Hierarchical adaptive sparse grids and quasi-Monte Carlo for option pricing under the rough Bergomi model

CHRISTIAN BAYER†, CHIHEB BEN HAMMOUDA ⬤*‡ and RAÚL TEMPONE‡§

†Weierstrass Institute for Applied Analysis and Stochastics (WIAS), Berlin, Germany
‡Computer, Electrical and Mathematical Sciences & Engineering Division (CEMSE), King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia
§Alexander von Humboldt Professor in Mathematics for Uncertainty Quantification, RWTH Aachen University, Aachen, Germany

The rough Bergomi (rBergomi) model, introduced recently in Bayer *et al.* [Pricing under rough volatility. *Quant. Finance*, 2016, **16**(6), 887–904], is a promising rough volatility model in quantitative finance. It is a parsimonious model depending on only three parameters, and yet remarkably fits empirical implied volatility surfaces. In the absence of analytical European option pricing methods for the model, and due to the non-Markovian nature of the fractional driver, the prevalent option is to use the Monte Carlo (MC) simulation for pricing. Despite recent advances in the MC method in this context, pricing under the rBergomi model is still a time-consuming task. To overcome this issue, we have designed a novel, hierarchical approach, based on: (i) adaptive sparse grids quadrature (ASGQ), and (ii) quasi-Monte Carlo (QMC). Both techniques are coupled with a Brownian bridge construction and a Richardson extrapolation on the weak error. By uncovering the available regularity, our hierarchical methods demonstrate substantial computational gains with respect to the standard MC method. They reach a sufficiently small relative error tolerance in the price estimates across different parameter constellations, even for very small values of the Hurst parameter. Our work opens a new research direction in this field, i.e. to investigate the performance of methods other than Monte Carlo for pricing and calibrating under the rBergomi model.

Keywords: Rough volatility; Monte Carlo; Adaptive sparse grids; Quasi-Monte Carlo; Brownian bridge construction; Richardson extrapolation

2010 Mathematics Subject Classification: 91G60, 91G20, 65C05, 65D30, 65D32

1. Introduction

Modeling volatility to be stochastic, rather than deterministic as in the Black-Scholes model, enables quantitative analysts to explain certain phenomena observed in option price data, in particular the implied volatility smile. However, this family of models has a main drawback in failing to capture the true steepness of the implied volatility smile close to maturity. Jumps can be added to stock price models to overcome this undesired feature, for instance by modeling the stock price process as an exponential Lévy process. However, the addition of jumps to stock price processes remains controversial (Christensen *et al.* 2014, Bajgrowicz *et al.* 2015).

Motivated by the statistical analysis of realized volatility by Gatheral *et al.* (2018) and the theoretical results on implied volatility (Alòs *et al.* 2007, Fukasawa 2011), rough stochastic volatility has emerged as a new paradigm in quantitative finance, overcoming the observed limitations of diffusive stochastic volatility models. In these models, the trajectories of the volatility have lower Hölder regularity than the trajectories of standard Brownian motion (Bayer *et al.* 2016, Gatheral *et al.* 2018). In fact, they are based on fractional Brownian motion (fBm), which is a centered Gaussian process whose covariance structure depends on the so-called Hurst parameter, H (we refer to Mandelbrot and Van Ness 1968, Coutin 2007, Biagini *et al.* 2008 for more details regarding fBm processes). In the rough volatility case, where $0 < H < 1/2$, the fBm has negatively correlated increments

*Corresponding author. Email: chiheb.benhammouda@kaust.edu.sa

and rough sample paths. Gatheral *et al.* (2018) empirically demonstrated the advantages of such models. For instance, they showed that the log-volatility in practice has a similar behavior to fBm with the Hurst exponent $H \approx 0.1$ at any reasonable time scale (see also Gatheral *et al.* 2014). These results were confirmed by Bennedsen *et al.* (2016), who studied over a thousand individual US equities and showed that H lies in $(0, 1/2)$ for each equity. Other works (Bayer *et al.* 2016, Bennedsen *et al.* 2016, Gatheral *et al.* 2018) showed additional benefits of such rough volatility models over standard stochastic volatility models in terms of explaining crucial phenomena observed in financial markets.

The rough Bergomi (rBergomi) model, proposed by Bayer *et al.* (2016), was one of the first developed rough volatility models. This model, which depends on only three parameters, shows a remarkable fit to empirical implied volatility surfaces. The construction of the rBergomi model was performed by moving from a physical to a pricing measure and by simulating prices under the model to fit the implied volatility surface well in the case of the S&P 500 index with only a few parameters. The model may be seen as a non-Markovian extension of the Bergomi variance curve model (Bergomi 2005).

Despite the promising features of the rBergomi model, pricing and hedging under such a model still constitutes a challenging and time-consuming task due to the non-Markovian nature of the fractional driver. In fact, standard numerical pricing methods, such as PDE discretization schemes, asymptotic expansions and transform methods, although efficient in the case of diffusions, are not easily carried over to the rough setting (with the remarkable exception of the rough Heston model (El Euch and Rosenbaum 2018, El Euch *et al.* 2018, Abi Jaber 2019, El Euch and Rosenbaum 2019) and its affine extensions (Gatheral and Keller-Ressel 2019, Jaber *et al.* 2019)). Furthermore, due to the lack of Markov and affine structure, conventional analytical pricing methods do not apply. To the best of our knowledge, the only prevalent method for pricing options under such models is Monte Carlo (MC) simulation. In particular, recent advances in simulation methods for the rBergomi model and different variants of pricing methods based on MC under such a model have been proposed in Bayer *et al.* (2016, 2017), Bennedsen *et al.* (2017), McCrickerd and Pakkanen (2018), and Jacquier *et al.* (2018a). For instance, in McCrickerd and Pakkanen (2018), the authors use a novel composition of variance reduction methods. When pricing under the rBergomi model, they achieved substantial computational gains over the standard MC method. Greater analytical understanding of option pricing and implied volatility under this model has been achieved in Jacquier *et al.* (2018b), Forde and Zhang (2017), and Bayer *et al.* (2019). It is crucial to note that hierarchical variance reduction methods, such as multi-level Monte Carlo (MLMC), are inefficient in this context, because of the poor behavior of the strong error which is the order of H (Neuenkirch and Shalaiko 2016).

Despite recent advances in the MC method, pricing under the rBergomi model is still computationally expensive. To overcome this issue, we design novel, fast option pricers for options whose underlyings follow the rBergomi model, based on (i) adaptive sparse grids quadrature (ASGQ), and (ii) quasi-Monte Carlo (QMC). Both techniques are coupled with a Brownian bridge construction and Richardson extrapolation. To use these two deterministic quadrature techniques (ASGQ and QMC) for our purposes, we solve two main issues that constitute the two stages of our newly designed method. In the first stage, we smooth the integrand by using the conditional expectation tools, as was proposed in Romano and Touzi (1997) in the context of Markovian stochastic volatility models, and in Bayer *et al.* (2018) in the context of basket options. In a second stage, we apply the deterministic quadrature method, to solve the integration problem. At this stage, before using the ASGQ or QMC method, we apply two hierarchical representations to overcome the issue of facing a high-dimensional integrand due to the discretization scheme used for simulating the rBergomi dynamics. Given that ASGQ and QMC benefit from anisotropy, the first representation consists in applying a hierarchical path generation method, based on a Brownian bridge construction, with the aim of reducing the effective dimension. The second technique consists in applying Richardson extrapolation to reduce the bias (weak error), which, in turn, reduces the number of time steps needed in the coarsest level to achieve a certain error tolerance and consequently the total number of dimensions needed for the integration problem. We emphasize that we are interested in the pre-asymptotic regime (corresponding to a small number of time steps), and that the use of Richardson extrapolation is justified by Conjectures 3.1 and 4.6, and our observed experimental results in that regime, which suggest, in particular, that we have convergence of order one for the weak error and that the pre-asymptotic regime is enough to achieve sufficiently accurate estimates for the option prices. Furthermore, we stress that no proper weak error analysis has been performed in the rough volatility context, which proves to be subtle, due to the absence of a Markovian structure. We also believe that we are the first to claim that for a particular class of payoff functions and in the pre-asymptotic regime both hybrid and exact schemes have a weak error of order one, which is at least justified numerically.

To the best of our knowledge, in the context of rough volatility models, we are also the first to propose (and design) a pricing method that is based on deterministic quadrature methods. As illustrated by our numerical experiments for different parameter constellations, our proposed methods appear to be competitive to the MC approach, which is the only prevalent method in this context. Assuming one targets price estimates with a sufficiently small relative error tolerance, our proposed methods demonstrate substantial computational gains over the standard MC method, even for very small values of H. However, we do not claim that these gains will hold in the asymptotic regime, which requires a higher accuracy. Furthermore, we limit ourselves in this work to comparing our novel proposed methods with standard MC. A more systematic comparison with the variant of MC proposed in McCrickerd and Pakkanen (2018) is left for future research.

We emphasize that applying deterministic quadrature for the family of (rough) stochastic volatility models is not straightforward, but we propose an original way to overcome the high dimensionality of the integration domain, by first reducing the total dimension using Richardson extrapolation and then coupling Brownian bridge construction with ASGQ

or QMC for the optimal performance of our proposed methods. Furthermore, our proposed methodology shows robust performance with respect to the values of the Hurst parameter H, as illustrated by different numerical examples, with even very low values of H. We stress that the performance of hierarchical variance reduction methods such as MLMC is very sensitive to the values of H, since rough volatility models are numerically tough in the sense that they admit low convergence rates for the mean square approximation (strong convergence rates) (see Neuenkirch and Shalaiko 2016).

While our focus is on the rBergomi model, our approach is applicable to a wide class of stochastic volatility models and in particular rough volatility models, since the approach that we propose does not use any particular property of the rBergomi model, and even the analytic smoothing step can be performed in an almost similar manner for any stochastic volatility model.

This paper is structured as follows: We begin in Section 2 by introducing the pricing framework that we are considering in this study. We provide some details about the rBergomi model, option pricing under this model and the simulation schemes used to simulate asset prices following the rBergomi dynamics. We also explain how we choose the optimal simulation scheme for the optimal performance of our approach. In Section 3, we discuss the weak error in the context of the rBergomi. Then, in Section 4, we explain the different building blocks that constitute our proposed methods, which are basically ASGQ, QMC, Brownian bridge construction, and Richardson extrapolation. Finally, in Section 5 we show the results obtained in different numerical experiments conducted with different parameter constellations for the rBergomi model. The reported results show the promising potential of our proposed methods in this context.

2. Problem setting

In this section, we introduce the pricing framework that we consider in this work. We start by giving some details on the rBergomi model proposed in Bayer *et al.* (2016). We then derive the formula of the price of a European call option under the rBergomi model in Section 2.2. Finally, we explain some details about the schemes that we use to simulate the dynamics of asset prices under the rBergomi model.

