# Hierarchical adaptive sparse grids for option pricing under the rough Bergomi model

Christian Bayer* Chiheb Ben Hammouda† Raul Tempone‡§

February 25, 2019

**Abstract**

The rough Bergomi (rBergomi) model, introduced recently in [4], is a promising rough volatility model in quantitative finance. This new model exhibits consistent results with the empirical fact of implied volatility surfaces being essentially time-invariant. This model also has the ability to capture the term structure of skew observed in equity markets. 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 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 design a novel, alternative, hierarchical approach, based on adaptive sparse grids quadrature, specifically using the same construction as multi-index stochastic collocation (MISC) [21], coupled with Brownian bridge construction and Richardson extrapolation. By uncovering the available regularity, our hierarchical method demonstrates substantial computational gains with respect to the standard MC method, when reaching a sufficiently small 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, Brownian bridge construction, Richardson extrapolation.

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

## 1 Details of Cholesky scheme coupled with hierarchical reresentation

Let us denote by the matrix $A$, the computable covariance matrix of $\widetilde{W}^H_{t_1}, \dots, \widetilde{W}_{t_N}, W^1_{t_1}, \dots, W^1_{t_N}$. We can use Cholesky decomposition of $A$ to produce exact samples of $W^1_{t_1}, \dots, W^1_{t_N}, \widetilde{W}^H_{t_1}, \dots, \widetilde{W}_{t_N}$.

*Weierstrass Institute for Applied Analysis and Stochastics (WIAS), Berlin, Germany.

†King Abdullah University of Science and Technology (KAUST), Computer, Electrical and Mathematical Sciences & Engineering Division (CEMSE), Thuwal $23955-6900$, Saudi Arabia (`chiheb.benhammouda@kaust.edu.sa`).

‡King Abdullah University of Science and Technology (KAUST), Computer, Electrical and Mathematical Sciences & Engineering Division (CEMSE), Thuwal $23955-6900$, Saudi Arabia (`raul.tempone@kaust.edu.sa`).

§Alexander von Humboldt Professor in Mathematics for Uncertainty Quantification, RWTH Aachen University, Germany.

In fact let us denote by $L$ the triangular matrix resulting from Cholesky decomposition such that

$$L = \left( \begin{array}{c|c} L_1 & 0 \\ L_2 & L_3 \end{array} \right),$$

where $L_1, L_2, L_3$ are $N \times N$ matrices, such that $L_1$ and $L_3$ are triangular.

Then, given a $2N \times 1$-dimensional Gaussian random input vector, $\mathbf{X} = (X_1, \dots, X_N, X_{N+1}, \dots, X_{2N})'$, we have

$$\mathbf{W}^{(1)} = L_1 \mathbf{X}_{1:N}, \quad \widetilde{\mathbf{W}} = \left( \begin{array}{c|c} L_2 & L_3 \end{array} \right) \mathbf{X}. \tag{1.1}$$

On the other hand, let us assume that we can construct $\mathbf{W}^{(1)}$ hierarchically through Brownian bridge construction defined by the linear mapping given by the matrix $G$, then given a $N$-dimensional Gaussian random input vector, $\mathbf{Z}'$, we can write

$$\mathbf{W}^{(1)} = G\mathbf{Z}',$$

and consequently

$$\mathbf{X}_{1:N} = L_1^{-1} G \mathbf{Z}'.$$

Therefore, given a $2N$-dimensional Gaussian random input vector, $\mathbf{Z} = (\mathbf{Z}', \mathbf{Z}'')$, we define our hierarchical representation by

$$\mathbf{X} = \left( \begin{array}{c|c} L_1^{-1} G & 0 \\ 0 & I_N \end{array} \right) \mathbf{Z}. \tag{1.2}$$

We need to make sure that $\mathbf{X}$ has Gaussian distribution as an outcome of the construction (4.2). Consequently, we need to compute carefully $L_1^{-1}$. Actually, I observed that $L_1 = I_{N \times N}$. Therefore, $\mathbf{X}$ has Gaussian distribution as an outcome of the construction (4.2).

**TO-DO** 1: Implement the appropriate Cholesky scheme, taking into account the above construction, and check if the hierarchical construction is giving good results.

## 2 Numerical experiments

In this section, we show the results obtained through the different numerical experiments, conducted across different parameter constellations for the rBergomi model. Details about these examples are presented in Table 5.1. The first set is the one that is closest to the empirical findings [8, 19], which suggest that $H \approx 0.1$. The choice of parameters values of $\nu = 1.9$ and $\rho = -0.9$ is justified by [4], 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 5.1, we wanted 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 Multi-level Monte Carlo (MLMC), are inefficient in this context, because of the poor behavior of the strong error, that is of the order of $H$ [30]. 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 5.1. 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 5.1.

