_0^{(d)}$	k_d	θ_d	σ_d	ρ_d	r				
1	0.0028	1.0738	0.1026	0.826	−0.2819	3%				
2	0.0059	0.0326	0.7078	1.5355	−0.687					

Notes. The Heston, 3/2 and 4/2 models are defined in (16)–(17). The CGMY-CIR is defined in (33)–(17). The Heston $\bigoplus$ 3/2 model and Double Heston models are defined, respectively, in (37)–(38) and (42)–(43) with $D = 2$. For the Heston, Double Heston, and CGMYCIR the parameters are taken, respectively, from Broadie and Kaya (2006, Table 1), Kyriakou et al. (2024, Table 2), Corsaro et al. (2019, Table 2). For the Heston $\bigoplus$ 3/2 model, 3/2 and 4/2 models, parameters are obtained from a calibration on a set of quoted option prices (see Section EC.7). The initial price is $S_0 = 100$ for all the parameter sets and the maturity is kept fix (1 year maturity).

Applying the tower property, we obtain the conditional characteristic function of log–returns

$$\hat{g}_{X_T \mid \left\{\zeta^{(d)}, V_T^{(d)}\right\}_{d=1}^{D}}(u) = \mathbb{E}\left[e^{iuX_T} \middle| \left\{\zeta^{(d)}, V_T^{(d)}\right\}_{d=1}^{D}\right] = e^{iu\sum_{d=1}^{D}\xi_1^{(d)}+\xi_2^{(d)}V_T^{(d)}} \prod_{d=1}^{D} \mathcal{L}_P^{(d)}\left(-iu\xi_4^{(d)} + \frac{u^2}{2}\xi_7^{(d)}\right). \tag{48}$$

We need at this stage to compute the conditional cumulants of the log-returns:

$$c_n = \frac{\partial^n}{\partial u^n} \log \mathbb{E}\left[e^{uX_T} \middle| \left\{\zeta^{(d)}, V_T^{(d)}\right\}_{d=1}^{D}\right]\Big|_{u=0} = \underbrace{\sum_{d=1}^{D} \frac{\partial^n}{\partial u^n} u(\xi_1^{(d)} + \xi_2^{(d)}V_T^{(d)})\Big|_{u=0}}_{=:C_1(n)} +$$

$$\underbrace{\sum_{d=1}^{D} \frac{\partial^n}{\partial u^n} \log \mathcal{L}_P^{(d)}\left(-u\xi_4^{(d)} - \frac{u^2}{2}\xi_7^{(d)}\right)\Big|_{u=0}}_{=:C_2(n)} = \begin{cases} \sum_{d=1}^{D}\xi_1^{(d)} + \xi_2^{(d)}V_T^{(d)} & \text{if } n = 1 \\ 0 & \text{if } n > 1 \end{cases} + C_2(n). \tag{49}$$

$C_2(n)$ can be computed running Algorithm 4 for $d = 1, \dots, D$ with $\bar{\mathcal{L}}(u) = \mathcal{L}_P^{(d)}\left(-u\xi_4^{(d)} - \frac{u^2}{2}\xi_7^{(d)}\right)$. The proposed simulation scheme for the multi-Heston model is summarized into Algorithm 9.

6. Numerical studies

In this section, we evaluate the performances of the proposed simulation algorithms, benchmarking with previous literature. Additional numerical experiments are provided in Section EC.3. All the computations are done using Matlab®(Version R2024b) 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. The parameters considered for the various experiments are given in Table 2. Based on the results in Section EC.3.1, we take $\tilde{q} = 10$, $s = 19$ and $n = 6$ when implementing Algorithms 5–9. Following standard literature (among others, Broadie & Kaya, 2006), we evaluate numerical performances based on computing times, bias and Root Mean Squared Error (RMSE), that are defined as follows. If C is the simulation estimator used for the European call option price and C_{true} is the true price, then the bias, variance and RMSE are given as follows: bias = $(C_{\text{true}} - \mathbb{E}[C])$, variance = $\mathbb{E}[(C_{\text{true}} - \mathbb{E}[C])^2]$, RMSE = $\sqrt{\text{bias}^2 + \frac{\text{variance}}{\mathcal{M}}}$, where $\mathcal{M}$ is the number of simulations. The standard error (henceforth, s.e.) is computed as s.e. = $\sqrt{\text{variance}/\mathcal{M}}$. The bias is estimated running 2×10^9 simulations.

6.1. Assessing the true bias and comparison with traditional exact simulation scheme

We compare the numerical performances of the proposed conditional COS approach for the exact simulation of stochastic volatility models with the traditional exact simulation scheme (summarized in Sections 2 and EC.5) and also evaluate the actual bias. First, our method presents only one

Table 3
True bias (in absolute value) and standard errors of the Monte Carlo simulation estimator (in parenthesis) of the ATM European call option price.