2.1. The rBergomi model

We consider the rBergomi model for the price process S_t as defined in Bayer *et al.* (2016), normalized to $r = 0$ (r is the interest rate), which is defined by

$$dS_t = \sqrt{v_t} S_t \, dZ_t,$$
$$v_t = \xi_0(t) \exp\left(\eta \widetilde{W}_t^H - \frac{1}{2}\eta^2 t^{2H}\right), \quad (1)$$

where the Hurst parameter $0 < H < 1/2$ and $\eta > 0$. We refer to v_t as the variance process, and $\xi_0(t) = \mathrm{E}[v_t]$ is the forward variance curve. Here, $\widetilde{W}^H$ is a certain Riemann-Liouville fBm

process (Marinucci and Robinson 1999, Picard 2011), defined by

$$\widetilde{W}_t^H = \int_0^t K^H(t-s) \, dW_s^1, \quad t \geq 0, \quad (2)$$

where the kernel $K^H : \mathbb{R}_+ \to \mathbb{R}_+$ is

$$K^H(t-s) = \sqrt{2H}(t-s)^{H-1/2}, \quad \forall \, 0 \leq s \leq t.$$

By construction, $\widetilde{W}^H$ is a centered, locally $(H - \epsilon)$- Hölder continuous Gaussian process with $\mathrm{Var}[\widetilde{W}_t^H] = t^{2H}$, and a dependence structure defined by

$$\mathrm{E}\left[\widetilde{W}_u^H \widetilde{W}_v^H\right] = u^{2H} C\left(\frac{v}{u}\right), \quad v > u,$$

where for $x \geq 1$ and $\gamma = \frac{1}{2} - H$

$$C(x) = 2H \int_0^1 \frac{ds}{(1-s)^\gamma (x-s)^\gamma}.$$

In (1) and (2), W^1, Z denote two *correlated* standard Brownian motions with correlation $\rho \in]-1, 0]$, so that we can represent Z in terms of W^1 as

$$Z = \rho W^1 + \bar{\rho} W^\perp = \rho W^1 + \sqrt{1-\rho^2} W^\perp,$$

where $(W^1, W^\perp)$ are two independent standard Brownian motions. Therefore, the solution to (1), with $S(0) = S_0$, can be written as

$$S_t = S_0 \exp\left(\int_0^t \sqrt{v(s)} \, dZ(s) - \frac{1}{2}\int_0^t v(s) \, ds\right), \quad S_0 > 0$$

$$v_u = \xi_0(u) \exp\left(\eta \widetilde{W}_u^H - \frac{\eta^2}{2} u^{2H}\right), \quad \xi_0 > 0. \quad (3)$$

2.2. Option pricing under the rBergomi model

We are interested in pricing European call options under the rBergomi model. Assuming $S_0 = 1$, and using the conditioning argument on the σ-algebra generated by W^1, and denoted by $\sigma(W^1(t), t \leq T)$ (an argument first used by Romano and Touzi (1997) in the context of Markovian stochastic volatility models), we can show that the call price is given by

$$C_{\mathrm{RB}}(T, K)$$
$$= \mathrm{E}\left[(S_T - K)^+\right]$$
$$= \mathrm{E}\left[\mathrm{E}\left[(S_T - K)^+ \mid \sigma(W^1(t), t \leq T)\right]\right]$$
$$= \mathrm{E}\left[C_{\mathrm{BS}}\left(S_0 = \exp\left(\rho \int_0^T \sqrt{v_t} \, dW_t^1 - \frac{1}{2}\rho^2 \int_0^T v_t \, dt\right),\right.$$
$$\left. k = K, \, \sigma^2 = (1-\rho^2) \int_0^T v_t \, dt\right)\right], \quad (4)$$

where $C_{\mathrm{BS}}(S_0, k, \sigma^2)$ denotes the Black–Scholes call price function, for initial spot price S_0, strike price k and volatility σ^2.

Equation (4) can be obtained by using the orthogonal decomposition of S_t into S_t^1 and S_t^2, where

$$S_t^1 = \mathcal{E}\left\{\rho \int_0^t \sqrt{v_s}\, dW_s^1\right\}, \quad S_t^2 = \mathcal{E}\left\{\sqrt{1-\rho^2}\int_0^t \sqrt{v_s}\, dW_s^\perp\right\},$$

and $\mathcal{E}(\cdot)$ denotes the stochastic exponential; then, by conditional log-normality, we have

$$\log S_t \mid \sigma\{W^1(s), s \le t\}$$
$$\sim \mathcal{N}\left(\log S_t^1 - \frac{1}{2}(1-\rho^2)\int_0^t v_s\, ds, (1-\rho^2)\int_0^t v_s\, ds\right).$$

We point out that the analytical smoothing, based on conditioning, performed in (4) enables us to uncover the available regularity, and hence get a smooth, analytic integrand inside the expectation. Therefore, applying a deterministic quadrature technique such as ASGQ or QMC becomes an adequate option for computing the call price, as we will investigate later. A similar conditioning was used in McCrickerd and Pakkanen (2018) but for variance reduction purposes only.

2.3. Simulation of the rBergomi model

One of the numerical challenges encountered in the simulation of rBergomi dynamics is the computation of $\int_0^T \sqrt{v_t}\, dW_t^1$ and $V = \int_0^T v_t\, dt$ in (4), mainly because of the singularity of the Volterra kernel $K^H(s-t)$ at the diagonal $s = t$. In fact, one needs to jointly simulate two Gaussian processes $(W_t^1, \widetilde{W}_t^H : 0 \le t \le T)$, resulting in $W_{t_1}^1, \ldots, W_{t_N}^1$ and $\widetilde{W}_{t_1}^H, \ldots, \widetilde{W}_{t_N}^H$ along a given time grid $t_1 < \cdots < t_N$. In the literature, there are essentially two suggested ways to achieve this:

(i) Covariance-based approach (exact simulation) (Bayer *et al.* 2016, 2019): $W_{t_1}^1, \ldots, W_{t_N}^1, \widetilde{W}_{t_1}^H, \ldots, \widetilde{W}_{t_N}^H$ together form a $(2N)$-dimensional Gaussian random vector with a computable covariance matrix, and therefore one can use Cholesky decomposition of the covariance matrix to produce exact samples of $W_{t_1}^1, \ldots, W_{t_N}^1, \widetilde{W}_{t_1}^H, \ldots, \widetilde{W}_{t_N}^H$ from $2N$-dimensional Gaussian random vector as an input. This method is exact but slow. The simulation requires $\mathcal{O}(N^2)$ flops. Note that the offline cost is $\mathcal{O}(N^3)$ flops.

(ii) The hybrid scheme of Bennedsen *et al.* (2017): This scheme uses a different approach, which is essentially based on Euler discretization, and approximates the kernel function in (2) by a power function near zero and by a step function elsewhere, which results in an approximation combining Wiener integrals of the power function and a Riemann sum (see (5) and Bennedsen *et al.* 2017 for more details). This approximation is inexact in the sense that samples produced here do not exactly have the distribution of $W_{t_1}^1, \ldots, W_{t_N}^1, \widetilde{W}_{t_1}^H, \ldots, \widetilde{W}_{t_N}^H$. However, they are much more accurate than the samples produced from a simple Euler discretization, and much faster than method (i). As in method (i), in this case, we need a 2N-dimensional Gaussian random input vector to produce one sample of $W_{t_1}^1, \ldots, W_{t_N}^1, \widetilde{W}_{t_1}^H, \ldots, \widetilde{W}_{t_N}^H$.

2.3.1. On the choice of the simulation scheme in our approach.
The choice of the simulation scheme in our approach was based on the observed behavior of the weak rates. Through our numerical experiments (see table 1 for the tested examples), we observe that, although the hybrid and exact schemes seem to converge with a weak error of order $\mathcal{O}(\Delta t)$, the pre-asymptotic behavior of the weak rate is different for both schemes (we provide a short discussion of the weak error in Section 3). As an illustration, from figure 1 for Set 1 parameter in table 1, the hybrid scheme has a consistent convergence behavior in the sense that it behaves in an asymptotic manner, basically right from the beginning, whereas the exact scheme does not. On the other hand, the constant seems to be considerably smaller for the exact scheme. These two features make the hybrid scheme the best choice to work within our context, since our approach is based on hierarchical representations involving the use of Richardson extrapolation (see Section 4.4).

2.3.2. The hybrid scheme.
Let us denote the number of time steps by N. As motivated in Section 2.3.1, in this work we use the hybrid scheme, which, on an equidistant grid $\{0, 1/N, 2/N, \ldots, NT/N\}$, is given by the following,

$$\widetilde{W}_{i/N}^H \approx \bar{W}_{i/N}^H$$
$$= \sqrt{2H}\left(\sum_{k=1}^{\min(i,\kappa)} \int_{i/N-k/N}^{i/N-k/N+1/N} \left(\frac{i}{N}-s\right)^{H-1/2} dW_s^1 \right.$$
$$\left. + \sum_{k=\kappa+1}^{i} \left(\frac{b_k}{N}\right)^{H-1/2} \int_{i/N-k/N}^{i/N-k/N+1/N} dW_s^1\right), \quad (5)$$

which results for $\kappa = 1$ in (6)

$$\widetilde{W}_{i/N}^H \approx \bar{W}_{i/N}^H$$
$$= \sqrt{2H}\left(W_i^2 + \sum_{k=2}^{i}\left(\frac{b_k}{N}\right)^{H-1/2}\left(W_{(i-(k-1))/N}^1 - W_{(i-k)/N}^1\right)\right),$$
$$1 \le i \le N, \quad (6)$$

where

$$W_i^2 = \int_{i-1/N}^{i/N}\left(\frac{i}{N}-s\right)^{H-1/2} dW_s^1,$$
$$b_k = \left(\frac{k^{H+1/2}-(k-1)^{H+1/2}}{H+\frac{1}{2}}\right)^{1/(H-1/2)}. \quad (7)$$

The sum in (6) requires the most computational effort in the simulation. Given that (6) can be seen as discrete convolution (see Bennedsen *et al.* 2017), we employ the fast Fourier transform to evaluate it, which results in $\mathcal{O}(N \log N)$ floating point operations.

We note that the variates $\bar{W}_0^H, \bar{W}_{1/N}^H, \ldots, \bar{W}_{[NT]/N}^H$ are generated by sampling N i.i.d draws from a $(\kappa + 1)$-dimensional Gaussian distribution and computing a discrete convolution. For the case of $\kappa = 1$, we denote these pairs of Gaussian random vectors from now on by $(\mathbf{W}^{(1)}, \mathbf{W}^{(2)})$, and we refer to Bennedsen *et al.* (2017) for more details.
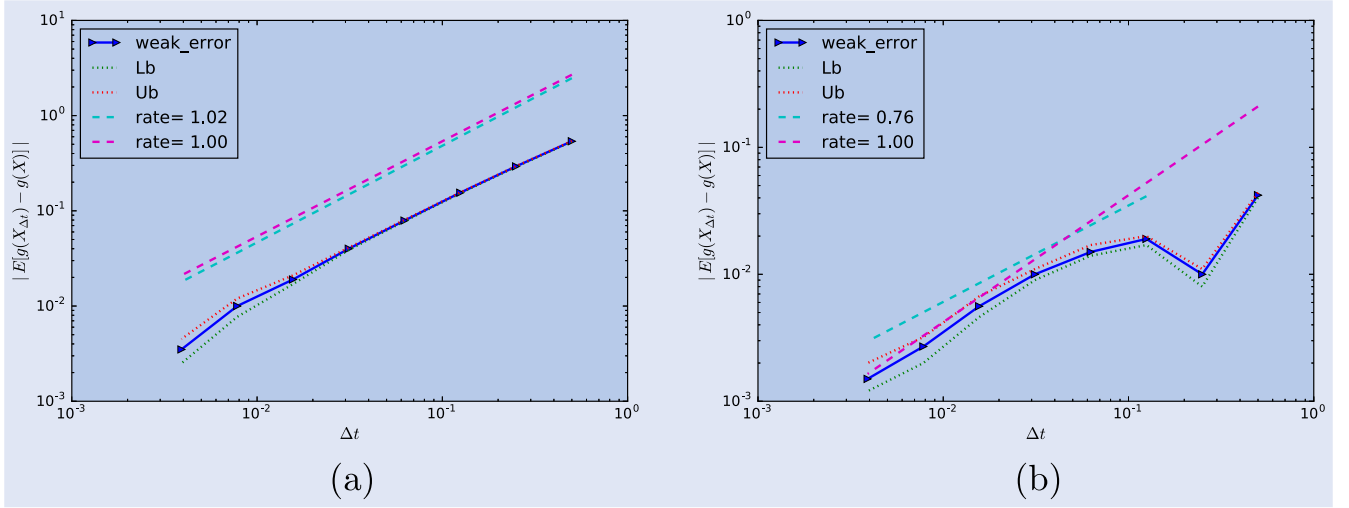
Figure 1. The convergence of the weak error $\mathcal{E}_B$, defined in (8), using MC with 6×10^6 samples, for Set 1 parameter in table 1. We refer to C_{RB} (as in (4)) for $\mathrm{E}\,[g(X)]$, and to C_{RB}^N (as in (9)) for $\mathrm{E}\,[g(X_{\Delta t})]$. The upper and lower bounds are 95% confidence intervals. (a) With the hybrid scheme (b) With the exact scheme.

