Efficient option pricing for Rough Bergomi model

1 The goal and outline of the project

The main goal of the project is to design a fast option pricer, based on multi-index stochastic collocation (MISC), for options whose dynamics follow rBergomi model.

2 The rBergomi model

We use the rBergomi model for the price process S_t as defined in [1], which is defined by

$$dS_t = \sqrt{v_t} S_t dZ_t, \tag{1}$$

$$v_t = \xi_0(t) \exp\left(\eta \tilde{W}_t - \frac{1}{2}\eta^2 t^{2H}\right), \tag{2}$$

where for $0 < H < 1$, we have $\tilde{W}^H$ is a certain Volterra process, defined by

$$\tilde{W}_t^H = \int_0^t K(t,s) dW_s^1, \quad K(t,s) = \sqrt{2H}(t-s)^{H-1/2}. \tag{3}$$

W_1, Z denote two *correlated* standard Brownian motions with correlation ρ, so that

$$Z := \rho W_1 + \overline{\rho} W_2 \equiv \rho W_1 + \sqrt{1-\rho^2} W_2 \tag{4}$$

Therefore, Eq 1 can be written as

$$S_t = S(0) \exp\left(\int_0^t \sqrt{v(s)} dZ(s) - \frac{1}{2}\int_0^t v(s) ds\right)$$
$$v(u) = \xi_0(u) \exp\left(\eta \tilde{W}_u^H - \frac{\eta^2}{2} u^H\right). \tag{5}$$

We refer to v_u as the variance process, where $\xi_0(u) = \mathrm{E}[v_u] \in \mathcal{F}_0$ a.s. the forward variance curve.

$\tilde{W}^H$ is a centered, locally $(H-\epsilon)$- Hølder continuous, Gaussian process with $\mathrm{var}\, \tilde{W}_t^H = t^{2H}$.

In [1], the approach consists in sampling the Gaussian process Z and $\tilde{W}^H$ on a discrete time grid using exact simulation and then approximating S and v using Euler discretization.

Assuming $S_0 = 1$, and using the conditioning argument on the σ-algebra generated by W_1 (argument first used by [4] in the context of Markovian SV models), we can show that the call price is given by

$$
\begin{aligned}
C_{RB}\left(T,K\right) &= E\left[(S_T-K)^+\right] \\
&= \mathrm{E}\left[\mathrm{E}\left[(S_T-K)^+ \mid \sigma(W^1(t), t\leq T)\right]\right] \\
&= E\left[C_{BS}\left(S_0=\exp\left(\rho\int_0^T \sqrt{v_t}dW_t^1-\frac{1}{2}\rho^2\int_0^T v_t dt\right),\ K=K,\ T=1,\ \sigma^2=(1-\rho^2)\int_0^T v_t dt\right)\right],
\end{aligned} \tag{6}
$$

where C_{BS} denotes the Black-Scholes price.

In fact, if we use the orthogonal decomposition of S_t into S_t^1 and S_t^2, where

$$
S_t^1 := \mathcal{E}\{\rho\int_0^t \sqrt{v_s}dW_s^1\},\ S_t^2 := \mathcal{E}\{\sqrt{1-\rho^2}\int_0^t \sqrt{v_s}dW_s^2\}, \tag{7}
$$

where $\mathcal{E}()$ denotes the stochastic exponential, then, we obtain

$$
\log S_t \mid \mathcal{F}_t^1 \sim \mathcal{N}\left(\log S_t^1-\frac{1}{2}(1-\rho^2)\int_0^t v_s ds, (1-\rho^2)\int_0^t v_s ds\right), \tag{8}
$$

where $\mathcal{F}_t^1=\sigma\{W_s^1 : s\leq t\}$. Therefore, we obtain (6).

The main challenge is the computation of $S=\int_0^T \sqrt{v_t}dW_t^1$ and $V=\int_0^T v_t dt$. As was mentioned in [2], we may try to avoid any sampling related to W^2 by a brute-force approach that consists in simulating a scalar Brownian motion W^1, followed by computing $\tilde{W}^H=\int K dW^1$ by Itô/Riemann Stieltjes approximations of (S,V). However, this is not advisable given the singularity of the Volterra kernel K. Therefore, one needs to simulate the two-dimensional Gaussian process $(W_t^1, \tilde{W}_t^H : 0\leq t\leq T)$.

3 Numerical examples

3.1 Numerical tests description

For our numerical tests, we coupled the C++ implementation used in [1] with the MISC library, and for comparison purposes we compare our results to the python code used in [3] with MC method, for $M=10^7$ paths. We note that both used methods have similar complexity for constructing the spot prices which is of order $\mathcal{O}\left(N\log(N)\right)$. Also, we use $S_0=1$, so the options will be prices in terms of the moneyness K, where K is the strike price.

We start our numerical tests by comparing the values of call options prices for different values of strikes $K=\{1.2,1,0.8,e^{-4}\}$ for $H=0.43$ (we note that this value of H is not realistic but it is a good starting point since this case the fractional Brownian motion (fBm) becomes simply Brownian motion) and $K=\{1.2,1,0.8\}$ for $H=0.07$ in Section 3.2. The used parameters are $H=\{0.43,0.07\}$, $\eta=1.9$, $\rho=-0.9$, $T=1$ and $\xi_0=0.235^2$. The values between parentheses in the tables are the standard errors for MC method. The results were reported for number of time steps $N\in\{2,4,8,16\}$. We note that we may need higher number of time steps N to achieve better accuracy but since we believe that we have order one convergence with respect to Δt (which is the limiting factor in the convergence), it is no so bad to test for few time steps. We are later going to use the Richardson extrapolation, which we hope to improve the convergence to quadratic (see figure 1 which shows that the weak error does not seem too bad already at this number of time steps).

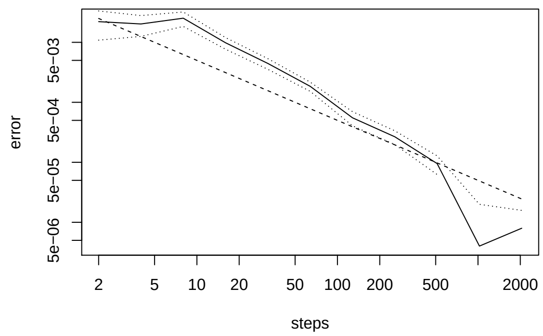


Figure 1: The weak error of rBergomi model

We note that for some cases, the convergence becomes extremely slow (either the bias stagnates at one value or it keeps increasing or decreasing without reaching the prescribed tolerance) specially when we are close to at-the money option (K close to 1), we do not put values for those cases. We also remark that we have better agreement between the results of MISC coupled with the C++ code and the MC method using the python code form [3], for small values of moneyness.

In Section 3.4, we show the convergence plots given by MISC library for the cases of $K = \{e^{-4}, 1.2\}$ and $H = 0.43$. We emphasize that in each case we have $2N$ stochastic parameters, the first N correspond to W_u^1, while the last N stochastic parameters correspond to W_u^2. Table 1 summarizes the obtained complexity rates for different number of time steps (for more details, see Section 3.4. For other cases, we could not obtain the convergence rates as we explained in the paragraph above. The table 1 supports our observation that for values of K close to at the money ($K = 1$), we observe a bad convergence behavior of MISC. In fact, the rates are much worse for $K = 1.2$ than $K = e^{-4}$. If we look at Section 3.4, we may see a potential explanation for this different complexity behavior with respect to K. In fact, it is clear by comparing the plots of the convergence rate of mixed differences per level for different value of K and the same number of time steps, we may notice that the convergence of mixed differences is much faster for the case of $K = e^{-4}$ than $K = 1.2$. This observation is supported by external tests (See Section 3.3), where we compare the convergence of first and second differences for $K = e^{-4}, = 1.2$, for 8 and 16 time steps for $H = 0.43, 0.07$. We still do not have a clear explanation why the rates are sensitive to the values of K.

