From elephant to goldfish (and back): memory in stochastic Volterra processes

Msik Ahmed Amine & Hamza Iqbi

Master Probabilités et Finance
Projet Informatique

Contents

Abstract

This paper examines the following article "From elephant to goldfish (and back): memory in stochastic Volterra processes" by O. Bonesini, G. Callegaro, M. Grasselli, and G. Pagès. It presents a mathematical exposition of the model, followed by a numerical analysis of its robustness and comparison with alternative methods. Implementation will be in Python, focusing on the model's discretization scheme. This study aims to deepen understanding of non-Markovian volatility dynamics and contribute practical insights for financial modeling and analysis.

1 Mathematical presentation of the model and the technique employed

1.1 Mathematical notations

Given a square-integrable stochastic process $Y \in \mathcal{L}^2$, define the family of random variables

$$(K * Y)_t := \int_0^t K(t-s)Y_s\, ds, \quad 0 \le t \le T, \tag{1}$$

with K a convolution Kernel such that the above integral exists finite p-a.s. Moreover, we introduce the following stochastic Itô integral of convolution type:

$$(K \overset{W}{*} Y)_t := \int_0^t K(t-s)Y_s\, dW_s, \quad 0 \le t \le T, \tag{2}$$

such that the above integral is well defined.

Definition 1.1 A pseudo-inverse co-kernel with respect to K is $\tilde{K}$, with $\rho \ge 0$, is a continuous kernel $\tilde{K} : \mathbb{R}_+ \to \mathbb{R}_+$, satisfying, for every $t \ge 0$,

$$(K * \tilde{K}t) = (\tilde{K} * K)(t) = e^{-\rho t}$$

Defintion 1.2 Given $K \in L^1_{loc}(\mathbb{R}_+)$, a function r belonging to $L^1_{loc}(\mathbb{R}_+)$ is called functional resolvent of the first kind of K if

$$(K * r)(t) = (r * K)(t) = 1,$$

for all $t \in \mathbb{R}_+$.

1.2 From path-dependent Volterra to standard SDEs

The recent debate about the inclusion of memory in the volatility process have also led some researchers to use Hurst parameter less than $\frac{1}{2}$ to capture the main stylized facts of historical volatility time series and fit option smiles observed in the SPX market. These works fall in the category of rough volatility.

One of the first results of this paper implies the transformation of a non-markovian, thus memory involving, Volterra stochastic process into a Markovian (depends only on the state of the process and not the historical values) process thanks to convolution kernels. This transformation is also reversible. The class of volterra SDEs this paper focuses on is

$$X_t = \xi^0 + \int_0^t K(t-s)b\left(s, (\tilde{K} * X)_s\right) ds + \int_0^t K(t-s)\sigma\left(s, (\tilde{K} * X)_s\right) dW_s,$$

where K is the original kernel involved in the non-Markovian process and $\tilde{K}$ is its ρ-pseudo-inverse co-kernel. The convolution $(K * \tilde{K})(t)$ or $(\tilde{K} * K)(t)$ represents the intertwining of these kernels resulting in an exponential decay factor $e^{-\rho t}$, which is essential in transforming the non-Markovian process to a Markovian process and vice versa. In the case where the kernels K and $\tilde{K}$ have the following properties:

- The kernel K satisfies an integrability condition for some $C_K > 0$ and $\delta \in (0, T)$, ensuring that its 2β-norm is finite over the interval $[0, T]$.
- The Borel functions b and σ on $[0, T] \times \mathbb{R}$ are Lipschitz in x uniformly in time and bounded at $x = 0$, satisfying the Lipschitz conditions in x with constants $[b]_{\text{Lip},x}$ and $[\sigma]_{\text{Lip},x}$, and the supremum of $b(t, 0)$ and $\sigma(t, 0)$ is finite.
- The initial condition ξ^0 is square-integrable, i.e., $\xi^0 \in L^2(\Omega)$.

