Fractional and Volterra processes in Finance

Challenge 1 - Simulation of Gaussian Volterra processes

Eduardo Abi Jaber

Challenge : Unlock the potential of Gaussian fractional processes and pave the way for more accurate simulations!

PLEASE ENTER YOUR FULL NAMES HERE:

- MEMBER 1
- MEMBER 2
- MEMBER 3

DEADLINE: 12 February before 10 AM to be sent by email to eduardo.abi-jaber@polytechnique.edu

The aim of the challenge is to figure out ways to efficiently simulate the Riemann-Liouville fractional Brownian motion:

$$X_t = \nu \int_0^t K(t,s) dW_s,$$

with

$$K(t,s) = \frac{1}{\Gamma(H+1/2)} (t-s)^{H-1/2} 1_{s<t}$$

and $H < 1/2$.

The covariance kernel of X is given in the following closed form

$$\Sigma_0(s,u) = \frac{\nu^2}{\Gamma(H+1/2)^2}\int_0^{s\wedge u}(s-z)^{H-1/2}(u-z)^{H-1/2}dz \tag{1}$$
$$= \frac{\nu^2}{\Gamma(\alpha)\Gamma(1+\alpha)}\frac{s^\alpha}{u^{1-\alpha}}\ {}_2F_1\left(1, 1-\alpha; 1+\alpha; \frac{s}{u}\right) \tag{2}$$

where $\alpha = H + 1/2$ and ${}_2F_1$ is the Gaussian hypergeometric function.

Guidelines

- Implement and briefly explain and comment the methods. We are interested in low regimes of H. Plot the sample paths on same gaussian increments, to compare paths by paths. You can take $T = 1.$ and $n_{steps} = 300$ time steps uniformly spaced on $[0, T]$. (set $\nu = 1$).
- Two metrics : running time (using "timeit) to simulate one trajectory and MSE error of the paths wrt to the exact path simulated using cholesky method:

$$MSE = \sqrt{\frac{1}{n_{steps}}\sum_{i=1}^{n_{steps}}\left(X_{t_i}^{\text{method}} - X_{t_i}^{\text{cholesky}}\right)^2}$$

Question: Detail the computations that lead to the covariance kernel. Is it valid for $H \leq 1/2$, $H \geq 1/2$? both?

Answer: ....

Several options and suggestions detailed below:

- Cholesky
- Different Euler schemes
- multifactor euler vs exact (cholesky on factors)
- mean reverting process
- time-dependent H
- Hybrid method
- Literature review

1. Exact simulation using Cholesky

Warm up with Cholesky, but aim to set the simulation field on fire with your efficient methods for Gaussian fractional processes.

In [8]:
```
import numpy as np
import matplotlib.pyplot as plt
import scipy.special as sc
from scipy.special import gamma, gammainc
from scipy.integrate import quad
```

The ${}_2F_1$ Gaussian hypergeometric function can be implemented using scipy `sc.hyp2f1`, pay close attention to the parameters, notably final parameter needs to be less than 1?

Question: Check the doc https://docs.scipy.org/doc/scipy/reference/generated/scipy.special.hyp2f1.html and explain.

In [7]:
```
#Write code
```

In [ ]:
```

```

2. Euler Schemes

Explore the unknown and discover the hidden potential of Euler methods through a performance evaluation.

We will consider three (modified) Euler schemes after writing

$$X_{t_i} = X_0 + \sum_{j=1}^{i} \underbrace{\int_{t_{j-1}}^{t_j} K(t_i, s) dW_s}_{Y_j^i}.$$

1. **EULER 1** Naive:

$$X_{t_i} = X_0 + \nu\sqrt{dt} \sum_{j=1}^{i} K(t_i, t_{j-1}) Z_j$$

   with $Z_j \sim \mathcal{N}(0, 1)$ iid.

2. **EULER 2** Write $dW_s \approx Z_j \frac{ds}{\sqrt{dt}}$ so that

$$X_{t_i} = X_0 + \nu \sum_{j=1}^{i} w_j^i Z_j$$

   with

$$w_j^i = \frac{1}{\sqrt{dt}} \int_{t_{j-1}}^{t_j} K(t_i, s) ds = \int_{t_{j-1}}^{t_j} K(t_i, s) ds = \frac{1}{\sqrt{dt}} \frac{1}{\Gamma(H+0.5)(H+0.5)} \left((t_i - t_{j-1})^{H+} \right.$$