Method \Steps	2	4	8	16
without Richardson extrapolation ($K = e^{-4}$)	$-3/20$	$-8/25$	$-4/5$	$-13/10$
without Richardson extrapolation ($K = 1.2$)	$-13/20$	$-9/10$	$-5/4$	$-$

Table 1: Complexity rates for different number of time steps for $H = 0.43$ and $K = \{e^{-4}, 1.2\}$

3.2 Comparing call options prices

Case $H = 0.43$

Method \Steps	2	4	8	16
MISC ($TOl = 10^{-1}$)	0.1057	0.0988	0.0836	0.0594
MISC ($TOl = 10^{-2}$)	0.1113	0.0939	0.0820	–
MISC ($TOl = 10^{-3}$)	0.1081	0.0918	0.0822	–
MISC ($TOl = 10^{-4}$)	0.1080	0.0921	–	–
MC method ($M = 10^7$)	0.0840 (4.14e−05)	0.0782 (3.23e−05)	0.0748 (2.84e−05)	0.0729 (8.36e−06)

Table 2: Call option price of the different methods for different number of time steps. Case $K = 1$

Method \Steps	2	4	8	16
MISC ($TOl = 10^{-1}$)	0.2230	0.2122	0.2038	0.1993
MISC ($TOl = 10^{-2}$)	0.2373	0.2355	0.2274	–
MISC ($TOl = 10^{-3}$)	0.2403	0.2331	–	–
MISC ($TOl = 10^{-4}$)	0.2405	0.2333	–	–
MC method ($M = 10^7$)	0.2228 (5.81e−05)	0.2237 (4.99e−05)	0.2238 (4.63e−05)	0.2236 (1.41e−05)

Table 3: Call option price of the different methods for different number of time steps. Case $K = 0.8$

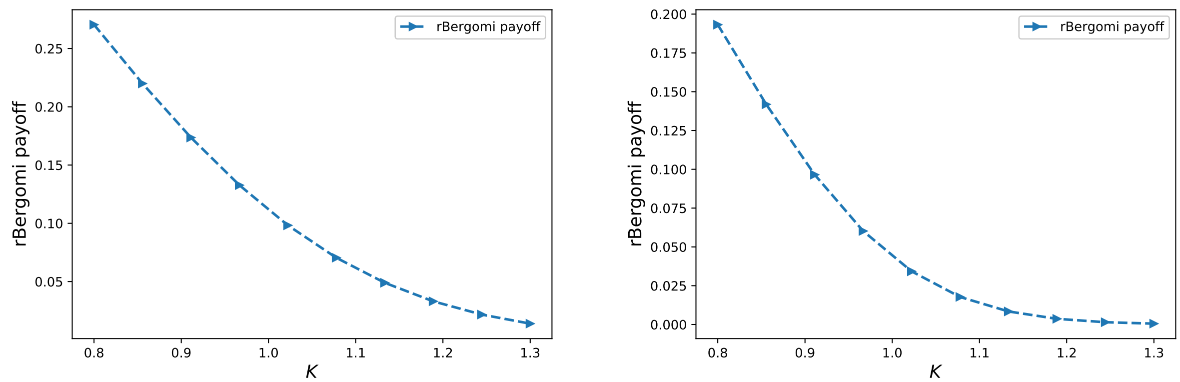
Method \Steps	2	4	8	16
MISC ($TOl = 10^{-1}$)	0.0271	0.0119	0.0044	0.0017
MISC ($TOl = 10^{-2}$)	0.0352	0.0133	0.0048	0.0017
MISC ($TOl = 10^{-3}$)	0.0345	0.0183	0.0093	0.0054
MISC ($TOl = 10^{-4}$)	0.0347	0.0181	0.0096	–
MC method ($M = 10^7$)	0.0184 (2.41e−05)	0.0105 (1.28e−05)	0.0063 (8.11e−06)	0.0043 (1.89e−06)

Table 4: Call option price of the different methods for different number of time steps. Case $K = 1.2$

Method \Steps	2	4	8	16
MISC ($TOl = 10^{-1}$)	0.9699	0.9730	0.9745	0.9752
MISC ($TOl = 10^{-2}$)	0.9699	0.9730	0.9789	0.9809
MISC ($TOl = 10^{-3}$)	0.9699	0.9818	0.9818	0.9819
MISC ($TOl = 10^{-4}$)	0.9816	0.9817	0.9817	–
MC method ($M = 10^7$)	0.9816	0.9817	0.9817	0.9817

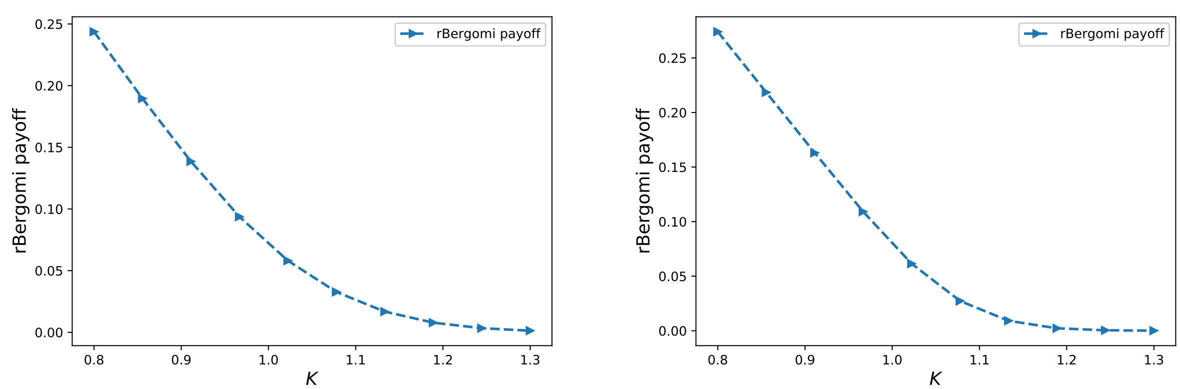
Table 5: Call option price of the different methods for different number of time steps. Case $K = e^{-4}$

Since we observed that it is difficult for MISC to converge for $N = 4, 8, 16$ for small tolerances $TOL = 10^{-3}, TOL = 10^{-4}$, we plot the final payoff we are using to check if we have a problem of regularity (See figures (2, 3) for $H = 0.43$ and figures (4, 5) for $H = 0.07$). I think the problem we observed is not related to regularity.



(a) (b)

Figure 2: Black Scholes payoff for rBergomi model as a function of moneyness for $H = 0.43$ a) 2 time steps, b) 4 time steps.



(a) (b)

Figure 3: Black Scholes payoff for rBergomi model as a function of moneyness for $H = 0.43$ a) 8 time steps, b) 16 time steps.

Case $H = 0.07$

Method \Steps	2	4	8	16
MISC ($TOl = 10^{-1}$)	0.1064	0.0899	0.0733	0.0956
MISC ($TOl = 10^{-2}$)	0.1226	0.1022	0.0933	–
MISC ($TOl = 10^{-3}$)	0.1215	0.1025	–	–
MISC ($TOl = 10^{-4}$)	0.1218	0.0924	–	–
MC method ($M = 10^7$)	0.0824 (1.01e−04)	0.0783 (4.63e−05)	0.0776 (3.95e−05)	0.0779 (3.64e−05)

Table 6: Call option price of the different methods for different number of time steps. Case $K = 1$

Method \Steps	2	4	8	16
MISC ($TOl = 10^{-1}$)	0.2156	0.2002	0.2002	0.1910
MISC ($TOl = 10^{-2}$)	0.2474	0.2378	0.2378	–
MISC ($TOl = 10^{-3}$)	0.2505	0.2377	–	–
MISC ($TOl = 10^{-4}$)	0.25	0.231	–	–
MC method ($M = 10^7$)	0.2199 (1.1e−04)	0.2212 (6.03e−05)	0.2225 (5.48e−05)	0.2233 (5.27e−05)

Table 7: Call option price of the different methods for different number of time steps. Case $K = 0.8$