Theorem 3.4 states that the stochastic Volterra equation has a pathwise continuous strong solution X_t given by

$$X_t = e^{-\rho t} \frac{d}{dt} \left[(e^{\rho \cdot} K) * \xi \right]_t, \quad t \in [0, T],$$

where ξ_t is defined as the unique (pathwise continuous) strong solution to the diffusion SDE

$$\xi_t = \xi^0 e^{\rho t} (\tilde{K} * \xi)_t + \int_0^t e^{\rho s} b(s, e^{-\rho s} \xi_s) ds + \int_0^t e^{\rho s} \sigma(s, e^{-\rho s} \xi_s) dW_s, \quad \xi_0 = 0,$$

with $\xi_0 = \xi^0$, representing the Markovian "memory process" within the dynamics of the non-Markovian process X_t. Using the convolution of X with $\tilde{K}$ and the property $(K * \tilde{K})(t) = e^{-\rho t}$, the non-Markovian process X can intuitively be transformed into the Markovian memory process ξ, as shown by the SDE $\xi_t = e^{\rho t} (\tilde{K} * X)_t$, which simplifies the complex dynamics into a more familiar diffusion process.
This basically refers to the title of the paper: we go from an elephant referring to the big task of taking into account all the historical values to a goldfish since it's more simpler to only use the state of a process.

1.3 Euler schemes

1.3.1 Euler scheme for ξ

Based on the previous work done, the paper introduces a continuous-time Euler scheme that has a strong order of convergence of $\frac{1}{2}$. More precisely, the following section describes a strong rate of convergence in $L^p(\mathbb{P})$ norm of order $\frac{1}{2} \wedge \gamma$, with γ being the Hölder-Lipschitz regularity of the coefficients b and σ for the continuous-time Euler scheme for ξ. This part also introduces a continuous-time Euler scheme for the elephant process with the same order of convergence.

Let's define

$$\bar{\xi}_t = \xi_0 \tilde{\phi}(t) + \int_0^t b(s, \xi_{\underline{s}}) ds + \int_0^t \sigma(s, \xi_{\underline{s}}) dW_s,$$

where $\underline{s} = t_k = \frac{kT}{n}$ for $s \in \left[\frac{kT}{n}, \frac{(k+1)T}{n} \right]$, where $k \in \{0, \ldots, n-1\}$.
where, $\tilde{\phi}(t) = \left(\tilde{K} * 1 \right)(t)$ and the time step $h = \frac{T}{n}$, where $n \in \mathbb{N}_{>0}$, is defined via

the pseudo-SDE with frozen coefficients. Let's assume now that coefficients b and σ satisfy the Hölder-Lipschitz time-space condition with $\gamma \in (0,1]$, $\tilde{\phi}$ is a non decreasing, concave and $\tilde{\phi}(0) = 0$ and $\xi^0 \in L^p(\mathbb{P})$, theorem 4.1 states that

$$\left\| \sup_{t\in[0,T]} |\xi_t - \overline{\xi}_t| \right\|_p \leq C_{b,\sigma,T,p} h^{1/2\wedge\gamma} \left(1 + \|\xi_0\|_p\right).$$

How to use this result? This shows that as n goes to infinty $\sup_{t\in[0,T]} |\xi_t - \overline{\xi}_t|$ goes to 0 in every $L^p(\mathbb{P})$. Also, as a consequence, γ rules the rate of convergence as soon as it's smaller than $\frac{1}{2}$. The proof of this result specified the two following cases: p greater than 2 and p smaller. In the first case, the inequalities held true based on Burkholder-Davis-Gundy (BDG), generalized Minkowski inequalities and a variant of Gronwall's lemma [30, Lemma 7.3, p.327] and the following decomposition:

$$\overline{\xi}_s - \overline{\xi}_{\underline{s}} = \xi_s - \xi_{\underline{s}} + \overline{\xi}_s - \overline{\xi}_{\underline{s}}, \quad s \in [0,T].$$

For the cas $0 < p < 2$, this was a consequence of consequence of Blagoven˘s˘cenkii-Freidlin Theorem [35, Theorem V.13.1].