3. Weak error discussion

To the best of our knowledge, no proper weak error analysis has been done in the rough volatility context, which proves to be subtle in the absence of a Markovian structure. However, we try in this Section to shortly discuss it in the context of the rBergomi model.

In this work, we are interested in approximating $\mathrm{E}\,[g(X_T)]$, where g is some smooth function and X is the asset price under the rBergomi dynamics such that $X_t = X_t(W^{(1)}_{[0,t]}, \widetilde{W}_{[0,t]})$, where $W^{(1)}$ is standard Brownian motion and $\widetilde{W}$ is the fractional Brownian motion as given by (2). Then we can express the approximation of $\mathrm{E}\,[g(X_T)]$ using the hybrid and exact schemes as the following

$$\mathrm{E}\left[g\left(X_T\left(W^{(1)}_{[0,T]}, \widetilde{W}_{[0,T]}\right)\right)\right]$$
$$\approx \mathrm{E}\left[g\left(\overline{X}_N\left(W^{(1)}_1, \ldots, W^{(1)}_N, \overline{W}_1, \ldots, \overline{W}_N\right)\right)\right]$$
(**Hybrid scheme**),

$$\mathrm{E}\left[g\left(X_T\left(W^{(1)}_{[0,T]}, \widetilde{W}_{[0,T]}\right)\right)\right]$$
$$\approx \mathrm{E}\left[g\left(\overline{X}_N\left(W^{(1)}_1, \ldots, W^{(1)}_N, \widetilde{W}_1, \ldots, \widetilde{W}_N\right)\right)\right]$$
(**Exact scheme**),

where $\overline{W}$ is the approximation of $\widetilde{W}$ as given by (5) and $\overline{X}_N$ is the approximation of X using N time steps. In the following, to simplify notation, let $\overline{\mathbf{W}} = (\overline{W}_1, \ldots, \overline{W}_N)$, $\mathbf{W}^1 = (W^{(1)}_1, \ldots, W^{(1)}_N)$ and $\widetilde{\mathbf{W}} = (\widetilde{W}_1, \ldots, \widetilde{W}_N)$. Then, the use of Richardson extrapolation in our methodology presented in Section 4 is mainly justified by the Conjecture 3.1.

CONJECTURE 3.1 *If we denote by $\mathcal{E}_B^{\mathrm{Hyb}}$ and $\mathcal{E}_B^{\mathrm{Chol}}$ the weak errors produced by the hybrid and Cholesky scheme respectively, then we have*

$$\mathcal{E}_B^{\mathrm{Chol}} = \mathcal{O}\left(\Delta t\right), \quad \mathcal{E}_B^{\mathrm{Hyb}} = \mathcal{O}\left(\Delta t\right).$$

We motivate Conjecture 3.1 by writing

$$\mathcal{E}_B^{\mathrm{Hyb}} = \left|\mathrm{E}\left[g\left(X_T\left(W^{(1)}_{[0,T]}, \widetilde{W}_{[0,T]}\right)\right)\right] - \mathrm{E}\left[g\left(\overline{X}_N\left(\mathbf{W}^1, \overline{\mathbf{W}}\right)\right)\right]\right|$$
$$\leq \left|\mathrm{E}\left[g\left(X_T\left(W^{(1)}_{[0,T]}, \widetilde{W}_{[0,T]}\right)\right)\right] - \mathrm{E}\left[g\left(\overline{X}_N\left(\mathbf{W}^1, \widetilde{\mathbf{W}}\right)\right)\right]\right|$$
$$+ \left|\mathrm{E}\left[g\left(\overline{X}_N\left(\mathbf{W}^1, \overline{\mathbf{W}}\right)\right)\right] - \mathrm{E}\left[g\left(\overline{X}_N\left(\mathbf{W}^1, \widetilde{\mathbf{W}}\right)\right)\right]\right|$$
$$\leq \mathcal{E}_B^{\mathrm{Chol}} + \left|\mathrm{E}\left[g\left(\overline{X}_N\left(\mathbf{W}^1, \overline{\mathbf{W}}\right)\right)\right] - \mathrm{E}\left[g\left(\overline{X}_N\left(\mathbf{W}^1, \widetilde{\mathbf{W}}\right)\right)\right]\right|.$$
(8)

From the construction of the Cholesky scheme, we expect that the weak error is purely the discretization error, that is

$$\mathcal{E}_B^{\mathrm{Chol}} = \mathcal{O}\left(\Delta t\right),$$

as it was observed by our numerical experiments (for illustration see figure 1(b) for the case of Set 1 in table 1). The second term in the right-hand side of (8) is basically related to approximating the integral (2) by (6). From our numerical experiments, it seems that this term is at least of order Δt and that its rate of convergence is independent of H (for illustration see figure 1(a) for the case of Set 1 in table 1).

4. Details of our hierarchical methods

We recall that our goal is to compute the expectation in (4), and we remind from Section 2.3.2 that we need $2N$-dimensional Gaussian inputs, $(\mathbf{W}^{(1)}, \mathbf{W}^{(2)})$ for the used hybrid scheme (N is the number of time steps in the time grid). We can rewrite (4) as

$$C_{\mathrm{RB}}\left(T, K\right)$$
$$= \mathrm{E}\left[C_{\mathrm{BS}}\left(S_0 = \exp\left(\rho \int_0^T \sqrt{v_t}\,\mathrm{d}W^1_t - \frac{1}{2}\rho^2 \int_0^T v_t\,\mathrm{d}t\right),\right.\right.$$
$$\left.\left. k = K, \sigma^2 = (1 - \rho^2)\int_0^T v_t\,\mathrm{d}t\right)\right]$$

$$\approx \int_{\mathbb{R}^{2N}} C_{BS} \left(G_{\mathrm{rB}}(\mathbf{w}^{(1)}, \mathbf{w}^{(2)}) \right) \rho_N(\mathbf{w}^{(1)}) \rho_N(\mathbf{w}^{(2)})$$
$$\times \, \mathrm{d}\mathbf{w}^{(1)} \, \mathrm{d}\mathbf{w}^{(2)}$$
$$:= C_{\mathrm{RB}}^N, \tag{9}$$

where G_{rB} maps $2N$ independent standard Gaussian random inputs, formed by $(\mathbf{W}^{(1)}, \mathbf{W}^{(2)})$, to the parameters fed to the Black-Scholes call price function, C_{BS} (defined in (4)), and ρ_N is the multivariate Gaussian density, given by

$$\rho_N(\mathbf{z}) = \frac{1}{(2\pi)^{N/2}} \, \mathrm{e}^{-(1/2)\mathbf{z}^T\mathbf{z}}.$$

Therefore, the initial integration problem that we are solving lives in $2N$-dimensional space, which becomes very large as the number of time steps N, used in the hybrid scheme, increases.

Our approach of approximating the expectation in (9) is based on hierarchical deterministic quadratures, namely (i) an ASGQ using the same construction as in Haji-Ali *et al.* (2016) and (ii) a randomized QMC based on lattice rules. In Section 4.1 we describe the ASGQ method in our context, and in Section 4.2 we provide details on the implemented QMC method. To make an effective use of either the ASGQ or the QMC method, we apply two techniques to overcome the issue of facing a high dimensional integrand due to the discretization scheme used for simulating the rBergomi dynamics. The first consists in applying a hierarchical path generation method, based on a Brownian bridge construction, with the aim of reducing the effective dimension, as described in Section 4.3. The second technique consists in applying Richardson extrapolation to reduce the bias, resulting in reducing the total number of dimensions needed for the integration problem. Details about the Richardson extrapolation are provided in Section 4.4.

If we denote by $\mathcal{E}_{\mathrm{tot}}$ the total error of approximating the expectation in (4), using the ASGQ estimator, Q_N (defined by (12)), then we obtain a natural error decomposition

$$\mathcal{E}_{\mathrm{tot}} \leq \left| C_{\mathrm{RB}} - C_{\mathrm{RB}}^N \right| + \left| C_{\mathrm{RB}}^N - Q_N \right| \leq \mathcal{E}_B(N) + \mathcal{E}_Q(\mathrm{TOL}_{\mathrm{ASGQ}}, N), \tag{10}$$

where $\mathcal{E}_Q$ is the quadrature error, $\mathcal{E}_B$ is the bias, $\mathrm{TOL}_{\mathrm{ASGQ}}$ is a user-selected tolerance for the ASGQ method, and C_{RB}^N is the biased price computed with N time steps, as given by (9).

On the other hand, the total error of approximating the expectation in (4) using the randomized QMC or MC estimator, $Q_N^{\mathrm{MC(QMC)}}$ can be bounded by

$$\mathcal{E}_{\mathrm{tot}} \leq \left| C_{\mathrm{RB}} - C_{\mathrm{RB}}^N \right| + \left| C_{\mathrm{RB}}^N - Q_N^{\mathrm{MC\,(QMC)}} \right| \leq \mathcal{E}_B(N) + \mathcal{E}_S(M, N), \tag{11}$$

where $\mathcal{E}_S$ is the statistical error,[†] M is the number of samples used for the MC or the randomized QMC method.

REMARK 4.1 We note that, thanks to (9), our approach, explained in the following sections, can be extended to any

(rough) stochastic volatility dynamics, with the only difference of using another function instead of G_{rB} in (9), with specifics of the considered model embedded in this function.

4.1. Adaptive sparse grids quadrature (ASGQ)

We assume that we want to approximate the expected value $\mathrm{E}[f(Y)]$ of an analytic function $f : \Gamma \to \mathbb{R}$ using a tensorization of quadrature formulas over Γ.

To introduce simplified notations, we start with the one-dimensional case. Let us denote by β a non-negative integer, referred to as a 'stochastic discretization level', and by $m : \mathbb{N} \to \mathbb{N}$ a strictly increasing function with $m(0) = 0$ and $m(1) = 1$, that we call 'level-to-nodes function'. At level β, we consider a set of $m(\beta)$ distinct quadrature points in $\mathbb{R}$, $\mathcal{H}^{m(\beta)} = \{y_\beta^1, y_\beta^2, \ldots, y_\beta^{m(\beta)}\} \subset \mathbb{R}$, and a set of quadrature weights, $\boldsymbol{\omega}^{m(\beta)} = \{\omega_\beta^1, \omega_\beta^2, \ldots, \omega_\beta^{m(\beta)}\}$. We also let $C^0(\mathbb{R})$ be the set of real-valued continuous functions over $\mathbb{R}$. We then define the quadrature operator as

$$Q^{m(\beta)} : C^0(\mathbb{R}) \to \mathbb{R}, \quad Q^{m(\beta)}[f] = \sum_{j=1}^{m(\beta)} f(y_\beta^j)\omega_\beta^j.$$

In our case, we have in (9) a multi-variate integration problem with, $f = C_{BS} \circ G_{\mathrm{rB}}$, $\mathbf{Y} = (\mathbf{W}^{(1)}, \mathbf{W}^{(2)})$, and $\Gamma = \mathbb{R}^{2N}$, in the previous notations. Furthermore, since we are dealing with Gaussian densities, using Gauss-Hermite quadrature points is the appropriate choice.

