Adaptive sparse grids and quasi Monte Carlo for option pricing under the rough Bergomi model

Chiheb Ben Hammouda

KAUST
King Abdullah University of Science and Technology

Christian Bayer

WIAS

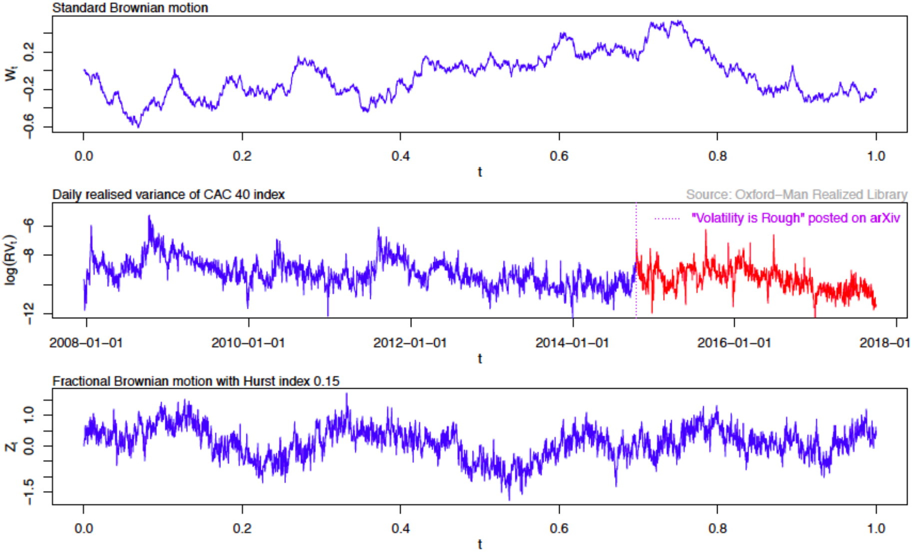
Raúl Tempone

RWTH AACHEN UNIVERSITY

3rd International Conference on Computational Finance (ICCF2019), A Coruña
8-12 July, 2019

Outline

Rough volatility [Gatheral et al., 2018]



The rough Bergomi model [Bayer et al., 2016]

This model, under a pricing measure, is given by

$$\begin{cases} dS_t &= \sqrt{v_t} S_t dZ_t, \\ v_t &= \xi_0(t) \exp\left(\eta \widetilde{W}_t^H - \frac{1}{2}\eta^2 t^{2H}\right), \\ Z_t &:= \rho W_t^1 + \bar{\rho} W_t^{\perp} \equiv \rho W^1 + \sqrt{1-\rho^2} W^{\perp}, \end{cases} \tag{1}$$

- $(W^1, W^{\perp})$: two independent standard Brownian motions
- $\widetilde{W}^H$ is Riemann-Liouville process, defined by

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

- $H \in (0, 1/2]$ ($H = 1/2$ for Brownian motion): controls the roughness of paths, $\rho \in [-1, 1]$ and $\eta > 0$.
- $t \mapsto \xi_0(t)$: forward variance curve, known at time 0.

Model challenges

- **Numerically:**
  - The model is non-affine and non-Markovian $\Rightarrow$ Standard numerical methods (PDEs, characteristic functions) seem inapplicable.
  - The only prevalent pricing method for mere vanilla options is Monte Carlo (MC) [Bayer et al., 2016, Bayer et al., 2017, McCrickerd and Pakkanen, 2018]: still a time consuming task.
  - Discretization methods have poor behavior of the strong error, that is the convergence rate is of order of $H \in [0, 1/2]$ [Neuenkirch and Shalaiko, 2016] $\Rightarrow$ Variance reduction methods, such as multilevel Monte Carlo (MLMC), are inefficient for very small values of H.
- **Theoretically:**
  - No proper weak error analysis done in the rough volatility context.