3. **EULER 3** Observe that $(Y_1^i, \ldots, Y_i^i)$ is a centered Gaussian vector with independent components such that the std of the j-th component is

$$\tilde{w}_j^i = \sqrt{\int_{t_{j-1}}^{t_j} K(t_i, s)^2 ds} = \frac{1}{\Gamma(H+0.5)} \sqrt{\frac{\left((t_i - t_{j-1})^{2H} - (t_i - t_j)^{2H}\right)}{2H}}$$

so that we use

$$X_{t_i} \approx X_0 + \nu \sum_{j=1}^{i} \tilde{w}_j^i Z_j.$$

Note that the simulation is not exact since

$$\mathbb{E}[Y_j^i Y_{j'}^{i'}] = \int_{t_{j-1}}^{t_j} K(t_i, s) K(t_{i'}, s) ds 1_{j=j'}$$

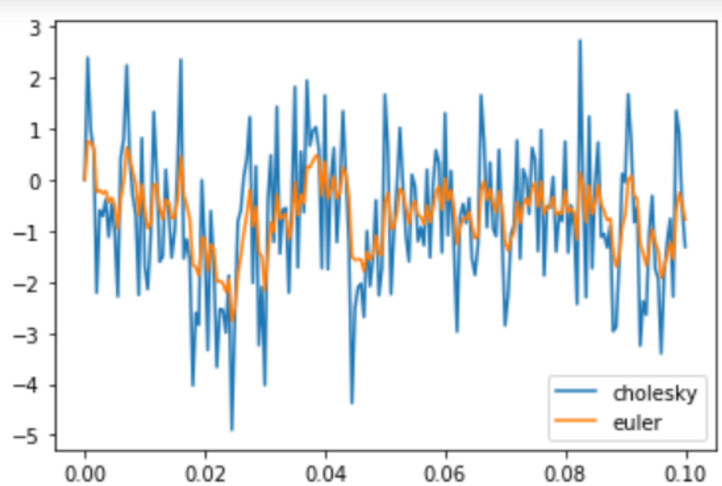
, whereas in the approximation $\mathbb{E}[\tilde{Y}_j^i \tilde{Y}_{j'}^{i'}] = w_j^i w_j^{i'}$. ($\boxed{\text{to double check}}$)

$\textbf{Reference}$: Rambaldi, S., & Pinazza, O. (1994). An accurate fractional Brownian motion generator. Physica A: Statistical Mechanics and its Applications, 208(1), 21-30.

Compare on graphs + MSE that the Naive Euler scheme is way off for small values of $H < 0.05$. Works fine for bigger values of $H > 0.3$... etc...

(!) Please stick to the names **EULER 1**, **EULER 2**, **EULER 3**.

Example of expected graph for sample path:



In [ ]:

In [ ]:

In [ ]:

In [ ]:

3. Multifactor approximations

Embrace the challenge, and push the boundaries of what's possible by making non-standard multifactor approximations work effectively.

Based on

- Abi Jaber, E., & El Euch, O. (2019). Multifactor approximation of rough volatility models. SIAM Journal on Financial Mathematics, 10(2), 309-349. https://arxiv.org/abs/1801.10359
- Abi Jaber, E. (2019). Lifting the Heston model. Quantitative Finance, 19(12), 1995-2013. https://arxiv.org/abs/1810.04868

$$X_t \approx X_0 + \nu \sum_{k=1}^{n} c_k Y_t^k$$

with

$$Y_t^k = \int_0^t e^{-x_k(t-s)} dW_s$$

$$Y_{t_i}^k = e^{-x_k h} Y_t^k + \xi_i^k, \quad \xi_i^k = \int_{t_{i-1}}^{t_i} e^{-x_k(t_i - s)} dW_s$$

with the parametrization:

$$c_i^n = \frac{(r_n^{(1-\alpha)} - 1) r_n^{(\alpha-1)(1+n/2)}}{\Gamma(\alpha)\Gamma(1-\alpha)(1-\alpha)} r_n^{(1-\alpha)i}, \quad x_i^n = \frac{1-\alpha}{2-\alpha} \frac{r_n^{2-\alpha} - 1}{r_n^{1-\alpha} - 1} r_n^{i-1-n/2},$$

where $\alpha := H + 1/2$, with a geometric repartition $\eta_i^n = r_n^i$ for some r_n such that

$$r_n \downarrow 1 \quad \text{and} \quad n \ln r_n \to \infty, \quad \text{as } n \to \infty.$$

We denote by

$$K_n(t) = \sum_{i=1}^{n} c_i e^{-x_i t}.$$

The first step is to determine a good value or r_n for a choice of n, H and T. For this, for a given H, n, T, we can choose r_n to minimize

$$\int_0^T |K_n(t) - K(t)|^2 dt$$

Question: Develop the expression (by developing the square) and show that it admits an explicit expression in terms of incomplete gamma function. Write a minimization function to find r and sanity check with the following table ($H = 0.1$, $T = 0.5$)

n	r_n	norm_n^2
4	50.5458	0.3699
10	18.0548	0.1125
20	8.8750	0.0325
40	4.4737	0.0076
200	1.6946	1.1166e-04

Answer:....