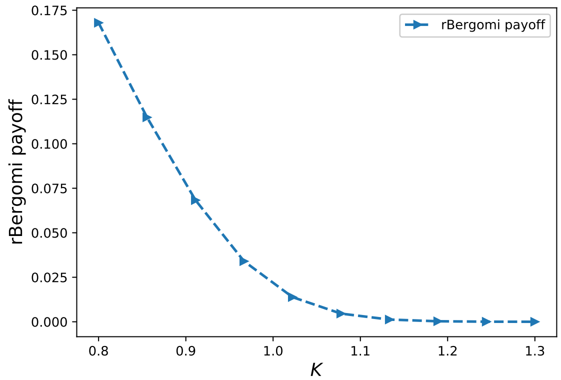
Method \Steps	2	4	8	16
MISC ($TOl = 10^{-1}$)	0.0288	0.0102	0.0025	0.0005
MISC ($TOl = 10^{-2}$)	0.0501	0.0161	0.0025	0.0005
MISC ($TOl = 10^{-3}$)	0.0515	0.0335	–	–
MISC ($TOl = 10^{-4}$)	0.0525	–	–	–
MC method ($M = 10^7$)	0.0207 (9.57e−05)	0.0165 (3.32e−05)	0.0144 (2.43e−05)	0.0130 (2e−05)

Table 8: Call option price of the different methods for different number of time steps. Case $K = 1.2$

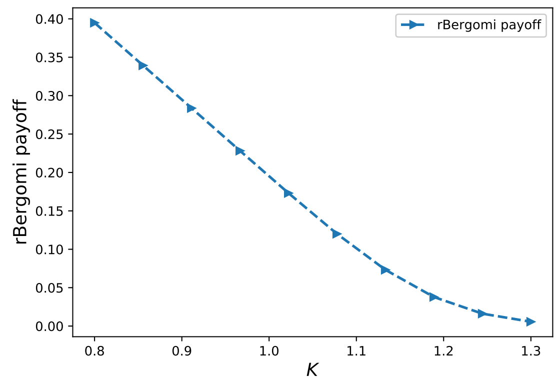
3.3 Investigating mixed differences

Case $H = 0.43$

$N = 8$

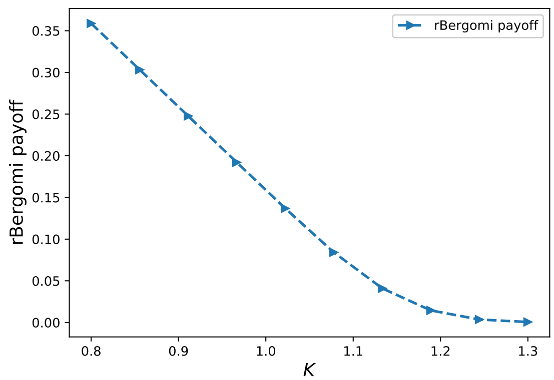


(a)

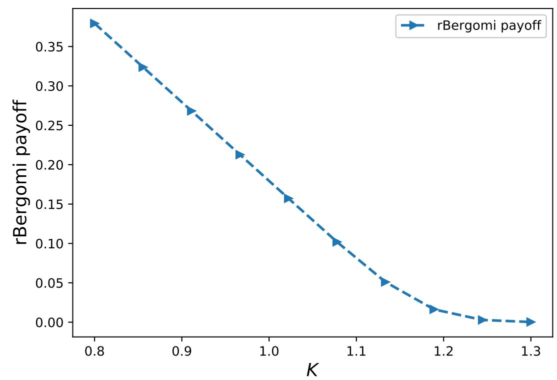


(b)

Figure 4: Black Scholes payoff for rBergomi model as a function of moneyness a) 2 time steps, b) 4 time steps.



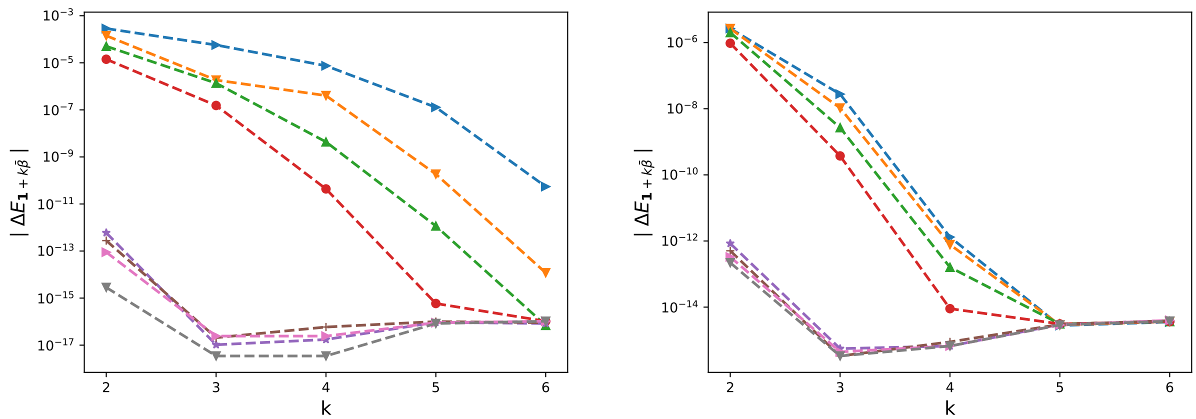
(a)



(b)

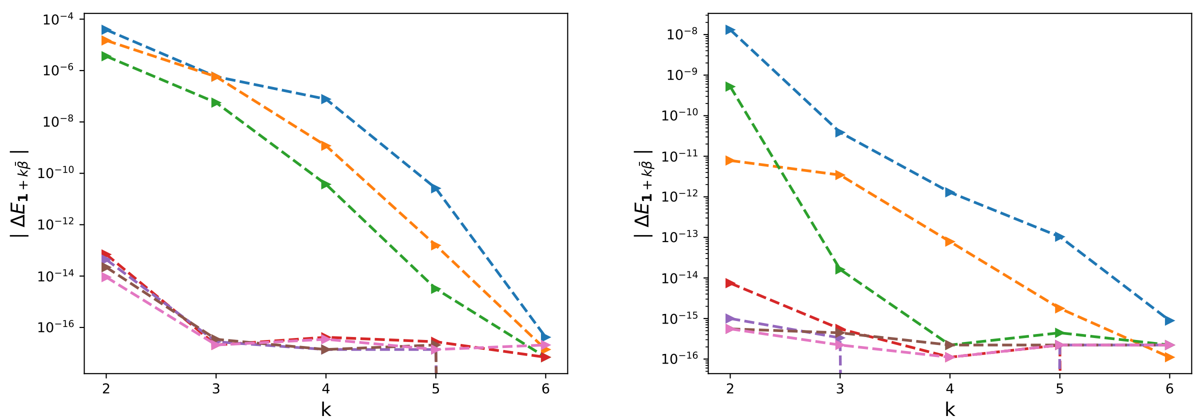
Figure 5: Black Scholes payoff for rBergomi model as a function of moneyness a) 8 time steps, b) 16 time steps.

$N = 16$



(a) (b)

Figure 6: The rate of convergence of first order differences $|\Delta E_\beta|$ ($\beta = \mathbf{1} + k\overline{\beta}$)): a) $K = 1$ b) $K = \exp(-4)$.