Option pricing challenges

The integration problem is challenging

- Issue 1: Time-discretization of the rough Bergomi process (large N (number of time steps)) $\Rightarrow$ S takes values in a high-dimensional space $\Rightarrow$ Curse of dimensionality when using numerical integration methods.
- Issue 2: The payoff function g is typically not smooth $\Rightarrow$ low regularity$\Rightarrow$ slow convergence of deterministic quadrature methods.

⚠ Curse of dimensionality: An integration error of order ε requires M function evaluations

$$M \geq c_{\varepsilon} \bar{d}^{-c \log \varepsilon},$$

unclear, I would just keep it in words, suggesting an exponential growth of the work in terms of the dimension of the input

where $\bar{d}$ depends on d and N.

Methodology

We design a hierarchical efficient pricing method based on

1. Analytic smoothing to uncover available regularity (inspired by [Romano and Touzi, 1997] in the context of stochastic volatility models). also on Bayer et al ;)
2. Approximating the option price using deterministic quadrature methods
   - **Adaptive sparse grids quadrature (ASGQ).**
   - **Quasi Monte Carlo (QMC).**

representations, exploiting the weak error 1

3. Coupling our methods with hierarchical transformations ⇒ Reduce the dimension of the problem.
   - **Brownian bridges** as a path generation method.
   - **Richardson Extrapolation** ⇒ Faster convergence of the weak error ⇒ ↘ number of time steps (smaller dimension).

Simulation of the rough Bergomi dynamics

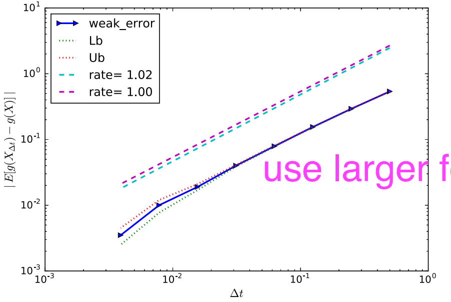
Goal: Simulate jointly $(W_t^1, \widetilde{W}_t^H : 0 \leq t \leq T)$, resulting in $W_{t_1}^1, \dots, W_{t_N}$ and $\widetilde{W}_{t_1}^H, \dots, \widetilde{W}_{t_N}^H$ along a given grid $t_1 < \dots < t_N$

1. **Covariance based approach** [Bayer et al., 2016]
   - Based on Cholesky decomposition of the covariance matrix of the $(2N)$-dimensional Gaussian random vector $W_{t_1}^1, \dots, W_{t_N}^1, \widetilde{W}_{t_1}^H, \dots, \widetilde{W}_{t_N}$.
   - Exact method but slow. at least O(N^2),
2. **The hybrid scheme** [Bennedsen et al., 2017]
   - Based on Euler discretization but crucially improved by moment matching for the singular term in the left point rule.
   - Accurate scheme that is much faster than the Covariance based approach.

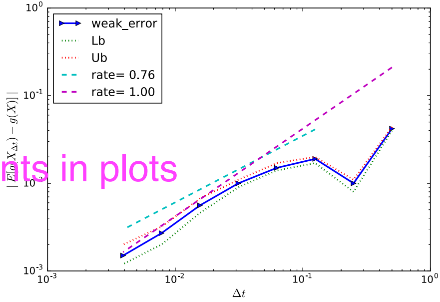
$\boxed{\text{O(N) up to log factors that depend on the desired error}}$

On the choice of the simulation scheme

Figure 1.1: The convergence of the weak error $\mathcal{E}_B$, using MC with 6×10^6 samples, for Set 1 parameter in Table 1. The upper and lower bounds are 95% confidence intervals. a) With the hybrid scheme b) With the exact scheme.