We define for any multi-index $\boldsymbol{\beta} \in \mathbb{N}^{2N}$

$$Q^{m(\boldsymbol{\beta})} : C^0(\mathbb{R}^{2N}) \to \mathbb{R}, \quad Q^{m(\boldsymbol{\beta})} = \bigotimes_{n=1}^{2N} Q^{m(\beta_n)},$$

where the nth quadrature operator is understood to act only on the nth variable of f. Practically, we obtain the value of $Q^{m(\boldsymbol{\beta})}[f]$ by using the grid $\mathcal{T}^{m(\boldsymbol{\beta})} = \prod_{n=1}^{2N} \mathcal{H}^{m(\beta_n)}$, with cardinality $\#\mathcal{T}^{m(\boldsymbol{\beta})} = \prod_{n=1}^{2N} m(\beta_n)$, and computing

$$Q^{m(\boldsymbol{\beta})}[f] = \sum_{j=1}^{\#\mathcal{T}^{m(\boldsymbol{\beta})}} f(\hat{y}_j)\bar{\omega}_j,$$

where $\hat{y}_j \in \mathcal{T}^{m(\boldsymbol{\beta})}$ and $\bar{\omega}_j$ are products of weights of the univariate quadrature rules. To simplify notation, hereafter, we replace $Q^{m(\boldsymbol{\beta})}$ by $Q^{\boldsymbol{\beta}}$.

A direct approximation $\mathrm{E}[f[\mathbf{Y}]] \approx Q^{\boldsymbol{\beta}}[f]$ is not an appropriate option, due to the well-known 'curse of dimensionality'. We use a hierarchical ASGQ[‡] strategy, specifically using the same construction as in Haji-Ali *et al.* (2016), and which uses stochastic discretizations and a classic sparsification approach to obtain an effective approximation scheme for $\mathrm{E}[f]$.

To be concrete, in our setting, we are left with a $2N$-dimensional Gaussian random input, which is chosen independently, resulting in $2N$ numerical parameters for ASGQ,

---

† The statistical error estimate of MC or randomized QMC is $C_\alpha(\sigma_M/\sqrt{M})$, where M is the number of samples and $C_\alpha = 1.96$ for 95% confidence interval.

‡ More details about sparse grids can be found in Bungartz and Griebel (2004).

which we use as the basis of the multi-index construction. For a multi-index $\boldsymbol{\beta} = (\beta_n)_{n=1}^{2N} \in \mathbb{N}^{2N}$, we denote by $Q_N^{\boldsymbol{\beta}}$ the result of approximating (9) with a number of quadrature points in the i-th dimension equal to $m(\beta_i)$. We further define the set of differences $\Delta Q_N^{\boldsymbol{\beta}}$ as follows: for a single index $1 \leq i \leq 2N$, let

$$\Delta_i Q_N^{\boldsymbol{\beta}} = \begin{cases} Q_N^{\boldsymbol{\beta}} - Q_N^{\boldsymbol{\beta}'}, & \text{with } \boldsymbol{\beta}' = \boldsymbol{\beta} - e_i, \quad \text{if } \boldsymbol{\beta}_i > 0, \\ Q_N^{\boldsymbol{\beta}}, & \text{otherwise}, \end{cases}$$

where e_i denotes the ith $2N$-dimensional unit vector. Then, $\Delta Q_N^{\boldsymbol{\beta}}$ is defined as

$$\Delta Q_N^{\boldsymbol{\beta}} = \left(\prod_{i=1}^{2N} \Delta_i \right) Q_N^{\boldsymbol{\beta}}.$$

For instance, when $N = 1$, then

$$\begin{aligned} \Delta Q_1^{\boldsymbol{\beta}} &= \Delta_2 \Delta_1 Q_1^{(\beta_1, \beta_2)} = \Delta_2 \left(Q_1^{(\beta_1, \beta_2)} - Q_1^{(\beta_1 - 1, \beta_2)} \right) \\ &= \Delta_2 Q_1^{(\beta_1, \beta_2)} - \Delta_2 Q_1^{(\beta_1 - 1, \beta_2)} \\ &= Q_1^{(\beta_1, \beta_2)} - Q_1^{(\beta_1, \beta_2 - 1)} - Q_1^{(\beta_1 - 1, \beta_2)} + Q_1^{(\beta_1 - 1, \beta_2 - 1)}. \end{aligned}$$

Given the definition of C_{RB}^N by (9), we have the telescoping property

$$C_{\mathrm{RB}}^N = Q_N^\infty = \sum_{\beta_1 = 0}^\infty \cdots \sum_{\beta_{2N} = 0}^\infty \Delta Q_N^{(\beta_1, \dots, \beta_{2N})} = \sum_{\boldsymbol{\beta} \in \mathbb{N}^{2N}} \Delta Q_N^{\boldsymbol{\beta}}.$$

The ASGQ estimator used for approximating (9), and using a set of multi-indices $\mathcal{I} \subset \mathbb{N}^{2N}$ is given by

$$Q_N^{\mathcal{I}} = \sum_{\boldsymbol{\beta} \in \mathcal{I}} \Delta Q_N^{\boldsymbol{\beta}}. \tag{12}$$

The quadrature error in this case is given by

$$\mathcal{E}_Q(\mathrm{TOL}_{\mathrm{ASGQ}}, N) = \left| Q_N^\infty - Q_N^{\mathcal{I}} \right| \leq \sum_{\boldsymbol{\beta} \in \mathbb{N}^{2N} \setminus \mathcal{I}} \left| \Delta Q_N^{\boldsymbol{\beta}} \right|. \tag{13}$$

We define the work contribution, $\Delta \mathcal{W}_{\boldsymbol{\beta}}$, to be the computational cost required to add $\Delta Q_N^{\boldsymbol{\beta}}$ to $Q_N^{\mathcal{I}}$, and the error contribution, $\Delta E_{\boldsymbol{\beta}}$, to be a measure of how much the quadrature error, defined in (13), would decrease once $\Delta Q_N^{\boldsymbol{\beta}}$ has been added to $Q_N^{\mathcal{I}}$, that is

$$\Delta E_{\boldsymbol{\beta}} = \left| Q_N^{\mathcal{I} \cup \{\boldsymbol{\beta}\}} - Q_N^{\mathcal{I}} \right| \tag{14}$$

$$\Delta \mathcal{W}_{\boldsymbol{\beta}} = \mathrm{Work}[Q_N^{\mathcal{I} \cup \{\boldsymbol{\beta}\}}] - \mathrm{Work}[Q_N^{\mathcal{I}}]. \tag{15}$$

The construction of the optimal $\mathcal{I}$ is done by profit thresholding (see figure 2 for illustration), that is, for a certain threshold value $\bar{T}$, and a profit of a hierarchical surplus defined by

$$P_{\boldsymbol{\beta}} = \frac{\left| \Delta E_{\boldsymbol{\beta}} \right|}{\Delta \mathcal{W}_{\boldsymbol{\beta}}}, \tag{16}$$

the optimal index set $\mathcal{I}$ for our ASGQ is given by $\mathcal{I} = \{\boldsymbol{\beta} \in \mathbb{N}_+^{2N} : P_{\boldsymbol{\beta}} \geq \bar{T}\}$.

REMARK 4.2 The choice of the hierarchy of quadrature points, $m(\boldsymbol{\beta})$, is flexible in the ASGQ algorithm and can be fixed by the user, depending on the convergence properties of the problem at hand. For instance, for the sake of reproducibility, in our numerical experiments, we used a linear hierarchy: $m(\beta) = 4(\beta - 1) + 1$, $1 \leq \beta$, for results of parameter set 1 in table 1. For the remaining parameter sets in table 1, we used a geometric hierarchy: $m(\beta) = 2^{\beta - 1} + 1$, $1 \leq \beta$.

REMARK 4.3 As emphasized in Haji-Ali *et al.* (2016), one important requirement to achieve the optimal performance of the ASGQ is to check the error convergence, defined by (15), of first and mixed difference operators. We checked this requirement in all our numerical experiments, and for illustration, we show in figures 3 and 4 the error convergence of first- and second-order differences for the case of parameter set 2 in table 1. These plots show that: (i) $\Delta E_{\boldsymbol{\beta}}$ decreases exponentially fast with respect to β_i, and (ii) $\Delta E_{\boldsymbol{\beta}}$ has a product structure since we observe a faster error decay for second differences, compared to corresponding first difference operators.

REMARK 4.4 The analiticity assumption, stated in the beginning of Section 4.1, is crucial for the optimal performance of our proposed method. In fact, although we face the issue of the 'curse of dimensionality' when increasing N, the analiticity of f implies a spectral convergence for sparse grids quadrature.

4.2. Quasi-Monte Carlo (QMC)

A second type of deterministic quadrature that we test in this work is the randomized QMC method. Specifically, we use the lattice rules family of QMC (Sloan 1985, Cools and Nuyens 2008, Nuyens 2014). The main input for the lattice rule is one integer vector with d components (d is the dimension of the integration problem).

In fact, given an integer vector $z = (z_1, \dots, z_d)$ known as *the generating vector*, a (rank-1) lattice rule with n points takes the form

$$Q_n(f) := \frac{1}{n} \sum_{k=0}^{n-1} f \left(\frac{kz \bmod n}{n} \right). \tag{17}$$

The quality of the lattice rule depends on the choice of the generating vector. Due to the modulo operation, it is sufficient to consider the values from 1 up to $n - 1$. Furthermore, we restrict the values to those relatively prime to n, to ensure that every one-dimensional projection of the n points yields n distinct values. Thus, we write $\mathbf{z} \in \mathbb{U}_n^d$, with $\mathbb{U}_n := \{z \in \mathbb{Z} : 1 \leq z \leq n - 1 \text{ and } \gcd(z, n) = 1\}$. For practical purposes, we choose n to be a power of 2. The total number of possible choices for the generating vector is then $(n/2)^d$.

To get an unbiased approximation of the integral, we use a randomly shifted lattice rule, which also allows us to obtain a practical error estimate in the same way as the MC method. It works as follows. We generate q independent random shifts $\Delta^{(i)}$ for $i = 0, \dots, q - 1$ from the uniform distribution of $[0, 1]^d$. For the same fixed lattice generating vector z, we compute the q different shifted lattice rule approximations and denote them by $Q_n^{(i)}(f)$ for $i = 0, \dots, q - 1$. We then take the
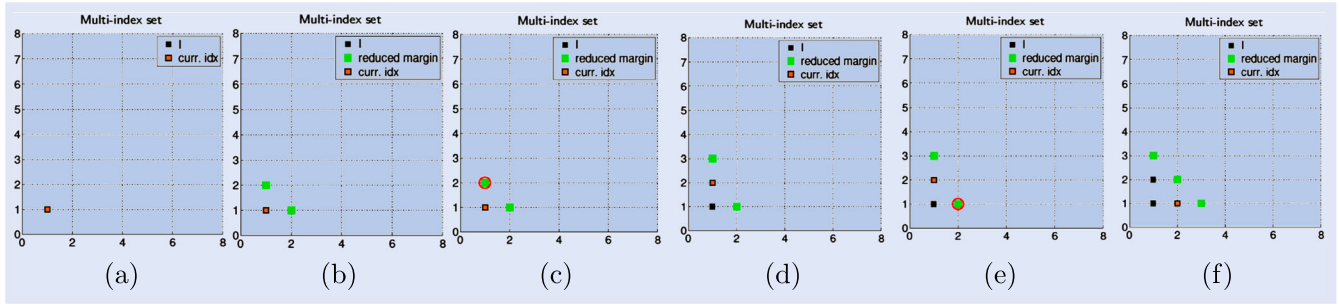
Figure 2. Snapshots of the greedy construction of the index set for ASGQ method. A posteriori, adaptive construction: Given an index set $\mathcal{I}_k$, compute the profits of the neighbor indices using (16), and select the most profitable one.
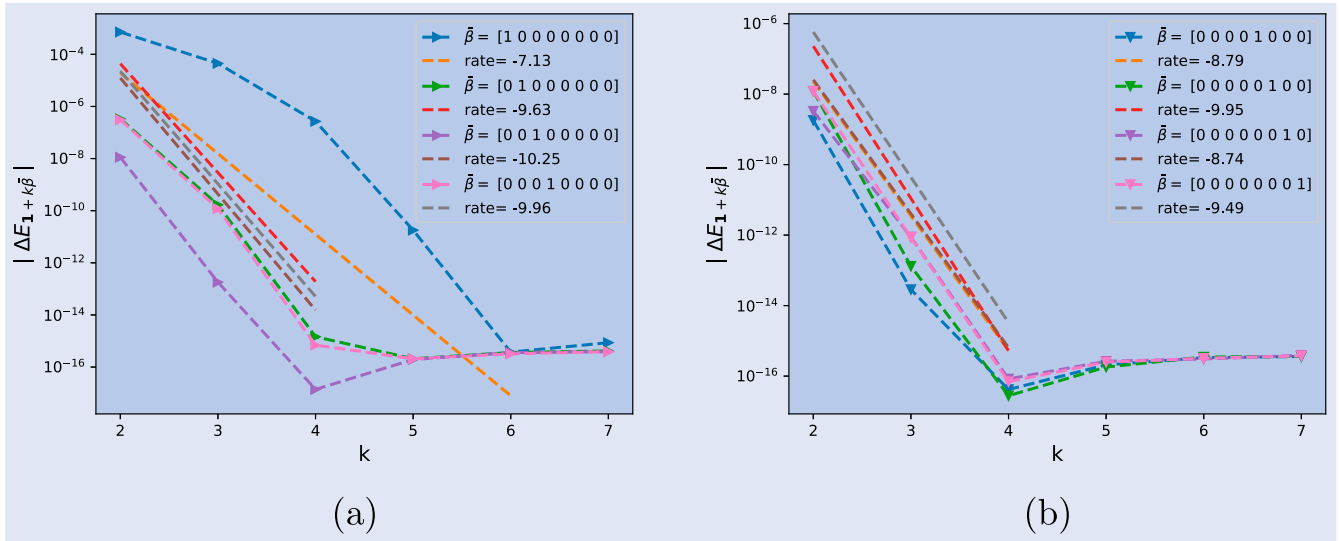