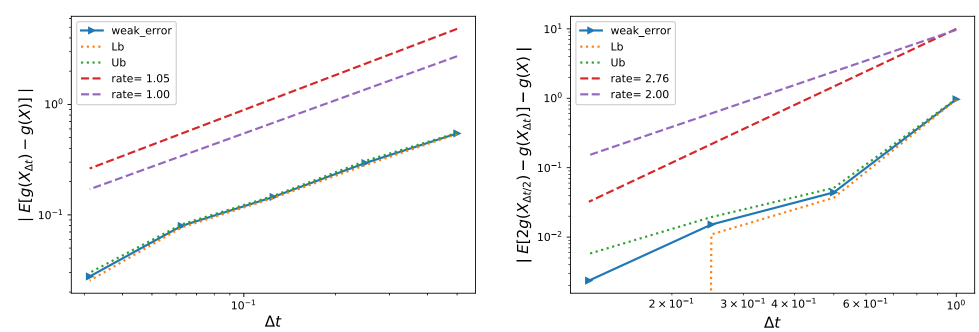
| Parameters | Reference solution |
|---|---|
| Set 1: $H = 0.07, K = 1, S_0 = 1, T = 1, \rho = -0.9, \eta = 1.9, \xi_0 = 0.235^2$ | 0.0791 (7.9e−05) |
| Set 2: $H = 0.02, K = 1, S_0 = 1, T = 1, \rho = -0.7, \eta = 0.4, \xi_0 = 0.1$ | 0.1248 (1.3e−04) |
| Set 3: $H = 0.02, K = 0.8, S_0 = 1, T = 1, \rho = -0.7, \eta = 0.4, \xi_0 = 0.1$ | 0.2407 (5.6e−04) |
| Set 4: $H = 0.02, K = 1.2, S_0 = 1, T = 1, \rho = -0.7, \eta = 0.4, \xi_0 = 0.1$ | 0.0568 (2.5e−04) |

Table 2.1: Reference solution, which is the approximation of the call option price under the rBergomi model, defined in (2.4), using MC with 500 time steps and number of samples, $M = 10^6$, for different parameter constellations. The numbers between parentheses correspond to the statistical errors estimates.

## 2.1 Weak error

We start our numerical experiments with accurately estimating the weak error (bias) for the different parameter sets in Table 5.1, with and without Richardson extrapolation.

For illustrative purposes, we only show the weak errors related to set 1 in Table 5.1 (see Figure 5.1). We note that we observed 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. Our results are purely experimental, and hence we cannot be sure what will happen in the asymptotic regime. 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 MISC method in order to estimate the quadrature error $\mathcal{E}_Q(\mathrm{TOL}_{\mathrm{MISC}}, N)$.




(a) (b)

Figure 2.1: The convergence of the weak error $\mathcal{E}_B(N)$, defined in (3.2), using MC, for set 1 parameter in Table 5.1. We refer to $C_{\mathrm{RB}}$ as $\mathrm{E}[g(X)]$, and to $C_{\mathrm{RB}}^N$ as $\mathrm{E}[g(X_{\Delta t})]$. The upper and lower bounds are 95% confidence intervals. a) without Richardson extrapolation. b) with Richardson extrapolation (level 1).

## 2.2 Comparing the different errors and computational time for MC and MISC

In this section, we conduct a comparison between MC and MISC in terms of errors and computational time. We show tables and plots reporting the different relative errors involved in the MC method (bias and statistical error[1] estimates), and in MISC (bias and quadrature error estimates). While fixing a sufficiently small error tolerance in the price estimates, we also compare the computational time needed for both 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.

Through our conducted numerical experiments for each parameter set, we follow these steps to achieve our 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 (3.2).

ii) The estimated biased solution, $C_{RB}^N$, is used as a reference solution to MISC to compute the quadrature error, $\mathcal{E}_Q(\text{TOL}_{\text{MISC}}, N)$, defined by (3.4).

iii) In order to compare with MC method, the number of samples, $M$, is chosen so that the statistical error of the Monte Carlo method, $\mathcal{E}_S(M)$, satisfies

$$\mathcal{E}_S(M) = \mathcal{E}_B(N) = \frac{\mathcal{E}_{\text{tot}}}{2}, \tag{2.1}$$

where $\mathcal{E}_B(N)$ is the bias as defined in (3.2) and $\mathcal{E}_{\text{tot}}$ is the total error.