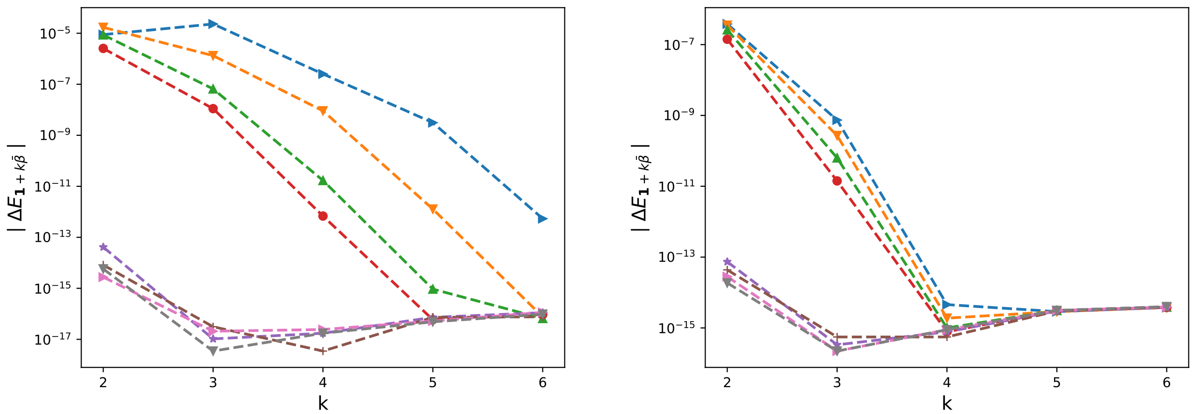


(a) (b)

Figure 7: The rate of convergence of second order differences $|\Delta E_\beta|$ ($\beta = \mathbf{1} + k\overline{\beta}$)): a) $K = 1$ b) $K = \exp(-4)$.

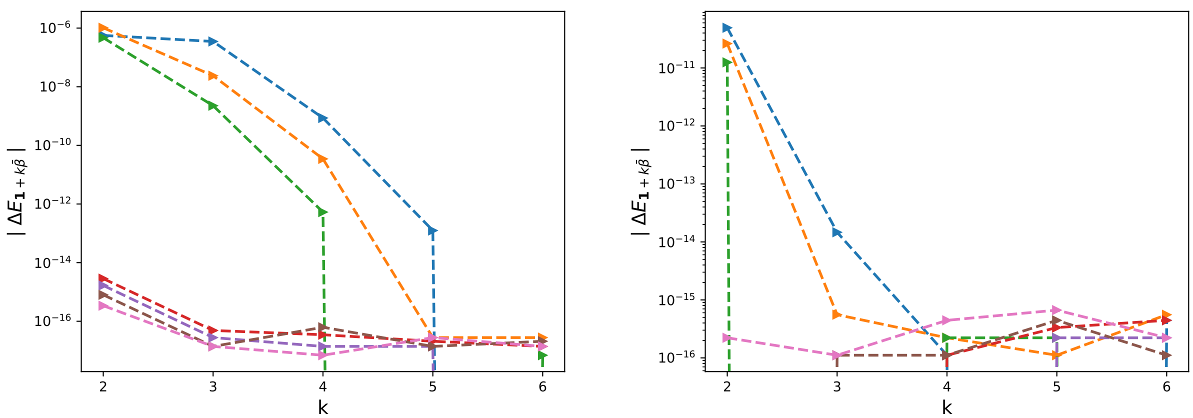
Case $H = 0.07$

$N = 8$



(a) (b)

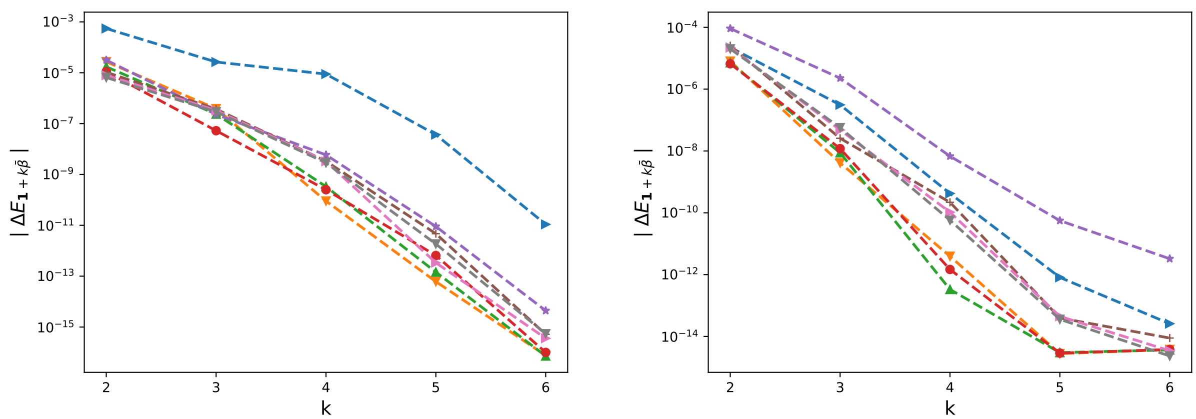
Figure 8: The rate of convergence of first order differences $|\Delta E_\beta|$ ($\beta = \mathbf{1} + k\overline{\beta}$)): a) $K = 1$ b) $K = \exp(-4)$.



(a) (b)

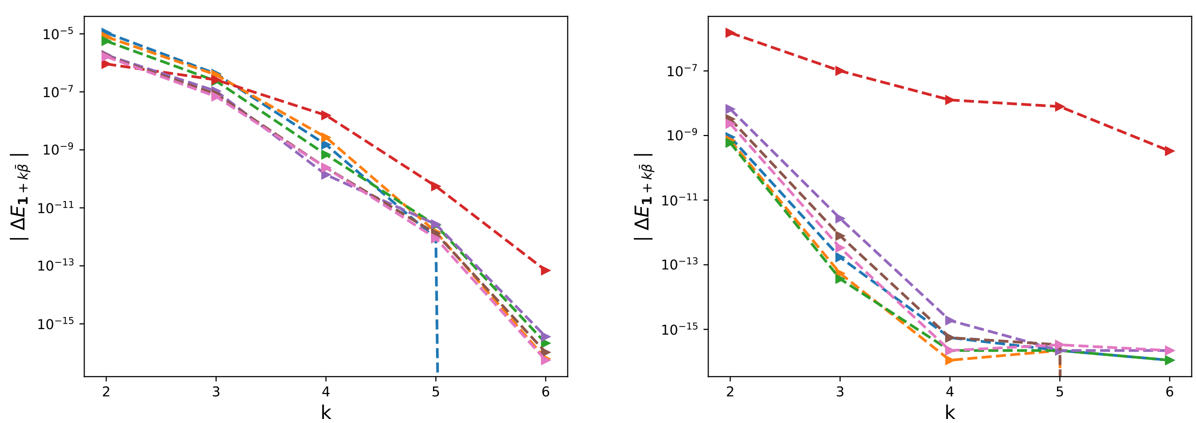
Figure 9: The rate of convergence of second order differences $|\Delta E_\beta|$ ($\beta = \mathbf{1} + k\overline{\beta}$)): a) $K = 1$ b) $K = \exp(-4)$.

$N = 16$



(a) (b)

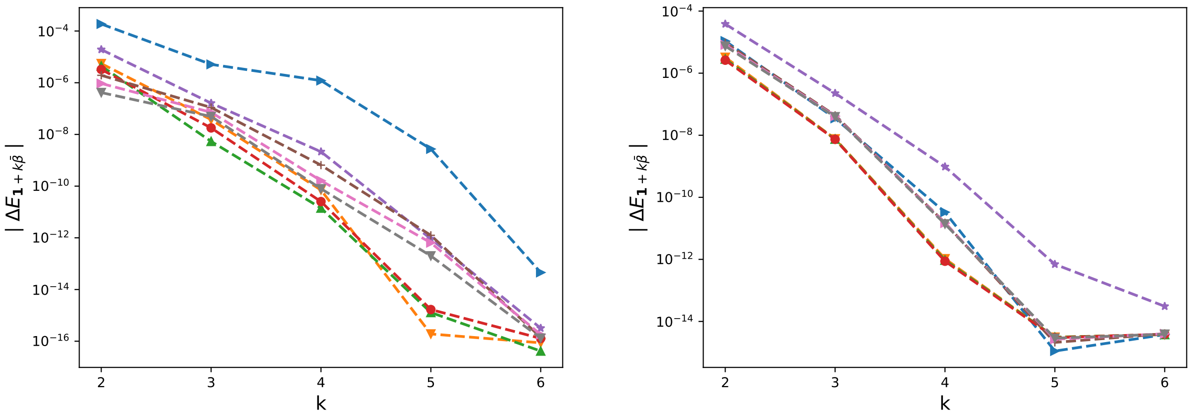
Figure 10: The rate of convergence of first order differences $|\Delta E_\beta|$ ($\beta = \mathbf{1} + k\overline{\beta}$)): a) $K = 1$ b) $K = \exp(-4)$.



