Learning Rough Volatility

Blanka Horvath
King's College London and
Imperial College London

Hammamet, 29th October 2018

International Conference on Control, Games and Stochastic Analysis
Hammamet, Oct. 29 to Nov. 01, 2018, Tunisia

- Rough volatility models have been around since October 2014 (see the Rough Volatility website for a chronicle of developments)
- These models have repeatedly proven to be superior to standard models in many areas: in volatility forecasting, in option pricing, close fits to the implied vol surface, . . .

- Rough volatility models have been around since October 2014 (see the Rough Volatility website for a chronicle of developments)
- These models have repeatedly proven to be superior to standard models in many areas: in volatility forecasting, in option pricing, close fits to the implied vol surface, . . .
- Relaxing the assumption of independence of volatility increments was crucial for the superior performance of rough volatility models $\Rightarrow$ but: several standard pricing methods no longer available & naive Monte Carlo methods slow
- Calibration time has been a bottleneck for rough volatility several advances have been made to speed up the calibration process [BLP '15, MP '17, HJM '17].

Today's talk:
Speedups for rough volatility models along two lines:

1. in pricing of vanilla options based on faster Monte Carlo approximations for a family of rough stochastic volatility models. [H-Jacquier-Muguruza '17])
2. in calibration by means of machine learning (ongoing with A. Muguruza and with M. Tomas).

Digression: Rough Volatility

Suppose a generic Itô process framework for the stock price $(S_t)_{t\geq 0}$:

$$\frac{dS_t}{S_t} = \mu_t dt + \sigma_t dB_t, \quad t \geq 0.$$

The phrase "**rough volatility**" refers to the idea that sample paths of the log volatility $\log(\sigma_t)$, $t \geq 0$ are rougher than the sample paths of Brownian motion.

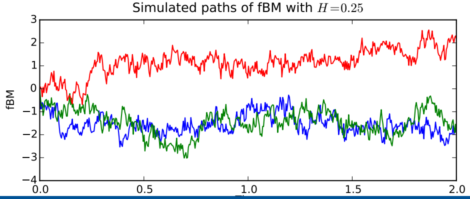
Volatility is Rough

Gatheral, Jaisson and Rosenbaum (2014) suggested that volatility is rough. The slogan "**volatility is rough**" refers to the idea that sample paths of the log volatility $\log(\sigma_t)$, $t \geq 0$ are rougher than the sample paths of Brownian motion (in terms of Hölder regularity).

Fractional Brownian motion

A fractional Brownian motion with Hurst parameter $H \in (0,1)$ is a continuous centered Gaussian process $(B_t^H)_{t\in\mathbb{R}}$ with covariance function

$$Cov(B_t^H, B_s^H) = \frac{1}{2}\left(|t|^{2H} + |s|^{2H} - |t-s|^{2H}\right), \quad s,t \in \mathbb{R}. \tag{1}$$



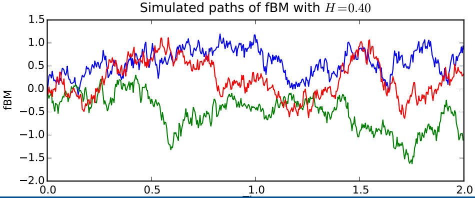
Volatility is Rough

Gatheral, Jaisson and Rosenbaum (2014) suggested that volatility is rough. The slogan "**volatility is rough**" refers to the idea that sample paths of the log volatility $\log(\sigma_t)$, $t \geq 0$ are rougher than the sample paths of Brownian motion (in terms of Hölder regularity).

Fractional Brownian motion

A fractional Brownian motion with Hurst parameter $H \in (0,1)$ is a continuous centered Gaussian process $(B_t^H)_{t \in \mathbb{R}}$ with covariance function

$$Cov(B_t^H, B_s^H) = \frac{1}{2}\left(|t|^{2H} + |s|^{2H} - |t-s|^{2H}\right), \quad s, t \in \mathbb{R}. \tag{2}$$



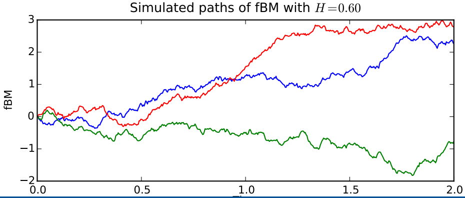
Volatility is Rough

Gatheral, Jaisson and Rosenbaum (2014) suggested that volatility is rough. The slogan "**volatility is rough**" refers to the idea that sample paths of the log volatility $\log(\sigma_t)$, $t \geq 0$ are rougher than the sample paths of Brownian motion (in terms of Hölder regularity).