(a)



(b)

use larger fonts in plots

mention the parameters here for the sake of clarity

Hybrid scheme [Bennedsen et al., 2017]

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

arising

- The hybrid scheme discretizes the $\widetilde{W}^H$ process into Wiener integrals of power functions and a Riemann sum, appearing from approximating the kernel by power functions near the origin and step functions elsewhere.

the upperscript notation is confusing

$$\widetilde{W}^H_{\frac{i}{N}} \approx \overline{W}^H_{\frac{i}{N}} = \sqrt{2H}\left(W_i^2 + \sum_{k=2}^{i}\left(\frac{b_k}{N}\right)^{H-\frac{1}{2}}\left(W^1_{\frac{i-(k-1)}{N}} - W^1_{\frac{i-k}{N}}\right)\right),$$

need to declare the correlation between W1 and W2 here

  - N is the number of time steps
  - $\{W_j^2\}_{j=1}^N$: Artificially introduced N Gaussian random variables that are used for left-rule points in the hybrid scheme.
  - $b_k = \left(\frac{k^{H+\frac{1}{2}} - (k-1)^{H+\frac{1}{2}}}{H+\frac{1}{2}}\right)^{\frac{1}{H-\frac{1}{2}}}$.

mention the error in this method

conditional expectation for

Analytic smoothing

$$
\begin{aligned}
C_{RB}(T,K) &= E\left[(S_T-K)^+\right] \\
&= \mathrm{E}\left[\mathrm{E}\left[(S_T-K)^+ \mid \sigma(W^1(t), t\le T)\right]\right] \\
&= E\left[C_{BS}\left(S_0 = \exp\left(\rho\int_0^T \sqrt{v_t}dW_t^1 - \frac{1}{2}\rho^2\int_0^T v_t dt\right),\right.\right. \\
&\qquad\qquad \left.\left. k=K,\ \sigma^2 = (1-\rho^2)\int_0^T v_t dt\right)\right] \\
&\approx \int_{\mathbb{R}^{2N}} C_{BS}\left(G(\mathbf{w}^{(1)},\mathbf{w}^{(2)})\right)\rho_N(\mathbf{w}^{(1)})\rho_N(\mathbf{w}^{(2)})d\mathbf{w}^{(1)}d\mathbf{w}^{(2)} \\
&= C_{RB}^N \qquad (2)
\end{aligned}
$$

again, the notation may confuse since this w_2 is not the w_2 from the hybrid scheme

- $C_{\mathrm{BS}}(S_0, k, \sigma^2)$: the Black-Scholes call price, for initial spot price S_0, strike price k, and volatility σ^2.
- G maps $2N$ independent standard Gaussian random inputs to the parameters fed to Black-Scholes formula.
- ρ_N: the multivariate Gaussian density, N: number of time steps.

Numerical integration methods

- **Plain Monte Carlo (MC)**

what is d in this context? Can we just use N instead?

  - $\varepsilon(M) = \mathcal{O}\left(M^{-1/2}\right)$
  - (+) insensitive to d, (−) slow convergence, no profit from regularity.
- **Classical Quasi-Monte Carlo (QMC)**
  - $\varepsilon(M) = \mathcal{O}\left(M^{-1}\log(M)^{d-1}\right)$
  - (+) better convergence, (−) sensitive to d, no profit from regularity.
- **Quadrature based on product approaches**
  - $\varepsilon(M) = \mathcal{O}\left(M^{-r/d}\right)$
  - (+) profits from regularity, faster than QMC if $r > d$, (−) highly sensitive to d.
- **Sparse grids quadrature (SGQ)**
  - $\varepsilon(M) = \mathcal{O}\left(M^{-s}\log(M)^{(d-1)(s+1)}\right)$
  - (+) profits from regularity, faster than QMC if $s > 1$, less sensitive to d.

ε: prescribed accuracy, M: the amount of work, d: dimension of problem, r, s: smoothness indices (bounded mixed (total) derivatives up to order $s(r)$).