Model	Algorithms 5–9			Traditional		
	$\varepsilon = 1$	$\varepsilon = 10^{-1}$	$\varepsilon = 10^{-2}$	$\hat{\zeta} = 10^{-2}$	$\hat{\zeta} = 10^{-3}$	$\hat{\zeta} = 10^{-4}$
Heston	0.0020	0.0005	0.0002	0.0098	0.0019	0.0006
	(1.66E−04)	(1.66E−04)	(1.66E−04)	(1.74E−04)	(1.66E−04)	(1.66E−04)
3/2	0.0010	0.0007	0.0003	0.0144	0.0047	0.0019
	(2.73E−04)	(2.73E−04)	(2.73E−04)	(2.77E−04)	(2.75E−04)	(2.74E−04)
4/2	0.0033	0.0015	0.0004	0.0268	0.0096	0.0062
	(2.60E−04)	(2.59E−04)	(2.59E−04)	(2.67E−04)	(2.76E−04)	(2.67E−04)
CGMYCIR	0.0005	0.0003	0.0002	0.0401	0.0120	0.0093
	(1.65E−04)	(1.65E−04)	(1.65E−04)	(1.66E−04)	(1.67E−04)	(1.65E−04)
Heston $\bigoplus$ 3/2	1.42E−04	1.18E−04	3.26E−05	0.0895	0.0141	0.0088
	(2.94E−04)	(2.93E−04)	(2.93E−04)	(3.18E−04)	(3.03E−04)	(2.98E−04)
Double Heston	0.0029	0.0003	0.0001	0.0028	0.0010	0.0005
	(3.19E−04)	(3.19E−04)	(3.20E−04)	(3.15E−04)	(3.20E−04)	(3.20E−04)

Notes. Parameters are as in Table 2. Reference price of the ATM European call options is: 6.80611 (Heston), 9.93256 (3/2), 10.40967 (4/2), 6.605306 (CGMYCIR), 10.44224 (Heston $\bigoplus$ 3/2), 9.36631 (Double Heston).

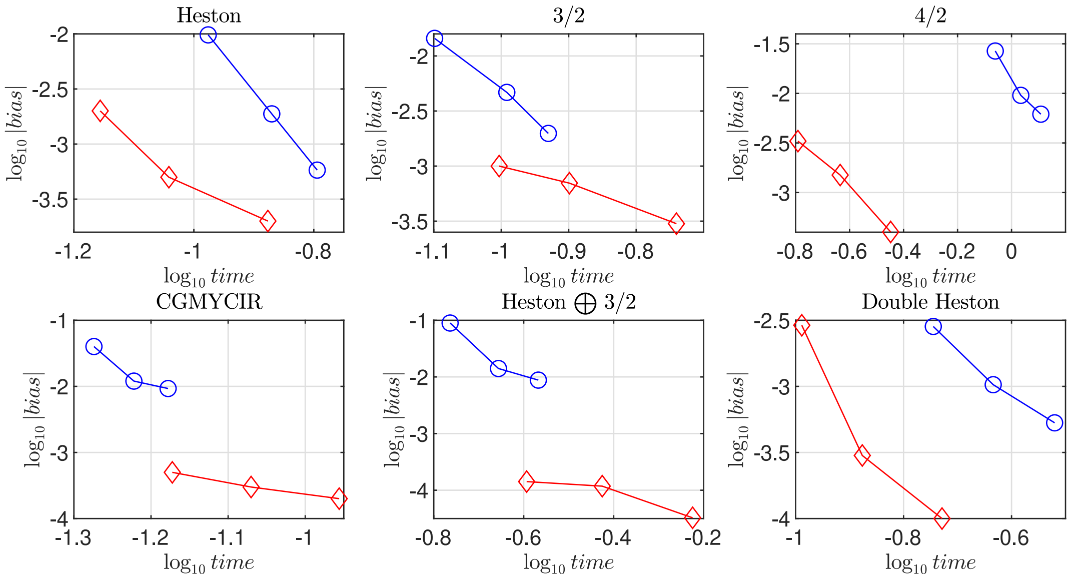

Fig. 1. Speed accuracy profiles: Algorithms 5–9 vs. Traditional exact simulation scheme (see Section 2).
Notes. Algorithms 5–9 plot with red diamond markers, benchmark (traditional method): plot with blue circle markers. Time is expressed in seconds. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

parameter controlling the error: ε. We take $\varepsilon \in \{1, 10^{-1}, 10^{-2}\}$ and run 2×10^9 simulations using Algorithms 5–9.[8] We report the biases in Table 3. Results show that the bias (in absolute value), as expected, decreases with ε. In addition, recalling that ε is maximum error tolerable in absolute value when pricing the European call option, the true bias appears to be much smaller than ε, even more than 1000 times smaller (depending on the model and the parameter set). In the same table, we also report the results of the same experiment with the traditional exact simulation scheme implemented with the Kang et al. (2017) adjustment (see Section 2). In this way, the error is controlled by three parameters, $\hat{\zeta}$, $\hat{c}_1$ and $\hat{c}_2$. Following the suggestion of Kang et al. (2017) we simply set $\hat{c}_1 = 0.1$ and $\hat{c}_2 = 0.5$ and study the bias for varying $\hat{\zeta}$. Results are reported in Table 3. Equipped with the bias, we can evaluate the speed-accuracy profiles. To this aim, we set $\mathcal{M} = 10^4$ and run Algorithms 5–9 and the traditional exact simulation scheme 50 times each, for the same values of ε and $\hat{\zeta}$ reported in Table 3. Then, we take the average computing times. Results are given in Fig. 1 and show that the proposed conditional COS approach largely outperforms the traditional exact simulation scheme and presents a superior runtime-accuracy trade-off. This is particularly evident for involved models such as CGMYCIR, Heston $\bigoplus$ 3/2 and Double Heston models.