Fractional Brownian motion

A fractional Brownian motion with Hurst parameter $H \in (0,1)$ is a continuous centered Gaussian process $(B_t^H)_{t\in\mathbb{R}}$ with covariance function

$$Cov(B_t^H, B_s^H) = \frac{1}{2}\left(|t|^{2H} + |s|^{2H} - |t-s|^{2H}\right), \quad s,t \in \mathbb{R}. \tag{3}$$



Under the physical measure, $\mathbb{P}$:

Gatheral, Jaisson and Rossenbaum proposed the following rough/fractional volatility model:

$$\begin{cases} dS_t = S_t \mu_t dt + S_t \sigma_t dW_t, & S_0 > 0 \\ \sigma_t = \sigma_0 \exp(W_t^H), & \sigma_0 > 0. \end{cases}$$

where W^H is a fractional Brownian motion with Hurst parameter $H \in (0, 1/2)$.

Implied volatility

- Asset price process: $(S_t = \mathrm{e}^{X_t})_{t\geq 0}$, with $X_0 = 0$.
- Black-Scholes-Merton (BSM) framework:

$$C_{\mathrm{BS}}(\tau, k, \sigma) := \mathbb{E}_0\left(\mathrm{e}^{X_\tau} - \mathrm{e}^k\right)_+ = \mathcal{N}(d_+) - \mathrm{e}^k \mathcal{N}(d_-),$$

$$d_\pm := -\frac{k}{\sigma\sqrt{\tau}} \pm \frac{1}{2}\sigma\sqrt{\tau}.$$

- Spot implied volatility $\sigma_\tau(k)$: the unique (non-negative) solution to

$$C_{\mathrm{observed}}(\tau, k) = C_{\mathrm{BS}}(\tau, k, \sigma_\tau(k)).$$

- Implied volatility: unit-free measure of option prices.

At the money skew

Let $\sigma_{BS}(k,\tau)$ denote the Black-Scholes implied volatility ($\tau := T - t$ and $k = \log\left(\frac{K}{S}\right)$) for an asset S. Then the at-the-money volatility skew is defined as

$$\psi(\tau) = \left|\frac{\partial}{\partial k}\sigma_{BS}(k,\tau)\right|_{k=0} \quad ; \quad \tau \geq 0.$$

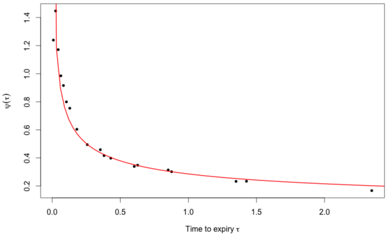


Figure 1.2: The black dots are non-parametric estimates of the S&P at-the-money (ATM) volatility skews as of August 14, 2013; the red curve is the power-law fit $\psi(\tau) = A\,\tau^{-0.407}$, τ measured in years.

Today's talk:
Speedups for rough volatility models along two lines:

1. in pricing of vanilla options based on faster Monte Carlo approximations for a family of rough stochastic volatility models. [H-Jacquier-Muguruza '17])
2. in calibration by means of machine learning (ongoing with A. Muguruza and with M. Tomas).

Our general framework

$$
\begin{aligned}
\mathrm{d}X_t &= -\frac{1}{2}V_t\mathrm{d}t + \sqrt{V_t}\mathrm{d}W_t, \quad && X_0 = 0, \\
V_t &= \Phi\left(\int_0^t g(t-s)\mathrm{d}Y_s\right), \quad && V_0 > 0,\ \alpha \in (-1/2, 1/2), \\
dY_t &= b(Y_t)\mathrm{d}t + \sigma(Y_t)\mathrm{d}Z_t, \quad && \mathrm{d}Z_t\mathrm{d}W_t = \rho\mathrm{d}t.
\end{aligned}
$$

where $\Phi \in \mathcal{C}^1$, $g \in \mathcal{L}^\alpha := \left\{u^\alpha L(u) : L \in \mathcal{C}_b^1([0,T]), \alpha \in \left(-\frac{1}{2}, \frac{1}{2}\right)\right\}$ and Y satisfies Yamada-Watanabe conditions for path-wise uniqueness.

Examples in this framework

$$
\begin{aligned}
\mathrm{d}X_t &= -\frac{1}{2}V_t\mathrm{d}t + \sqrt{V_t}\mathrm{d}W_t, & X_0 &= 0,\\
V_t &= \Phi\left(\int_0^t g(t-s)\mathrm{d}Y_s\right), & V_0 &> 0,\ \alpha \in (-1/2, 1/2),\\
dY_t &= b(Y_t)\mathrm{d}t + \sigma(Y_t)\mathrm{d}Z_t, & \mathrm{d}Z_t\mathrm{d}W_t &= \rho\mathrm{d}t.
\end{aligned}
$$

$\Downarrow$ $g(u) = u^\alpha,\ \Phi(x) = \mathcal{E}(x),\ Y_t = Z_t$

Examples in this framework

$$
\begin{aligned}
\mathrm{d}X_t &= -\frac{1}{2}V_t\mathrm{d}t + \sqrt{V_t}\mathrm{d}W_t, && X_0 = 0,\\
V_t &= \Phi\left(\int_0^t g(t-s)\mathrm{d}Y_s\right), && V_0 > 0,\ \alpha \in (-1/2, 1/2),\\
dY_t &= b(Y_t)\mathrm{d}t + \sigma(Y_t)\mathrm{d}Z_t, && \mathrm{d}Z_t\mathrm{d}W_t = \rho\mathrm{d}t.
\end{aligned}
$$

$\Downarrow$ $g(u) = u^\alpha,\ \Phi(x) = \mathcal{E}(x),\ Y_t = Z_t$

Rough Bergomi:

$$
\begin{aligned}
\mathrm{d}X_t &= -\frac{1}{2}V_t\mathrm{d}t + \sqrt{V_t}\mathrm{d}W_t, && X_0 = 0\\
V_t &= \xi_0(t)\mathcal{E}\left(2\nu C_H\int_0^t \frac{dZ_u}{(t-u)^{1/2-H}}\right), && \nu, \xi_0(\cdot) > 0\\
dZ_t dW_t &= \rho dt, && \rho \in (0,1)
\end{aligned}
$$

Examples in this framework

$$
\begin{aligned}
\mathrm{d}X_t &= -\frac{1}{2} V_t \mathrm{d}t + \sqrt{V_t}\mathrm{d}W_t, & X_0 = 0, \\
V_t &= \Phi\left(\int_0^t g(t-s)\mathrm{d}Y_s\right), & V_0 > 0,\ \alpha \in (-1/2, 1/2), \\
dY_t &= b(Y_t)\mathrm{d}t + \sigma(Y_t)\mathrm{d}Z_t, & \mathrm{d}Z_t \mathrm{d}W_t = \rho \mathrm{d}t.
\end{aligned}
$$

$\Downarrow$ $g(u) = u^{\alpha}$, $\Phi(x) = \eta + Id.$, $\mathrm{d}Y_t = \kappa(\theta - Y_t)\mathrm{d}t + \xi\sqrt{Y_t}\mathrm{d}Z_t$

Examples in this framework

$$
\begin{aligned}
\mathrm{d}X_t &= -\frac{1}{2}V_t\mathrm{d}t + \sqrt{V_t}\mathrm{d}W_t, && X_0 = 0,\\
V_t &= \Phi\left(\int_0^t g(t-s)\mathrm{d}Y_s\right), && V_0 > 0,\ \alpha \in (-1/2, 1/2),\\
dY_t &= b(Y_t)\mathrm{d}t + \sigma(Y_t)\mathrm{d}Z_t, && \mathrm{d}Z_t\mathrm{d}W_t = \rho\mathrm{d}t.
\end{aligned}
$$