We show the summary of our numerical findings in Table 5.2, which highlights the computational gains achieved by MISC over MC method to meet a certain error tolerance, which we set approximately to 1%. More detailed results for each case of parameter set, as in Table 5.1, are provided in Sections 5.2.1, 5.2.2, 5.2.3 and 5.2.4.

| Parameter set | Level of Richardson extrapolation | Total relative error | Ratio of CPU time (MC/MISC) |
|---|---|---|---|
| Set 1 | level 1 | 3% | 1.6 |
| | level 2 | 1% | $> 9$ |
| Set 2 | without | 0.2% | 5 |
| Set 3 | without | 0.4% | 7 |
| Set 4 | without | 2% | 1.3 |

Table 2.2: Summary of relative errors and computational gains, achieved by the different methods. In this table, we highlight the computational gains achieved by MISC over MC method to meet a certain error tolerance. As expected, these gains are improved when applying Richardson extrapolation as observed for the case of parameters set 1. We provide details about the way we compute these gains for each case in the following sections.

[1] The statistical error estimate of MC is $\frac{\sigma_M}{\sqrt{M}}$, which is the standard deviation estimate of the MC estimator, where $M$ is the number of samples.

### 2.2.1 Case of parameters in Set 1, in Table 5.1

In this section, we conduct our numerical experiments for three different scenarios: i) without Richardson extrapolation (see Tables 5.3 and 5.4), ii) with (level 1) Richardson extrapolation (see Tables 5.5 and 5.6), and iii) with (level 2) Richardson extrapolation (see Tables 5.7 and 5.8). Our numerical experiments show that MISC coupled with (level 1) Richardson extrapolation requires approximately 60% of the work of MC coupled with (level 1) Richardson extrapolation, to achieve a total relative error of around 3%. This gain is improved further when applying level 2 Richardson extrapolation. In fact, MISC coupled with (level 2) Richardson extrapolation requires approximately less than 10% of the work of MC coupled with (level 2) Richardson extrapolation, to achieve a total relative error below 1%. Applying Richardson extrapolation brought a significant improvement for MISC (see Figure 5.2 and Tables 5.3,5.4,5.5,5.6,5.7,5.8).

| Method | Steps | | | |
|---|---|---|---|---|
| | 2 | 4 | 8 | 16 |
| MISC ($\text{TOL}_{\text{MISC}} = 10^{-1}$) | **0.69** (0.54,0.15) | **0.42** (0.29,0.13) | **0.31** (0.15,0.16) | **0.11** (0.07,0.04) |
| MISC ($\text{TOL}_{\text{MISC}} = 10^{-2}$) | **0.66** (0.54,0.12) | **0.29** (0.29,$6e{-}04$) | **0.16** (0.15,0.01) | **0.08** (0.07,0.01) |
| MC | **1.05** (0.54,0.51) | **0.59** (0.295,0.295) | **0.31** (0.155,0.155) | **0.14** (0.07,0.07) |
| M(# MC samples) | $2 \times 10$ | $4 \times 10$ | $10^2$ | $4 \times 10^2$ |

Table 2.3: Total relative error of MISC, without Richardson extrapolation, with different tolerances, and MC to compute the call option prices for different numbers of time steps. The values between parentheses correspond to the different errors contributing to the total relative error: for MISC we report the bias and quadrature errors and for MC we report the bias and the statistical errors estimates. The number of MC samples, $M$, is chosen to satisfy (5.1).

| Method | Steps | | | |
|---|---|---|---|---|
| | 2 | 4 | 8 | 16 |
| MISC ($\text{TOL}_{\text{MISC}} = 10^{-1}$) | 0.08 | 0.13 | 0.7 | 163 |
| MISC ($\text{TOL}_{\text{MISC}} = 10^{-2}$) | 0.2 | 5 | 333 | 1602 |
| MC method | 0.001 | 0.003 | 0.02 | 0.2 |

Table 2.4: Comparison of the computational time (in seconds) of MC and MISC, to compute the call option price of the rBergomi model for different numbers of time steps. The average MC CPU time is computed over 100 runs.

| Method | Steps | | |
| --- | --- | --- | --- |
| | $1-2$ | $2-4$ | $4-8$ |
| MISC ($\text{TOL}_{\text{MISC}} = 10^{-1}$) | **1.33** (0.96,0.37) | **0.18** (0.07,0.11) | **0.144** (0.015,0.129) |
| MISC ($\text{TOL}_{\text{MISC}} = 5.10^{-2}$) | **1.33** (0.96,0.37) | **0.23** (0.07,0.16) | **0.025** (0.015,0.010) |
| MISC ($\text{TOL}_{\text{MISC}} = 10^{-2}$) | **1.08** (0.96,0.12) | **0.08** (0.07,0.01) | **0.025** (0.015,0.010) |
| MC | **1.88** (0.96,0.92) | **0.14** (0.07,0.07) | **0.03** (0.015,0.015) |
| M(# MC samples) | 10 | $2 \times 10^3$ | $4 \times 10^4$ |