(a) (b)

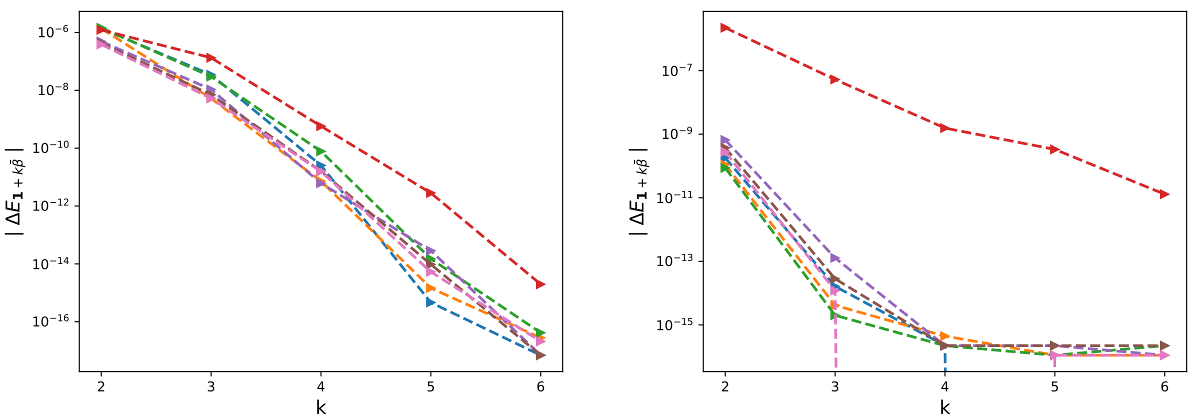
Figure 11: The rate of convergence of second order differences $|\Delta E_\beta|$ ($\beta = \mathbf{1} + k\overline{\beta}$)): a) $K = 1$ b) $K = \exp(-4)$.

3.4 Convergence plots using MISC ($H = 0.43$)



(a) (b)

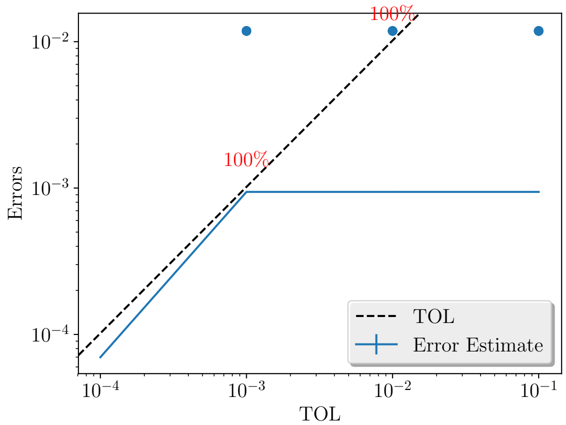
Figure 12: The rate of convergence of first order differences $|\Delta E_\beta|$ ($\beta = \mathbf{1} + k\overline{\beta}$)): a) $K = 1$ b) $K = \exp(-4)$.



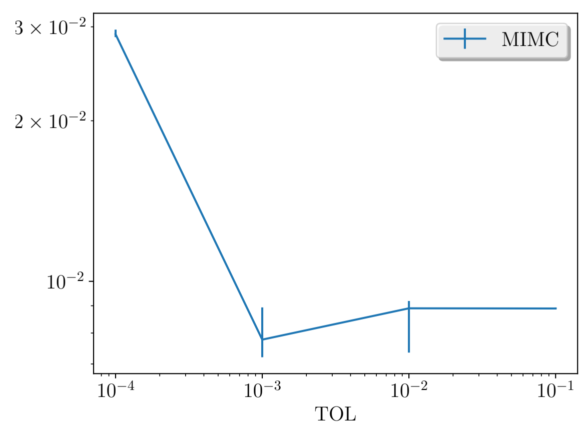
(a) (b)

Figure 13: The rate of convergence of second order differences $|\Delta E_\beta|$ ($\beta = \mathbf{1} + k\overline{\beta}$)): a) $K = 1$ b) $K = \exp(-4)$.

Case of 2 time steps, $K = e^{-4}$

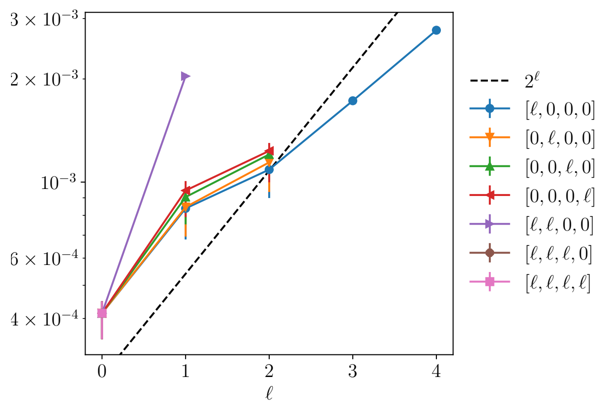


(a) Error estimate

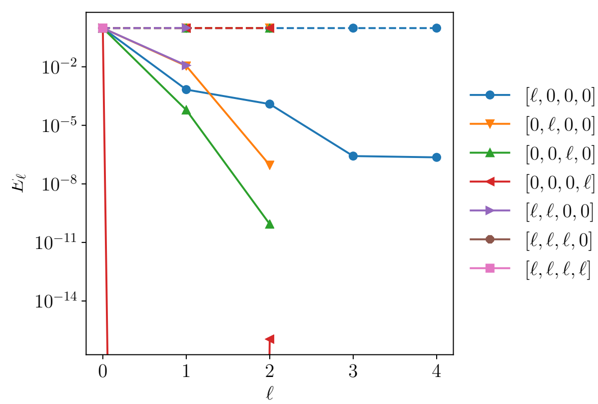


(b) Average running time as a function of TOL

Figure 14: Convergence and complexity results for the call payoff with rBergomi model.



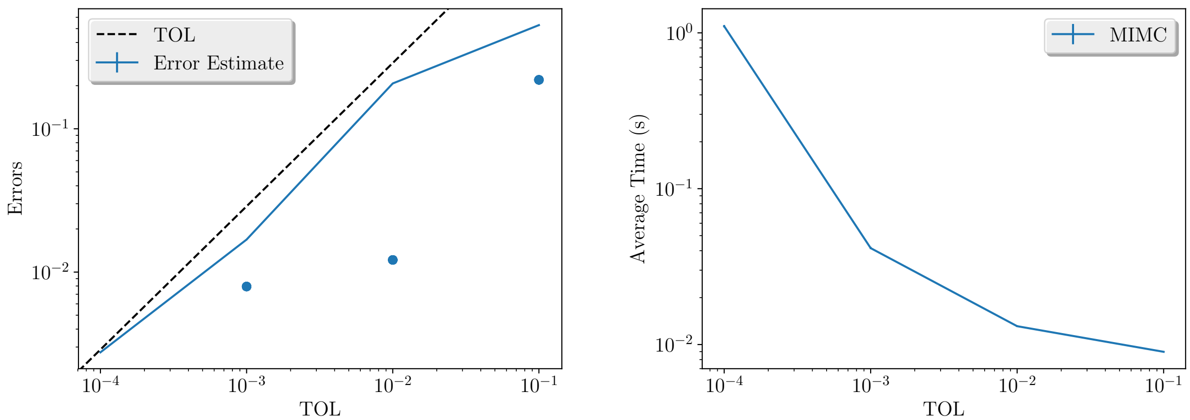
(a) Average Computational time per level



(b) The convergence rate of mixed differences per level

Figure 15: Convergence and work rates for discretization levels the call payoff with rBergomi model.