$\Downarrow$ $g(u) = u^\alpha$, $\Phi(x) = \eta + Id.$, $\mathrm{d}Y_t = \kappa(\theta - Y_t)\mathrm{d}t + \xi\sqrt{Y_t}\mathrm{d}Z_t$

Rough Heston:

$$
\begin{aligned}
\mathrm{d}X_t &= -\frac{1}{2}V_t\mathrm{d}t + \sqrt{V_t}\mathrm{d}W_t, && X_0 = 0,\\
Y_t &= \int_0^t \kappa(\theta - Y_s)\mathrm{d}t + \int_0^t \xi\sqrt{Y_s}\mathrm{d}Z_s && V_0, \kappa, \xi, \theta > 0,\ 2\kappa\theta > \xi^2\\
V_t &= \eta + \int_0^t (t-s)^\alpha \mathrm{d}Y_s, && \eta > 0,\ \alpha \in (-1/2, 1/2).
\end{aligned}
$$

FCLT for Hölder cont. processes:

$$
\begin{tikzcd}
(\mathcal{C}^{1/2-}([0,T]), \|\cdot\|_{1/2}) \arrow[r, bend left, "\mathcal{G}^\alpha"] & (\mathcal{C}^{\alpha+1/2}([0,T]), \|\cdot\|_{\alpha+1/2}) \arrow[r, bend left, "\Phi"] & (\mathcal{C}^{\alpha+1/2}([0,T]), \|\cdot\|_{\alpha+1/2}) \\
Y_n(t) \arrow[r, "\mathcal{G}^\alpha"] \arrow[d, Rightarrow, "{(\mathcal{L}, \|\cdot\|_{1/2})}"'] & \mathcal{B}_n(t) = \mathcal{G}^\alpha(Y_n(t)) \arrow[r, "\Phi"] \arrow[d, Rightarrow, "{(\mathcal{L}, \|\cdot\|_{\alpha+1/2})}"'] & \Phi\left(\mathcal{B}_n(t)\right) \arrow[d, Rightarrow, "{(\mathcal{L}, \|\cdot\|_{\alpha+1/2})}"'] \\
Y(t) \arrow[r, "\mathcal{G}^\alpha"] & \mathcal{B}(t) = \int_0^t g(t-s)\mathrm{d}Y_s \arrow[r, "\Phi"] & \Phi\left(\mathcal{B}(t)\right)
\end{tikzcd}
$$

FCLT for Hölder continuous processes

Theorem (rough Donsker theorem)

Consider the sequence $(W_n(t))_{n\geq 1}$ *and* W *its weak limit in* $\left(\mathcal{C}^{1/2}([0,T]), \|\cdot\|_{1/2}\right)$.
Then $(\mathcal{G}^{\alpha} W_n)_{n\geq 1}$ *converges weakly to* $\int_0^{\cdot} g(\cdot - s)\mathrm{d}W_s$ *in* $\left(\mathcal{C}^{\alpha+1/2}([0,T]), \|\cdot\|_{\alpha+1/2}\right)$ *for* $\alpha \in (-\frac{1}{2}, \frac{1}{2})$.

FCLT for rough volatility models

FCLT for rough volatility models

Define recursively in time, for any $n \geq 1$, $t \in [0, T]$, $t_k = \frac{k}{N}$

$$X_n(t) := -\frac{1}{2}\frac{T}{n}\sum_{k=1}^{\lfloor nt \rfloor} \Phi\left(\left(\mathcal{G}^{\alpha} Y_n\right)(t_k)\right) + \sqrt{\frac{T}{\sigma n}}\sum_{k=1}^{\lfloor nt \rfloor} \sqrt{\Phi\left(\left(\mathcal{G}^{\alpha} Y_n\right)(t_k)\right)}\left(W_n(t_{k+1}) - W_n(t_k)\right)$$

FCLT for rough volatility models

Define recursively in time, for any $n \geq 1$, $t \in [0, T]$, $t_k = \frac{k}{N}$

$$X_n(t) := -\frac{1}{2}\frac{T}{n}\sum_{k=1}^{\lfloor nt \rfloor} \Phi\left(\left(\mathcal{G}^{\alpha} Y_n\right)(t_k)\right) + \sqrt{\frac{T}{\sigma n}}\sum_{k=1}^{\lfloor nt \rfloor} \sqrt{\Phi\left(\left(\mathcal{G}^{\alpha} Y_n\right)(t_k)\right)}\left(W_n(t_{k+1}) - W_n(t_k)\right)$$