Table 2.5: Total relative error of MISC, coupled with Richardson extrapolation (level 1), with different tolerances, and MC, coupled with Richardson extrapolation (level 1), to compute the call option price for different numbers of time steps. The values between parentheses correspond to the different errors contributing to the total relative error: for MISC we report the bias and quadrature errors and for MC we report the bias and the statistical errors. The number of MC samples, $M$, is chosen to satisfy (5.1). The values marked in red correspond to the values used for computational work comparison against MC method, reported in Table 5.2.

| Method | Steps | | |
| --- | --- | --- | --- |
| | $1-2$ | $2-4$ | $4-8$ |
| MISC ($\text{TOL}_{\text{MISC}} = 10^{-1}$) | 0.1 | 0.2 | 1.6 |
| MISC ($\text{TOL}_{\text{MISC}} = 5.10^{-2}$) | 0.1 | 0.6 | 37 |
| MISC ($\text{TOL}_{\text{MISC}} = 10^{-2}$) | 1.3 | 6 | 2382 |
| MC | 0.003 | 2 | 60 |

Table 2.6: Comparison of the computational time (in seconds) of MC and MISC, using Richardson extrapolation (level 1), to compute the call option price of the rBergomi model for different numbers of time steps. The average MC CPU time is computed over 100 runs. The values marked in red correspond to the values used for computational work comparison against MC method, reported in Table 5.2.

| Method | Steps | |
|---|---|---|
| | $1-2-4$ | $2-4-8$ |
| MISC ($TOL_{MISC}=10^{-1}$) | **0.54** (0.24,0.30) | **0.113** (0.006,0.107) |
| MISC ($TOL_{MISC}=5.10^{-2}$) | **0.49** (0.24,0.25) | **0.009** (0.006,0.003) |
| MISC ($TOL_{MISC}=10^{-2}$) | **0.27** (0.24,0.03) | **0.009** (0.006,0.003) |
| MC | **0.45** (0.24,0.21) | **0.012** (0.006,0.006) |
| M(# MC samples) | $4\times10^2$ | $4\times10^5$ |

Table 2.7: Total relative error of MISC, coupled with Richardson extrapolation (level 2), with different tolerances, and MC, coupled with Richardson extrapolation (level 2), to compute the call option price for different numbers of time steps. The values between parentheses correspond to the different errors contributing to the total relative error: for MISC we report the bias and quadrature errors and for MC we report the bias and the statistical errors. The number of MC samples, $M$, is chosen to satisfy (5.1). The values marked in red correspond to the values used for computational work comparison against MC method, reported in Table 5.2.

| Method | Steps | |
|---|---|---|
| | $1-2-4$ | $2-4-8$ |
| MISC ($TOL_{MISC}=10^{-1}$) | 0.2 | 2 |
| MISC ($TOL_{MISC}=5.10^{-2}$) | 0.5 | 74 |
| MISC ($TOL_{MISC}=10^{-2}$) | 9 | 3455 |
| MC | 0.2 | 690 |

Table 2.8: Comparison of the computational time (in seconds) of MC and MISC, using Richardson extrapolation (level 2), to compute the call option price of the rBergomi model for different numbers of time steps. The average MC CPU time is computed over 100 runs. The values marked in red correspond to the values used for computational work comparison against MC method, reported in Table 5.2.

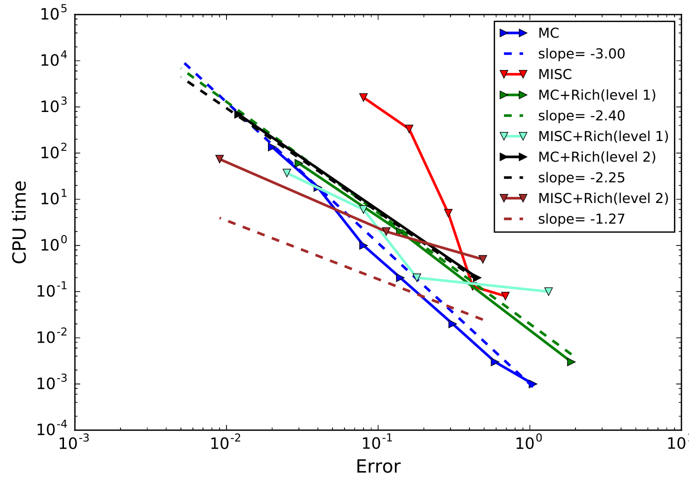



Figure 2.2: Computational work comparison for MISC and MC methods (with and without) Richardson extrapolation, for the case of parameter set 1 in Table 5.1. This plot shows that to achieve a relative error below 1%, MISC coupled with level 2 of Richardson extrapolation is the best option in terms of computational time. Furthermore, applying Richardson extrapolation brings a significant improvement for MISC and MC methods, in terms of numerical complexity.