6.2. Comparison with fast approximated sampling schemes

We compare our exact simulation approach with the alternative fast approximated sampling schemes proposed by Kyriakou et al. (2024) and Zeng et al. (2023). The speed-accuracy profile is evaluated by comparing the RMSE and the computing time. When implementing our conditional COS approach, we set the following tuning parameters: $n = 6$, $s = 19$, $\tilde{q} = 10$ and $\varepsilon = 10^{-1}$.

[8] We have also tested the methodology considering smaller values of ε (e.g., 10^{-3}), but the resulting bias is extremely small and hard to evaluate accurately since needing a huge number of simulations.

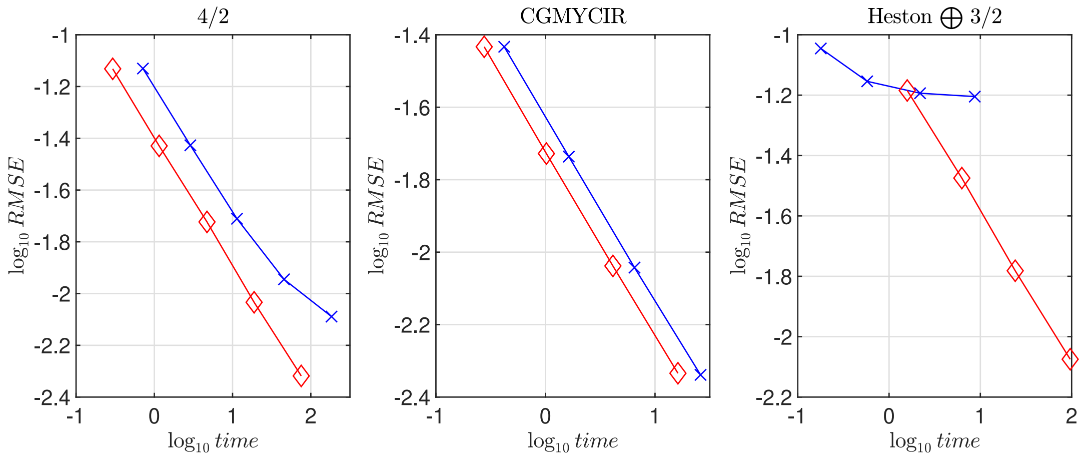

Fig. 2. Speed accuracy profiles: Algorithms 5–9 vs. Kyriakou et al. (2024).
Notes. Algorithms 5, 7, 8: plot with red diamond markers; method of Kyriakou et al. (2024): plot with blue cross markers. Time is expressed in seconds. Number of simulations: $\mathcal{M} = \tilde{m} \times 10^4$ with $\tilde{m} \in \{4, 16, 64, 256\}$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

6.2.1. *Comparison with Kyriakou et al. (2024)*

The methodology in Kyriakou et al. (2024) allows to obtain accurate estimates for the price of an option at a reduced computational cost. Additionally, it is very general and can be applied to all the models considered in this paper. To save space, we summarize the methodology in Section EC.6. However, our methodology is expected to perform better in most of the models considered in this paper, as we illustrate next. First, we consider the CGMYCIR model where the methodology of Kyriakou et al. (2024) suffers from some drawbacks. After simulating fast $\left(\int_0^T V_s ds, V_T\right)$ one still needs to simulate the log-returns. This means simulating the CGMY model, which can be done only by numerically inverting the characteristic function of the log-returns (Ballotta & Kyriakou, 2014). In this regard, it is clear that the benefit of simulating fast $\left(\int_0^T V_s ds, V_T\right)$ is reduced because after that one still needs to perform simulation through characteristic function inversion and root finding algorithms, introducing other sources of error (depending on which methodology is applied) and increasing the computing times. Similar reasoning applies to other Lévy models whose simulation is involved (e.g., Meixner, tempered stable, generalized hyperbolic models). Algorithm 7 is therefore expected to perform better in this case, since it still requires only one characteristic function inversion but does not require sampling the conditional integrated variance. Running 2×10^9 simulations through the method of Kyriakou et al. (2024), we obtained an European call option price equal to 6.6054 (5.8E–04). The benchmark methodology therefore appears very accurate under this model. However, when comparing the RMSE and the computing time (see central subplot of Fig. 2) for various number of simulations it is clear that Algorithm 7 outperforms, being faster for the same level of accuracy. We stress that the benchmark methodology of Kyriakou et al. (2024) is very effective. Indeed, it is very fast and the bias is small (around 10^{-4} for the considered parameters). Thus, it is very difficult to beat it. Even if the difference between the two methods is very small, the improvement is still remarkable considering the general excellent performances of the benchmark method. Second, we consider the 4/2 model. To facilitate comparison with Kyriakou et al. (2024), let us consider for this model the same parameters used in their original paper, i.e.: $V_0 = 0.04$, $k = 1.8$, $\theta = 0.04$, $\sigma = 0.2$, $\rho = -0.7$, $a = 0.3$, $b = 0.025$, $r = 2\%$.[9] As reported in Kyriakou et al. (2024, Table 3), with such parameters the absolute bias in their approach is equal to 0.0067. In contrast, through Algorithm 5, we obtained an absolute bias equal to 0.0002. Equipped with the bias, we compute the RMSE and take computing times for varying number of simulations. Results are reported in the left subplot of Fig. 2. Also in this case, we find that our proposed Algorithm 5 outperforms the benchmark. Third, let us consider the Heston ⊕ 3/2 model. We implement the methodology of Kyriakou et al. (2024) for the simulation of such model and estimate the bias using 2×10^9. The bias is very large. Indeed, the option price computed via the methodology of Kyriakou et al. (2024) is: 10.3804 (2.97E–04). We report the RMSE and computing times in the right subplot of Fig. 2. Results indicate that our approach largely outperforms the benchmark method of Kyriakou et al. (2024) under the Heston ⊕ 3/2 model.