Theorem (rDonsker for rough volatility models)

$(X_n)_{n\geq 1}$, converges weakly to X in $(\mathcal{C}^{1/2-}(\mathbb{T}), \|\cdot\|_{1/2_-})$,

$$\begin{aligned}
\mathrm{d}X_t &= -\frac{1}{2}V_t\mathrm{d}t + \sqrt{V_t}\mathrm{d}W_t, & X_0 = 0,\\
V_t &= \Phi\left(\int_0^t g(t-s)\mathrm{d}Y_s\right), & V_0 > 0,\ \alpha \in \left(-1/2, 1/2\right),\\
dY_t &= b(Y_t)\mathrm{d}t + \sigma(Y_t)\mathrm{d}Z_t, & \mathrm{d}Z_t\mathrm{d}W_t = \rho\mathrm{d}t.
\end{aligned}$$

Example: rough Bergomi smiles

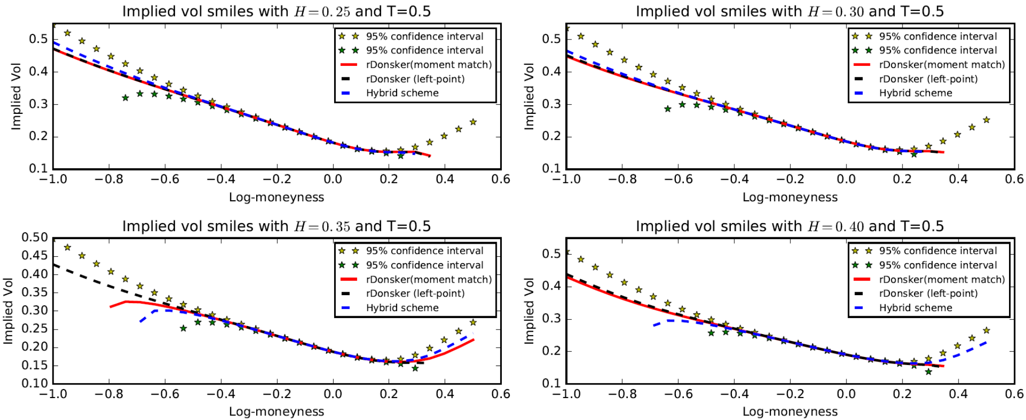
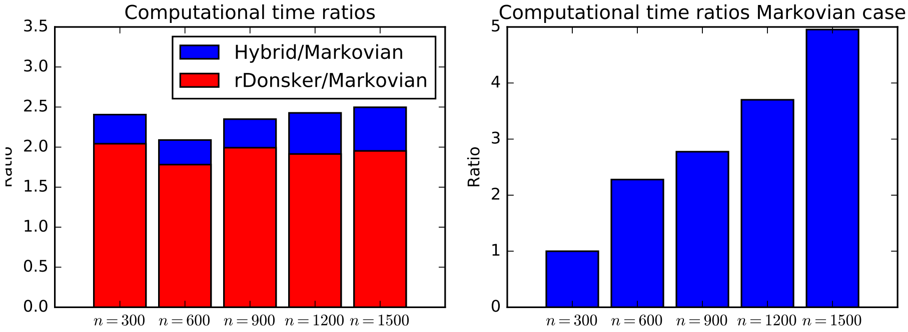


Figure 1: Parameters: $\nu = 1, \rho = -0.7, \xi_0 = 0.04$, $n = 468$ steps

Performance

- rDonsker is $1.25\times$ faster than Hybrid scheme (because we omit the Cholesky bit)



Speedups for rough volatility models along two lines:

Part 1: in pricing of vanilla options based on faster Monte Carlo approximations for a family of rough stochastic volatility models. [H-Jacquier-Muguruza '17])

Part 2: in calibration by means of machine learning techniques (ongoing with A. Muguruza and M. Tomas)

Part 2: Speed-ups on calibration

Part 2: Speed-ups on calibration

- one step away from of-the-shelf optimizers to explore the parameter space more efficiently, limiting the number of function evaluations for calibration. Tests on this with Amir Sani and Aitor Muguruza.
- Main idea: prior to calibration, approximate the implied volatility function

$$\sigma : (\underbrace{\alpha, \beta, \rho, H}_{\text{parameters}}) \times (\tau, k) \mapsto \sigma_\tau(k;\ \alpha, \beta, \rho, H)$$

by a deterministic function, learned by a neural network.
Two parts of the neural network (i) Approximation network (2) Calibration network on top.
See also calibration by neural networks: Recent work of Bayer and Stemper:
Both works rely on the crucial observation of separation the **approximation** and the **calibration** networks.