### 2.2.2 Case of parameters in Set 2, in Table 5.1

In this section, we only conduct our numerical experiments for the case without Richardson extrapolation, since the results show that we meet a small enough error tolerance without the need to apply Richardson extrapolation. Our numerical experiments show that MISC requires approximately 20% of the work of MC method, to achieve a total relative error of around 0.2% (see Figure 5.3 and Tables 5.10 and 5.9).

| Method | Steps | | | |
| --- | --- | --- | --- | --- |
| | 2 | 4 | 8 | 16 |
| MISC ($\text{TOL}_{\text{MISC}} = 10^{-1}$) | **0.03** (0.02,0.01) | **0.022** (0.008,0.014) | **0.022** (0.004,0.018) | **0.017** (0.001,0.016) |
| MISC ($\text{TOL}_{\text{MISC}} = 10^{-2}$) | **0.03** (0.02,0.01) | **0.017** (0.008,0.009) | **0.008** (0.004,0.004) | **0.001** (0.001,$4e{-}04$) |
| MISC ($\text{TOL}_{\text{MISC}} = 10^{-3}$) | **0.02** (0.02,$8e{-}04$) | **0.009** (0.008,$8e{-}04$) | **0.005** (0.004,$8e{-}04$) | **0.001** (0.001,$4e{-}04$) |
| 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) | $4 \times 10^3$ | $2 \times 10^4$ | $10^5$ | $10^6$ |

Table 2.9: Total relative error of MISC, without Richardson extrapolation, with different tolerances, and MC to compute the call option price for different numbers of time steps. The values between parentheses correspond to the different errors contributing to the total relative error: for MISC we report the bias and quadrature errors and for MC we report the bias and the statistical errors estimates. The number of MC samples, $M$, is chosen to satisfy (5.1). The values marked in red correspond to the values used for computational work comparison against MC method, reported in Table 5.2.

| Method | Steps | | | |
| --- | --- | --- | --- | --- |
| | 2 | 4 | 8 | 16 |
| MISC ($\text{TOL}_{\text{MISC}} = 10^{-1}$) | 0.1 | 0.1 | 0.2 | 0.8 |
| MISC ($\text{TOL}_{\text{MISC}} = 10^{-2}$) | 0.1 | 0.5 | 8 | 92 |
| MISC ($\text{TOL}_{\text{MISC}} = 10^{-3}$) | 0.5 | 3 | 24 | 226 |
| MC method | 0.15 | 1.6 | 16.5 | 494 |

Table 2.10: Comparison of the computational time (in seconds) of MC and MISC, to compute the call option price of the rBergomi model for different numbers of time steps. The average MC CPU time is computed over 100 runs. The values marked in red correspond to the values used for computational work comparison against MC method, reported in Table 5.2.

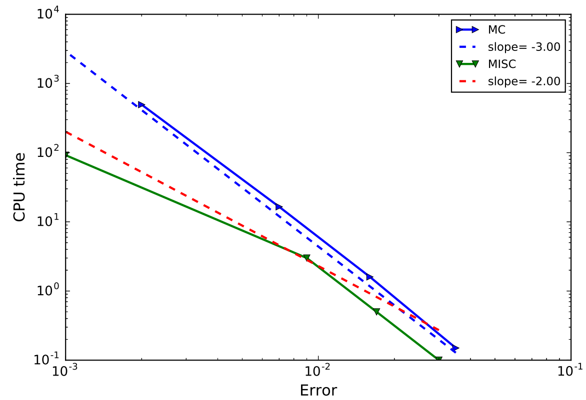



Figure 2.3: Computational work comparison for MISC and MC methods, for the case of parameter set 2 in Table 5.1. This plot shows that to achieve a relative error below 1%, MISC outperforms MC method in terms of computational time.

### 2.2.3 Case of parameters in Set 3, in Table 5.1

In this section, we only conduct our numerical experiments for the case without Richardson extrapolation, since the results show that we meet a small enough error tolerance without the need to apply Richardson extrapolation. Our numerical experiments show that MISC requires approximately 14% of the work of MC method, to achieve a total relative error of around 0.4% (see Figure 5.4 and Tables 5.12 and 5.11).