Based on these results, the advantages of the proposed method over the one proposed in Kyriakou et al. (2024) can be summarized as follows: (i) it allows for a simplified control of the error; (ii) it presents a better runtime-accuracy profile when considering the simulation of very involved models with challenging parametrizations, such as CGMYCIR, 4/2 and Heston ⊕ 3/2 models.

6.2.2. *Comparison with Zeng et al. (2023)*

Under the models (16)–(17) and (33)–(17), Zeng et al. (2023, Eq. 28) suggest approximating the conditional cumulative distribution function of the log-returns via Hilbert transform:

$$G_{X_T|V_T}(y) \approx \tilde{G}_{X_T|V_T}(y) := \frac{1}{2} + \frac{i}{2} \sum_{m=-M}^{M} \frac{e^{-i(m-1/2)hy}}{(m-1/2)\pi} \hat{g}_{X_T|V_T}(i(m-1/2)h), \tag{50}$$

where h is the discretization step size and M is the truncation level, that are the first two parameters controlling the error. Given (50), the simulation is performed as follows. (i) Truncate and discretize the range of X_T into $\tilde{K}+1$ levels from x_0 to $x_{\tilde{K}}$; (ii) Truncate and discretize the range of $\log V_T$ into $\tilde{L}+1$ levels from $\log v_0$ to $\log v_{\tilde{L}}$; (iii) Evaluate the conditional characteristic function $\hat{g}_{X_T|V_T=v_{\tilde{L}}}(i(m-1/2)h)$ and the approximated conditional

[9] When considering the model parameters reported in Table 2 for the 4/2 model, we found a very large bias when computing option prices using the methodology of Kyriakou et al. (2024). We have thus opted for considering the same model parameters considered in the original paper to facilitate comparison.

Table 4
Choice of the parameters controlling the error when implementing the method of Zeng et al. (2023) for benchmark comparison.

Model	$\log v_0$	$\log v_{\tilde{L}}$	$\tilde{L}$	x_0	$x_{\tilde{K}}$	$\tilde{K}$	M	Bias
3/2	−7.6933	−1.1708	200	0.6931	8.5172	2001	1000	−0.0194
4/2	−9.1916	−1.6187	200	0.6931	8.5172	961	480	0.0038

cumulative distribution function $\tilde{G}_{X_T|V_T}(x_{\tilde{K}})$ for $k = 1, \dots, \tilde{K}$, $l = 1, \dots, \tilde{L}$ and $m = -M, \dots, M$ from (50); (iv) Generate a sample from $(V_T|V_0)$ from (18); (v) Simulate $(X_T|V_T)$ using enhanced interpolation techniques (see Zeng et al., 2023, Eqs. 29-30 and Remark 3.1).

This technique is extremely fast since steps (i)–(iii) can be performed outside the Monte Carlo loop (so called, initialization) and because step (v) is based on interpolation techniques, that are faster than using root finding algorithms. Therefore, this methodology is expected to be faster than our method for the same number of simulations (especially for a high number of simulations) and constitutes a very solid benchmark for our approach. As we discuss below, the main advantage of our proposed approach is a simplified control of the error. Indeed, under the methodology of Zeng et al. (2023), the error is controlled by 7 parameters, i.e., $\tilde{K}$, $\tilde{L}$, x_0, $x_{\tilde{K}}$, v_0, $v_{\tilde{L}}$, M.[10] In contrast, with our method the error is controlled by only one parameter. Additionally, Zeng et al. (2023, Theorem 4.5) present a computable upper error bound for the bias implicit in their methodology. In principle, one could compute such bound for various combinations of the seven parameters controlling the error until finding sufficient accuracy. However, it is clear that Zeng et al. (2023, Formula 41) is challenging to implement and potentially slow to run. In contrast, in our method the upper error bound for the bias is an input specified by the user. In addition, we stress that our method is better extendable to multifactor models. Consider for example a model with D variance factors. To implement the method of Zeng et al. (2023), one should discretize the D variance processes, introducing $D \times 3$ additional parameters controlling the error. In total, one would have $4 + 3 \times D$ parameters governing the error. Thus, for the triple Heston model (i.e., $D = 3$), one has 13 parameters controlling the error. In contrast, with our proposed method the error is controlled by only one parameter, independently on the number of factors D.