I would introduce an error split to motivate the quadrature

Sparse grids I

cite Griebel acta numerica

Goal: Given $F : \mathbb{R}^d \to \mathbb{R}$ and a multi-index $\boldsymbol{\beta} \in \mathbb{N}_+^d$, approximate

$$\mathrm{E}\left[F\right] \approx Q^{m(\boldsymbol{\beta})}[F],$$

where $Q^{m(\boldsymbol{\beta})}$ a Cartesian quadrature grid with $m(\beta_n)$ points along y_n.
Idea: Denote $Q^{m(\boldsymbol{\beta})}[F] = F_{\boldsymbol{\beta}}$ and introduce the first difference operator

$$\Delta_i F_{\boldsymbol{\beta}} \begin{cases} F_{\boldsymbol{\beta}} - F_{\boldsymbol{\beta}-e_i}, & \text{if } \beta_i > 1 \\ F_{\boldsymbol{\beta}} & \text{if } \beta_i = 1 \end{cases}$$

where e_i denotes the ith d-dimensional unit vector, and mixed difference operators

$$\Delta[F_{\boldsymbol{\beta}}] = \otimes_{i=1}^{d} \Delta_i F_{\boldsymbol{\beta}}$$

Sparse grids II

A quadrature estimate of $E[F]$ is

$$\mathcal{M}_{\mathcal{I}_\ell}[F] = \sum_{\boldsymbol{\beta} \in \mathcal{I}_\ell} \Delta[F_{\boldsymbol{\beta}}], \tag{3}$$

- Product approach: $\mathcal{I}_\ell = \{\max\{\beta_1, \dots, \beta_d\} \le \ell;\ \boldsymbol{\beta} \in \mathbb{N}_+^d\}$
- Regular SG: $\mathcal{I}_\ell = \{|\ \boldsymbol{\beta}\ |_1 \le \ell + d - 1;\ \boldsymbol{\beta} \in \mathbb{N}_+^d\}$

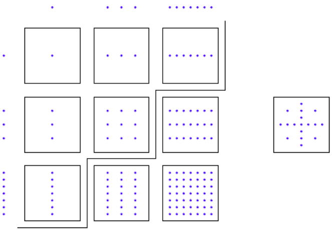


Figure 2.1: Left are product grids $\Delta_{\beta_1} \otimes \Delta_{\beta_2}$ for $1 \le \beta_1, \beta_2 \le 3$. Right is the corresponding SG construction.

- ASGQ: $\mathcal{I}_\ell = \mathcal{I}^{\text{ASGQ}}$.

ASGQ in practice

- The construction of $\mathcal{I}^{\text{ASGQ}}$ is done by profit thresholding

$$\mathcal{I}^{\text{ASGQ}} = \{\boldsymbol{\beta} \in \mathbb{N}_+^d : P_{\boldsymbol{\beta}} \geq \overline{T}\}.$$

- **Profit of a hierarchical surplus** $P_{\boldsymbol{\beta}} = \frac{|\Delta E_{\boldsymbol{\beta}}|}{\Delta \mathcal{W}_{\boldsymbol{\beta}}}$.
- **Error contribution**: $\Delta E_{\boldsymbol{\beta}} = \left| \mathcal{M}^{\mathcal{I} \cup \{\boldsymbol{\beta}\}} - \mathcal{M}^{\mathcal{I}} \right|$.
- **Work contribution**: $\Delta \mathcal{W}_{\boldsymbol{\beta}} = \text{Work}[\mathcal{M}^{\mathcal{I} \cup \{\boldsymbol{\beta}\}}] - \text{Work}[\mathcal{M}^{\mathcal{I}}]$.

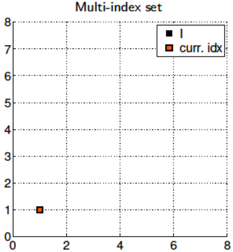


Figure 2.2: A posteriori, adaptive construction: Given an index set $\mathcal{I}_k$, compute the profits of the neighbor indices and select the most profitable one

ASGQ in practice

- The construction of $\mathcal{I}^{\text{ASGQ}}$ is done by profit thresholding

$$\mathcal{I}^{\text{ASGQ}} = \{\boldsymbol{\beta} \in \mathbb{N}_+^d : P_{\boldsymbol{\beta}} \geq \overline{T}\}.$$

- **Profit of a hierarchical surplus** $P_{\boldsymbol{\beta}} = \frac{|\Delta E_{\boldsymbol{\beta}}|}{\Delta \mathcal{W}_{\boldsymbol{\beta}}}$.
- **Error contribution**: $\Delta E_{\boldsymbol{\beta}} = \left|\mathcal{M}^{\mathcal{I}\cup\{\boldsymbol{\beta}\}} - \mathcal{M}^{\mathcal{I}}\right|$.
- **Work contribution**: $\Delta \mathcal{W}_{\boldsymbol{\beta}} = \text{Work}[\mathcal{M}^{\mathcal{I}\cup\{\boldsymbol{\beta}\}}] - \text{Work}[\mathcal{M}^{\mathcal{I}}]$.