| Method | Steps | | | |
|---|---|---|---|---|
| | 2 | 4 | 8 | 16 |
| MISC ($\text{TOL}_{\text{MISC}} = 10^{-1}$) | **0.008** (0.006,0.002) | **0.009** (0.004,0.005) | **0.008** (0.003,0.005) | **0.009** (0.002,0.007) |
| MISC ($\text{TOL}_{\text{MISC}} = 10^{-2}$) | **0.008** (0.006,0.002) | **0.009** (0.004,0.005) | **0.005** (0.003,0.002) | **0.002** (0.002,$1e{-}04$) |
| MISC ($\text{TOL}_{\text{MISC}} = 10^{-3}$) | **0.008** (0.006,0.002) | **0.006** (0.004,0.002) | **0.003** (0.003,$1e{-}04$) | **0.002** (0.002,$1e{-}04$) |
| MISC ($\text{TOL}_{\text{MISC}} = 10^{-4}$) | **0.006** (0.006,$4e{-}04$) | **0.004** (0.004,$2e{-}04$) | **0.003** (0.003,$1e{-}04$) | − |
| 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) | $2 \times 10^4$ | $4 \times 10^4$ | $6 \times 10^4$ | $8 \times 10^4$ |

Table 2.11: Total relative error of MISC, without Richardson extrapolation, with different tolerances, and MC to compute the call option price for different numbers of time steps. The values between parentheses correspond to the different errors contributing to the total relative error: for MISC we report the bias and quadrature errors and for MC we report the bias and the statistical errors estimates. The number of MC samples, $M$, is chosen to satisfy (5.1). The values marked in red correspond to the values used for computational work comparison against MC method, reported in Table 5.2.

| Method | Steps | | | |
|---|---|---|---|---|
| | 2 | 4 | 8 | 16 |
| MISC ($\mathrm{TOL}_{\mathrm{MISC}} = 10^{-1}$) | 0.1 | 0.1 | 0.1 | 1 |
| MISC ($\mathrm{TOL}_{\mathrm{MISC}} = 10^{-2}$) | 0.1 | 0.15 | 9 | 112 |
| MISC ($\mathrm{TOL}_{\mathrm{MISC}} = 10^{-3}$) | 0.2 | 2 | 27 | 2226 |
| MISC ($\mathrm{TOL}_{\mathrm{MISC}} = 10^{-4}$) | 1 | 6 | 136 | − |
| MC method | 1 | 3 | 10 | 40 |

Table 2.12: Comparison of the computational time (in seconds) of MC and MISC, to compute the call option price of the rBergomi model for different numbers of time steps. The average MC CPU time is computed over 100 runs. The values marked in red correspond to the values used for computational work comparison against MC method, reported in Table 5.2.

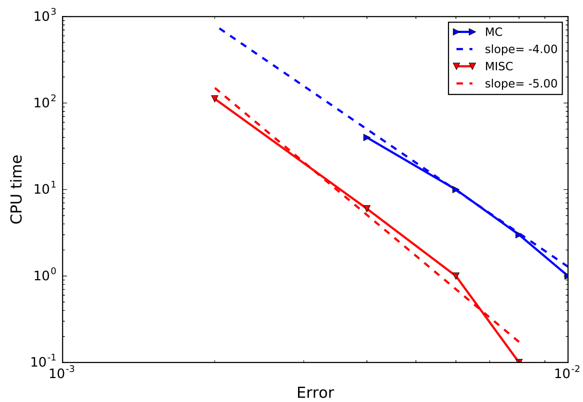



Figure 2.4: Comparison of computational work for MC and MISC methods, for the case of parameter set 3 in Table 5.1. This plot shows that to achieve a relative error below 1%, MISC outperforms MC method in terms of computational time.

### 2.2.4 Case of parameters in Set 4, in Table 5.1

In this section, we only conduct our numerical experiments for the case without Richardson extrapolation. Our numerical experiments show that MISC requires approximately 75% of the work of MC method, to achieve a total relative error of around 2% (see Figure 5.5 and Tables 5.14 and 5.13). Similar to the case of set 1 parameters illustrated in section 5.2.1, we believe that Richardson extrapolation will improve the performance of MISC method.

| Method | Steps | | | |
|---|---|---|---|---|
| | 2 | 4 | 8 | 16 |
| MISC ($\mathrm{TOL}_{\mathrm{MISC}} = 10^{-1}$) | **0.09** (0.07,0.05) | **0.07** (0.03,0.04) | **0.07** (0.02,0.05) | **0.06** (0.01,2$e$−04) |
| MISC ($\mathrm{TOL}_{\mathrm{MISC}} = 10^{-2}$) | **0.09** (0.07,5$e$−04) | **0.07** (0.03,0.04) | **0.02** (0.02,3$e$−04) | **0.02** (0.01,2$e$−04) |
| MISC ($\mathrm{TOL}_{\mathrm{MISC}} = 10^{-3}$) | **0.07** (0.07,5$e$−04) | **0.03** (0.03,4$e$−04) | **0.02** (0.02,3$e$−04) | **0.01** (0.01,2$e$−04) |
| 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) | $6 \times 10^2$ | $2 \times 10^3$ | $8 \times 10^3$ | $2 \times 10^4$ |