Approximation by neural networks I

General setup: two parts of the network:

1. Generator: Input (parameters) Output (implied volatilities)
2. Calibrator: Input (implied volatilities) Output (*optimal* parameters).

Both feed-forward neural networks for the generator three hidden layers (1000-800-600)-nodes. Calibrator 1 layer on top.

Approximation by neural networks I

General setup: two parts of the network:

1. Generator (approximation of IV surfaces via NN)

In order to train the network we first need to build a training set (supervised learning).

Approximation by neural networks I

General setup: two parts of the network:

1. Generator (approximation of IV surfaces via NN)

In order to train the network we first need to build a training set (supervised learning).

- For this we can use numerical valuation functions (Bergomi model, Rough Bergomi, Heston, ... Part 1): We generate 20,000 surfaces for each model, using a fixed grid of strikes and tenors.
- Though training time consuming, it can be done offline.
- We sample uniformly points in the parameter set $\theta \in \Theta$, then compute and save $f(\theta)$. Those samples will constitute our training set. We repeat this procedure until we reach enough samples for our surrogate function to be a good approximation.

Approximation by neural networks I

General setup: two parts of the network:

2. Calibrator

In order to train the network we first need to build a training set. Conclusions:

Approximation by neural networks I

General setup: two parts of the network:

2. Calibrator

In order to train the network we first need to build a training set. Conclusions:

- This can be done online, fast (within range of ˜ 1 second already unoptimized)
- Evaluation of parameters now more direct than via Monte Carlo. One minimizes now the distance between the (approximator) surrogate functions $\hat{f}(\theta^*)$ and the volatility surface.

Approximation by neural networks II

We see that after learning, calibrating many parameters is fast

Approximation by neural networks II

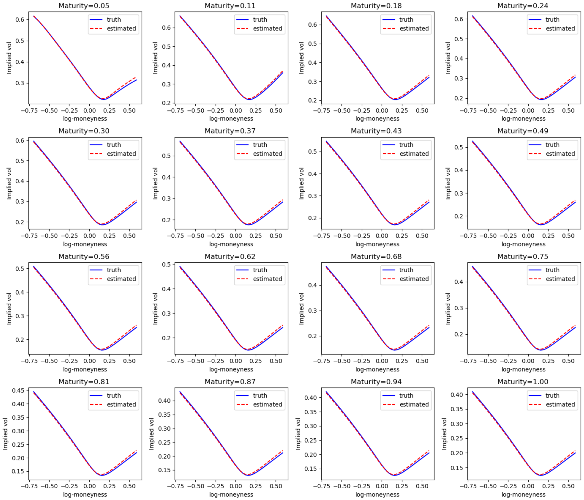
We see that after learning, calibrating many parameters is fast
$\Rightarrow$ approximate several models at the same time.

Approximation by neural networks II

We see that after learning, calibrating many parameters is fast
$\Rightarrow$ approximate several models at the same time.
New learning procedure:

- Train the generator on several models at the same time (here Parameters from Heston and Bergomi parameters) in Monte Carlo experiments as before.
- Calibrate several models at the same time $\Rightarrow$ determine the best-fit model to a given data (flag).
- Controlled experiments: train on both Bergomi and Heston $\Rightarrow$ test on data generated by Heston.

Approximation experiment via NN (Bergomi)



Thank you for your attention!