Figure 3. The rate of error convergence of first-order differences $\left|\Delta E_{\boldsymbol{\beta}}\right|$, defined by (15), ($\boldsymbol{\beta} = \mathbf{1} + k\bar{\boldsymbol{\beta}}$) for parameter set 2 in table 1. The number of quadrature points used in the ith dimension is $N_i = 2^{\beta_i - 1} + 1$. a) With respect to $\mathbf{W}^{(1)}$. b) With respect to $\mathbf{W}^{(2)}$.
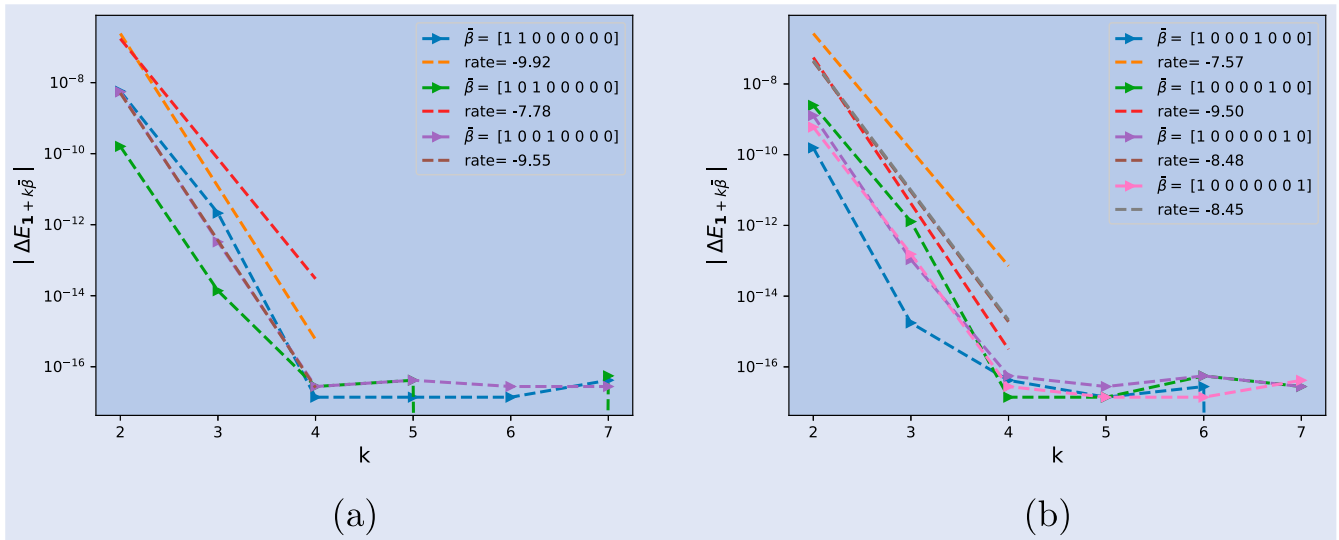
Figure 4. The rate of error convergence of second-order differences $\left|\Delta E_{\boldsymbol{\beta}}\right|$, defined by (15), ($\boldsymbol{\beta} = \mathbf{1} + k\bar{\boldsymbol{\beta}}$) for parameter set 2 in table 1. The number of quadrature points used in the ith dimension is $N_i = 2^{\beta_i - 1} + 1$. a) With respect to $\mathbf{W}^{(1)}$. b) With respect to $\mathbf{W}^{(2)}$.

average

$$\overline{Q}_{n,q}(f) = \frac{1}{q} \sum_{i=0}^{q-1} Q_n^{(i)}(f)$$

$$= \frac{1}{q} \sum_{i=0}^{q-1} \left(\frac{1}{n} \sum_{k=0}^{n-1} f\left(\frac{kz + \Delta^{(i)} \bmod n}{n} \right) \right) \quad (18)$$

as our final approximation to the integral and the total number of samples of the randomized QMC method is $M^{\mathrm{QMC}} = q \times n$.

We note that, since we are dealing with Gaussian randomness and with integrals in infinite support, we use the inverse of the standard normal cumulative distribution function as a pre-transformation to map the problem to $[0, 1]$, and then use the randomized QMC. Furthermore, in our numerical test, we use a pre-made point generator using latticeseq_b2.py in python from https://people.cs.kuleuven.be/~dirk.nuyens/qmc-generators/.

REMARK 4.5 We emphasize that we opt for one of the most simple rules (very easy to specify and implement) for QMC, that is the lattice rule, which, thanks to the hierarchical representations that we implemented (Richardson extrapolation and Brownian bridge construction) appears to give already a very good performance. Furthermore, we believe that using more sophisticated rules such as lattice sequences or interlaced Sobol sequences may have a similar or even better performance.

4.3. Brownian bridge construction

In the literature on ASGQ and QMC, several hierarchical path generation methods (PGMs) have been proposed to reduce the effective dimension. Among these techniques, we mention Brownian bridge construction (Morokoff and Caflisch 1994, Moskowitz and Caflisch 1996, Caflisch *et al.* 1997), principal component analysis (PCA) (Acworth *et al.* 1998) and linear transformation (LT) (Imai and Tan 2004).

In our context, the Brownian motion on a time discretization can be constructed either sequentially using a standard random walk construction, or hierarchically using other PGMs, as listed above. For our purposes, to make an effective use of ASGQ or QMC methods, which benefit from anisotropy, we use the Brownian bridge construction that produces dimensions that are not equally important, contrary to a random walk procedure for which all the dimensions of the stochastic space have equal importance. In fact, Brownian bridge uses the first several coordinates of the low-discrepancy points to determine the general shape of the Brownian path, and the last few coordinates influence only the fine detail of the path. Consequently, this representation reduces the effective dimension of the problem, resulting in an acceleration of the ASGQ and QMC methods by reducing the computational cost.

Let us denote $\{t_i\}_{i=0}^N$ as the grid of time steps. Then the Brownian bridge construction (Glasserman 2004) consists of the following: given a past value B_{t_i} and a future value B_{t_k}, the value B_{t_j} (with $t_i < t_j < t_k$) can be generated according to

$$B_{t_j} = (1 - \rho)B_{t_i} + \rho B_{t_k} + \sqrt{\rho(1 - \rho)(k - i)\Delta t} z, \ z \sim \mathcal{N}(0, 1),$$

where $\rho = (j - i)/(k - i)$.

4.4. Richardson extrapolation

Another representation that we couple with the ASGQ and QMC methods is the Richardson extrapolation (Talay and Tubaro 1990). In fact, applying level K_{R} (level of extrapolation) of the Richardson extrapolation dramatically reduces the bias, and as a consequence, reduces the number of time steps N needed in the coarsest level to achieve a certain error tolerance. As a consequence, the Richardson extrapolation directly reduces the total dimension of the integration problem for achieving some error tolerance.

CONJECTURE 4.6 *Let us denote by* $(X_t)_{0 \leq t \leq T}$ *a stochastic process following the rBergomi dynamics given by* (1), *and by* $(\hat{X}_{t_i}^{\Delta t})_{0 \leq t_i \leq T}$ *its approximation using the hybrid scheme (described in Section* 2.3.2*) with a time step* Δt. *Then, for sufficiently small* Δt, *and a suitable smooth function f, we assume that*

$$\mathrm{E}\left[f(\hat{X}_T^{\Delta t}) \right] = \mathrm{E}\left[f(X_T) \right] + c\Delta t + \mathcal{O}\left(\Delta t^2 \right). \quad (19)$$

Applying (19) with discretization step $2\Delta t$, we obtain

$$\mathrm{E}\left[f(\hat{X}_T^{2\Delta t}) \right] = \mathrm{E}\left[f(X_T) \right] + 2c\Delta t + \mathcal{O}\left(\Delta t^2 \right),$$

implying

$$2\mathrm{E}\left[f(\hat{X}_T^{2\Delta t}) \right] - \mathrm{E}\left[f(\hat{X}_T^{\Delta t}) \right] = \mathrm{E}\left[f(X_T) \right] + \mathcal{O}\left(\Delta t^2 \right).$$

For higher levels of extrapolations, we use the following: Let us denote by $\Delta t_J = \Delta t_0 2^{-J}$ the grid sizes (where Δt_0 is the coarsest grid size), by K_{R} the level of the Richardson extrapolation, and by $I(J, K_{\mathrm{R}})$ the approximation of $\mathrm{E}\left[f(X_T) \right]$ by terms up to level K_{R} (leading to a weak error of order K_{R}), then we have the following recursion

$$I(J, K_{\mathrm{R}}) = \frac{2^{K_{\mathrm{R}}} I(J, K_{\mathrm{R}} - 1) - I(J - 1, K_{\mathrm{R}} - 1)}{2^{K_{\mathrm{R}}} - 1},$$

$$J = 1, 2, \ldots, K_{\mathrm{R}} = 1, 2, \ldots$$

REMARK 4.7 We emphasize that, throughout our work, we are interested in the pre-asymptotic regime (a small number of time steps), and the use of Richardson extrapolation is justified by Conjectures 3.1 and 4.6, and our observed experimental results in that regime (see Section 5.1), which suggest a convergence of order one for the weak error.

5. Numerical experiments

In this section, we show the results obtained through the different numerical experiments, conducted across different

Table 1. Reference solution, which is the approximation of the call option price under the rBergomi model, defined in (4), using MC with 500 time steps and number of samples, $M = 8 \times 10^6$, for different parameter constellations.

Parameters	Reference solution
Set 1: $H = 0.07, K = 1, S_0 = 1, T = 1, \rho = -0.9, \eta = 1.9, \xi_0 = 0.235^2$	0.0791 (5.6e−05)
Set 2: $H = 0.02, K = 1, S_0 = 1, T = 1, \rho = -0.7, \eta = 0.4, \xi_0 = 0.1$	0.1246 (9.0e−05)
Set 3: $H = 0.02, K = 0.8, S_0 = 1, T = 1, \rho = -0.7, \eta = 0.4, \xi_0 = 0.1$	0.2412 (5.4e−05)
Set 4: $H = 0.02, K = 1.2, S_0 = 1, T = 1, \rho = -0.7, \eta = 0.4, \xi_0 = 0.1$	0.0570 (8.0e−05)

Note: The numbers between parentheses correspond to the statistical error estimates.

parameter constellations for the rBergomi model. Details about these examples are presented in table 1. The first set is the one closest to the empirical findings (Bennedsen *et al.* 2016, Gatheral *et al.* 2018), which suggest that $H \approx 0.1$. The choice of parameter values of $\nu = 1.9$ and $\rho = -0.9$ is justified by Bayer *et al.* (2016), where it is shown that these values are remarkably consistent with the SPX market on 4th February 2010. For the remaining three sets in table 1, we want to test the potential of our method for a very rough case, that is $H = 0.02$, for three different scenarios of moneyness, S_0/K. In fact, hierarchical variance reduction methods, such as MLMC, are inefficient in this context because of the poor behavior of the strong error, which is of the order of H (Neuenkirch and Shalaiko 2016). We emphasize that we checked the robustness of our method for other parameter sets, but for illustrative purposes, we only show results for the parameters sets presented in table 1. For all our numerical experiments, we consider a number of time steps $N \in \{2, 4, 8, 16\}$, and all reported errors are relative errors, normalized by the reference solutions provided in table 1.

5.1. Weak error

We start our numerical experiments by accurately estimating the weak error (bias) discussed in Section 3, for the different sets of parameters in table 1, with and without the Richardson extrapolation. For illustrative purposes, we only show the weak errors related to set 1 in table 1 (see figure 5). We note that we observed a similar behavior for the other parameter sets, with slightly worse rates for some cases. We emphasize that the reported weak rates correspond to the pre-asymptotic regime that we are interested in. We are not interested in estimating the rates specifically but rather obtaining a sufficiently precise estimate of the weak error (bias), $\mathcal{E}_B(N)$, for different numbers of time steps N. For a fixed discretization, the corresponding estimated biased solution will be set as a reference solution to the ASGQ method in order to estimate the quadrature error $\mathcal{E}_Q(\text{TOL}_{\text{ASGQ}}, N)$.

5.2. Comparing the errors and computational time for MC, QMC and ASGQ

In this section, we conduct a comparison between MC, QMC and ASGQ, in terms of errors and computational time. We show tables and plots reporting the different relative errors

involved in the MC and QMC methods (bias and statistical error estimates), and in ASGQ (bias and quadrature error estimates). While fixing a sufficiently small relative error tolerance in the price estimates, we compare the computational time needed for all methods to meet the desired error tolerance. We note that, in all cases, the actual work (runtime) is obtained using an Intel(R) Xeon(R) CPU E5-268 architecture.

For the numerical experiments we conducted for each parameter set, we follow these steps to achieve the reported results:

(i) For a fixed number of time steps, N, we compute an accurate estimate, using a large number of samples, M, of the biased MC solution, C_{RB}^N. This step also provides us with an estimate of the bias error, $\mathcal{E}_B(N)$, defined by (10).

(ii) The estimated biased solution, C_{RB}^N, is used as a reference solution to the ASGQ method to compute the quadrature error, $\mathcal{E}_Q(\text{TOL}_{\text{ASGQ}}, N)$, defined by (13).

(iii) In order to compare the different methods, the number of samples, M^{QMC} and M^{MC}, are chosen so that the statistical errors of randomized QMC, $\mathcal{E}_{S,\text{QMC}}(M^{\text{QMC}})$, and MC, $\mathcal{E}_{S,\text{MC}}(M^{\text{MC}})$, satisfy

$$\mathcal{E}_{S,\text{QMC}}(M^{\text{QMC}}) = \mathcal{E}_{S,\text{MC}}(M^{\text{MC}}) = \mathcal{E}_B(N) = \frac{\mathcal{E}_{\text{tot}}}{2},$$
(20)

where $\mathcal{E}_B(N)$ is the bias as defined in (10) and $\mathcal{E}_{\text{tot}}$ is the total error.

We show the summary of our numerical findings in table 2, which highlights the computational gains achieved by ASGQ and QMC over the MC method to meet a certain error tolerance, which we set approximately to 1%. We note that the results are reported using the best configuration with Richardson extrapolation for each method. More detailed results for each case of parameter set, as in table 1, are provided in Sections 5.2.1–5.2.4.

5.2.1. Case of parameters in Set 1 in table 1. In this section, we conduct our numerical experiments for three different scenarios: (i) without Richardson extrapolation, (ii) with (level 1) Richardson extrapolation, and (iii) with (level 2) Richardson extrapolation. Figure 6 shows a comparison of the numerical complexity for each method under the three different scenarios. From this figure, we conclude that, to
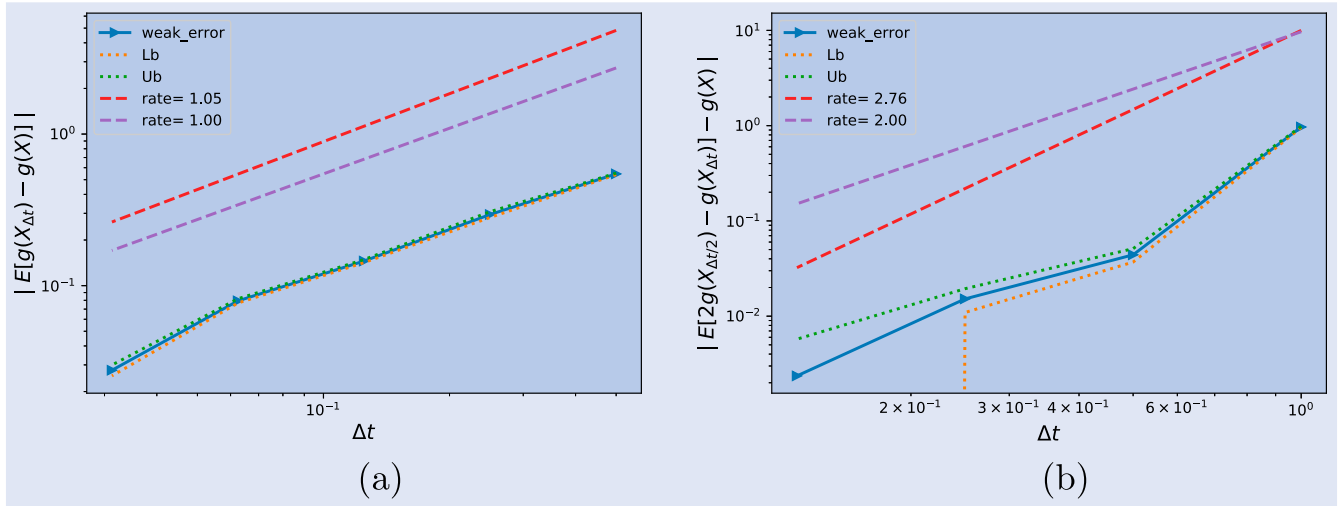
Figure 5. The convergence of the weak error $\mathcal{E}_B(N)$, defined in (8), using MC, for set 1 parameter in table 1. We refer to C_{RB} as $E[g(X)]$, and to C_{RB}^N as $E[g(X_{\Delta t})]$. The upper and lower bounds are 95% confidence intervals. (a) without Richardson extrapolation. (b) with Richardson extrapolation (level 1).

Table 2. Summary of relative errors and computational gains, achieved by the different methods.

Parameter set	Total relative error	CPU time ratio (ASGQ/MC)	CPU time ratio (QMC/MC)
Set 1	1%	6.7%	10%
Set 2	0.2%	4.7%	1.4%
Set 3	0.4%	3.8%	4.7%
Set 4	2%	20%	10%

Note: In this table, we highlight the computational gains achieved by ASGQ and QMC over the MC method to meet a certain error tolerance. We note that the ratios are computed for the best configuration with the Richardson extrapolation for each method. We provide details about how we compute these gains, for each case, in Sections 5.2.1–5.2.4.

achieve a relative error of 1%, the level 1 of the Richardson extrapolation is the optimal configuration for both the MC and the randomized QMC methods, and that the level 2 of the Richardson extrapolation is the optimal configuration for the ASGQ method.

We compare these optimal configurations for each method in figure 7, and we show that both ASGQ and QMC outperform MC, in terms of numerical complexity. In particular, to achieve a total relative error of 1%, ASGQ coupled with the level 2 of the Richardson extrapolation requires approximately 6.7% of the work of MC coupled with the level 1 of the Richardson extrapolation, and QMC coupled with the level 1 of the Richardson extrapolation requires approximately 10% of the work of MC coupled with the level 1 of the Richardson extrapolation. We show more detailed outputs for the methods that are compared in figure 7 in Appendix A.1.

5.2.2. Case of parameters in Set 2 in table 1. In this section, we only conduct our numerical experiments for the case without the Richardson extrapolation, since the results show that we meet a small enough relative error tolerance without the need to apply the Richardson extrapolation. We compare the different methods in figure 8, and we determine that both ASGQ and QMC outperform MC, in terms of numerical complexity. In particular, to achieve a total relative

error of approximately 0.2%, ASGQ requires approximately 4.7% of the work of MC, and that QMC requires approximately 1.4% of the work of MC. We show more detailed outputs for the methods that are compared in figure 8 in Appendix A.2.

5.2.3. Case of parameters in Set 3 in table tab1. In this section, we only conduct our numerical experiments for the case without the Richardson extrapolation, since the results show that we meet a small enough relative error tolerance without the need to apply the Richardson extrapolation. We compare the different methods in figure 9, and we determine that both ASGQ and QMC outperform MC, in terms of numerical complexity. In particular, to achieve a total relative error of approximately 0.4%, ASGQ requires approximately 3.8% of the work of MC, and that QMC requires approximately 4.7% of the work of MC. We show more detailed outputs for the methods that are compared in figure 9 in Appendix A.3.

5.2.4. Case of parameters in Set 4 in table 1. In this section, we only conduct our numerical experiments for the case the without Richardson extrapolation. We compare the different methods in figure 10, and we determine that both
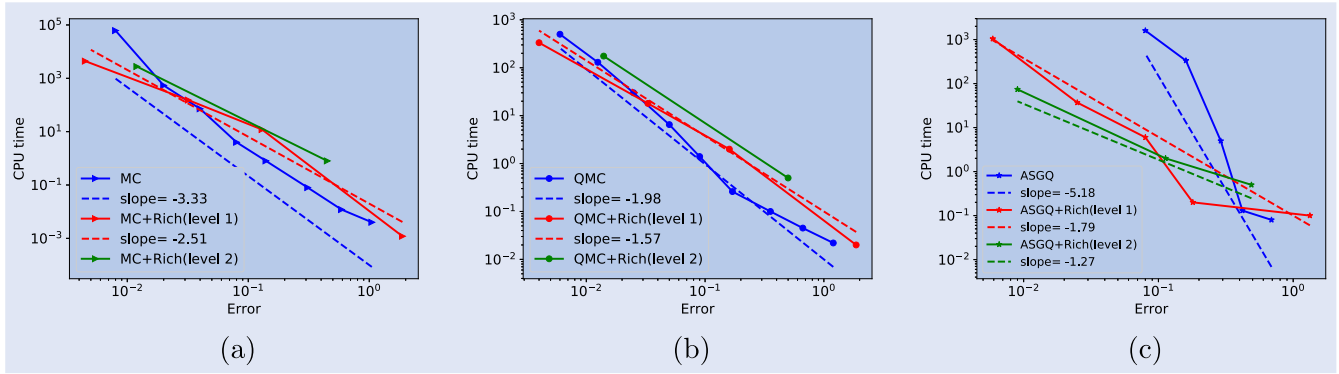
Figure 6. Comparing the numerical complexity of the different methods with the different configurations in terms of the level of Richardson extrapolation, for the case of parameter set 1 in table 1. (a) MC methods. (b) QMC methods. (d) ASGQ methods.
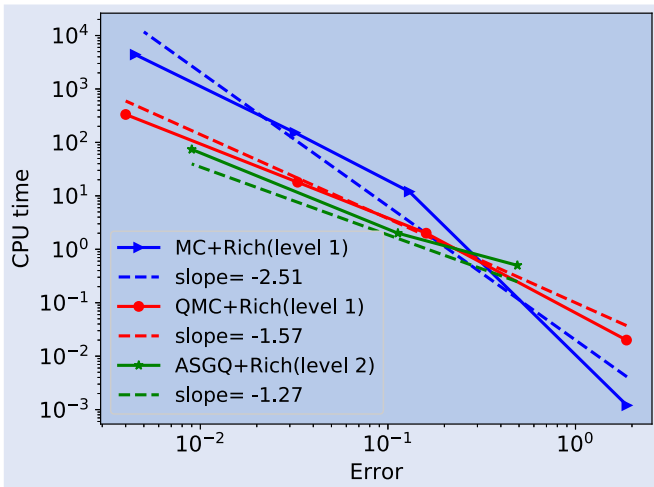

Figure 7. Computational work comparison for the different methods with the best configurations, as concluded from figure 6, for the case of parameter set 1 in table 1. This plot shows that, to achieve a relative error below 1%, ASGQ coupled with the level 2 of the Richardson extrapolation and QMC coupled with the level 1 of the Richardson extrapolation have a similar performance. Furthermore, they significantly outperform the MC method coupled with the level 1 of the Richardson extrapolation.

Figure 9. Comparison of computational work for the different methods, for the case of parameter set 3 in table 1.

Figure 8. Computational work comparison for the different methods, for the case of parameter set 2 in table 1. This plot shows that, to achieve a relative error below 1%, ASGQ and QMC have a similar performance and that they outperform the MC method significantly, in terms of computational time.
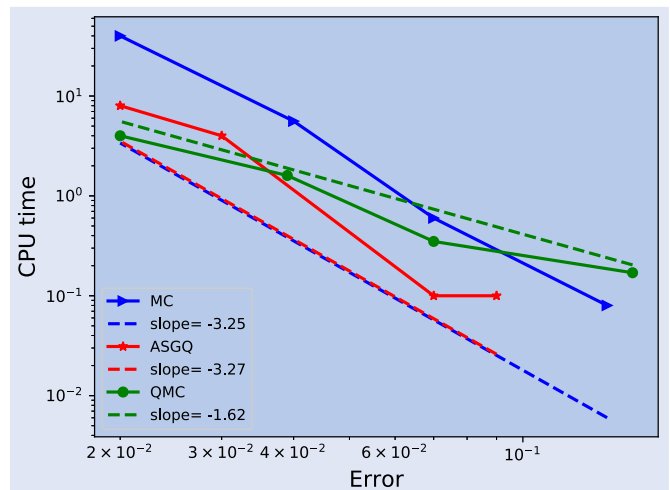
Figure 10. Comparison of computational work for the different methods, for the case of parameter set 4 in table 1.

ASGQ and QMC outperform MC, in terms of numerical complexity. In particular, to achieve a total relative error of approximately 2%, ASGQ requires approximately 20% of the work of MC, and QMC requires approximately 10% of the

work of MC. We show more detailed outputs for the methods that are compared in figure 10 in Appendix A.4. Similar to the case of set 1 parameters, illustrated in Section 5.2.1, we believe that the Richardson extrapolation will improve the performance of the ASGQ and QMC methods. We should also point out that, since we are in the out of the money regime in this case, a fairer comparison of the methods may be done after coupling them with an importance sampling method, so that more points are sampled in the right region of the payoff function.

6. Conclusions and future work

In this work, we propose novel fast option pricers for options whose underlyings follow the rBergomi model, as in Bayer *et al.* (2016). The new methods are based on hierarchical deterministic quadrature methods: (i) ASGQ using the same construction as in Haji-Ali *et al.* (2016), and (ii) the QMC method. Both techniques are coupled with a Brownian bridge construction and Richardson extrapolation of the weak error.

Given that in this context the only prevalent option is to use different variants of the MC method, which is computationally expensive, our main contribution is to propose a competitive alternative to the MC approach that is based on deterministic quadrature methods. We believe that we are the first to propose (and design) a pricing method in the context of rough volatility models that is based on deterministic quadrature, and that our approach opens a new research direction to investigate the performance of methods other than MC for pricing and calibrating rough volatility models.

Assuming one targets price estimates with a sufficiently small relative error tolerance, our proposed methods demonstrate substantial computational gains over standard MC when pricing under the rBergomi model, even for very small values of the Hurst parameter. We show these gains through our numerical experiments for different parameter constellations. We clarify that we do not claim that these gains will hold in the asymptotic regime, i.e. for higher accuracy requirements. Furthermore, the use of the Richardson extrapolation is justified in the pre-asymptotic regime, where our observed experimental results suggest a convergence of order one for the weak error. We emphasize that, to the best of our knowledge, no proper weak error analysis has been done in the rough volatility context, but we claim that both hybrid and exact schemes have a weak error of order one, for a particular class of payoff functions and in the pre-asymptotic regime, which justified numerically. While our focus is on the rBergomi model, our approach is applicable to a wide class of stochastic volatility models, in particular involving rough volatility.

In this work, we limit ourselves to compare our novel proposed method against standard MC. A more systematic comparison against the variant of MC proposed in McCrickerd and Pakkanen (2018) can be carried out, but this remains for future study. Another future research direction is to provide a reliable method for controlling the quadrature error for ASGQ which is, to the best of our knowledge, still an open research problem. This is even more challenging in our context, especially for low values of H. We emphasize that the main aim of this work is to illustrate the high potential of deterministic quadrature coupled with hierarchical representations for pricing options under the rBergomi model. Lastly, we note that accelerating our novel approach can be achieved using better versions of the ASGQ and QMC methods, or an alternative weak approximation of rough volatility based on a Donsker type approximation, as recently suggested in Horvath *et al.* (2017). The ASGQ and QMC methods proposed in this work can also be applied in the context of this scheme; here we have only an N-dimensional input rather than the $(2N)$-dimensional input required for the integration problem (9), with N being the number of time steps. Further analysis, in particular regarding the Richardson approximation, will be done in future research.

ORCID

Chiheb Ben Hammouda http://orcid.org/0000-0002-8386-0406

References

Abi Jaber, E., Lifting the Heston model. *Quant. Finance*, 2019, **19**(12), 1995–2013.

Acworth, P.A., Broadie, M. and Glasserman, P., A comparison of some Monte Carlo and quasi Monte Carlo techniques for option pricing. In *Monte Carlo and Quasi-Monte Carlo Methods 1996*, pp. 1–18, 1998 (Springer: New York).

Alòs, E., León, J.A. and Vives, J., On the short-time behavior of the implied volatility for jump-diffusion models with stochastic volatility. *Finance Stoch.*, 2007, **11**(4), 571–589.

Bajgrowicz, P., Scaillet, O. and Treccani, A., Jumps in high-frequency data: Spurious detections, dynamics, and news. *Manage. Sci.*, 2015, **62**(8), 2198–2217.

Bayer, C., Friz, P. and Gatheral, J., Pricing under rough volatility. *Quant. Finance*, 2016, **16**(6), 887–904.

Bayer, C., Friz, P.K., Gassiat, P., Martin, J. and Stemper, B., A regularity structure for rough volatility. Preprint, 2017. arXiv:1710.07481.

Bayer, C., Siebenmorgen, M. and Tempone, R., Smoothing the payoff for efficient computation of basket option pricing. *Quant. Finance*, 2018, **18**(3), 491–505.

Bayer, C., Friz, P.K., Gulisashvili, A., Horvath, B. and Stemper, B., Short-time near-the-money skew in rough fractional volatility models. *Quant. Finance*, 2019, **19**(5), 779–798.

Bennedsen, M., Lunde, A. and Pakkanen, M.S., Decoupling the short-and long-term behavior of stochastic volatility. Preprint, 2016. arXiv:1610.00332.

Bennedsen, M., Lunde, A. and Pakkanen, M.S., Hybrid scheme for Brownian semistationary processes. *Finance Stoch.*, 2017, **21**(4), 931–965.

Bergomi, L., Smile dynamics II. *Risk*, 2005, **18**, 67–73.

Biagini, F., Hu, Y., Øksendal, B. and Zhang, T.,*Stochastic Calculus for Fractional Brownian Motion and Applications*. Probability and its Applications, 2008 (Springer: London).

Bungartz, H.-J. and Griebel, M., Sparse grids. *Acta Numerica*, 2004, **13**, 147–269.

Caflisch, R.E., Morokoff, W.J. and Owen, A.B., Valuation of mortgage backed securities using Brownian bridges to reduce effective dimension, 1997.

Christensen, K., Oomen, R.C.A. and Podolskij, M., Fact or friction: Jumps at ultra high frequency. *J. Financ. Econ.*, 2014, **114**(3), 576–599.

Cools, R. and Nuyens, D., A Belgian view on lattice rules. In *Monte Carlo and Quasi-Monte Carlo Methods 2006*, pp. 3–21, 2008 (Springer: Berlin, Heidelberg).

Coutin, L., An introduction to (stochastic) calculus with respect to fractional Brownian motion. In *Séminaire de Probabilités XL*, pp. 3–65, 2007 (Springer: Berlin, Heidelberg).

El Euch, O. and Rosenbaum, M., Perfect hedging in rough Heston models. *Ann. Appl. Probab.*, 2018, **28**(6), 3813–3856.

El Euch, O. and Rosenbaum, M., The characteristic function of rough Heston models. *Math. Finance*, 2019, **29**(1), 3–38.

El Euch, O., Gatheral, J. and Rosenbaum, M., Roughening Heston, 2018. Available at SSRN 3116887.

Forde, M. and Zhang, H., Asymptotics for rough stochastic volatility models. *SIAM J. Financ. Math.*, 2017, **8**(1), 114–145.

Fukasawa, M., Asymptotic analysis for stochastic volatility: Martingale expansion. *Finance Stoch.*, 2011, **15**(4), 635–654.

Gatheral, J. and Keller-Ressel, M., Affine forward variance models. *Finance Stoch.*, 2019, **23**, 501–533.

Gatheral, J., Jaisson, T., Lesniewski, A. and Rosenbaum, M., Volatility is rough, part 2: Pricing. Workshop on Stochastic and Quantitative Finance, Imperial College London, London, 2014.

Gatheral, J., Jaisson, T. and Rosenbaum, M., Volatility is rough. *Quant. Finance*, 2018, **18**(6), 933–949.

Glasserman, P., *Monte Carlo Methods in Financial Engineering*, 2004 (Springer: New York).

Haji-Ali, A.-L., Nobile, F., Tamellini, L. and Tempone, R., Multi-index stochastic collocation for random PDEs. *Comput. Methods Appl. Mech. Eng.*, 2016, **306**, 95–122.

Horvath, B., Jacquier, A. and Muguruza, A., Functional central limit theorems for rough volatility, 2017. Available at SSRN 3078743.

Imai, J. and Tan, K.S., Minimizing effective dimension using linear transformation. In *Monte Carlo and Quasi-Monte Carlo Methods 2002*, pp. 275–292, 2004 (Springer: Berlin, Heidelberg).

Jaber, E.A., Larsson, M. and Pulido, S., Affine Volterra processes. *Ann. Appl. Probab.*, 2019, **29**(5), 3155–3200.

Jacquier, A., Martini, C. and Muguruza, A., On VIX futures in the rough Bergomi model. *Quant. Finance*, 2018a, **18**(1), 45–61.