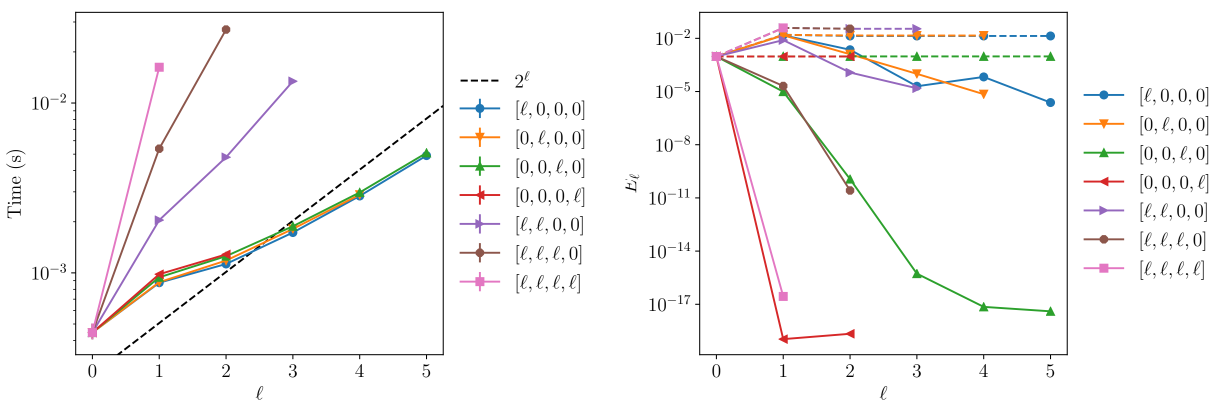
Case of 2 time steps, $K = 1.2$



(a) Error estimate

(b) Average running time as a function of TOL

Figure 16: Convergence and complexity results for the call payoff with rBergomi model.

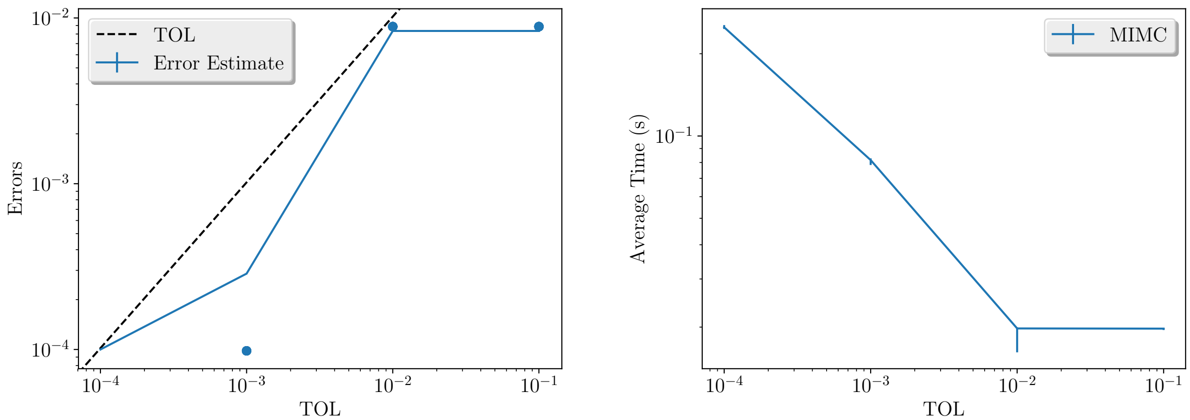


(a) Average Computational time per level

(b) The convergence rate of mixed differences per level

Figure 17: Convergence and work rates for discretization levels the call payoff with rBergomi model.

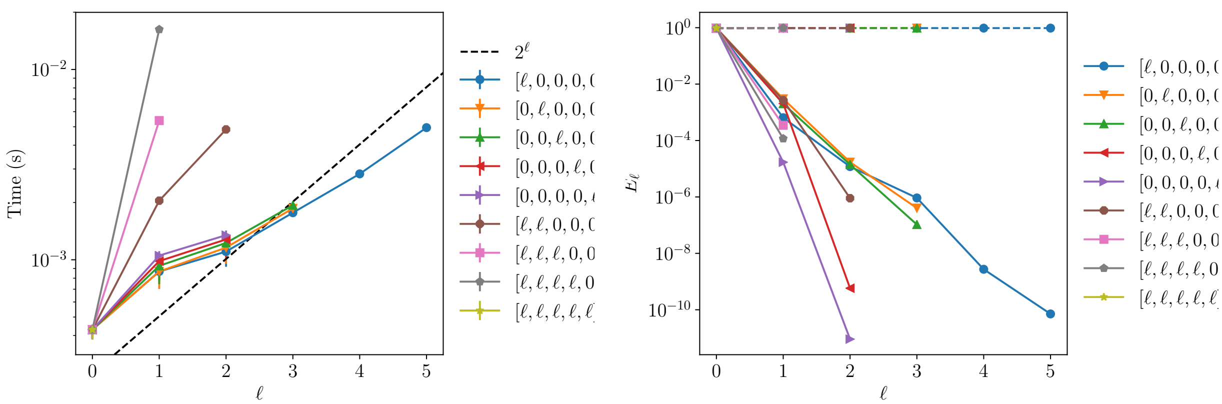
Case of 4 time steps, $K = e^{-4}$



(a) Error estimate

(b) Average running time as a function of TOL

Figure 18: Convergence and complexity results for the call payoff with rBergomi model.

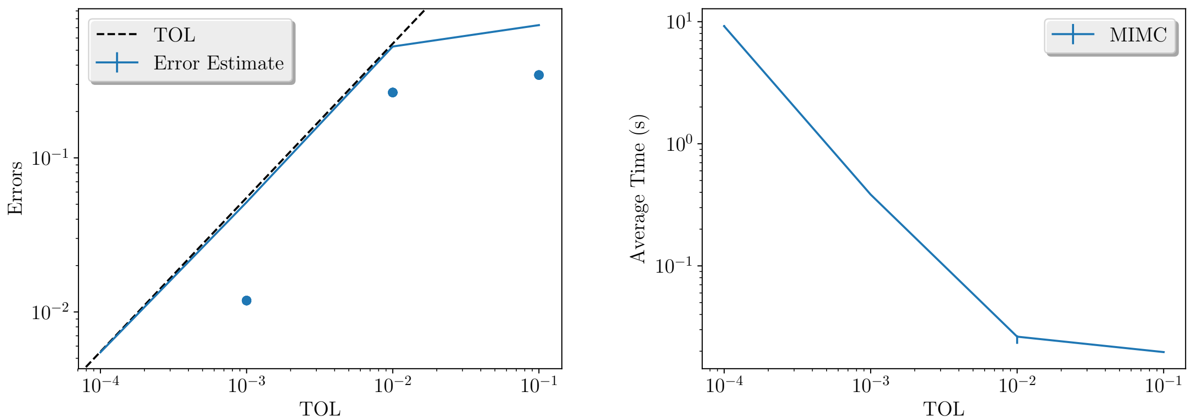


(a) Average Computational time per level

(b) The convergence rate of mixed differences per level

Figure 19: Convergence and work rates for discretization levels the call payoff with rBergomi model.