Finally, we proceed with a numerical comparison between the two methodologies.[11] We consider the 3/2 and 4/2 models, with the parameters reported in Table 2. We start by computing the bias. First, "playing" with the aforementioned tuning parameter set, we find an optimal set up. The tuning parameters are reported in Table 4. Note that this step can be rather time consuming since such parameters are hard to determine via trial and error. Moreover, optimal tuning parameters may differ substantially when considering different models. Following Zeng et al. (2023), we have set $\tilde{K} = 2M + 1$ and determined the lower and upper bound of the variance, v_0 and $v_{\tilde{L}}$, by Monte Carlo simulation taking the maximum and minimum value of the log-variance after 10^7 simulations. Moreover, $[x_0, x_{\tilde{K}}] := \log(100) \mp \log(50)$. In the case of the 4/2 model we found numerical issues when setting $M > 480$. This is why we do not further increase M to increase accuracy. Second, given the optimal tuning parameters we compute the bias using 10^9 simulations (it is reported in Table 4). We notice that, despite the abundant usage of approximations, the methodology is extremely accurate, especially in the 4/2 model. However, the bias is larger compared to Algorithm 5, as expected since our method can be considered exact and limits the amount of numerical approximations. Given the bias, we can study the speed accuracy trade-off. More specifically, we compute RMSE and running times in seconds for various number of simulations. Results are reported (in log–log scale) in Fig. 3, and show that our approach largely outperforms that of Zeng et al. (2023) under the parametrization considered for the 3/2 model. For what concerns the 4/2 model, our approach outperforms only when a high number of simulations (and thus high accuracy) is required. We find that when high accuracy is not necessary, the methodology of Zeng et al. (2023) is preferable because it runs faster. However, when increasing the number of simulations, Algorithm 5 outperforms the method of Zeng et al. (2023) since the standard error of the Monte Carlo simulation estimator for the option price decreases to 0 and the RMSE coincides with the bias. Since the bias obtained through Algorithm 5 is smaller, our approach outperforms (Zeng et al., 2023) for a high number of simulations.

In summary, our method is expected to be slower (for the same number of simulations) than the one proposed by Zeng et al. (2023), especially for a high number of simulations. However, our method presents the following advantages. (i) It allows for an improved control of the error, which is controlled by only one parameter, instead of seven. (ii) It is easier to implement, in the sense that the user must specify a smaller number of tuning parameters and the error upper bound is supplied by the user, which is easier than estimating it numerically through Zeng et al. (2023, Formula 41) or by trial and error for various model parameters configurations. (iii) It is better extendable to multifactor models, in the sense that for a D factor model the error is still controlled by only one parameter instead of $4 + 3 \times D$. (iv) Under challenging model parameters, our approach presents a smaller bias (in principle, the final user can set the maximum tolerable bias as small as desired), entailing an improved speed-accuracy profile when a high number of simulations is employed.

6.3. *Pricing forward starting options using Corollary 4*

Generally, when pricing European options it is possible to obtain reasonable estimates of the true option price and thus computing the bias of a simulation scheme. However, when pricing forward starting options, the true price is not available, and thus it is not possible to estimate the bias (e.g., Broadie & Kaya, 2006, Section 7). However, through Corollary 4, it is still possible to easily control the error when pricing forward starting options even if the true price is not known. Indeed, note that through Corollary 4 it is possible to obtain a price approximation that differs by no more than ϵ with probability q from the true price (both q and ϵ are supplied by the user, rendering the error control very easy).

As illustrative examples, let us consider the Heston, CGMYCIR and 4/2 models. Following Broadie and Kaya (2006) and Li and Wu (2019), we consider two dates: $T_1 = 1$ and $T_2 = 2$ and price forward starting call options with payoff: $e^{-rT_2} \max(0, S_{T_2} - S_{T_1})$. We set the following parameters: $s_1 = s_2 = 19$, $\tilde{q}_1 = \tilde{q}_2 = 10$, $n_1 = n_2 = 6$. Then, we set two levels of the probability $q \in \{0.85, 0.95\}$ and maximum error tolerance $\epsilon \in \{0.1, 0.01\}$. In Table 5 we report the price of the forward starting options, along with computing times and the number of simulations obtained through Corollary 4. Some comments are in order. To implement our methodology, we compute the price of the forward starting by pricing an European call option at

[10] Following Zeng et al. (2023), for numerical experiments we set $h = \frac{2\pi}{x_{\tilde{K}} - x_0}$. Theoretically, h could be considered as an additional parameter controlling the error since alternative choices are possible.

[11] To implement the methodology of Zeng et al. (2023), we have used the codes available at https://github.com/zxubl/Option-Pricing-Hilbert-Interpolation.

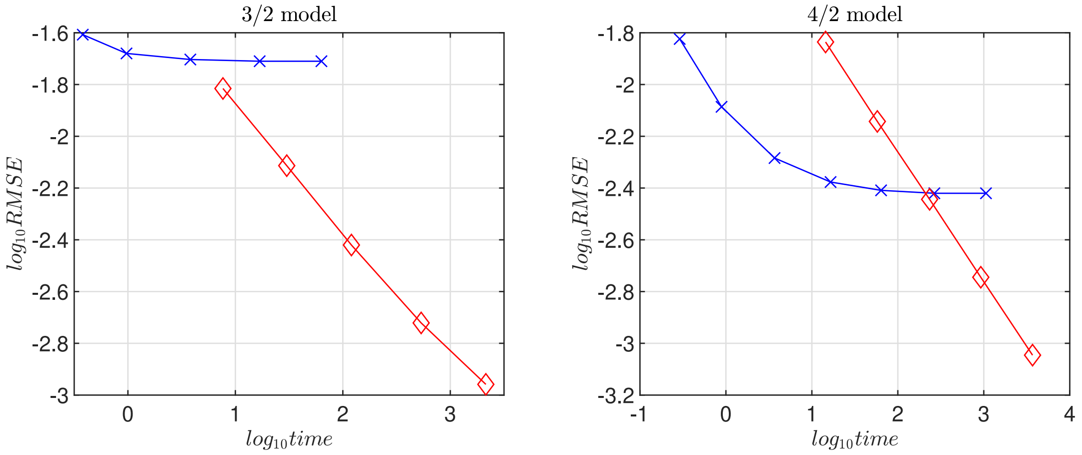

Fig. 3. Speed accuracy profiles: Algorithm 5 vs. Zeng et al. (2023).
Notes. Algorithm 5: plot with red diamond markers; method of Zeng et al. (2023): plot with blue cross markers. Time is expressed in seconds. Number of simulations: $\mathcal{M} = \tilde{m} \times 10^4$ with $\tilde{m} \in \{64, 256, 1024, 4096, 16384\}$. Additionally, for the method of Zeng et al. (2023) under the 4/2 model we also show the RMSE and computing time for $\mathcal{M} = \tilde{m} \times 4096 \times 10^4$ with $\tilde{m} \in \{16, 64\}$ simulations. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 5
Simulation estimates for a forward starting option based on Corollary 4.