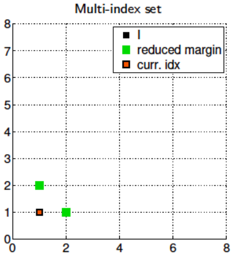


Figure 2.3: A posteriori, adaptive construction: Given an index set $\mathcal{I}_k$, compute the profits of the neighbor indices and select the most profitable one.

ASGQ in practice

- The construction of $\mathcal{I}^{\text{ASGQ}}$ is done by profit thresholding

$$\mathcal{I}^{\text{ASGQ}} = \{\boldsymbol{\beta} \in \mathbb{N}_{+}^{d} : P_{\boldsymbol{\beta}} \geq \overline{T}\}.$$

- **Profit of a hierarchical surplus** $P_{\boldsymbol{\beta}} = \frac{|\Delta E_{\boldsymbol{\beta}}|}{\Delta \mathcal{W}_{\boldsymbol{\beta}}}$.
- **Error contribution:** $\Delta E_{\boldsymbol{\beta}} = \left|\mathcal{M}^{\mathcal{I} \cup \{\boldsymbol{\beta}\}} - \mathcal{M}^{\mathcal{I}}\right|$.
- **Work contribution:** $\Delta \mathcal{W}_{\boldsymbol{\beta}} = \text{Work}[\mathcal{M}^{\mathcal{I} \cup \{\boldsymbol{\beta}\}}] - \text{Work}[\mathcal{M}^{\mathcal{I}}]$.

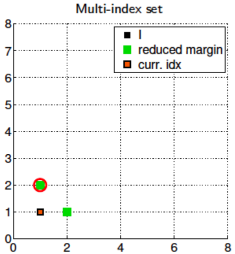


Figure 2.4: A posteriori, adaptive construction: Given an index set $\mathcal{I}_k$, compute the profits of the neighbor indices and select the most profitable one.

ASGQ in practice

- The construction of $\mathcal{I}^{\text{ASGQ}}$ is done by profit thresholding

$$\mathcal{I}^{\text{ASGQ}} = \{\boldsymbol{\beta} \in \mathbb{N}_+^d : P_{\boldsymbol{\beta}} \geq \overline{T}\}.$$

- **Profit of a hierarchical surplus** $P_{\boldsymbol{\beta}} = \frac{|\Delta E_{\boldsymbol{\beta}}|}{\Delta \mathcal{W}_{\boldsymbol{\beta}}}$.
- **Error contribution:** $\Delta E_{\boldsymbol{\beta}} = \left|\mathcal{M}^{\mathcal{I}\cup\{\boldsymbol{\beta}\}} - \mathcal{M}^{\mathcal{I}}\right|$.
- **Work contribution:** $\Delta \mathcal{W}_{\boldsymbol{\beta}} = \text{Work}[\mathcal{M}^{\mathcal{I}\cup\{\boldsymbol{\beta}\}}] - \text{Work}[\mathcal{M}^{\mathcal{I}}]$.