Table 2.13: Total relative error of MISC, without Richardson extrapolation, with different tolerances, and MC to compute the call option price for different numbers of time steps. The values between parentheses correspond to the different errors contributing to the total relative error: for MISC we report the bias and quadrature errors and for MC we report the bias and the statistical errors estimates. The number of MC samples, $M$, is chosen to satisfy (5.1). The values marked in red correspond to the values used for computational work comparison against MC method, reported in Table 5.2.

| Method | Steps | | | |
|---|---|---|---|---|
| | 2 | 4 | 8 | 16 |
| MISC ($\mathrm{TOL}_{\mathrm{MISC}} = 10^{-1}$) | 0.1 | 0.1 | 0.2 | 0.5 |
| MISC ($\mathrm{TOL}_{\mathrm{MISC}} = 10^{-2}$) | 0.1 | 0.1 | 8 | 97 |
| MISC ($\mathrm{TOL}_{\mathrm{MISC}} = 10^{-3}$) | 0.7 | 4 | 26 | 1984 |
| MC method | 0.02 | 0.15 | 1.4 | 10 |

Table 2.14: Comparison of the computational time (in seconds) of MC and MISC, to compute the call option price of rBergomi model for different numbers of time steps. The average MC CPU time is computed over 100 runs. The values marked in red correspond to the values used for computational work comparison against MC method, reported in Table 5.2.

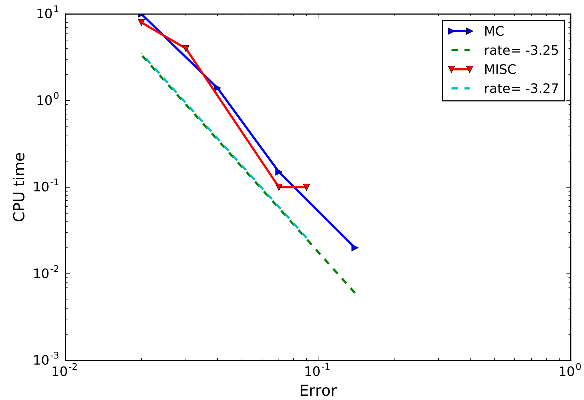



Figure 2.5: Comparison of computational work for MC and MISC methods, for the case of parameter set 4 in Table 5.1. This plot shows that to achieve a relative error around 1%, MISC and MC methods have similar performance in terms of computational time.

**Acknowledgments** C. Bayer gratefully acknowledges support from the German Research Foundation (DFG, grant BA5484/1). This work was supported by the KAUST Office of Sponsored Research (OSR) under Award No. URF/1/2584-01-01 and the Alexander von Humboldt Foundation. C. Ben Hammouda and R. Tempone are members of the KAUST SRI Center for Uncertainty Quantification in Computational Science and Engineering. The authors would like to thank Joakim Beck, Eric Joseph Hall and Erik von Schwerin for their helpful and constructive comments that greatly contributed to improving the final version of the paper.

# References Cited

[1] Peter A Acworth, Mark Broadie, and Paul Glasserman. A comparison of some Monte Carlo and quasi Monte Carlo techniques for option pricing. In *Monte Carlo and Quasi-Monte Carlo Methods 1996*, pages 1–18. Springer, 1998.

[2] Elisa Alòs, Jorge A León, and Josep Vives. On the short-time behavior of the implied volatility for jump-diffusion models with stochastic volatility. *Finance and Stochastics*, 11(4):571–589, 2007.

[3] Pierre Bajgrowicz, Olivier Scaillet, and Adrien Treccani. Jumps in high-frequency data: Spurious detections, dynamics, and news. *Management Science*, 62(8):2198–2217, 2015.

[4] Christian Bayer, Peter Friz, and Jim Gatheral. Pricing under rough volatility. *Quantitative Finance*, 16(6):887–904, 2016.

[5] Christian Bayer, Peter K Friz, Paul Gassiat, Joerg Martin, and Benjamin Stemper. A regularity structure for rough volatility. *arXiv preprint arXiv:1710.07481*, 2017.

[6] Christian Bayer, Peter K Friz, Archil Gulisashvili, Blanka Horvath, and Benjamin Stemper. Short-time near-the-money skew in rough fractional volatility models. *Quantitative Finance*, pages 1–20, 2018.

[7] CHRISTIAN BAYER, MARKUS SIEBENMORGEN, and RAUL TEMPONE. Smoothing the payoff for efficient computation of basket option pricing.