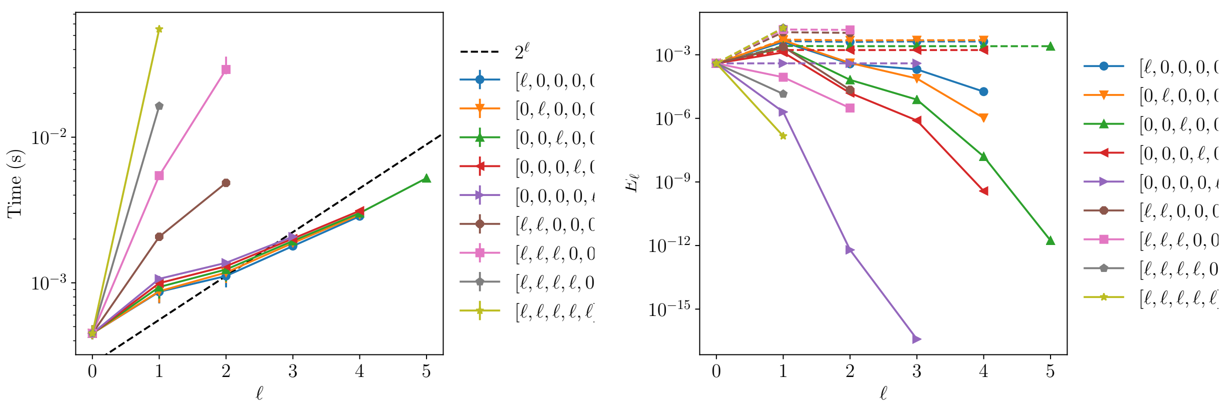
Case of 4 time steps, $K = 1.2$



(a) Error estimate

(b) Average running time as a function of TOL

Figure 20: Convergence and complexity results for the call payoff with rBergomi model.

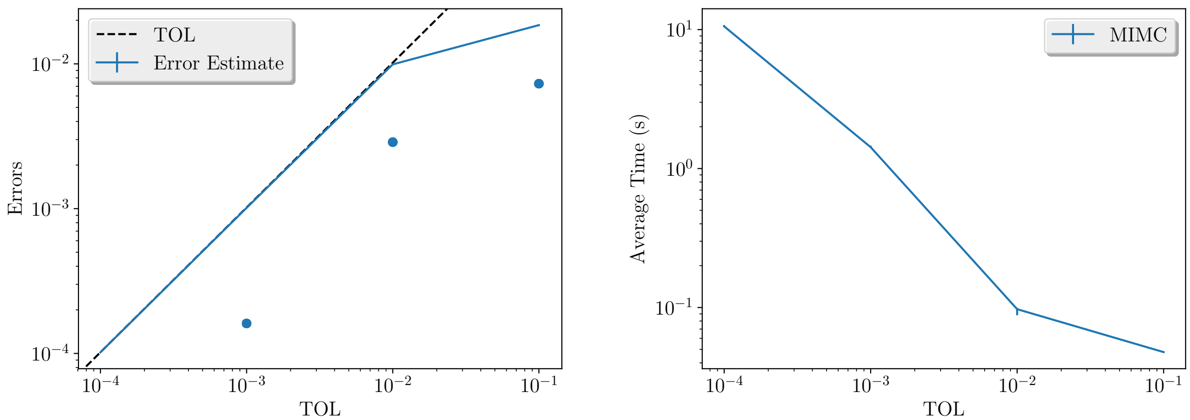


(a) Average Computational time per level

(b) The convergence rate of mixed differences per level

Figure 21: Convergence and work rates for discretization levels the call payoff with rBergomi model.

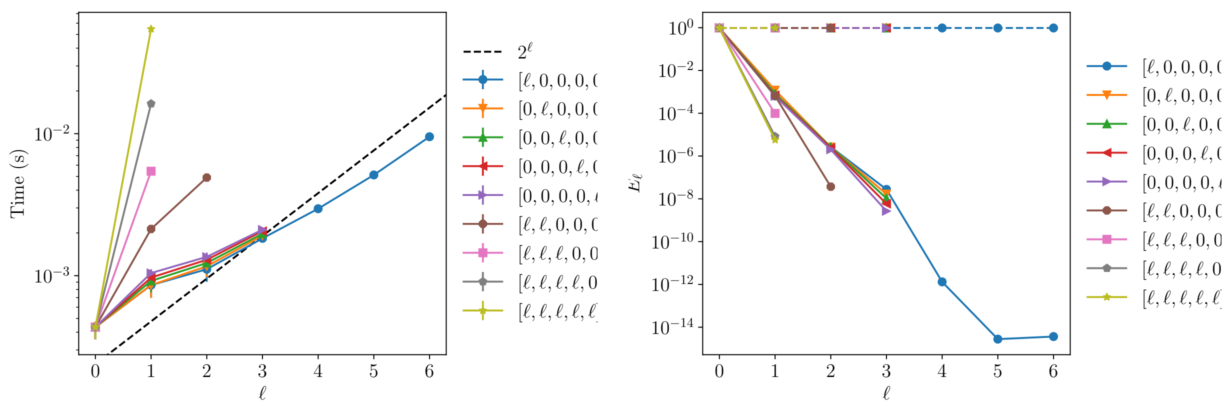
Case of 8 time steps, $K = e^{-4}$



(a) Error estimate

(b) Average running time as a function of TOL

Figure 22: Convergence and complexity results for the call payoff with rBergomi model.

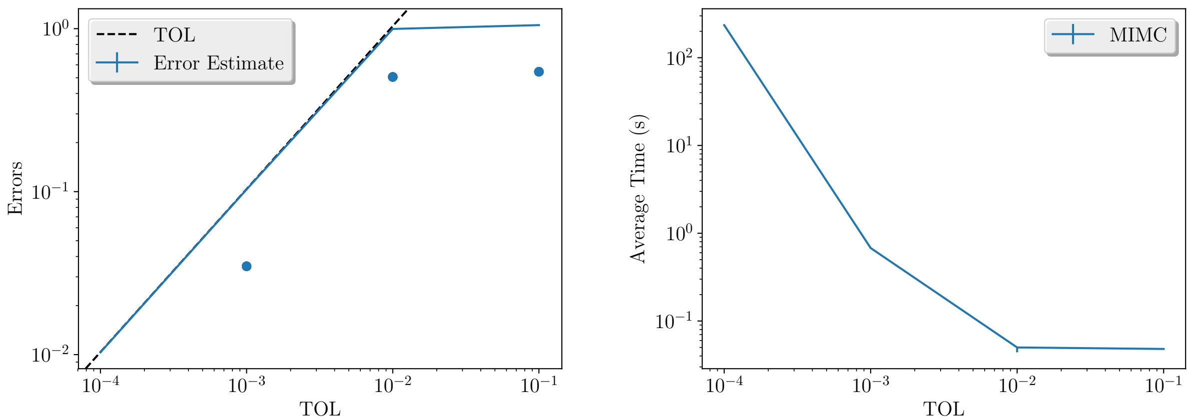


(a) Average Computational time per level

(b) The convergence rate of mixed differences per level

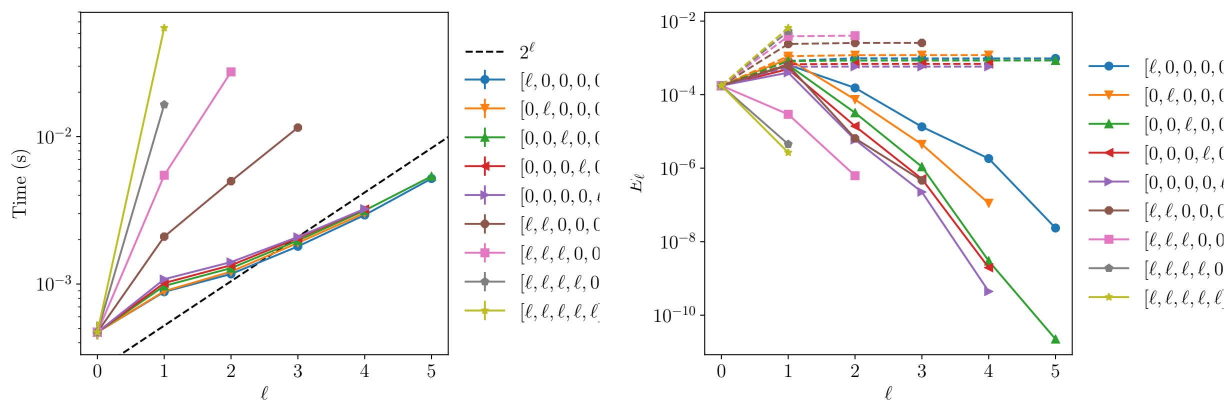
Figure 23: Convergence and work rates for discretization levels the call payoff with rBergomi model.