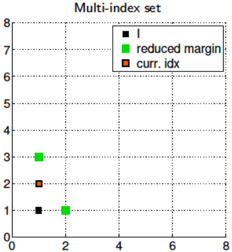


Figure 2.5: A posteriori, adaptive construction: Given an index set $\mathcal{I}_k$, compute the profits of the neighbor indices and select the most profitable one.

link to misc

ASGQ in practice

- The construction of $\mathcal{I}^{\text{ASGQ}}$ is done by profit thresholding

$$\mathcal{I}^{\text{ASGQ}} = \{\boldsymbol{\beta} \in \mathbb{N}_+^d : P_{\boldsymbol{\beta}} \geq \overline{T}\}.$$

- **Profit of a hierarchical surplus** $P_{\boldsymbol{\beta}} = \frac{|\Delta E_{\boldsymbol{\beta}}|}{\Delta \mathcal{W}_{\boldsymbol{\beta}}}$.
- **Error contribution**: $\Delta E_{\boldsymbol{\beta}} = \left|\mathcal{M}^{\mathcal{I}\cup\{\boldsymbol{\beta}\}} - \mathcal{M}^{\mathcal{I}}\right|$.
- **Work contribution**: $\Delta \mathcal{W}_{\boldsymbol{\beta}} = \text{Work}[\mathcal{M}^{\mathcal{I}\cup\{\boldsymbol{\beta}\}}] - \text{Work}[\mathcal{M}^{\mathcal{I}}]$.

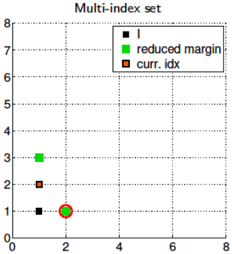


Figure 2.6: A posteriori, adaptive construction: Given an index set $\mathcal{I}_k$, compute the profits of the neighbor indices and select the most profitable one.

ASGQ in practice

- The construction of $\mathcal{I}^{\text{ASGQ}}$ is done by profit thresholding

$$\mathcal{I}^{\text{ASGQ}} = \{\boldsymbol{\beta} \in \mathbb{N}_+^d : P_{\boldsymbol{\beta}} \geq \overline{T}\}.$$

- **Profit of a hierarchical surplus** $P_{\boldsymbol{\beta}} = \frac{|\Delta E_{\boldsymbol{\beta}}|}{\Delta \mathcal{W}_{\boldsymbol{\beta}}}$.
- **Error contribution**: $\Delta E_{\boldsymbol{\beta}} = \left|\mathcal{M}^{\mathcal{I} \cup \{\boldsymbol{\beta}\}} - \mathcal{M}^{\mathcal{I}}\right|$.
- **Work contribution**: $\Delta \mathcal{W}_{\boldsymbol{\beta}} = \text{Work}[\mathcal{M}^{\mathcal{I} \cup \{\boldsymbol{\beta}\}}] - \text{Work}[\mathcal{M}^{\mathcal{I}}]$.

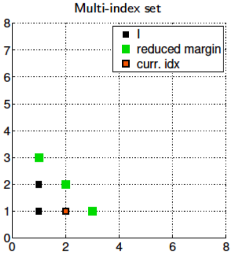


Figure 2.7: A posteriori, adaptive construction: Given an index set $\mathcal{I}_k$, compute the profits of the neighbor indices and select the most profitable one.

ASGQ in practice

- The construction of $\mathcal{I}^{\text{ASGQ}}$ is done by profit thresholding

$$\mathcal{I}^{\text{ASGQ}} = \{\boldsymbol{\beta} \in \mathbb{N}_+^d : P_{\boldsymbol{\beta}} \geq \overline{T}\}.$$

- **Profit of a hierarchical surplus** $P_{\boldsymbol{\beta}} = \frac{|\Delta E_{\boldsymbol{\beta}}|}{\Delta \mathcal{W}_{\boldsymbol{\beta}}}$.
- **Error contribution**: $\Delta E_{\boldsymbol{\beta}} = \left|\mathcal{M}^{\mathcal{I}\cup\{\boldsymbol{\beta}\}} - \mathcal{M}^{\mathcal{I}}\right|$.
- **Work contribution**: $\Delta \mathcal{W}_{\boldsymbol{\beta}} = \text{Work}[\mathcal{M}^{\mathcal{I}\cup\{\boldsymbol{\beta}\}}] - \text{Work}[\mathcal{M}^{\mathcal{I}}]$.