ε	q	Heston			CGMYCIR			4/2		
		$\mathcal{M}$	Price	Time	$\mathcal{M}$	Price	Time	$\mathcal{M}$	Price	Time
0.1	0.85	1256	6.9686	0.04	1423	6.7764	0.04	51 231	9.9272	2.64
0.01	0.85	125 189	6.9555	9.49	135 567	6.7853	9.25	4 791 221	9.9156	538.19
0.1	0.95	2342	6.9608	0.06	2583	6.7886	0.06	98 009	9.9406	4.96
0.01	0.95	230 611	6.9530	12.52	252 011	6.7802	11.66	9 449 077	9.9148	790.43

Notes. Time is expressed in seconds. s.e. denotes the standard error. Parameters as in Table 2.

time T_1 given the variance level and the asset price at time T_1. To price efficiently European options in this case, we utilize the approach of Junike (2024) from which we can control the error. To implement the methodology of Junike (2024) in our context, we need (i) the characteristic function of $(X_{T_2}|X_{T_1}, V_{T_1})$; (ii) the high order cumulants of $(X_{T_2}|X_{T_1}, V_{T_1})$. Both can be easily obtained under the Heston and CGMYCIR model. We take the characteristic function from Cui et al. (2017) and compute cumulants following the same approach of Feunou and Okou (2018) (see also Brignone & Sgarra, 2020, p. 109 and Brignone et al., 2023, Appendix B). Similar reasoning applies also to the Double Heston model. For the other models considered in this paper (i.e., 3/2, 4/2 and Heston $\bigoplus$ 3/2), this approach is not very efficient because the required characteristic function depends on the hypergeometric function. Since the European option price must be computed by characteristic function inversion for each simulation, this algorithm will be slow. Thus, for such models, we suggest a different approach. We simulate first $(V_{T_2}|V_{T_1})$ and then price for each simulation the European call option given the characteristic function of $(X_{T_2}|X_{T_1}, V_{T_2}, V_{T_1})$. In this way, the relevant characteristic function does not depend anymore on the hypergeometric function, and computing time will be drastically reduced. Results reported in Table 5 show the effectiveness of our approach when pricing forward starting options, with small computing times for the Heston and CGMYCIR models. Computing times increase drastically when considering the 4/2 model. This is for two main reasons: (i) the necessity of evaluating the modified Bessel function of the first kind when pricing the European call option given the characteristic function of $(X_{T_2}|X_{T_1}, V_{T_2}, V_{T_1})$; (ii) the higher number of simulations required to obtain the price estimate. The latter is certainly more important. When we price forward starting options by simulating the stock price and the variance values at time T_1, and then using the European option pricing formula to evaluate the option payoff, we eliminate the variance that occurs due to the uncertainty after T_1. However, when pricing options conditioning also on V_{T_2} we are introducing uncertainty after T_1. Therefore, this estimator is expected to have higher variance than when using directly European option pricing formula without simulating V_{T_2} (we refer to Broadie & Kaya, 2006, Section 7 for a similar discussion).

6.4. A variant of Algorithms 5 –9: fast approximated sampling scheme for path-dependent options based on the moments-matched metalog

While exact simulation schemes are theoretically appealing, they are difficult to implement in practice for pricing path dependent options via Monte Carlo simulation, especially for path-dependent derivatives whose payoff is a function of frequently observed prices along the path where the simulation requires multiple steps with small time increments. For example Li and Wu (2019, Table 5) price Asian, Barrier and forward starting options by exact simulation, but they consider very few monitoring dates. Under these circumstances, exact simulation is potentially inferior to other approximate schemes, which are less accurate but much faster. For smaller time gaps (and larger numbers of discretization steps), exact simulation is slower. Therefore, in what follows, we illustrate the numerical performances of an alternative simulation scheme which is obtained as special case of Algorithms 5–9. More specifically, we simulate log-returns conditionally on the variance via the moment-matched metalog distribution (i.e., skipping the COS method).[12] Since the time consuming path-wise Fourier inversion step is sidestepped with such an approach, the resulting

[12] We thank an anonymous referee for suggesting us to test such simulation approach.

Table 6
Pricing discretely monitored fixed strike Asian call options using the metalog approximation.