Case of 8 **time steps,** $K = 1.2$



(a) Error estimate (b) Average running time as a function of TOL

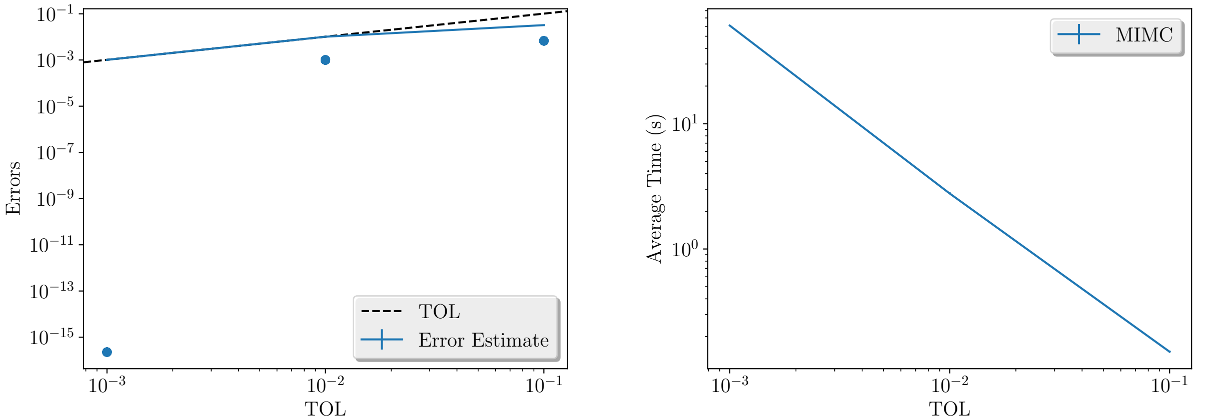
Figure 24: Convergence and complexity results for the call payoff with rBergomi model.



(a) Average Computational time per level (b) The convergence rate of mixed differences per level

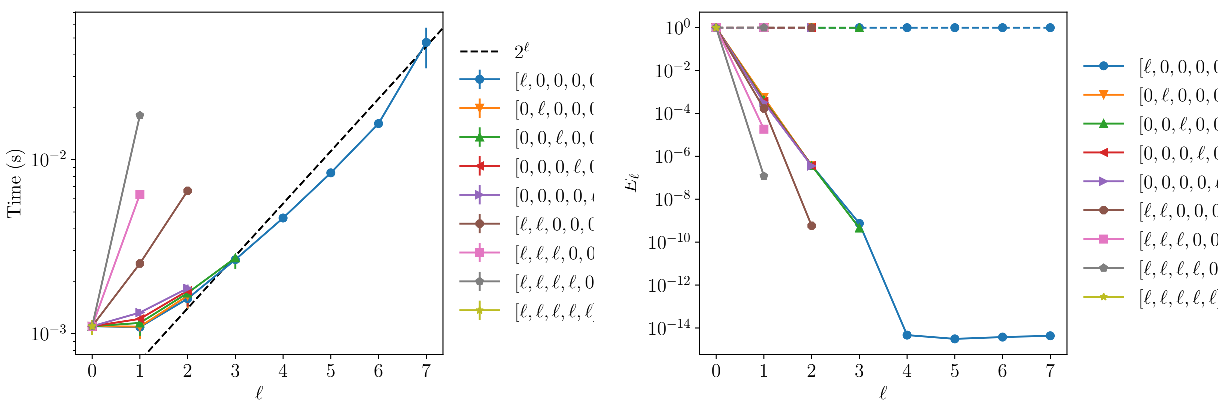
Figure 25: Convergence and work rates for discretization levels the call payoff with rBergomi model.

Case of 16 time steps, $K = e^{-4}$



(a) Error estimate (b) Average running time as a function of TOL

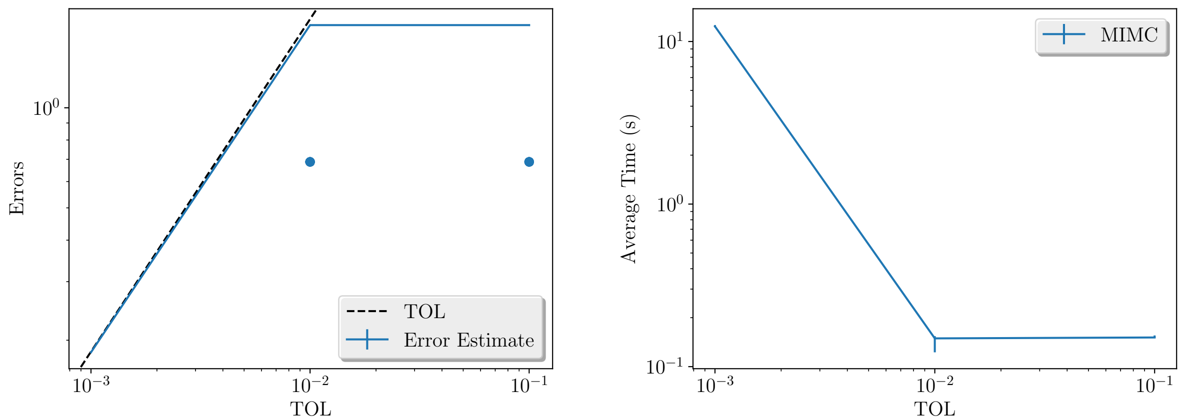
Figure 26: Convergence and complexity results for the call payoff with rBergomi model.



(a) Average Computational time per level (b) The convergence rate of mixed differences per level

Figure 27: Convergence and work rates for discretization levels the call payoff with rBergomi model.

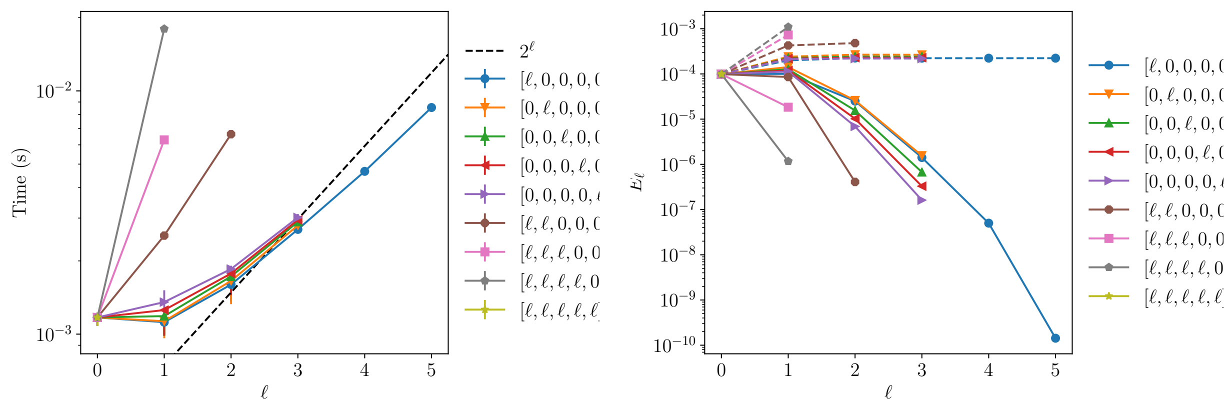
Case of 16 time steps, $K = 1.2$



(a) Error estimate

(b) Average running time as a function of TOL

Figure 28: Convergence and complexity results for the call payoff with rBergomi model.



(a) Average Computational time per level

(b) The convergence rate of mixed differences per level

Figure 29: Convergence and work rates for discretization levels the call payoff with rBergomi model.

References Cited

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

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

[3] Ryan McCrickerd and Mikko S Pakkanen. Turbocharging monte carlo pricing for the rough bergomi model. *arXiv preprint arXiv:1708.02563*, 2017.

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