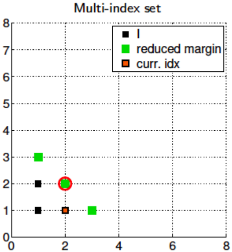


Figure 2.8: A posteriori, adaptive construction: Given an index set $\mathcal{I}_k$, compute the profits of the neighbor indices and select the most profitable one.

Randomized QMC

- A (rank-1) lattice rule [Sloan, 1985, Nuyens, 2014] with n points

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

  where $z = (z_1, \ldots, z_d) \in \mathbb{N}^d$.
- A randomly shifted lattice rule

$$\overline{Q}_{n,q}(f) = \frac{1}{q} \sum_{i=0}^{q-1} Q_n^{(i)}(f) = \frac{1}{q} \sum_{i=0}^{q-1} \left(\frac{1}{n} \sum_{k=0}^{n-1} f\left(\frac{kz + \Delta^{(i)} \bmod n}{n}\right)\right), \quad (4)$$

  where $\{\Delta^{(i)}\}_{i=1}^{q}$: independent random shifts, and $M^{\mathrm{QMC}} = q \times n$.
  - Unbiased approximation of the integral.
  - Practical error estimate.
- We use a pre-made point generators using latticeseq_b2.py from `https://people.cs.kuleuven.be/~dirk.nuyens/qmc-generators/`.

Wiener Path generation methods

$\{t_i\}_{i=0}^{N}$: Grid of time steps, $\{B_{t_i}\}_{i=0}^{N}$: Brownian motion increments

- **Random Walk**
  - Proceeds incrementally, given B_{t_i},

$$B_{t_{i+1}} = B_{t_i} + \sqrt{\Delta t} z_i, \; z_i \sim \mathcal{N}(0,1).$$

  - All components of $\mathbf{z} = (z_1, \ldots, z_N)$ have the same scale of importance: isotropic.
- **Hierarchical Brownian Bridge** [Glasserman, 2004]
  - 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 ($\rho = \frac{j-i}{k-i}$)

$$B_{t_j} = (1-\rho) B_{t_i} + \rho B_{t_k} + \sqrt{\rho(1-\rho)(k-i)\Delta t} z_j, \; z_j \sim \mathcal{N}(0,1). \tag{5}$$

  - The most important values (determine the large scale structure of Brownian motion) are the first components of $\mathbf{z} = (z_1, \ldots, z_N)$.
  - $\searrow$ the effective dimension (# important dimensions) and $\nearrow$ anisotropy between different directions $\Rightarrow$ Faster ASGQ and QMC convergence.

Error comparison

$\mathcal{E}_{\text{tot}}$: the total error of approximating the expectation in (2).

- When using ASGQ estimator, Q_N

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

  where $\mathcal{E}_Q$ is the quadrature error, $\mathcal{E}_B$ is the bias, TOL_{ASGQ} is a user selected tolerance for ASGQ method.

- When using randomized QMC or MC estimator, $Q_N^{\text{MC (QMC)}}$

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

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

- M^{QMC} and M^{MC}, are chosen so that $\mathcal{E}_{S,\text{QMC}}(M^{\text{QMC}})$ and $\mathcal{E}_{S,\text{MC}}(M^{\text{MC}})$ satisfy

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

Numerical experiments

Table 1: Reference solution (using MC with 500 time steps and number of samples, $M = 8 \times 10^6$) of call option price under the rough Bergomi model, for different parameter constellations. The numbers between parentheses correspond to the statistical errors estimates.

Parameters	Reference solution
Set 1: $H = 0.07, K = 1, S_0 = 1, T = 1, \rho = -0.9, \eta = 1.9, \xi_0 = 0.235$	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)

Set 1

- The first set is the closest to the empirical findings [Gatheral et al., 2018, Bennedsen et al., 2016], suggesting that $H \approx 0.1$. The choice of values $\nu = 1.9$ and $\rho = -0.9$ is justified by [Bayer et al., 2016].
- For the remaining three sets, we wanted to test the potential of our method for a very rough case, where variance reduction methods are inefficient.