FCLT for Hölder continuous processes

FCLT for Hölder continuous processes

Define for any $\omega \in \Omega$, $n \geq 1$, $t \in [0, T]$, the approximating sequence

$$W_n(t,\omega) := \frac{1}{\sigma\sqrt{n}} \sum_{k=1}^{j} \xi_k(\omega) + \frac{nt - j}{\sigma\sqrt{n}} \xi_{j+1}(\omega), \quad \text{whenever } t \in \left[\frac{j}{n}, \frac{j+1}{n}\right), \text{ for } j = 0, \dots, n-1.$$

where the family $(\xi_i)_{i\geq 1}$ forms an iid sequence of centered random variables with finite moments of all orders and $\mathbb{E}(\xi_1^2) = \sigma^2 > 0$.

FCLT for Hölder continuous processes

Define for any $\omega \in \Omega$, $n \geq 1$, $t \in [0, T]$, the approximating sequence

$$W_n(t,\omega) := \frac{1}{\sigma\sqrt{n}} \sum_{k=1}^{j} \xi_k(\omega) + \frac{nt-j}{\sigma\sqrt{n}} \xi_{j+1}(\omega), \quad \text{whenever } t \in \left[\frac{j}{n}, \frac{j+1}{n}\right), \text{ for } j = 0, \ldots, n-1.$$

where the family $(\xi_i)_{i\geq 1}$ forms an iid sequence of centered random variables with finite moments of all orders and $\mathbb{E}(\xi_1^2) = \sigma^2 > 0$.

Theorem (Donsker-Lamperti Theorem)

The sequence $(W_n)_{n\geq 1}$ converges weakly to a Brownian motion in $(\mathcal{C}^\alpha([0, T]), \|\cdot\|_\alpha)$ for all $\alpha < \frac{1}{2}$.

Monte-Carlo

Monte-Carlo

Monte-Carlo

The left-point approximation may be modified e.g.

$$\int_0^{\frac{Ti}{n}} g\left(\frac{Ti}{n} - s\right) \mathrm{d}W_s \approx \frac{1}{\sqrt{n\sigma}} \sum_{k=1}^{j-1} g\left(t_k^*\right) \xi_k, \quad j = 0, \ldots, n$$

where t_k^* is chosen optimally to match first and second moments

Monte-Carlo

The left-point approximation may be modified e.g.

$$\int_0^{\frac{Ti}{n}} g\left(\frac{Ti}{n} - s\right) \mathrm{d}W_s \approx \frac{1}{\sqrt{n\sigma}} \sum_{k=1}^{j-1} g\left(t_k^*\right) \xi_k, \quad j = 0, \ldots, n$$

where t_k^* is chosen optimally to match first and second moments , i.e.,

$$g(t_k^*) = \sqrt{n \int_{\frac{T(k-1)}{n}}^{\frac{Tk}{n}} g(t-s)^2 \mathrm{d}s}, \quad k = 1, \ldots, n.$$

Monte-Carlo

The left-point approximation may be modified e.g.

$$\int_0^{\frac{Ti}{n}} g\left(\frac{Ti}{n} - s\right) \mathrm{d}W_s \approx \frac{1}{\sqrt{n}\sigma} \sum_{k=1}^{j-1} g\left(t_k^*\right) \xi_k, \quad j = 0, \dots, n$$

where t_k^* is chosen optimally to match first and second moments , i.e.,

$$g(t_k^*) = \sqrt{n \int_{\frac{T(k-1)}{n}}^{\frac{Tk}{n}} g(t-s)^2 \mathrm{d}s}, \quad k = 1, \dots, n.$$

- This simple trick improves substantially the simulation (specially when α is close to $-1/2$)

Monte-Carlo

The left-point approximation may be modified e.g.

$$\int_0^{\frac{Ti}{n}} g\left(\frac{Ti}{n} - s\right) \mathrm{d}W_s \approx \frac{1}{\sqrt{n\sigma}} \sum_{k=1}^{j-1} g\left(t_k^*\right) \xi_k, \quad j = 0, \dots, n$$

where t_k^* is chosen optimally to match first and second moments , i.e.,

$$g(t_k^*) = \sqrt{n \int_{\frac{T(k-1)}{n}}^{\frac{Tk}{n}} g(t-s)^2 \mathrm{d}s}, \quad k = 1, \dots, n.$$

- This simple trick improves substantially the simulation (specially when α is close to $-1/2$)
- The hybrid scheme also admits this trick