1.3.2 Euler scheme for X

Two propositions of schemes were introduced in this section of the paper. The first one

$$\overline{X}_t := \xi^0 + \int_0^t K(t-s) b(\underline{s}, \overline{\xi_{\underline{s}}}) ds + \int_0^t K(t-s)\sigma(\underline{s}, \overline{\xi_{\underline{s}}}) dW_s,$$

so that at the discretization points we get $\overline{X}_0 = \xi^0$ and

$$\overline{X}_{t_{k+1}} = \xi^0 + \sum_{\ell=0}^{k} \left(\int_{t_\ell}^{t_{\ell+1}} K(t_{k+1}-s) b(t_\ell, \overline{\xi}_{t_\ell}) ds + \int_{t_\ell}^{t_{\ell+1}} K(t_{k+1}-s)\sigma(t_\ell, \overline{\xi}_{t_\ell}) dW_s \right).$$

The second proposition replaces in the kernel function s by $\underline{s}$ so that in the previous sum we can factorize by $K(t_{k+1} - t_l)$ and the integrals become easy to compute as:

$$\overline{\tilde{X}}_{t_{k+1}} = \xi^0 + \sum_{\ell=0}^{k} K(t_{k+1}-t_\ell) \left(\frac{1}{n} b(t_\ell, \tilde{\xi}_{t_\ell}) + \sigma(t_\ell, \tilde{\xi}_{t_\ell})(W_{t_{\ell+1}} - W_{t_\ell}) \right), \quad \overline{\tilde{X}}_0 = \xi^0.$$

What to deduce? If the work one has to do only uses the final value of the process $\overline{X}^n_T$ (typically for European options) we can have this quantity with a complexity of O(n) while we would have needed $O(n^2)$ typically for standard volterra SDEs. On the other hand if we need all the path of the process (for path dependant options for example), the complexity remains the same.

Theorem 4.4 in the paper also gives some insights about the convergence rate of the first scheme introduced above: $\overline{X}$ converges towards X in $L^p(\mathbb{P})$ at a strong rate order of $\frac{1}{2} \wedge \gamma$ with $\gamma \in (0,1)$. γ is the same as introduced for the Euler scheme of ξ in the Hölder-Lipschitz assumption. Let's note that there also some additional assumptions regarding the Kernel functions . In fact in the proof of this convergence rate, the

authors of the paper are confronted to the following quantity after applying Minkowski inequality and other transformations to $\left\|X_t - \tilde{X}_t\right\|_p$:

$$\int_0^t K(t-s) \left\|\xi_s - \tilde{\xi}_s\right\|_p ds.$$

To use Corollary 4.2 (see below) with $r = \frac{2\beta}{\beta-1}$, assume exists $\beta > 1$ such that $\mathcal{K}_\beta$ defined as= $\int_0^T K(t)^{2\beta} dt$ is finite and using Hölder inequality the authors retrieve the convergence rate in the case where $p < 2$.

Corollary 4.2 Under the assumptions of Theorem 4.1, for every $r > 0$ and $p > 0$ there exists a real constant $C_{b,\sigma,T,r,p} > 0$, such that

$$\left(\int_0^T \|\xi_s - \tilde{\xi}_s\|_p \, dt\right)^{1/r} \leq C_{b,\sigma,T,r,p} \left(1 + \|\xi_0\|_p\right) \left(\left(\frac{T}{n}\right)^{\gamma \frac{r-1}{r}} + \left(\int_0^T (\phi(t) - \tilde{\phi}(t))^r \, dt\right)^{1/r}\right).$$

1.4 The case of fractional kernels: a relevant example

This section in the paper applies the previous results to specific kernel function of rough volatility models, typically for $t \in [0,T]$ define $K(t) = K_{1,\alpha,0}(t) := K_{1,\alpha}(t) = \frac{t^{\alpha-1}}{\Gamma(\alpha)}$ where we choose as a Hurst parameter $H = \alpha + \frac{1}{2}$ and $H \in (0, \frac{1}{2})$ so that we can fall in the case of rough volatility. These kernel functions indeed check the condition $\mathcal{K}_\beta$ for every $\beta \in \left(1, \frac{1}{2(1-\alpha)}\right)$.

1.4.1 Existence of X

Lemma 5.1. Let $\xi^0 \in L^p(\Omega)$, with $p \geq 2$. If $\sup_{t\in[0,T]} \|\xi_t\|_p < +\infty$, then there exists a unique $(\mathcal{F}_t)_{0\leq t\leq T}$-adapted process X such that, for almost all $t \geq 0$,

$$X_t = \xi^0 + (K \overset{W}{*} Y)_t, \quad \text{P-a.s.}$$

Moreover, if there exists $p > \max\left\{\frac{1}{\alpha-1}, 2\right\}$ such that $\sup_{t\in[0,T]} \|\xi_t\|_p < +\infty$, then the stochastic process X has a path-wise continuous modification and there exists a modification with locally α-Hölder continuous trajectories for any $\alpha \in \left(0, \alpha - \frac{1}{2} - \frac{1}{p}\right)$.

Note that this lemma also shows that p should me greater than $\max(\frac{1}{H}, 2)$ in order to apply Kolmogorov criterion and gives the exmaple of $H = 0.1$, typical for rough volatilities model, we should have a p-integrablity in minimum of L^{10}.

Furthermore, using the fact that $\tilde{K}_{1,\alpha} = K_{1,1-\alpha}$ and noticing that it satisfies the conditions of Theorem 3.4 (see page 3)

$$X_t = \xi^0 + \int_0^t \frac{(t-s)^{\alpha-1}}{\Gamma(\alpha)} b(s, \xi^{1-\alpha}(X)_s) ds + \int_0^t \frac{(t-s)^{\alpha-1}}{\Gamma(\alpha)} \sigma(s, \xi^{1-\alpha}(X)_s) dW_s,$$

and

$$\xi_t = \xi^0 \frac{t^{1-\alpha}}{\Gamma(2-\alpha)} + \int_0^t b(s, \xi_s) ds + \int_0^t \sigma(s, \xi_s) dW_s.$$

Note that we retrieve these formulas as ρ is supposed to be equal to 0 and the first term of the expression of ξ comes from direct computation of $(K_{1,1-\alpha} * 1)(t)$.

At the end, the document discusses Euler schemes in the case of fractional kernel $K_{1,\alpha}$ for $1/2 < \alpha < 1$ so that volatility is rough (Hurst parameter lower than $\frac{1}{2}$.

1.4.2 Improvement made on the convergence rate

The improvement made is that since $\tilde{\phi}(t) = \frac{1-\alpha}{\Gamma(2-\alpha)}$ is $(1-\alpha)-$Hölder we get a convergence rate of $O(\frac{T}{n})^{(1-\alpha)\wedge\gamma}$ (not good in rough volatility cases since α is close to 0.9 typically). The improvement made is shown thanks to theorem 5.3 that improves the convergence rate of the Euler scheme of X.

Theorem 5.3. Let $K = K_{1,\alpha}$, with $\alpha \in \left(\frac{1}{2}, 1\right)$, and let b and σ satisfy the time-space Hölder-Lipschitz condition in Equation (4.3), with Hölder parameter $\gamma \in (0, 1)$. Then, for every $p > 0$,

$$\sup_{t\in[0,T]} \|X_t - \widetilde{X}_t\|_p \leq C_{\alpha,\gamma,T,p}(1 + \|\xi^0\|_p) \left(\frac{T}{n}\right)^{\gamma\wedge\frac{1}{2}},$$

for some real constant $C_{\alpha,\gamma,T,p} > 0$.

2 Numerical study of the proposed method

To illustrate the theoretical framework above, a numerical simulation is poposed. We focus on the fractional kernel

$$K(t) = \frac{t^{\alpha-1}}{\Gamma(\alpha}, t \in [0, T].$$

This choice is due to the recent interest on rough (non-Markovian) volatility. This is the case when the Hurst parameter $H \in (0, \frac{1}{2})$, which means $\alpha \in (\frac{1}{2}, 1)$.

2.1 An inspiring model

We recall the quadratic rough Heston model, where the dynamics of the traded asset are given by :

$$d\mathcal{S}_t = \mathcal{S}_t\sqrt{\mathcal{V}_t}dB_t, \qquad \mathcal{S}_0 = s_0 > 0$$

,

with B a standard Brownian motion and the variance process $\mathcal{V}$ is defined as follows:

$$\begin{cases} \mathcal{V}_t = a(\mathcal{Z}_t - b)^2 + c \\ \mathcal{Z}_t = \mathcal{Z}_0 + \int_0^t (t-s)^{\alpha-1}\frac{\lambda}{\Gamma(\alpha)}(\theta_0(s) - \mathcal{Z}_s)\, ds + \int_0^t (t-s)^{\alpha-1}\frac{\eta}{\Gamma(\alpha)}\sqrt{\mathcal{V}_s}dB_s \end{cases}$$

with $a > 0, b \geq 0, c \geq 0$ and $\alpha \in (\frac{1}{2}, 1), \lambda > 0, \eta > 0$ and θ_0 a deterministic function of time.

Let us stress that since $z \mapsto \sqrt{a(z-b)^2 + c}$ is convex and Lipschitz so that the above stochastic Volterra equation has a unique strong solution.

2.2 An elephant and a goldfish

We introduce the SDE for the elephant process Z

$$Z_t = \xi^0 + \int_0^t \frac{(t-s)^{H-\frac{1}{2}}}{\Gamma(H+\frac{1}{2})}(\mu - \lambda Y_s)ds + \eta \int_0^t \frac{(t-s)^{H-\frac{1}{2}}}{\Gamma(H+\frac{1}{2})}\sigma(Y_s)dW_s, \qquad t \in [0,T]$$

where we have defined

$$\sigma(y) = \sqrt{a(y-b)^2 + c}, \qquad a > 0, b \geq 0, c \geq 0.$$

The goldfish process is such that

$$Y_t = \xi^0 \frac{t^{\frac{1}{2}-H}}{\Gamma(\frac{3}{2}-H)} + \int_0^t (\mu - \lambda Y_s)ds + \eta \int_0^t \sigma(Y_s)dW_s, \qquad t \in [0,T]$$

and

$$V_t = a(Z_t - b)^2 + c, \qquad t \in [0,T].$$

The Markovian processes Y (and also V) can be simulated with a high degree of efficiency. It is anticipated that the early behavior of these processes will be characterized by a noticeable initial surge in memory, which is a direct result of the initial conditions set at time t = 0. In contrast, the elephant process, denoted as Z, is expected to exhibit trajectories that are comparatively more erratic. This behavior is consistent with its inherent non-Markovian properties, indicating a deviation from the predictability typically associated with Markovian processes.

Lemma 5.4. Given the stochastic processes Z and Y defined above, we have

$$\mathbb{E}[Y_t] \to \frac{\mu}{\lambda} \quad \text{and} \quad \mathbb{E}[Z_t] \to 0, \quad \text{as} \quad t \to +\infty.$$

This is the asymptotic behaviour of the average of Y and Z that we expect to observe in the simulations yet to come.

2.3 Some simulations

To simulate Y, we use a discrete Euler scheme:

$$\overline{Y}_{t_{k+1}} - \overline{Y}_{t_k} = \xi^0 \left(\frac{t_{k+1}^{\frac{1}{2}-H}}{\Gamma(\frac{3}{2}-H)} - \frac{t_k^{\frac{1}{2}-H}}{\Gamma(\frac{3}{2}-H)} \right) + h(\mu - \lambda Y_{t_k}) + \eta\sigma(Y_{t_k})\Delta W_{t_{k+1}}$$

with $t_k = \frac{kT}{N}, k = 1, ..., N$ and $\Delta W_{t_{k+1}} = W_{t_{k+1}} - W_{t_k}$.
The simulation of Z is more laborious:

- We can use a discretization of Euler scheme that freezes the kernel:

$$\overline{Z}_{t_{k+1}} = \xi^0 + \sum_{\ell=0}^{k} K(t_{k+1} - t_\ell) \left(\frac{T}{N}(\mu - \lambda \overline{Y}_{t_\ell}) + \sigma(\overline{Y}_{t_\ell})\Delta W_{t_\ell+1} \right)$$

- Otherwise, the -regular- Euler scheme discretization reads:

$$\tilde{Z}_{t_{k+1}} = \xi^0 + \sum_{\ell=0}^{k} \left(\int_{t_\ell}^{t_{\ell+1}} K(t_{k+1}-s)(\mu - \lambda \overline{Y}_{t_\ell}) + \int_{t_\ell}^{t_{\ell+1}} K(t_{k+1}-s)\sigma(\overline{Y}_{t_\ell}) dW_s \right)$$

Thus, we need to simulate the vectors

$$G^{n,\ell} = \left[\int_{t_{\ell-1}}^{t_\ell} (t_k - u)^\alpha \frac{dW_u}{\Gamma(\alpha+1)} \right]_{k=\ell,\dots,n}, \quad \ell = 1, \dots, n$$

.

The covariance matrix of these vectors is:

$$\Sigma^{n,\ell} = \frac{1}{\Gamma(\alpha+1)} \left[\int_{t_\ell - 1}^{t_\ell} (t_k - u)^\alpha (t_{k'} - u)^\alpha du \right]_{\ell \le k,k' \le n} \tag{3}$$

$$= \frac{1}{\Gamma(\alpha+1)} \left(\frac{T}{n} \right)^{2\alpha+1} [C_{k-\ell,k'-\ell}]_{\ell \le k,k' \le n}, \ell = 1, ..., n, \tag{4}$$

with the infinite symmetric matrix C is :

$$C = \left[\int_0^1 ((i+v)(j+v))^\alpha dv \right]_{i,j \ge 0}$$

.

We will use LDL^t decomposition instead of Cholesky to simulate the Gaussian vectors in question.

3 Simulation results

In this section, we simulate the trajectories of the processes Z, Y and V. We set the parameters

$$a = 0.384, b = 0.095, c = 0.0025$$

3.1 Freezing the kernel

We simulate the processes over the time window $[0, T]$ with $T = 5$ on the grid $t_k = \frac{kT}{N}$.

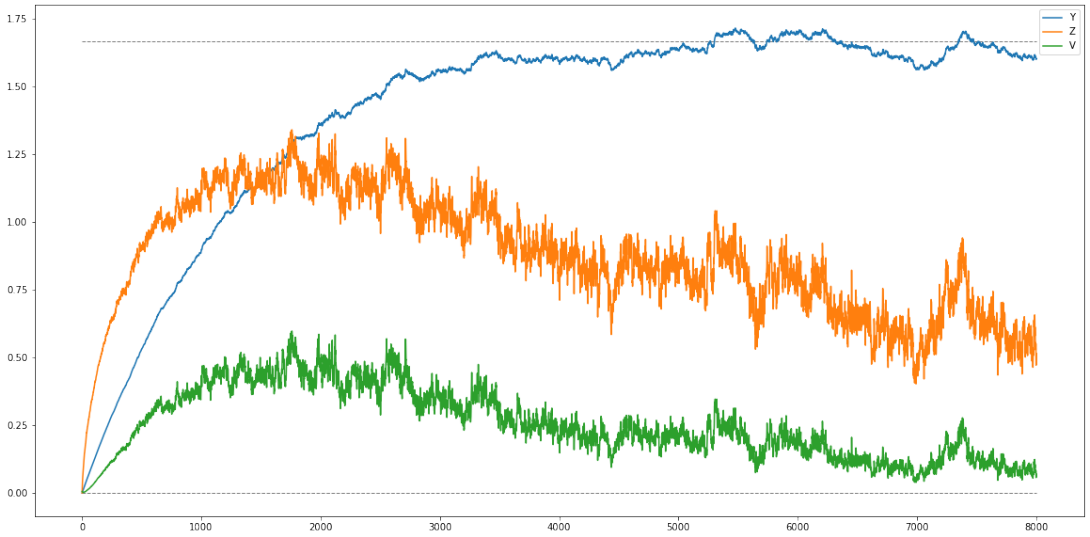

Figure 1: $\xi_0 = 0$, $\mu = 2$, $\lambda = 1.2$, $\eta = 0.1$

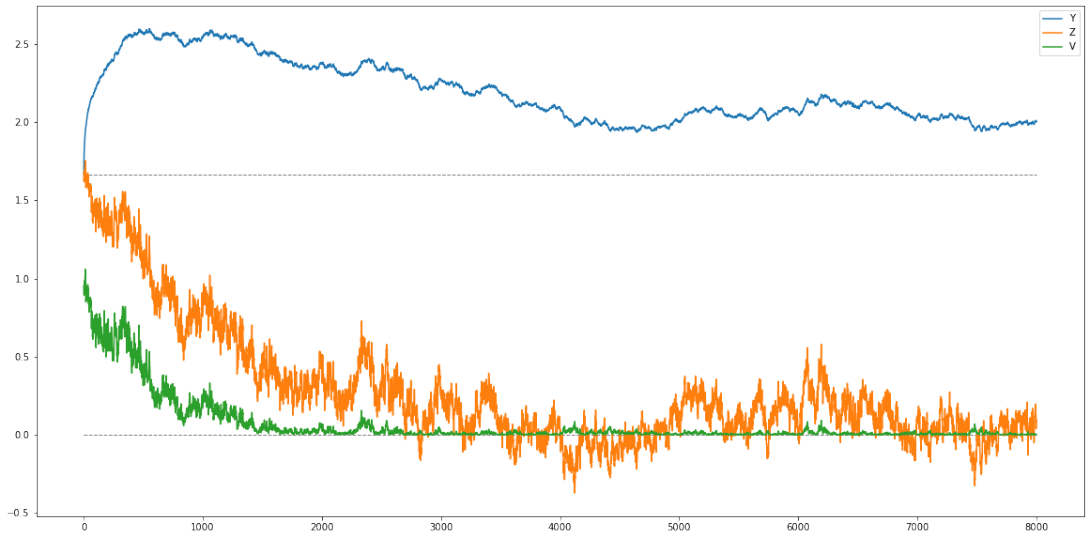

Figure 2: $\xi_0 = \frac{\mu}{\lambda}$, $\mu = 2$, $\lambda = 20$, $\eta = 0.01$

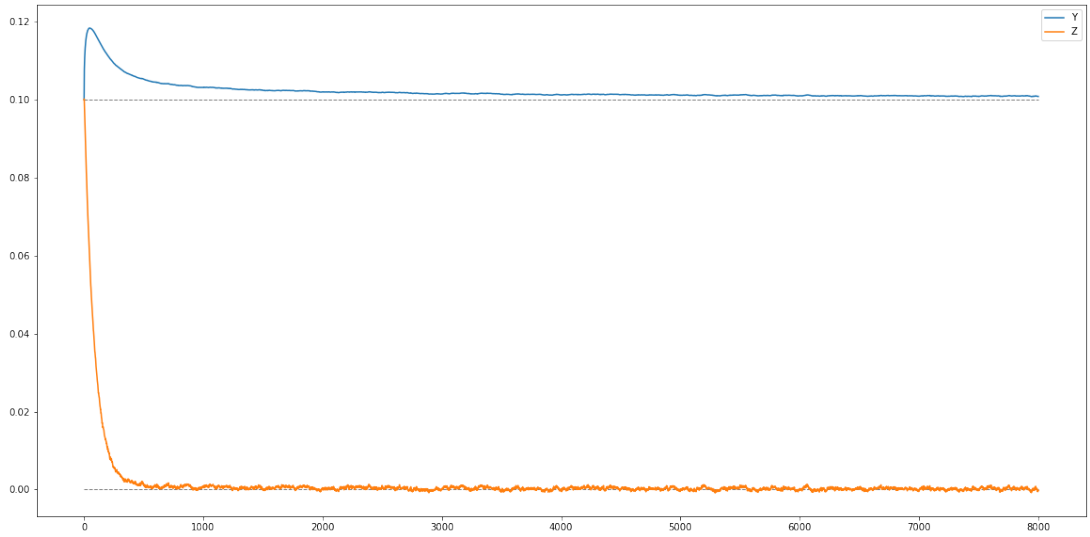

Figure 3: $\xi_0 = \frac{\mu}{\lambda}$, $\mu = 2$, $\lambda = 1.2$, $\eta = 0.1$

These simulations with different sets of parameters exhibit the theoretically expected behaviour asymptotically and at $t = 0^+$:

- The trajectories of the process Y are less rough than those of Z and V.
- We can see, visually, the convergence $\mathbb{E}[Y_t] \to \frac{\mu}{\lambda}$ and $\mathbb{E}[Z_t] \to 0$, as $t \to +\infty$.
- When $\xi_0 = 0$, the burst of memory is not observed because the term $\xi_0 \frac{t^{\frac{1}{2}}}{\Gamma(\frac{3}{2}-H)}$ vanishes in the dynamics of Y.
- When $\xi_0 \neq 0$, an initial abrupt upward shift is observed in the trajectory of Y, as seen in Figures 2 and 3. This is attributed to the presence of the initial condition, interpreted as a burst of memory enabling Y to retain a record of events preceding the starting point.

3.2 Straightforward approach

This approach is time-consuming (and memory-consuming): we were unable to perform simulations with $N = 8000$, but rather with $N = 1000$.

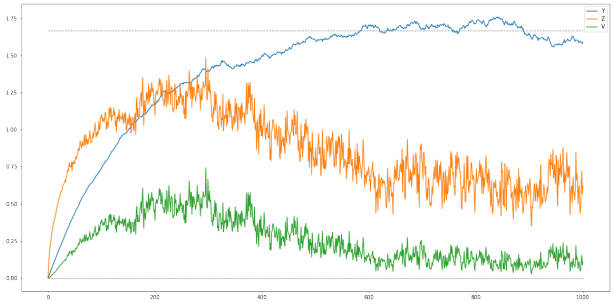

Figure 4: $\xi_0 = 0$, $\mu = 2$, $\lambda = 1.2$, $\eta = 0.1$

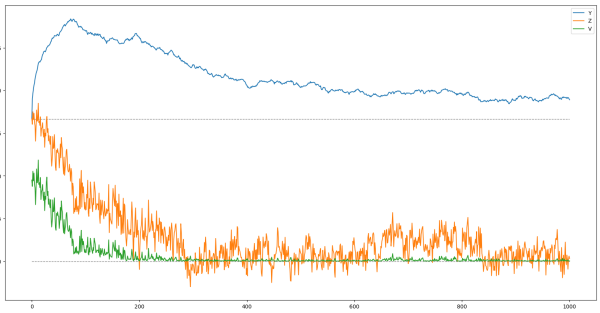

Figure 5: $\xi_0 = \frac{\mu}{\lambda}$, $\mu = 2$, $\lambda = 20$, $\eta = 0.01$

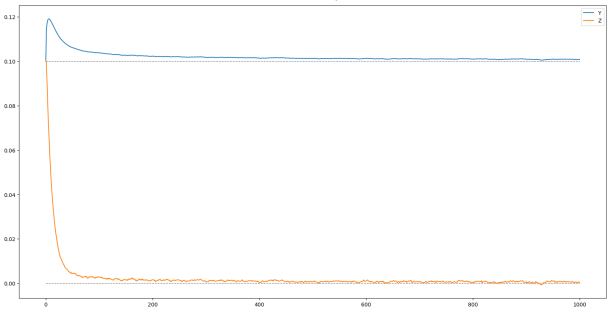

Figure 6: $\xi_0 = \frac{\mu}{\lambda}$, $\mu = 2$, $\lambda = 1.2$, $\eta = 0.1$

The same conclusions about the previous simulations hold for these ones. Nevertheless, one must stress on the fact that this second approach is (in general) time consuming.

These are the execution times on CPU:

	Simulation A)	Simulation B)	Simulaiton C)
Approach 1 with $N = 8000$	$4m$ $50.7s$	$4m$ $55.1s$	$5m$ $5.1s$
Approach 2 with $N = 1000$	$19m$ $16.5s$	$18m$ $41.7s$	$4.7s$

Table 1: A comparison of execution time between the two approaches for all three simulations

4 Choice of numerical implementation

Regarding the choice of numerical implementation we have chosen Python over C++ for several reasons. First, we were more comfortable using Python since we had more experience and projects where we had to implement Python code. Then, the fact that we don't have many types of objects and the use of the classical features that make C++ interesting (polymorphism, inheritance and adaptability) made us think that time of execution would have been the only difference between a C++ code and Python code. C++ code is compiled, which means it is transformed into machine code that can be executed directly by the computer's processor. In a sense it's more understandable for

a machine so that it executes faster the code.

Facing the dilema between time execution and our skillset we have decided to code in Python especially because we were intending to use parallelisation techniques to make it even faster and using Professor Abbas-Turki class (where we have coded in python) to make our code faster.

Finally, our knowledge and experience with python libraries was also a reason why we have chosen to code in Python.

5 Conclusion

In conclusion, this paper successfully explores the transformation of a non-Markovian process to a Markovian one, through the utilization of convolution kernels, provides significant insights into memory effect under the rough volatility framework. The numerical implementation, focused on the Euler schemes, proved the efficiency of such modelisations. This should encourage the development of researches in line with rough volatility.

References

[1] O. BONESINI, G. CALLEGARO, M. GRASSELLI, AND G. PAGÈS, *From elephant to goldfih (and back): memory in stochastic Volterra pocesses*, arXiv:2306.02708.