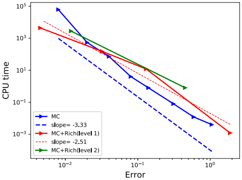
Relative errors and computational gains

Table 2: In this table, we highlight the computational gains achieved by ASGQ and QMC over MC method to meet a certain error tolerance. We note that the ratios are computed for the best configuration with Richardson extrapolation for each method.

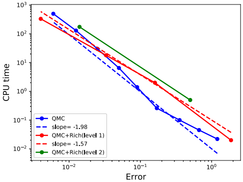
Parameter set	**Relative error**	**CPU time ratio** (**MC/ASGQ**)	**CPU time ratio** (**MC/QMC**)
Set 1	1%	15	10
Set 2	0.2%	21.5	73.2
Set 3	0.4%	26.7	21.3
Set 4	2%	5	10

Complexity of the different methods

Computational work

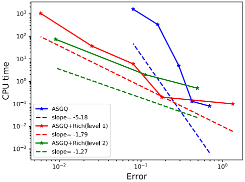


(a)



(b)

use larger fonts, this is impossible to read



computational work

(c)

Figure 3.1: Numerical complexity of the different methods with the different configurations in terms of Richardson extrapolation 's level. Case of parameter set 1 in Table 1. a) MC methods. b) QMC methods. d) ASGQ methods.

Complexity of the different methods with their best configurations

comutational work

make the fig larger, you have space

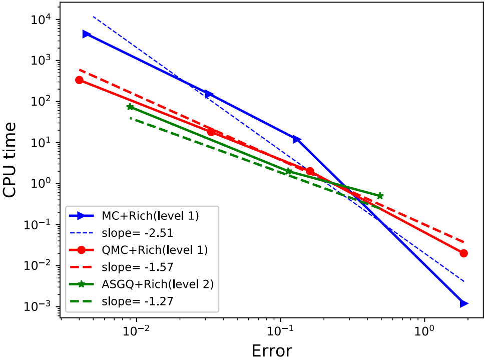


Figure 3.2: Computational work comparison for the different methods with the best configurations concluded from Figure 3.1, for the case of parameter set 1 in Table 1.

Conclusions

Using conditional expectations for numerical smoothing

- Proposed novel, fast option pricers, based on hierarchical deterministic quadrature methods, for options whose underlyings follow the rBergomi model.
- Given a sufficiently small relative error tolerance, our proposed methods demonstrate substantial computational gains over the standard MC method, for different parameter constellations.
- Accelerating our novel methods can be achieved by using more optimal hierarchical path generation method than Brownian bridge construction, such as PCA or LT transformations.

unclear

mention also that one may use better QMC or ASG as well

Thank you for your attention

References I

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

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

Bayer, C., Hammouda, C. B., and Tempone, R. (2018).
Hierarchical adaptive sparse grids for option pricing under the rough bergomi model.
arXiv preprint arXiv:1812.08533.

References II

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

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

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

Glasserman, P. (2004).
Monte Carlo methods in financial engineering.
Springer, New York.

References III

McCrickerd, R. and Pakkanen, M. S. (2018).
Turbocharging Monte Carlo pricing for the rough Bergomi model.
Quantitative Finance, pages 1–10.

Morokoff, W. J. and Caflisch, R. E. (1994).
Quasi-random sequences and their discrepancies.
SIAM Journal on Scientific Computing, 15(6):1251–1279.

Moskowitz, B. and Caflisch, R. E. (1996).
Smoothness and dimension reduction in quasi-Monte Carlo methods.
Mathematical and Computer Modelling, 23(8):37–54.

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

References IV

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

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

Sloan, I. H. (1985).
Lattice methods for multiple integration.
Journal of Computational and Applied Mathematics, 12:131–143.

Talay, D. and Tubaro, L. (1990).
Expansion of the global error for numerical schemes solving stochastic differential equations.
Stochastic analysis and applications, 8(4):483–509.