	Heston			3/2			4/2		
	Price	s.e.	Time	Price	s.e.	Time	Price	s.e.	Time
European	6.8066	0.0235	0.51	9.9452	0.0386	2.40	10.8602	0.0399	2.35
Asian	3.5680	0.0122	–	5.7177	0.0217	2.40	6.0784	0.0214	–
	CGMYCIR			Heston $\bigoplus$ 3/2			Double Heston		
	Price	s.e.	Time	Price	s.e.	Time	Price	s.e.	Time
European	6.6227	0.0260	0.71	11.6545	0.0440	2.51	9.3672	0.0466	0.49
Asian	3.4615	0.0136	–	6.0875	0.0225	2.51	4.3884	0.0200	–

Notes. Time is expressed in seconds. s.e. denotes the standard error. Parameters as in Table 2. Additional parameters: $K = 100$, $m = 12$. Prices, s.e. and computing time refer to $\mathcal{M} = 10^5$ simulations. True prices for European options are reported in Table 3. The true price of the Asian option under the Heston model is equal to 3.5665 (see Fusai & Kyriakou, 2016).

simulation scheme is extremely fast. Moreover, it is numerically stable and can be applied for pricing path-dependent options with many monitoring dates. To illustrate performances, we consider the problem of pricing Asian options whose discounted payoff is $e^{-rt_m}\max\left(0, \frac{1}{m+1}\sum_{j=0}^{m} S_{t_j} - K\right)$, where m is the number of monitoring dates. In Table 6, we report European and Asian options prices (along with the computing times) computed with such an approach.[13] We notice that the methodology is generally quite accurate, except for the 4/2 and Heston $\bigoplus$ 3/2 model. This is not surprising based on the numerical results presented in Table C.7, where we find that for such models the metalog approximation (still being more accurate than the normal distribution) presents smaller accuracy than the other models. In conclusion, the proposed methodology gives a reasonably accurate estimate of path–dependent options at a small computing time.

7. Conclusions

In this paper, we present a new exact simulation scheme for analytically solvable (in the sense of Zeng et al., 2023) stochastic volatility option pricing models based on the COS method. Building blocks of the new algorithm are the COS method (Fang & Oosterlee, 2009; Junike, 2024; Junike & Pankrashkin, 2022), the use of the metalog distribution to speed up the root-finding algorithms, a fast method for computing the conditional cumulants of log-returns that significantly extends the one proposed in Kyriakou et al. (2024). The new algorithm improves the traditional exact simulation scheme (e.g., Broadie & Kaya, 2006) in several directions. First, the maximum error tolerance is an input specified by the end user, rather than having to be numerically estimated through time-consuming numerical simulations. Second, it provides better numerical performance than the traditional exact simulation scheme. Third, the error analysis is extended to forward starting options, a task that is unreachable with previous methods. In addition, we propose exact simulation schemes for the Heston $\bigoplus$ 3/2 model and the Multi-Heston model. The methodology can be applied to simulate models other than the ones considered in this paper, see Section EC.8 for more details. Finally, we show that the proposed approach exhibits an improved speed-accuracy profile even when compared to state-of-the-art fast approximated sampling schemes, such as Kyriakou et al. (2024) and Zeng et al. (2023). The main limitation of our approach is that our error analysis does not extend to models with jumps in the variance process (see the discussion in Section EC.8) and to path-dependent options with more involved payoff than forward starting. Extending such error analysis to more complicated path-dependent options, such as Asian or barrier options, and models with jumps in the variance process, is very complicated and beyond the scope of this work. However, it will be the subject of future research. Additionally, Corollaries 3 and 4 links the maximum error tolerance with the variance of the Monte Carlo simulation estimator and the minimal number of simulations required to be sure at a given probability that the error is below the tolerance. Therefore, the computing time depends on the variance of the Monte Carlo simulation estimators. It would be interesting to study variance reduction techniques to speed up the pricing procedure. This would be particularly useful when pricing path-dependent options and will be considered as future research.

CRediT authorship contribution statement

Riccardo Brignone: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Software, Validation, Writing – original draft, Writing – review & editing. **Gero Junike:** Conceptualization, Formal analysis, Investigation, Methodology, Project administration, Validation, Writing – original draft, Writing – review & editing.

Acknowledgments

We thank the Editor and four anonymous reviewers for fruitful comments that undoubtedly helped improve our paper. We are thankful to Pingping Jiang for fruitful discussions and the provision of code material.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.ejor.2025.08.061.

[13] European option prices are reported to check the accuracy of the proposed approach.