[8] Mikkel Bennedsen, Asger Lunde, and Mikko S Pakkanen. Decoupling the short-and long-term behavior of stochastic volatility. *arXiv preprint arXiv:1610.00332*, 2016.

[9] Mikkel Bennedsen, Asger Lunde, and Mikko S Pakkanen. Hybrid scheme for Brownian semistationary processes. *Finance and Stochastics*, 21(4):931–965, 2017.

[10] Lorenzo Bergomi. Smile dynamics II. *Risk*, 18:67–73, 2005.

[11] F. Biagini, Y. Hu, B. Øksendal, and T. Zhang. *Stochastic Calculus for Fractional Brownian Motion and Applications*. Probability and Its Applications. Springer London, 2008.

[12] Hans-Joachim Bungartz and Michael Griebel. Sparse grids. *Acta numerica*, 13:147–269, 2004.

[13] Russel E Caflisch, William J Morokoff, and Art B Owen. *Valuation of mortgage backed securities using Brownian bridges to reduce effective dimension*. 1997.

[14] Kim Christensen, Roel CA Oomen, and Mark Podolskij. Fact or friction: Jumps at ultra high frequency. *Journal of Financial Economics*, 114(3):576–599, 2014.

[15] Laure Coutin. An introduction to (stochastic) calculus with respect to fractional Brownian motion. In *Séminaire de Probabilités XL*, pages 3–65. Springer, 2007.

[16] Martin Forde and Hongzhong Zhang. Asymptotics for rough stochastic volatility models. *SIAM Journal on Financial Mathematics*, 8(1):114–145, 2017.

[17] Masaaki Fukasawa. Asymptotic analysis for stochastic volatility: martingale expansion. *Finance and Stochastics*, 15(4):635–654, 2011.

[18] Jim Gatheral, Thibault Jaisson, Andrew Lesniewski, and Mathieu Rosenbaum. Volatility is rough, part 2: Pricing. Workshop on Stochastic and Quantitative Finance, Imperial College London, London, 2014.

[19] Jim Gatheral, Thibault Jaisson, and Mathieu Rosenbaum. Volatility is rough. *Quantitative Finance*, 18(6):933–949, 2018.

[20] Paul Glasserman. *Monte Carlo methods in financial engineering*. Springer, New York, 2004.

[21] Abdul-Lateef Haji-Ali, Fabio Nobile, Lorenzo Tamellini, and Raul Tempone. Multi-index stochastic collocation for random pdes. *Computer Methods in Applied Mechanics and Engineering*, 306:95–122, 2016.

[22] Junichi Imai and Ken Seng Tan. Minimizing effective dimension using linear transformation. In *Monte Carlo and Quasi-Monte Carlo Methods 2002*, pages 275–292. Springer, 2004.

[23] Antoine Jacquier, Claude Martini, and Aitor Muguruza. On VIX futures in the rough Bergomi model. *Quantitative Finance*, 18(1):45–61, 2018.

[24] Antoine Jacquier, Mikko S Pakkanen, and Henry Stone. Pathwise large deviations for the rough Bergomi model. *arXiv preprint arXiv:1706.05291*, 2017.

[25] Benoit B Mandelbrot and John W Van Ness. Fractional Brownian motions, fractional noises and applications. *SIAM review*, 10(4):422–437, 1968.

[26] Domenico Marinucci and Peter M Robinson. Alternative forms of fractional Brownian motion. *Journal of statistical planning and inference*, 80(1-2):111–122, 1999.

[27] Ryan McCrickerd and Mikko S Pakkanen. Turbocharging Monte Carlo pricing for the rough Bergomi model. *Quantitative Finance*, pages 1–10, 2018.

[28] William J Morokoff and Russel E Caflisch. Quasi-random sequences and their discrepancies. *SIAM Journal on Scientific Computing*, 15(6):1251–1279, 1994.

[29] Bradley Moskowitz and Russel E Caflisch. Smoothness and dimension reduction in quasi-Monte Carlo methods. *Mathematical and Computer Modelling*, 23(8):37–54, 1996.

[30] Andreas Neuenkirch and Taras Shalaiko. The order barrier for strong approximation of rough volatility models. *arXiv preprint arXiv:1606.03854*, 2016.

[31] Jean Picard. Representation formulae for the fractional Brownian motion. In *Séminaire de Probabilités XLIII*, pages 3–70. Springer, 2011.

[32] Marc Romano and Nizar Touzi. Contingent claims and market completeness in a stochastic volatility model. *Mathematical Finance*, 7(4):399–412, 1997.

[33] Denis Talay and Luciano Tubaro. Expansion of the global error for numerical schemes solving stochastic differential equations. *Stochastic analysis and applications*, 8(4):483–509, 1990.