Jacquier, A., Pakkanen, M.S. and Stone, H., Pathwise large deviations for the rough Bergomi model. *J. Appl. Probab.*, 2018b, **55**(4), 1078–1092.

Mandelbrot, B.B. and Van Ness, J.W., Fractional Brownian motions, fractional noises and applications. *SIAM Rev.*, 1968, **10**(4), 422–437.

Marinucci, D. and Robinson, P.M., Alternative forms of fractional Brownian motion. *J. Stat. Plan. Inference.*, 1999, **80**(1–2), 111–122.

McCrickerd, R. and Pakkanen, M.S., Turbocharging Monte Carlo pricing for the rough Bergomi model. *Quant. Finance*, 2018, **18**(11), 1877–1886.

Morokoff, W.J. and Caflisch, R.E., Quasi-random sequences and their discrepancies. *SIAM. J. Sci. Comput.*, 1994, **15**(6), 1251–1279.

Moskowitz, B. and Caflisch, R.E., Smoothness and dimension reduction in quasi-Monte Carlo methods. *Math. Comput. Model.*, 1996, **23**(8), 37–54.

Neuenkirch, A. and Shalaiko, T., The order barrier for strong approximation of rough volatility models. Preprint, 2016. arXiv:1606.03854.

Nuyens, D., The construction of good lattice rules and polynomial lattice rules, 2014.

Picard, J., Representation formulae for the fractional Brownian motion. In *Séminaire de Probabilités XLIII*, pp. 3–70, 2011 (Springer: Berlin, Heidelberg).

Romano, M. and Touzi, N., Contingent claims and market completeness in a stochastic volatility model. *Math. Finance*, 1997, **7**(4), 399–412.

Sloan, I.H., Lattice methods for multiple integration. *J. Comput. Appl. Math.*, 1985, **12**, 131–143.

Talay, D. and Tubaro, L., Expansion of the global error for numerical schemes solving stochastic differential equations. *Stoch. Anal. Appl.*, 1990, **8**(4), 483–509.

A. Appendix

A.1. Case of set 1 parameters in table 1

Table A1. Total relative error of MC and randomized QMC coupled with the Richardson extrapolation (level 1), to compute the call option price for different numbers of time steps.

Method	Steps			
	1–2	2–4	4–8	8–16
QMC + level 1 of Richardson extrapolation	**1.87** (0.96,0.91)	**0.16** (0.07,0.09)	**0.033** (0.015,0.018)	**0.0044** (0.002,0.002)
M(# QMC samples)	128	8192	131 072	2 097 152
MC + level 1 of Richardson extrapolation	**1.88** (0.96,0.92)	**0.14** (0.07,0.07)	**0.03** (0.015,0.015)	**0.0044** (0.002,0.0024)
M(# MC samples)	4×10	8×10^3	16×10^4	5×10^5

Notes: The values between parentheses correspond to the different errors contributing to the total relative error: the bias and the statistical error estimates. The number of MC and QMC samples, M, are chosen to satisfy (20).

Table A2. Comparison of the computational time (in seconds) of MC and randomized QMC coupled with the Richardson extrapolation (level 1) to compute the call option price of the rBergomi model for different numbers of time steps.

Method	Steps			
	1–2	2–4	4–8	8–16
QMC + level 1 of Richardson extrapolation	0.018	2	18	333
MC + level 1 of Richardson extrapolation	0.0012	12	152	4400

Note: The average MC CPU time is computed over 100 runs.

Table A3. Total relative error of ASGQ, coupled with the Richardson extrapolation (level 2), to compute the call option price for different numbers of time steps.

Method	Steps	
	1–2–4	2–4–8
ASGQ + level 2 of Richardson extrapolation ($\text{TOL}_{\text{ASGQ}} = 10^{-1}$)	**0.54** (0.24,0.30)	**0.113** (0.006,0.107)
ASGQ + level 2 of Richardson extrapolation ($\text{TOL}_{\text{ASGQ}} = 5.10^{-2}$)	**0.49** (0.24,0.25)	**0.009** (0.006,0.003)

Note: The values between parentheses correspond to the different errors contributing to the total relative error: the bias and quadrature errors.

Table A4. Comparison of the computational time (in seconds) of ASGQ coupled with the Richardson extrapolation (level 2) to compute the call option price of the rBergomi model for different numbers of time steps.

Method	Steps	
	1–2–4	2–4–8
ASGQ + level 2 of Richardson extrapolation ($\text{TOL}_{\text{ASGQ}} = 10^{-1}$)	0.2	2
ASGQ + level 2 of Richardson extrapolation ($\text{TOL}_{\text{ASGQ}} = 5.10^{-2}$)	0.5	74

A.2. Case of set 2 parameters in table 1

Table A5. Total relative error of the different methods without the Richardson extrapolation, to compute the call option price for different numbers of time steps.

Method	Steps			
	2	4	8	16
ASGQ ($\text{TOL}_{\text{ASGQ}} = 10^{-1}$)	**0.03** (0.02,0.01)	**0.022** (0.008,0.014)	**0.022** (0.004,0.018)	**0.017** (0.001,0.016)
ASGQ ($\text{TOL}_{\text{ASGQ}} = 10^{-2}$)	**0.03** (0.02,0.01)	**0.017** (0.008,0.009)	**0.008** (0.004,0.004)	**0.001** (0.001,4e−04)
QMC	**0.04** (0.02,0.02)	**0.017** (0.008,0.009)	**0.008** (0.004,0.004)	**0.002** (0.001,0.001)
M(# QMC samples)	4096	8192	32 768	262 144
MC	**0.04** (0.02,0.02)	**0.016** (0.008,0.008)	**0.007** (0.004,0.003)	**0.002** (0.001,0.001)
M(# MC samples)	16×10^3	8×10^4	4×10^5	4×10^6

Notes: The values between parentheses correspond to the different errors contributing to the total relative error; for ASGQ, we report the bias and quadrature errors, and for MC and QMC, we report the bias and the statistical error estimates. The number of MC and QMC samples, M, are chosen to satisfy (20).

Table A6. Comparison of the computational time (in seconds) of the different methods to compute the call option price of the rBergomi model for different numbers of time steps.

Method	Steps			
	2	4	8	16
ASGQ ($\text{TOL}_{\text{ASGQ}} = 10^{-1}$)	0.1	0.1	0.2	0.8
ASGQ ($\text{TOL}_{\text{ASGQ}} = 10^{-2}$)	0.1	0.5	8	92
QMC method	0.3	0.7	3.25	27
MC method	0.6	6.4	66	1976

Note: The average MC CPU time is computed over 100 runs.

A.3. Case of set 3 parameters in table 1

Table A7. Total relative error of the different methods without the Richardson extrapolation, to compute the call option price for different numbers of time steps.

Method	Steps			
	2	4	8	16
ASGQ ($\text{TOL}_{\text{ASGQ}} = 10^{-1}$)	**0.008** (0.006,0.002)	**0.009** (0.004,0.005)	**0.008** (0.003,0.005)	**0.009** (0.002,0.007)
ASGQ ($\text{TOL}_{\text{ASGQ}} = 10^{-2}$)	**0.008** (0.006,0.002)	**0.009** (0.004,0.005)	**0.005** (0.003,0.002)	**0.002** (0.002,1e−04)
ASGQ ($\text{TOL}_{\text{ASGQ}} = 10^{-3}$)	**0.008** (0.006,0.002)	**0.006** (0.004,0.002)	**0.003** (0.003,1e−04)	**0.002** (0.002,1e−04)
ASGQ ($\text{TOL}_{\text{ASGQ}} = 10^{-4}$)	**0.006** (0.006,4e−04)	**0.004** (0.004,2e−04)	**0.003** (0.003,1e−04)	−
QMC	**0.015** (0.006,0.009)	**0.008** (0.004,0.004)	**0.0066** (0.003,0.0036)	**0.004** (0.002,0.002)
M(# QMC samples)	$2^3 \times 2^{10} = 8192$	$2^3 \times 2^{11} = 16\,384$	$2^3 \times 2^{12} = 32\,768$	$2^3 \times 2^{13} = 65\,536$
MC	**0.01** (0.006,0.005)	**0.008** (0.004,0.004)	**0.006** (0.003,0.003)	**0.004** (0.002,0.002)
M(# MC samples)	8×10^4	16×10^4	24×10^4	32×10^4

Notes: The values between parentheses correspond to the different errors contributing to the total relative error; for ASGQ, we report the bias and quadrature errors, and for MC and QMC, we report the bias and the statistical error estimates. The number of MC and QMC samples, M, are chosen to satisfy (20).

Table A8. Comparison of the computational time (in seconds) of the different methods to compute the call option price of the rBergomi model for different numbers of time steps.

Method	Steps			
	2	4	8	16
ASGQ ($\text{TOL}_{\text{ASGQ}} = 10^{-1}$)	0.1	0.1	0.1	1
ASGQ ($\text{TOL}_{\text{ASGQ}} = 10^{-2}$)	0.1	0.15	9	112
ASGQ ($\text{TOL}_{\text{ASGQ}} = 10^{-3}$)	0.2	2	27	2226
ASGQ ($\text{TOL}_{\text{ASGQ}} = 10^{-4}$)	1	6	136	–
QMC method	0.65	1.4	3.25	7.5
MC method	4	12	40	160

Note: The average MC CPU time is computed over 100 runs.

A.4. Case of set 4 parameters in table 1

Table A9. Total relative error of the different methods without the Richardson extrapolation, to compute the call option price for different numbers of time steps.

Method	Steps			
	2	4	8	16
ASGQ ($\text{TOL}_{\text{ASGQ}} = 10^{-1}$)	**0.09** (0.07,0.05)	**0.07** (0.03,0.04)	**0.07** (0.02,0.05)	**0.06** (0.01,2e−04)
ASGQ ($\text{TOL}_{\text{ASGQ}} = 10^{-2}$)	**0.09** (0.07,5e−04)	**0.07** (0.03,0.04)	**0.02** (0.02,3e−04)	**0.02** (0.01,2e−04)
ASGQ ($\text{TOL}_{\text{ASGQ}} = 10^{-3}$)	**0.07** (0.07,5e−04)	**0.03** (0.03,4e−04)	**0.02** (0.02,3e−04)	**0.01** (0.01,2e−04)
QMC	**0.155** (0.07,0.085)	**0.07** (0.03,0.04)	**0.039** (0.02,0.019)	**0.02** (0.01,0.01)
M(# QMC samples)	$2^3 \times 2^8 = 2048$	$2^3 \times 2^9 = 4096$	$2^3 \times 2^{11} = 16\,384$	$2^3 \times 2^{12} = 32\,768$
MC	**0.14** (0.07,0.07)	**0.07** (0.03,0.04)	**0.04** (0.02,0.02)	**0.02** (0.01,0.01)
M(# MC samples)	24×10^2	8×10^3	32×10^3	8×10^4

Notes: The values between parentheses correspond to the different errors contributing to the total relative error; for ASGQ, we report the bias and quadrature errors, and for MC and QMC, we report the bias and the statistical error estimates. The number of MC and QMC samples, *M*, are chosen to satisfy (20).

Table A10. Comparison of the computational time (in seconds) of the different methods to compute the call option price of rBergomi model for different numbers of time steps.

Method	Steps			
	2	4	8	16
ASGQ ($\text{TOL}_{\text{ASGQ}} = 10^{-1}$)	0.1	0.1	0.2	0.5
ASGQ ($\text{TOL}_{\text{ASGQ}} = 10^{-2}$)	0.1	0.1	8	97
ASGQ ($\text{TOL}_{\text{ASGQ}} = 10^{-3}$)	0.7	4	26	1984
QMC method	0.17	0.35	1.6	4
MC method	0.08	0.6	5.6	40

Note: The average MC CPU time is computed over 100 runs.