References

Abate, J., & Whitt, W. (1992). The Fourier-series method for inverting transforms of probability distributions. *Queueing Systems*, *10*(1), 5–88.
Baldeaux, J. (2012). Exact simulation of the $3/2$ model. *International Journal of Theoretical and Applied Finance*, *15*.
Baldeaux, J., Grasselli, M., & Platen, E. (2015). Pricing currency derivatives under the benchmark approach. *Journal of Banking and Finance*, *53*, 34–48.
Ballotta, L., & Kyriakou, I. (2014). Monte Carlo simulation of the CGMY process and option pricing. *Journal of Futures Markets*, *34*(12), 1095–1121.
Ballotta, L., & Rayée, G. (2022). Smiles & smirks: Volatility and leverage by jumps. *European Journal of Operational Research*, *298*(3), 1145–1161.
Brignone, R. (2024). Exact simulation of the multifactor Ornstein–Uhlenbeck driven stochastic volatility model. *SIAM Journal on Scientific Computing*, *46*(3), A1441–A1460.
Brignone, R., & Gonzato, L. (2024). Exact simulation of the Hull and White stochastic volatility model. *Journal of Economic Dynamics & Control*, *163*, Article 104861.
Brignone, R., Gonzato, L., & Lütkebohmert, E. (2023). Efficient Quasi-Bayesian estimation of affine option pricing models using risk-neutral cumulants. *Journal of Banking and Finance*, *148*, Article 106745.
Brignone, R., & Sgarra, C. (2020). Asian options pricing in Hawkes-type jump-diffusion models. *Annals of Finance*, *16*(1), 101–119.
Broadie, M., & Kaya, Ö. (2006). Exact simulation of stochastic volatility and other affine jump diffusion processes. *Operations Research*, *54*(2), 217–231.
Cai, N., Song, Y., & Chen, N. (2017). Exact simulation of the SABR model. *Operations Research*, *65*(4), 931–951.
Callegaro, G., Fiorin, L., & Grasselli, M. (2019). Quantization meets Fourier: a new technology for pricing options. *Annals of Operations Research*, *282*, 59–86.
Carr, P., Geman, H., Madan, D., & Yor, M. (2003). Stochastic volatility for Lévy processes. *Mathematical Finance*, *13*(3), 345–382.
Choi, J., & Kwok, Y. (2024). Simulation schemes for the Heston model with Poisson conditioning. *European Journal of Operational Research*, *314*(1), 363–376.
Choudhury, G. L., & Lucantoni, D. M. (1996). Numerical computation of the moments of a probability distribution from its transform. *Operations Research*, *44*(2), 368–381.
Christoffersen, P., Heston, S., & Jacobs, K. (2009). The shape and term structure of the index option smirk: Why multifactor stochastic volatility models work so well. *Management Science*, *55*(12), 1914–1932.
Corsaro, S., Kyriakou, I., Marazzina, D., & Marino, Z. (2019). A general framework for pricing Asian options under stochastic volatility on parallel architectures. *European Journal of Operational Research*, *272*, 1082–1095.
Cortazar, G., Lopez, M., & Naranjo, L. (2017). A multifactor stochastic volatility model of commodity prices. *Energy Economics*, *67*, 182–201.
Cui, Y., del Bano Rollin, S., & Germano, G. (2017). Full and fast calibration of the Heston stochastic volatility model. *European Journal of Operational Research*, *263*(2), 625–638.
De Col, A., Gnoatto, A., & Grasselli, M. (2013). Smiles all around: FX joint calibration in a Multi-Heston model. *Journal of Banking and Finance*, *37*(10), 3799–3818.
Fabozzi, F. J., Recchioni, M., & Renò, R. (2025). Fifty years at the interface between financial modeling and operations research. *European Journal of Operational Research*.
Fang, F., & Oosterlee, C. (2009). A novel pricing method for European options based on Fourier-Cosine series expansions. *SIAM Journal on Scientific Computing*, *31*(2), 826–848.
Feunou, B., & Okou, C. (2018). Risk-neutral moment-based estimation of affine option pricing models. *Journal of Applied Econometrics*, *33*, 1007–1025.
Fusai, G., & Kyriakou, I. (2016). General optimized lower and upper bounds for discrete and continuous arithmetic Asian options. *Mathematics of Operations Research*, *41*(2), 531–559.
Gil-Pelaez, J. (1951). Note on the inversion theorem. *Biometrika*, *38*(3–4), 481–482.
Gnoatto, A., Grasselli, M., & Platen, E. (2022). Calibration to FX triangles of the 4/2 model under the benchmark approach. *Decisions in Economics and Finance*, *45*(1), 1–34.
Grasselli, M. (2017). The 4/2 stochastic volatility model: A unified approach for the Heston and the 3/2 model. *Mathematical Finance*, *27*(4), 1013–1034.
Hughett, P. (1998). Error bounds for numerical inversion of a probability characteristic function. *SIAM Journal on Numerical Analysis*, *35*(4), 1368–1392.
Junike, G. (2024). On the number of terms in the COS method for European option pricing. *Numerische Mathematik*, *156*, 533–564.
Junike, G., & Pankrashkin, K. (2022). Precise option pricing by the COS method - How to choose the truncation range. *Applied Mathematics and Computation*, *421*, Article 126935.
Junike, G., & Stier, H. (2023). The multidimensional COS method for option pricing. arXiv preprint arXiv:2307.12843.
Kang, C., Kang, W., & Lee, J. (2017). Exact simulation of the Wishart multidimensional stochastic volatility model. *Operations Research*, *65*(5), 1190–1206.
Keelin, T. W. (2016). The metalog distributions. *Decision Analysis*, *13*(4), 243–277.
Kyriakou, I., Brignone, R., & Fusai, G. (2024). Unified moment–based modelling of integrated stochastic processes. *Operations Research*, *72*(4), 1630–1653.
Li, C., & Wu, L. (2019). Exact simulation of the Ornstein-Uhlenbeck driven stochastic volatility model. *European Journal of Operational Research*, *275*, 768–779.
Recchioni, M., Iori, G., Tedeschi, G., & Ouellette, M. (2021). The complete Gaussian kernel in the multi-factor Heston model: Option pricing and implied volatility applications. *European Journal of Operational Research*, *293*(1), 336–360.
Yuan, L., & Kalbfleisch, J. (2000). On the Bessel distribution and related problems. *Annals of the Institute of Statistical Mathematics*, *52*, 438–447.
Yuan, D., & Lei, L. (2016). Some results following from conditional characteristic functions. *Communications in Statistics. Theory and Methods*, *45*(12), 3706–3720.
Zeng, P., Xu, Z., Jiang, P., & Kwok, Y. K. (2023). Analytical solvability and exact simulation in models with affine stochastic volatility and Lévy jumps. *Mathematical Finance*, *33*(3), 842–890.