In [ ]:

Now that we know how to determin r.

3.1 Multifactor with Euler methods on factors

We will consider several Euler-type approximations for factors:

1. **Factor-Euler 1** :

$$\xi_i^k \approx e^{-x_k dt} \sqrt{dt} Z_i$$

2. **Factor-Euler 2**: writing $dW_s = Z_i ds / \sqrt{dt}$

$$\xi_i^k \approx \frac{1}{\sqrt{dt}} \int_{t_{i-1}}^{t_i} e^{-x_k(t_i - s)} ds = \frac{1}{\sqrt{dt}} \frac{1 - e^{-x_k dt}}{x_k}$$

3. **Factor-Euler 3** : using that ξ_i^k is gaussian with variance $\frac{1-e^{-2x_k h}}{2x_k}$, so that

$$\xi_i^k \approx \sqrt{\frac{1-e^{-2x_k h}}{2x_k}} Z_i$$

4. **Factor-Euler 4** :: implicit scheme as in lifting heston paper in the appendix
5. **Factor-Euler 5** : modified variance:

$$X_{t_{i+1}} = X_0 + \nu \sum_k c_k e^{-x_k dt} Y_{t_i}^k + \nu \int_{t_i}^{t_{i+1}} K_n(t_{i+1}, s) dW_s$$

approximate second term by variance of original kernel K.

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

3.2 Multifactor exact simulation with Cholesky

$$X_t \approx X_0 + \nu \sum_{k=1}^n c_k Y_t^k$$

with

$$Y_t^k = \int_0^t e^{-x_k(t-s)} dW_s$$

$$Y_{t_i}^k = e^{-x_k h} Y_t^k + \xi_i^k, \quad \xi_i^k = \int_{t_{i-1}}^{t_i} e^{-x_k(t_i - s)} dW_s$$

We will use exact approximation using Cholseky to simulate $(\xi_i^1, \ldots, \xi_i^n)^\top \sim \mathcal{N}(0, \Sigma)$ with

$$\Sigma_{kl} = \int_{t_i}^{t_{i+1}} e^{-(x_k + x_l)(t_{i+1} - s)} ds = \frac{1 - e^{-(x_k + x_l)dt}}{x_k + x_l}$$

For each t_i generate $Z_i = (Z_i^1, \ldots, Z_i^n)^\top$ independant standard Gaussian and set

$$\xi_i = L Z_i \quad \text{with } LL^\top \Sigma.$$

Set

$$E_{dt} = \exp(-\text{diag}(x_1, \ldots, x_n)dt)$$

Then,

$$X_{t_{i+1}} = X_0 + \nu * c^\top E_{dt} Y_{t_i} + \nu * c^\top L Z_i = X_0 + \nu * c^\top E_{dt} Y_{t_i} + \nu * \sqrt{c^\top \Sigma c} U_i$$

with

$$U_i := \frac{c^\top L Z_i}{\sqrt{c^\top \Sigma c}} \sim \mathcal{N}(0, 1)$$

$\mathbf{Q:}$ What is the difference and main advantage of such method compared to Cholesky of part 1?

In [ ]:

In [ ]:

In [ ]:

In [ ]:

4. Going beyond

Simulation is not a one-size-fits-all solution. Break free from the mold and discover new methods to solve the problem at hand.

4.0 Literature review

Do a Literature review and try to find/implement other methods? and compare with above

4.1 Mean reverting

Figure out how to adapt/compare all above methods to the mean reverting process

$$X_t = X_0 - \int_0^t K(t, s)\lambda X_s ds + \int_0^t \nu K(t, s) dW_s,$$

with K the fractional kernel as above and $\lambda > 0$.

4.2 More H

- Extend the above methods to time-dependent H with $H(t) \in (0, 1/2)$.
- Extend the above methods to the case $H \geq 1/2$.

4.3 Hybrid method

Implement and compare the Hybrid method https://arxiv.org/pdf/1507.03004.pdf

In [ ]:

In [ ]:

In [ ]: