INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

UNIVERSITÀ DEGLI STUDI DI PADOVA

Master 1 internship report
Volterra processes and memory : a numerical approach

EDGAR NEBOIT
Tutors : GIORGIA CALLEGARO and MARTINO GRASSELLI

Summer 2025

Abstract

The internship started on the 3rd of June and finished on the 31st of August. It took place at the Mathematics Department of Padova, under the supervision of Giorgia Callegaro and Martino Grasselli. The main goal of the internship was to code an efficient algorithm to numerically implement an Euler-type scheme for a new stochastic model designed to address non-Markovian processes.

This work is related to the paper [10], and the primary objective was to implement the Euler-type scheme presented in Section 4. Once the algorithm was optimized and correctly implemented, I conducted numerical Monte Carlo tests to study the strong convergence order of the scheme, which achieved an order of 1/2 in a special case (fractional kernel case).

I also performed numerical experiments on two other schemes and compared the results among these three schemes. This report concludes with some experiments regarding the Hölder regularity of our process.

Contents

1 Introduction

The main object is linked to the mathematical formulation of time dependency and, more precisely, to non-Markovian evolution. This subject is an active research field among many others in probability theory and has numerous applications, such as in quantum systems: see, for instance, [11] or [30].

One way to mathematically model these problems is through Stochastic Volterra Equations (SVEs). Such formulations arise in various fields, including biology, physics, and mathematical finance; see [1]. Despite numerous attempts over the years, their simulation still poses significant challenges. In particular, the rate of (strong and weak) convergence rapidly deteriorates with the regularity of the paths. For a thorough exposition of the current state of the art and contributions to the topic of SVEs simulation, we refer the interested reader to [3].

The recent work of O. Bonesini, G. Callegaro, M. Grasselli, and G. Pagès in [10] introduces a new framework. This framework exploits the possibility of transforming, via a convolution kernel, a Volterra path-dependent (non-Markovian) stochastic process into a standard (Markovian) diffusion process without memory. This novel approach achieves a remarkable strong convergence order of 1/2 for an Euler-type scheme in the case where the kernel is fractional of order $H \in]0, 1/2[$. The most significant aspect is that this convergence order is independent of the parameter H, which characterizes the roughness of the paths of the Volterra process. This represents a substantial improvement compared to the performance of Euler schemes for the solution of Volterra SDEs arising from rough volatility models, whose strong rate of convergence is typically H (with H often close to 0.1 in financial applications), see, e.g., [32] and [20].

2 Stochastic Volterra equation

2.1 Volterra equation

Lets start with some introduction about *Volterra equation* (VE). VE are equations of the form

$$x(t) + \int_0^t k(t,s)g(s,x(t))ds = f(t), \ t \geq 0. \tag{2.1}$$

(2.1) is precisely a *non-linear Volterra equation* with g, f, k given functions. From [17], and precisely the Theorem 1.1 of Chapter 12, we can obtain a continuous maximal solution on $[0, T]$ with $T \in [0, T_\infty[$ with $g \in \mathcal{C}([0, T_\infty[\times\mathbb{R}^n, \mathbb{R}^n), f \in \mathcal{C}([0, T_\infty[, \mathbb{R}^n)$ continuous functions and k be a $\mathbb{R}^{n\times n}$-valued Volterra kernel of continuous type on $[0, T_\infty[$. For a definition of a continuous Volterra kernel, see Definition 9.5.2 of [17].
We can also choose $k(s,t) = k(t-s)$, the kernel is known as *convolution kernel* and when $k \in L^1_{loc}([0,T], \mathbb{R}^{n\times n})$, k is a continuous Volterra kernel. For more details, see [17] or [33].
There are many examples of the use of Volterra equations in physics, engineering, or biology. See the examples section of [17].

We recall here some useful properties of the convolution.

Definition 2.1 - Convolution.
The convolution of two functions $f, g : \mathbb{R} \to \mathbb{R}$ is denoted $f * g$ and it's the following function

$$(f * g)(t) = \int_{\mathbb{R}} f(t-s)g(s)ds, \ \forall t \in \mathbb{R}. \tag{2.2}$$

The definition holds for $t \in \mathbb{R}$ such that the integral is defined.

Here are some first properties when $f, g \in L^1$.

Proposition 2.1 - Convolution properties.
Take $I \in \{\mathbb{R}, \mathbb{R}^+, \{[0,T] | 0 < T < +\infty\}\}$, $f, g \in L^1(I)$ and $p \in [1, +\infty]$, then

1. $f * g = g * f$.
2. $f * g \in L^1(I)$ and $\|f * g\|_{L^1} \leq \|f\|_{L^1} \|g\|_{L^1}$.
3. If $f \in L^p(I)$ then $f * g \in L^p(I)$.

A complete list of other properties about the convolution can be found in [17] Chapter 2 section 2.

2.2 Kernels

In this part we develop families of kernels and some of the most common ones based on [17] and [33]. The first step is to take a 2-parameters function k defined on an interval $J \subset \mathbb{R}$ such that $k : J^2 \to \mathbb{K}$, where $\mathbb{K}$ is either $\mathbb{R}$ or $\mathbb{C}$.

Definition 2.2 - Fredholm and Volterra Kernel.
If k is measurable, then k is called a *Fredholm kernel* on J. If k is a Fredholm kernel and such that

$$\begin{cases} k(0,0) = 0, \\ \forall (s,t) \in J^2, \ \text{s.t } s > t, \ k(s,t) = 0, \end{cases} \tag{2.3}$$

then k is a *Volterra kernel*.

Remark 2.1.
One can also add a L^p assumption on a Volterra kernel which means

$$\int_J \int_J |k(s,t)|^p dt ds < +\infty,$$

in some cases, it only requires L^p integrability on one variable.

We now give the definition of *kernel of type* L^p.

Definition 2.3 - Kernel of type L^p.
Let $p \in [1, +\infty]$. k is called *kernel of type* L^p is k is measurable and $\|\|k\|\|_{L^p(J)} < +\infty$, where

$$|||k|||_{L^p} := \sup_{\|f\|_{L^p} \leq 1} \||k| \star |f|\|_{L^p(J)}, \tag{2.4}$$

with $(k \star f) : t \to \int_0^t k(t,s)f(s)ds$, which leads to

$$|||k|||_{L^p} := \sup_{\|f\|_{L^p} \leq 1} \int_J \left[\int_J |k(t,s)||f(s)|ds \right]^p dt. \tag{2.5}$$

Remark 2.2.
If needed, we will use again the notation $\star$ to name the product : $(k \star f)(t) = \int_J |k(t,s)||f(t)|ds$ and $(k \star k')(t) = \int_J |k(t,s)||k'(t,s)|ds$.
Using indicator functions, we can show that

$$\|k\|_{L^1(J^2)} \leq C|||k|||_{L^p(J)}, \ C > 0. \tag{2.6}$$

We also defined the notion of kernel of type $L^p_{loc}(J)$ is for every compact subinterval of J K, the restriction of k to K^2 is a kernel of type L^p.

Kernels of type L^p have other useful properties, and there are some sufficient conditions on the integrability of k to be of type L^p, see for instance proposition 2.7 of Chapter 9 of [17].

Proposition 2.2 - Volterra and L^p type.
A Volterra kernel such that $k \in L^2(J \times J)$ is of type L^2 on J.

Remark 2.3.
In most case, the interval J is of the form $[0, T]$ with $T > 0$ and we can only compute k on $\Delta^2_T := \{(s,t) | 0 \leq s \leq t \leq T\}$.

Another useful definition is the definition of *bounded kernel*.

Definition 2.4 - Bounded kernel.
k is called a kernel of bounded type on J if k is measurable and satisfies the following conditions :

1. For every $t \in J,\ s \to k(t,s)$ is integrable.
2. For every $f \in L^\infty(J),\ t \to \int_J k(t,s)f(t)ds$ is a Borel and bounded function over J.
3. $|||k|||_{B^\infty} < \infty$ with
$$|||k|||_{B^\infty} := \sup_{t\in J} \int_J |k(t,s)|ds. \tag{2.7}$$

This previous definition leads to the definition of *continuous kernel*.

Definition 2.5 - Continuous kernel.
k is a kernel of continuous type on J if k is measurable and if $\forall t \in J,\ s \to k(t,s)$ is integrable. And if the function $t \to (s \to k(t,s))$ is, respectively , a continuous from J to $L^1(J)$.

Remark 2.4.
The continuity assumption in the previous Definition means that, for every $t \in J$
$$\lim_{h\mapsto 0} \int_J |k(t+h,s) - k(t,s)|ds = 0. \tag{2.8}$$
This last property is common in many works and we have the following properties that provide sufficient conditions on k to be of continuous type. Cf assumption (2.3) Theorem 2.1 in [18].

Proposition 2.3 - Volterra and continuity.
Every Volterra type convolution kernel in $L^1_{loc}(\mathbb{R}^+)$ is of continuous type on every interval which is bounded to the left.

Now, we move on to the convolution kernel.

Definition 2.6 - Convolution kernel.
Take this time a one-variable function $K : \mathbb{R}^+ \to \mathbb{R}^+$ such that
$$\begin{cases} \forall T > 0,\ \forall (s,t) \in \Delta_T^2,\ K(s,t) = K(t-s,0) = K(t-s), \\ \forall T > 0,\ \int_0^T K(t)dt \in]0, +\infty[, \end{cases} \tag{2.9}$$
then K is called a *convolution kernel*.

We recall some commom kernels.

Kernel name	**Kernel form**
cosine	$\cos(2\pi h)$
sinc	$\frac{\sin(\pi h)}{\pi h}$
Squared exponential	$k(h) = \exp\left(-\frac{1}{2}\frac{h^2}{\ell^2}\right)$
Exponential	$\exp\left(-\frac{\lvert h\rvert}{\ell}\right)$
Matérn 3/2	$\left(1+\sqrt{3}\frac{\lvert h\rvert}{\ell}\right)\exp\left(-\sqrt{3}\frac{\lvert h\rvert}{\ell}\right)$
Matérn 5/2	$\left(1+\sqrt{5}\frac{\lvert h\rvert}{\ell}+\frac{5}{3}\frac{h^2}{\ell^2}\right)\exp\left(-\sqrt{5}\frac{\lvert h\rvert}{\ell}\right)$

One of the most important kernels is the *Gamma kernel*, see for instance [6] for more details.

Definition 2.7 - Gamma Kernel.
The *Gamma kernel* a 3-parameter kernel with $c > 0, \alpha > 0, \rho > 0$ such that

$$K_{c,\alpha,\rho}(t) = ce^{-\rho t}\frac{t^{\alpha-1}}{\Gamma(\alpha)}, \tag{2.10}$$

with Γ the *Gamma* function such that $\Gamma(z) = \int_0^{+\infty} t^{z-1}e^{-t}dt$.

Remark 2.5.
The Gamma kernel is very useful because it includes some other kernel types like constant : $K_{c,0,0}$ $(\rho = \alpha = 0)$ or fractional $K_{c,\alpha,0}$ $(\rho = 0)$ or even exponential : $K_{c,0,\rho}$

Finally, we introduce the notion of ρ–pseudo-inverse co-kernel which is useful for the Gamma kernel.

Definition 2.8 - ρ–pseudo-inverse co-kernel.
A ρ–pseudo-inverse co-kernel with respect another convolution kernel K, denoted $\tilde{K}$, is also a convolution kernel from $\mathbb{R}^+$ t $\mathbb{R}^+$ which verified the condition

$$(K \star \tilde{K})(t) = (\tilde{K} \star K)(t) = e^{-\rho t}. \tag{2.11}$$

Remark 2.6.
For a Gamma kernel $K_{c,\alpha,\rho}$, we have a easy computation for the ρ–pseudo-inverse co-kernel, which is also a Gamma kernel with parameter $(1/c, 1-\alpha, \rho)$

$$\tilde{K}_{c,\alpha,\rho}(t) = K_{1/c,1-\alpha,\rho}(t) = \frac{1}{c}e^{-\rho t}\frac{t^{-\alpha}}{\Gamma(1-\alpha)}. \tag{2.12}$$

2.3 Integral representation of Gaussian Process

Now that we introduced some kernels and some useful definitions and properties, we are interested in defining a process of the form

$$B_t = \int_0^t K(t,s)dW_t. \tag{2.13}$$

With K a kernel and W a standard Brownian motion. We first introduce the notion of *Volterra process*.

2.3.1 Volterra process

We use the definition used in [12]. Let $T > 0$ and consider $K : [0,T] \times [0,T] \to \mathbb{R}^+$ such that K is a Volterra kernel, see (2.2) and $K(t,.) \in L^2([0,T]),\ \forall t \in [0,T]$. Then we define the function

$$R(s,t) := \int_0^{t\wedge s} K(s,r)K(s,t)dr,\ s,t \in [0,T]. \tag{2.14}$$

Definition 2.9 - Volterra Process.
Let $V = \{V_t, t \in [0,T]\}$ a centered stochastic process be the function R defined in (2.14) and such that (H) :" there is $\delta > 0$ such that b is ε–Hölder continuous for every $\varepsilon \in]0,\delta[$".
Then V is called a *Volterra process* with kernel K.

Remark 2.7.
If either of the two following conditions is satisfied, then (H) holds.

1. There exists $C > 0$ and $\beta > 1$ such that

$$\int_0^T (K(t,r) - K(r,s))^2 dr \leq C|t-s|^\beta,\ \forall s,t \in [0,T], \tag{2.15}$$

   then $\delta = \frac{\beta-1}{2}$.

2. If V is Gaussian and (2.15) holds with some $C > 0$ and $\beta > 0$, then $\delta = \frac{\beta}{2}$.

2.3.2 Gaussian Volterra processes

As Remark 2.7 points out, a Volterra process which is Gaussian seems to have better properties. In fact, when a Volterra process is Gaussian, we can establish very useful properties.

Consider $B = \{B_t, t \in [0,T]\}$ a continuous Gaussian centered process with $\mathbb{E}(B_s B_t) = R(s,t)$ his covariance function which admits the form (2.14). It has been shown that B can be expressed as

$$B_t = \int_0^t K(t,s)dW_s, t \in [0,T], \tag{2.16}$$

with K a $L^2([0,T])$ kernel, see [4] or [12]. We can also defined *Gaussian Fredholm processes* by taking the integral representation (2.16) and a Fredholm kernel instead of a Volterra kernel, see [34]

for more details and Definition 2.2.

Remark 2.8.
In this case, condition (2.15) became $\forall s < t,\ \varepsilon > 0$

1. $\int_s^t K(t,r)^2 dr \leq c_\varepsilon |t-s|^{2H-\varepsilon}$,
2. $\int_0^s |K(t,r) - K(s,r)|^2 dr \leq c_\varepsilon |t-s|^{2H-\varepsilon}$.

See [5] for more detail, if these two conditions are verified, then B is $a-$Hölder continuous for $a < H$. Alòs et al. [4] established sufficient conditions on K to B be $a-$Hölder continuous for $a < H$ which are

1. $\int_s^t K(t,r)^2 dr \leq c|t-s|^{2H}$,
2. K is differentiable in t and $\left|\frac{\partial K}{\partial t}(t,s)\right| \leq c(t-s)^{H-3/2}$.

2.3.3 Kernel representation of the fractional Brownian motion

A way to look for kernel which leads to processes of the form (2.16) is to look for kernel of the form

$$K(s,t) = a(s)\int_s^t b(u)c(u-s)du. \tag{2.17}$$

With a, b, c a least measurable functions. This form had been studied in [24], see also [25] and [26] for the particular case of power-type kernels. This form is due to the integral representation of the fractional Brownian motion, see section 7.6.1 for reminders on the fBm. For any $H \in]0,1[$, we can show that

$$B_H(t) = \int_0^t K_H(t,s)dW_s, \tag{2.18}$$

with K_H a Volterra kernel of the form of (2.17). See [4], [15], [24] or [19] for more details. K_H admits the following expression

$$K_H(t,s) = c_H s^{1/2-H}\left((t(t-s))^{H-1/2} - (H-1/2)\int_s^t u^{H-3/2}(u-s)^{H-1/2}du\right)\mathbf{1}_{0<s<t}, \tag{2.19}$$

with $c_H = \left(\frac{2H\Gamma(3/2-H)}{\Gamma(H+1/2)\Gamma(2-2H)}\right)^{1/2}$ a constant, Γ the Gamma function, see Definition 2.7, and $\mathbf{1}_{0<s<t}$ the indicative function over $\{(s,t) \in [0,T]^2 | 0 < s < t\}$.
In the form (2.19), it clear that this is a Volterra kernel. We can rewrite K_H as

$$K_H(t,s) = c_H(t-s)^{H-1/2} + c_H\left(\frac{1}{2} - H\right)\int_s^t (u-s)^{H-3/2}\left(1 - \left(\frac{s}{u}\right)^{1/2-H}\right)du. \tag{2.20}$$

There also some others ways to represent the fBm, see [23], for more details.

Remark 2.9. 1. When $H = 1/2$, we have $K_H(t,s) = c_H = 1$ and the representation is the Brownian motion.

2. When we have $H > 1/2$, the expression became

$$K_H(t,s) = (H - \frac{1}{2})c_H s^{1/2-H} \int_s^t u^{H-1/2}(u-s)^{H-3/2}du. \tag{2.21}$$

2.4 Stochastic Volterra equation

2.4.1 Introduction

Lets now move to the *stochastic Volterra equation* (SVE), a SVE is a stochastic equation of the form

$$X_t = \varphi_t + \int_0^t g(t,s,X_s)ds + \int_0^t f(t,s,X_s)dW_s, \tag{2.22}$$

with W a Brownian motion or a fBm, f, g two deterministic functions that are at least measurable, and φ another stochastic process. This kind of equation was first introduced by Berger and Mizel in [[7], [8]]. A more common way to write them is in the following form

$$X_t = x_0 + \int_0^t K(t,s)b(s,X_s)ds + \int_0^t K(t,s)\sigma(s,X_s)dW_s. \tag{2.23}$$

with K a Volterra kernel and b, σ two deterministic functions measurable as well, and x_0 a constant or a L^p stochastic process, for instance. We can show that under the Lipschitz condition on b and σ, the equation (2.23) admits a unique solution; see [14] or [18]. One can also see [36] for the singular kernel and non-Lipschitz assumption, or [2] for a study of the affine case.
This kind of equation has several potential applications, such as in finance (see [1]), and it can be very interesting to look for some parametrization:

$$X_t^\theta = x_0 + \int_0^t K_\theta(t,s)b(s,X_s^\theta)ds + \int_0^t K_\theta(t,s)\sigma(s,X_s^\theta)dW_s. \tag{2.24}$$

with $\theta \in \mathbb{R}^d$ a set of $d-$parameters with $d \geq 1$. To address this kind of problem, one can refer to the following paper by Yanqing Wang : [35]. Some interesting research has been conducted concerning predictions given a grid of parameters with some Neural Networks; see, for instance : [9].

2.4.2 Discrete approximation

In this section, we focus on the discrete approximation of Stochastic Volterra Equations (SVEs). Many papers address this problem; see, for example, [3] and [32] for Euler-type schemes or even Milstein schemes.

The Euler scheme is obtained by freezing all variables, similar to standard Stochastic Differential Equations (SDEs). For a subdivision of the interval $[0,T]$ into $n+1$ points, denoted as $0 = t_0^n < t_1^n < \cdots < t_n^n = T$, the Euler scheme freezes time between two consecutive points t_k^n and t_{k+1}^n. For simplicity, we fix n and denote t_k^n as t_k.

The Euler scheme for the equation (2.23) is given by:

$$\bar{X}_{k+1} = x_0 + \sum_{i=0}^{k} K(t_k, t_i) b(t_i, \bar{X}_i) \Delta t_i + \sum_{i=0}^{k} K(t_k, t_i) \sigma(t_i, \bar{X}_i) \Delta W_{i+1}, \tag{2.25}$$

where $\Delta t_i := t_{i+1} - t_i$, $\Delta W_{i+1} := W_{t_{i+1}} - W_{t_i}$, and $\bar{X}_k$ is the approximation process, with the goal that $\bar{X}_i \approx X_{t_i}$.

This is the simplest way to implement a numerical scheme for SVEs and presents no particular difficulties in terms of numerical implementation. Recently, a new method called the **balanced Euler method** has been introduced which is a implicit method and provided more stability. in [31].

3 A new framework to address non-Markovian problem

O. Bonesini, G. Callegaro, M. Grasselli and G. Pagès recently presented a new framework concerning non-Markovian SDE, which is similar in some ways to SVE, but is entirely new, see [10]. Let's start with some notation, we fixed a time final instant $T > 0$ and we will work on a filtered complete probability space $(\Omega, \mathcal{F}; \{\mathcal{F}_t\}_{t\geq 0}, \mathbb{P})$ and we consider two stochastic processes $X = \{X_t, t \in [0,T]\}$ and $\xi = \{\xi_t, t \in [0,T]\}$, namely (resp.) the "Elephant" and the "Goldfish" processes. We will also denote, for any convolution kernel s.t $K(t,s) = K(t-s)$ (see Definition 2.6) and any stochastic process Y such as X or ξ, the convolution operation denoted by $\star$:

$$(K \star Y)_t := \int_0^t K(t-s)Y_s ds,\ 0 \leq s \leq t \leq T. \tag{3.1}$$

The operator $(K \star .)$ is well defined on the set $\mathcal{L}([0,T]) = \mathcal{L}$, which is the set of stochastic process Y which are $\{\mathcal{F}_t\}_{t\geq 0}-$progressively measurable such that $\|Y\|_{\mathcal{L}} < +\infty$ with

$$\|Y\|_{\mathcal{L}} := \mathbb{E}\left[\int_0^T |Y_s| ds\right]^{1/2}. \tag{3.2}$$

We also take our probability space such as a Brownian motion $W = \{W_t,\ t \in [0,T]\}$ is well defined on it. We can now properly define our framework.

The **Elephant (non-Markovian) process** X is the solution of the following SDE:

$$X_t = \xi^0 + \int_0^t K(t-s)b(s, Y_s)\, ds + \int_0^t K(t-s)\sigma(s, Y_s)\, dW_s,\ t \in [0,T], \tag{3.3}$$

with $Y_s := (\widetilde{K} * X)_s$.

We also introduce the **Goldfish (Markovian diffusion) process** $\xi = \{\xi_t | t \in [0,T]\}$, defined as $\xi_t := e^{\rho t}(\widetilde{K} * X)_t$, which solves the stochastic differential equation (SDE):

$$\xi_t = \xi^0 \varphi(t) + \int_0^t \widetilde{b}(s, \xi_s) ds + \int_0^t \widetilde{\sigma}(s, \xi_s) dW_s,\ \xi_0 = 0, \tag{3.4}$$

with $\varphi(t) := e^{\rho t}(\widetilde{K} * \mathbf{1})(t)$ and $\widetilde{b} = e^{\rho u} b(u, e^{-\rho u} x)$, $\widetilde{\sigma} = e^{\rho u}\sigma(u, e^{-\rho u}x)$. The parameter $\rho \geq 0$ is fixed and relates the kernel $K =: K_{.,.,\rho}$ with its co-kernel $\widetilde{K}$, such that

$$(K * \widetilde{K})_t = (\widetilde{K} * K)_t = e^{-\rho t},$$

see Section 3.1 of [10] for more details. Equations (3.3) and (3.4) are well-posed (admits a unique strong continuous solution) if those following assumptions are fulfilled, for $\beta > 1,\ \theta \in]0,1[$

1. The kernel K satisfies, for some $C_K > 0$ and $\delta \in]0,T[$

$$\int_0^T K^{2\beta}(t) ds \quad \text{and} \quad \left(\int_0^T [K(\delta + v) \wedge T) - K(v)]^{2\beta} dv\right) < C_k \delta^{\theta}. \tag{3.5}$$

2. The function b, σ are Borel defined on $[0,T] \times \mathbb{R}$ are Lipchitz in x uniformly in $t \in [0,T]$, and bounded at $x = 0$

(i) For all $t \in [0,T]$, for all $x, y \in \mathbb{R}$

$$|b(t,x) - b(t,y)| \leq [b]_{Lip,x}|x-y| \quad \text{and} \quad |\sigma(t,x) - \sigma(t,y)| \leq [\sigma]_{Lip,x}|x-y| \tag{3.6}$$

(ii) $\sup_{t\in[0,T]} |b(t,0)| + |\sigma(t,0)| < +\infty$

3. $\xi_0 \in L_0(\mathbb{P})$,i,e, it is $\mathbb{P}$–a.s. finite.

For more details, see Theorem 3.4 of [10].

Up to now, we will work under the following assumption:

Assumption 3.1.
In all the presented numerical experiments we consider $\rho = 0$.

3.1 The numerical scheme

Let us introduce a time grid $\pi = \{t_0 = 0 < ... < t_N = T\}$ with $t_i = i \cdot \delta t,\ \forall i = 0, \dots, N$ and $\delta t = T/N$. If we denote with $X^N = \{X_k^N | k = 0, \dots, N\}$ the approximation of the process X, the goal is to have a good approximation $X_k^N \approx X_{t_k}$, in a sense that will be made precise in the sequel. We also denote by $\overline{\xi}^N = \{\overline{\xi}_k^N | k = 0, \dots, N\}$ the process approximating ξ.

The general numerical schemes for (3.3) and (3.4) are given by

$$X_{k+1}^N = \xi^0 + \sum_{l=0}^{k} b(t_l, \overline{\xi}_l^N)\left(\int_{t_l}^{t_{l+1}} K(t_{k+1}-s)ds\right) + \sigma(t_l, \overline{\xi}_l^N)\left(\int_{t_l}^{t_{l+1}} K(t_{k+1}-s)dW_s\right), X_0^N = \xi_0 \tag{3.7}$$

$$\overline{\xi}_{k+1}^N = \overline{\xi}_k^N + \xi^0(\varphi(t_{k+1}) - \varphi(t_k)) + b(t_k, \overline{\xi}_k^N)\delta t + \sigma(t_k, \overline{\xi}_k^N)\Delta W_{k+1},\ \xi_0 = 0. \tag{3.8}$$

Remark 3.1.
Here we use the notation X^N (instead of $\overline{X}^N$) because (3.7) is not a numerical scheme for a fully discretised process, due to the presence of the path dependent integrals in the definition of the process X. We will then implement numerically Equation (3.7) in Section 3 in three different ways.

The scheme (3.8) is simply the usual Euler-Maruyama discretization method for traditional SDE. If needed, one can switch to higher order schemes (like e.g. Milstein), but numerically speaking, simulating the process ξ is standard and does not represent an issue.

3.2 Goal

We want to provide a discretization scheme for (3.3) using different methods, in the special case where the kernel is given by $K = K_{1,\alpha,0}$ (**fractional kernel**), with $\alpha \in]0,1[$, namely $K(t) = \frac{t^{\alpha-1}}{\Gamma(\alpha)}$. Thanks to the standing Assumption 3.1 $\rho = 0$ we have $(K * \widetilde{K})_t = 1,\ \forall t \geq 0$, thus $\widetilde{K} = K_{1,1-\alpha,0}$ so that $\widetilde{K}(t) = \frac{t^{-\alpha}}{\Gamma(\alpha)}$.
We also assume the following expressions for b and σ, $\forall t \in [0,T]$ and $\forall x \in \mathbb{R}$:

$$b(t,x) = b(x) = (\mu - \lambda x);$$

$$\sigma(t,x) = \sigma(x) = \eta\sqrt{a(x-b)^2 + c},$$

with η, μ, λ, a, c positive real constants (here η is somehow redundant but it will be useful when investigating the speed of convergence of the discretization error with respect to σ). Let us remark that as $\rho = 0$ we have $(\widetilde{K} * X)_s = \xi_s$, see Section 3.3 of [10]. The Equation (3.3) becomes

$$X_t = \xi^0 + \int_0^t \frac{(t-s)^{\alpha-1}}{\Gamma(\alpha)}(\mu - \lambda\xi_s)\,ds + \eta\int_0^t \frac{(t-s)^{\alpha-1}}{\Gamma(\alpha)}\sqrt{c + a(\xi_s - b)^2}\,dW_s,$$

while the Goldfish process (ξ_s) solves

$$\xi_t = \xi_0\frac{t^{1-\alpha}}{\Gamma(2-\alpha)} + \int_0^t (\mu - \lambda\xi_s)ds + \eta\int_0^t \sqrt{a(\xi_s - b)^2 + c}\,dW_s.$$

Remark 3.2.
Notice that the functions b and σ satisfy the time-space Hölder-Lipschitz condition (4.3) of [10]. Indeed, $\forall\,(t,s) \in [0,T]^2, \forall\,(x,y) \in \mathbb{R}^2$:

$$|b(t,y) - b(s,x)| = |\mu - \lambda x - \mu + \lambda y| = \lambda|x-y|,$$

and

$$|\sigma(s,x) - \sigma(t,y)| = |\sqrt{a(x-b)^2 + c} - \sqrt{a(y-b)^2 + c}| = \sqrt{a}\left|\sqrt{(x-b)^2 + (\sqrt{\frac{c}{a}})^2} - \sqrt{(y-b)^2 + (\sqrt{\frac{c}{a}})^2}\right|,$$

$$= \sqrt{a}\left|\|(x-b, \sqrt{\frac{c}{a}})\|_2 - \|(y-b, \sqrt{\frac{c}{a}})\|_2\right|.$$

Then, by the triangle inequality of the Euclidean norm in $\mathbb{R}^2$: $|\sigma(s,x) - \sigma(t,y)| \le \sqrt{a}\|(x - y, 0)\|_2 = \sqrt{a}\sqrt{(x-y)^2} = \sqrt{a}|x-y|$ and therefore:

$$|b(t,y) - b(s,x)| + |\sigma(s,x) - \sigma(t,y)| \le (\sqrt{a} + \lambda)|x-y|,$$

$$|b(t,y) - b(s,x)| + |\sigma(s,x) - \sigma(t,y)| \le (\sqrt{a} + \lambda)((1 + |x| + |y|)|t-s|^\gamma + |x-y|),$$

with $\gamma \in (0,1]$ arbitrary.

Let us now recall that the time grid is $\pi = \{t_0 = 0 < \cdots < t_N = T\}$ with $t_i = i\delta t,\ \forall i = 0,\dots,N$ and $\delta t = T/N$. The process approximating X is denoted with $\overline{X}^N = \{\overline{X}^N_k\,|k = 0,\dots,N\}$, and the goal is finding a good approximation such that $\overline{X}^N_k \approx X_{t_k}$. We also denote by $b_i := b(\overline{\xi}^N_i)$ and $\sigma_i := \sigma(\overline{\xi}^N_i)$ where $\overline{\xi}^N = \{\overline{\xi}^N_k\,|k = 0,\dots,N\}$ denotes the approximation process of (ξ) such that

$$\overline{\xi}^N_{k+1} = \xi^0\left(\frac{t_{k+1}^{1-\alpha}}{\Gamma(2-\alpha)} - \frac{t_k^{1-\alpha}}{\Gamma(2-\alpha)}\right) + \overline{\xi}^N_k + \delta t\cdot b\left(\overline{\xi}^N_k\right) + \sigma\left(\overline{\xi}^N_k\right)\Delta W_{k+1},$$

with the initial condition $\overline{\xi}^N_0 = 0$, where $\Delta W = \{\Delta W_k := W_{k+1} - W_k | k = 1,\dots,N-1\}$ is the sequence of Brownian motion increments.

3.3 Three simulation schemes

3.3.1 Full Euler scheme (all kernels are "freezed")

The simplest way to implement (3.7) consists in the following discretization:

$$\overline{X}_{k+1}^N = \xi_0 + \sum_{i=0}^{k} K(t_{k+1}-t_i) b(t_i, \overline{\xi}_i^N)\delta t + \sum_{i=0}^{k} K(t_{k+1}-t_i)\sigma(t_i, \overline{\xi}_i^N)\Delta W_{i+1},$$

based on Equation (4.9) in [10], so that the numerical scheme reads

$$\overline{X}_{k+1}^N = \xi^0 + \frac{\delta t^\alpha}{\Gamma(\alpha)} \sum_{i=0}^{k} ((k+1)-i)^{\alpha-1} b_i + \frac{\delta t^{\alpha-1}}{\Gamma(\alpha)} \sum_{i=0}^{k} ((k+1)-i)^{\alpha-1} \sigma_i \Delta W_{i+1}.$$

3.3.2 General method (Cholesky goes wild and drives everybody crazy)

We recall the scheme (3.7): $\forall k = 0, \dots, N-1$ we have $X_0^N = \xi^0$ and

$$X_{k+1}^N = \xi^0 + \sum_{l=0}^{k} b(t_l, \overline{\xi}_l^N) \left(\int_{t_l}^{t_{l+1}} K(t_{k+1}-s) ds \right) + \sum_{l=0}^{k} \sigma(t_l, \overline{\xi}_l^N) \left(\int_{t_l}^{t_{l+1}} K(t_{k+1}-s) dW_s \right),$$

while the scheme (3.8) reads:

$$\overline{\xi}_{k+1}^N = \overline{\xi}_k^N + \xi^0(\varphi(t_{k+1}) - \varphi(t_k)) + b(t_k, \overline{\xi}_k^N)\delta t + \sigma(t_k, \overline{\xi}_k^N)\Delta W_{k+1}, \ \xi_0 = 0.$$

The first deterministic terms can be computed using high numerical integration methods such as *quad*, see Library doc. We denote by C the matrix with the deterministic terms, such that

$$C = \begin{pmatrix} \int_{t_0}^{t_1} K(t_1-s)\,ds & 0 & 0 & \dots & 0 \\ \int_{t_0}^{t_1} K(t_2-s)\,ds & \int_{t_1}^{t_2} K(t_2-s)\,ds & 0 & \dots & 0 \\ \int_{t_0}^{t_1} K(t_3-s)\,ds & \int_{t_1}^{t_2} K(t_3-s)\,ds & \int_{t_2}^{t_3} K(t_3-s)\,ds & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \int_{t_0}^{t_1} K(t_N-s)\,ds & \int_{t_1}^{t_2} K(t_N-s)\,ds & \int_{t_2}^{t_3} K(t_N-s)\,ds & \dots & \int_{t_{N-1}}^{t_N} K(t_N-s)\,ds \end{pmatrix} \in \mathbb{R}^{N\times N}.$$

The matrix C can be rewritten as follows:

$$C = (C_{ij})_{i,j=1\dots N} = \begin{cases} 0, & \text{if } j > i \\ \int_{t_{i-1}}^{t_i} K(t_j - s) ds, & \text{if } j \le i \end{cases}$$

The main problem, nevertheless, is the simulation of the random terms $\int_{t_l}^{t_{l+1}} K(t_{k+1}-s)dW_s$. We need to store the different coefficients in a matrix that we will denote with G and which takes the form

$$G = \begin{bmatrix} \int_{t_0}^{t_1} K(t_1-s)\,dW_s & 0 & 0 & \dots & 0 \\ \int_{t_0}^{t_1} K(t_2-s)\,dW_s & \int_{t_1}^{t_2} K(t_2-s)\,dW_s & 0 & \dots & 0 \\ \int_{t_0}^{t_1} K(t_3-s)\,W_s & \int_{t_1}^{t_2} K(t_3-s)\,dW_s & \int_{t_2}^{t_3} K(t_3-s)\,dW_s & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \int_{t_0}^{t_1} K(t_N-s)\,dW_s & \int_{t_1}^{t_2} K(t_N-s)\,dW_s & \int_{t_2}^{t_3} K(t_N-s)\,dW_s & \dots & \int_{t_{N-1}}^{t_N} K(t_N-s)\,dW_s \\ \int_{t_0}^{t_1} 1\,dW_s = \Delta W_1 & \int_{t_1}^{t_2} 1\,dW_s = \Delta W_2 & \int_{t_2}^{t_3} 1\,dW_s = \Delta W_3 & \dots & \int_{t_N-1}^{t_N} 1\,dW_s = \Delta W_N \end{bmatrix} \in \mathbb{R}^{(N+1)\times N}.$$

The matrix G can be rewritten in a more compact way as follows

$$G = (G_{ij})_{i=1..N+1, j=1...N} = \begin{cases} 0, & \text{if } j > i \\ \int_{t_{i-1}}^{t_i} K(t_j - s) dW_s, & \text{if } j \le i \le N \\ \Delta W_j, & \text{if } i = N+1 \end{cases}$$

We added the last line of the matrix G for simulating ξ and for sake of compactness. Then the following relation holds: $X_0^N = \xi_0$ and

$$\forall k = 0, \dots, N-1,\ X_{k+1}^N = \xi^0 + \sum_{l=0}^{k} b(t_l, \overline{\xi}_l^N) C_{k+1,l} + \sum_{l=0}^{k} \sigma(t_l, \overline{\xi}_l^N) G_{k+1,l}.$$

$$\overline{\xi}_{k+1}^N = \overline{\xi}_k^N + \xi^0(\varphi(t_{k+1}) - \varphi(t_k)) + b(t_k, \overline{\xi}_k^N)\delta t + \sigma(t_k, \overline{\xi}_k^N) G_{N+1,k},\ \xi_0 = 0.$$

The matrix G is of size $(N+1) \times N$ and we now apply the Cholesky decomposition method on each column to simulate the corresponding vectors: let us denote by G^ℓ, $\ell \in \{1, ..., N\}$ the ℓ-th column of the matrix G without the zero-elements, so that we have a Gaussian vector of size $(N - \ell + 2)$.

$$G^\ell = \begin{pmatrix} \int_{t_{\ell-1}}^{t_\ell} K(t_\ell - s)\,dW_s \\ \int_{t_{\ell-1}}^{t_\ell} K(t_{\ell+1} - s)\,dW_s \\ \dots \\ \int_{t_{\ell-1}}^{t_\ell} K(t_N - s)\,dW_s \\ \Delta W_l \end{pmatrix}$$

The covariance matrix of the vector G^ℓ, denoted by Σ^l, is given by:

$$\Sigma^\ell = \begin{bmatrix} \int_{t_{\ell-1}}^{t_\ell} K(t_\ell - s)K(t_\ell - s)\,ds & \dots & \int_{t_{\ell-1}}^{t_\ell} K(t_\ell - s)K(t_N - s)\,ds & \int_{t_{\ell-1}}^{t_\ell} K(t_\ell - s)\,ds \\ \int_{t_{\ell-1}}^{t_\ell} K(t_{\ell+1} - s)K(t_\ell - s)\,ds & \dots & \int_{t_{\ell-1}}^{t_\ell} K(t_{\ell+1} - s)K(t_N - s)\,ds & \int_{t_{\ell-1}}^{t_\ell} K(t_{\ell+1} - s)\,ds \\ \int_{t_{\ell-1}}^{t_\ell} K(t_{\ell+2} - s)K(t_\ell - s)\,ds & \dots & \int_{t_{\ell-1}}^{t_\ell} K(t_{\ell+2} - s)K(t_N - s)\,ds & \int_{t_{\ell-1}}^{t_\ell} K(t_{\ell+2} - s)\,ds \\ \vdots & \ddots & \vdots & \vdots \\ \int_{t_{\ell-1}}^{t_\ell} K(t_N - s)K(t_\ell - s)\,ds & \dots & \int_{t_{\ell-1}}^{t_\ell} K(t_N - s)K(t_N - s)\,ds & \int_{t_{\ell-1}}^{t_\ell} K(t_N - s)\,ds \\ \int_{t_{\ell-1}}^{t_\ell} K(t_\ell - s)\,ds & \dots & \int_{t_{\ell-1}}^{t_\ell} K(t_N - s)\,ds & \frac{T}{N} \end{bmatrix} \in \mathbb{R}^{(N-l+2)\times(N-l+2)}.$$

We can write Σ^l as

$$\Sigma^l = \begin{pmatrix} B^l & V_l \\ V_l^\top & T/N \end{pmatrix},$$

where $B^l = (B^l_{ij})_{i,j=1...N-l+2} \in \mathbb{R}^{(N-l+2)\times(N-l+2)}$, such that $B^l_{ij} = \int_{t_{\ell-1}}^{t_\ell} K(t_{l+i-1} - s)K(t_{l+j-1} - s)\, ds$ and with $V_l = (\int_{t_{\ell-1}}^{t_\ell} K(t_\ell - s)\, ds, ..., \int_{t_{\ell-1}}^{t_\ell} K(t_N - s)) \in \mathbb{R}^{N-l+1}$. Note that Σ^l is a positive definite covariance matrix, so that its Cholesky decomposition exists. Let us denote by L the (lower) triangular matrix such that

$$\Sigma = LL^\top.$$

The Cholesky decomposition of Σ^ℓ can be calculated as follows:

- We denote by L^B_ℓ the lower triangular matrix in the Cholesky decomposition of the matrix B^l. This matrix can be efficiently obtained by calculating the Cholesky decomposition L^B of the matrix B. Indeed, we have $L^B_\ell = [L^B_{i,j}]_{1\le i,j\le N-\ell+1}$.
- We calculate the vector: $\delta_\ell = (L^B_\ell)^{-1} V_\ell$.
- We then obtain the Cholesky decomposition of the matrix Σ^ℓ by taking:

$$\widetilde{L}_\ell = \begin{bmatrix} L^B_\ell & 0 \\ \delta_\ell^\top & \kappa \end{bmatrix}$$

In fact, we get

$$\Sigma^\ell = \begin{bmatrix} B^\ell & V_\ell \\ V_\ell^\top & \frac{T}{N} \end{bmatrix} = \widetilde{L}_\ell \widetilde{L}_\ell^\top = \begin{bmatrix} L^B_\ell & 0 \\ \delta_\ell^\top & \kappa \end{bmatrix} \begin{bmatrix} (L^B_\ell)^\top & \delta_\ell \\ 0 & \kappa \end{bmatrix} = \begin{bmatrix} L^B_\ell (L^B_\ell)^\top & L^B_\ell \delta_\ell \\ \delta_\ell^\top (L^B_\ell)^\top & \delta_\ell^\top \delta_\ell + \kappa^2 \end{bmatrix},$$

where

- $\delta_\ell = (L^B_\ell)^{-1} V_\ell$
- $\kappa = \sqrt{\frac{T}{N} - \delta_\ell{}^\top \delta_\ell}$.

This method provides an efficient calculation of the Cholesky decomposition of Σ^ℓ in terms of the decomposition of B_ℓ, thus reducing the number of calculations. Finally, we compute G^l via the relation (valid in distribution)

$$G^l \overset{\mathcal{L}}{=} \widetilde{L}_l Z_l, \text{ with } Z_l \sim \mathcal{N}(0, \mathcal{I}_{N-l+1}).$$

Remark 3.3.
From an algorithmic point of view, we only have to compute one Cholesky decomposition, namely the one of the matrix B. In fact, let $l \in \{1, ..., N\}$ and $N_l = N - l + 2$ such that $B^l \in \mathbb{R}^{(N_l-1)\times(N_l-1)}$. Take $i, j = 1...N_l - 1$ then

$$\begin{aligned} B^l_{i,j} &= \int_{t_{l-1}}^{t_l} K(t_{l+i-1} - s)K(t_{l+j-1} - s)ds, \\ &= \int_0^{\delta t} K(t_{l+i-1} - t_{l-1} - u)K(t_{l+j-1} - t_{l-1} - u)du, \quad (u = s - t_{l-1}) \\ &= \int_0^{\delta t} K(t_i - u)K(t_j - u)du. \end{aligned}$$

Then all matrices B^l are sub-matrix of the matrix $B \in \mathbb{R}^{N\times N}$ such that

$$B_{i,j} = \int_0^{\delta t} K(t_i - u)K(t_j - u)du. \tag{3.9}$$

Although theoretically sound, the pure Cholesky decomposition doesn't work in practice when some eigenvalues are very close to zero, namely when the matrix becomes singular. To avoid this situation, we used a more stable decomposition called the *extended Cholesky decomposition* (or *decomposition* $TDT^\top$) that only requires the positive semi-definite property, i.e. some eigenvalue may be equal to zero. We now briefly recall the method here: consider a symmetric positive semi-definite matrix

$$C = \left[C_{ij}\right]_{0\le i,j\le n-1}.$$

The **extended Cholesky decomposition** seeks to write C as

$$C = TDT^\top,$$

where:

- $T \in \mathbb{R}^{n\times n}$ is a unit lower triangular matrix (i.e. $T_{ii} = 1$).
- $D \in \mathbb{R}^{n\times n}$ is a diagonal matrix with $D_{ii} \ge 0$.
- $T^\top$ denotes the transpose of T.

Advantages:

- Existence is guaranteed when C is symmetric and positive semi-definite (unlike classical Cholesky, which requires strict definiteness).
- Applicable even when some eigenvalues are zero.
- More numerically stable in degenerate cases.

Algorithm: For $k = 0, 1, \ldots, n-1$,

$$D_{kk} = C_{kk} - \sum_{s=0}^{k-1} T_{ks}^2 D_{ss},$$

$$\text{for } i > k, \quad T_{ik} = \frac{1}{D_{kk}}\Big(C_{ik} - \sum_{s=0}^{k-1} T_{is} T_{ks} D_{ss}\Big),$$

while $T_{ii} = 1$ for all i and $T_{ij} = 0$ for $j > i$. Once this decomposition is found, we recover the classical Cholesky decomposition taking $L = T\sqrt{D}$.

Remark 3.4.
Once we compute the Cholesky decomposition (or its extended version), one checks that the triangular matrix $L_B^l \in \mathbb{R}^{N_l \times N_l}$ only has few columns that are considerable, in the sense that starting from the r-th column ($r \in \{1....N_l\}$), one has

$$[L_B^l]_{i,r} := l_{ir} < 10^{-4},\ i = 1...N_l.$$

Moreover, the coefficients beyond the r-th column decay quickly to 10^{-14}. Thus, to speed up the computation of the product $G^l = \widetilde{L}_l Z,\ Z, \sim \mathcal{N}(0, \mathcal{I}_{N_l})$, with

$$\widetilde{L}_l = \begin{bmatrix} L_\ell^B & 0 \\ \delta_\ell^\top & \kappa \end{bmatrix},$$

one can truncate the matrix by taking the first r columns. It turns out that taking $r = 3$ reveals to be a good choice, so that the matrix L_l^B reduces to

$$L_l^B = \begin{bmatrix} l_{1,1} & 0 & 0 & \cdots & \cdots & 0 \\ l_{2,1} & l_{2,2} & 0 & \cdots & \cdots & 0 \\ l_{3,1} & l_{3,2} & l_{3,3} & 0 & \cdots & 0 \\ l_{4,1} & l_{4,2} & l_{4,3} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{N_l-1,1} & l_{N_l-1,2} & l_{N_l-1,3} & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{(N_l-1)\times(N_l-1)}$$

Then the product $L_l^B X = Y$ with $X, Y \in \mathbb{R}^{N_l-1}$ is reduced to the following simple linear system

$$\begin{cases} y_1 = l_{1,1}x_1 \\ y_2 = l_{2,1}x_1 + l_{2,2}x_2 \\ y_3 = l_{3,1}x_1 + l_{3,2}x_2 + l_{3,3}x_3 \\ \vdots \\ y_i = l_{i,1}x_1 + l_{i,2}x_2 + l_{i,3}x_3,\ \forall i = 3...N_l - 1. \end{cases} \tag{3.10}$$

We now show some trajectories of the two different processes. In the rest of the plots we will take the following values for our parameters, except for explicit values.

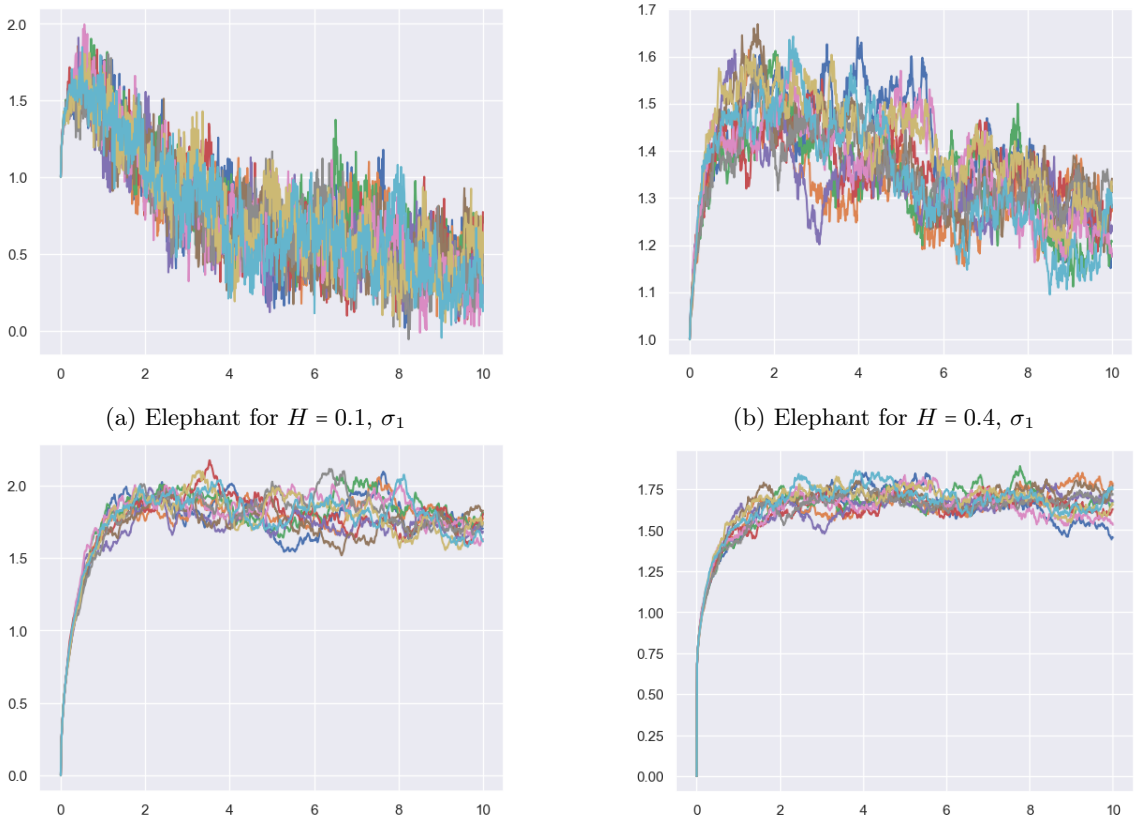
Assumption 3.2.
We will denote $\sigma_1(x) = \eta\sqrt{a(x-b)^2 + c}$ and $\sigma_2(x) = \eta\sqrt{C + \tanh(x)}$. We will take the following set of parameters depending on which σ_i we are working on, for σ_1

$$\mu = 2,\ \lambda = 1.2,\ a = 0.384,\ b = 0.095,\ c = 0.0025,\ \xi_0 = 1,\ \eta = 0.1,$$

and for σ_2

$$\mu = 2,\ \lambda = 1.2, C = 1.2,\ \xi_0 = 1,\ \eta = 0.1.$$

In Figure 1 we display 10 trajectories for different values of H.



(a) Elephant for $H = 0.1$, σ_1

(b) Elephant for $H = 0.4$, σ_1

(c) Goldfish for $H = 0.1$, σ_1

(d) Goldfish for $H = 0.4$, σ_1

Figure 1: 10 trajectories for different values of H

3.3.3 Hybrid method (a good tradeoff?)

The goal of this method is to replace the Wiener integral: $\int_{t_l}^{t_{l+1}} K(t_k - s)dW_s$ with its L^2-projection on the σ-algebra generated by a discrete Gaussian noise:

$$\mathbb{E}\left[\int_{t_l}^{t_{l+1}} K(t_k - s)dW_s \mid \Delta W_{l+1}\right] =: C_{l,k}\Delta W_{l+1} \quad \text{(regression)}$$

Set $C_{l,k} > 0$ a constant defined as

$$C_{l,k} = \frac{\mathbb{E}\left[\int_{t_l}^{t_{l+1}} K(t_k - s)dW_s \Delta W_{l+1}\right]}{\mathbb{E}\left[\Delta W_{l+1}^2\right]},$$

then

$$C_{l,k} = \frac{N}{T}\int_{t_l}^{t_{l+1}} K(t_k - s)ds$$

and the scheme becomes

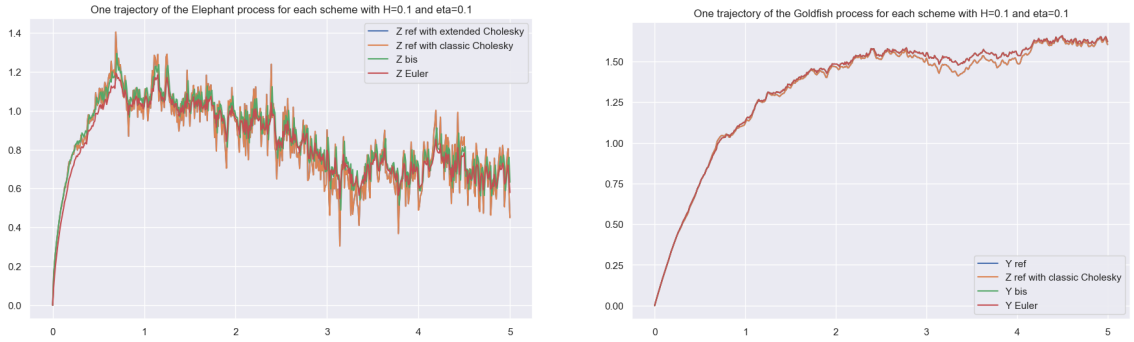
$$\overline{X}_k^N = \xi^0 + \sum_{l=0}^{k-1}\int_{t_l}^{t_{l+1}} K(t_k - s)ds\left(b(\overline{\xi}_l^N) + \frac{N}{T}\sigma(\overline{\xi}_{l+1}^N)\Delta W_l\right).$$

3.3.4 Comparison of the three schemes

To speed up the Python code, we use a JIT (Just-In-Time) compiler from the `Numba` library, which is a very simple way to compile code quickly. For more details about it, see the Numba site.

The main advantage of Numba is that the compiler skips the *Python object* mode and instead focuses on the specific type of a variable (float, int, `np.ndarray`, etc.), much like faster programming languages such as *C* or *Fortran*.

The library also allows us to use many other features, such as parallelization, vectorization, and even GPU acceleration (only with NVIDIA hardware) if needed.



(a) Comp Elephant (b) Comp Goldfish

Figure 2: Comparison of the 3 methods. Here "ref" is relative to the general scheme with an extended version of the Cholesky decomposition (namely the decomposition $LDL^\intercal$). Here the solutions from classic or extended Cholesky are the same (less than 10^{-5} difference between those two in L^2 and with max norm).

From Figure 2 a) we see that the General simulation method provides trajectories with more randomness than the other two schemes. The Hybrid scheme and the Full Euler scheme are the fastest algorithms, as one could easily have guessed. We provide here a couple of examples in order to give the flavor of the computational costs:

(N; number of traj.)	General	Full Euler	Hybrid
$(1000; 10)$	2.3	1.5	0.74
$(250; 1000)$	0.47	16.2	0.49

Table 1: Computational costs, in seconds, for the corresponding number of discretization points N and the number of trajectories, as made precise in the first column.

Remark 3.5.
We did not optimize the Full Euler algorithm using Numba, this is why the computational cost is high in the second case.

Note that, even though we have the same scheme in the 3 methods for the Goldfish process (here Y), it is natural to end up with slightly different trajectories, since in the general method we compute an approximation of the Cholesky matrix which brings little differences.

Comparison	**Y**	**Z**
Relative L2 difference : $\|X^{ref} - X\|_2 = \sqrt{\sum_{i=0}^{N} \|X^{ref}[i] - X[i]\|^2} / \|X^{ref}\|_2$		
Euler vs General	1.933e-02	7.775e-02
Hybrid vs General	1.933e-02	1.164e-01
Relative max difference : $\|X^{ref} - X\|_\infty = \max_i \|X^{ref}[i] - X[i]\| / \|X^{ref}\|_\infty$		
Euler vs General	4.375e-02	1.716e-01
Hybrid vs General	4.375e-02	2.288e-01

Table 2: Relative errors summed up (L2 and max) for Y and Z

In conclusion, all the proposed methods are promising insofar as they allow for the computation of many trajectories while maintaining a reasonable computational time, which represents a crucial point in view of possible applications, such as Monte Carlo pricing.

4 Numerical convergence of the general scheme

We now illustrate the results of the convergence of the general numerical scheme by specifying the coefficients of the process. Remarkably, we shall show that, in our examples, the convergence rate of the strong error is indeed 1/2 as predicted in Ref paper. We set $\eta = 1$ and we consider two specifications for the function σ, namely:

$$\sigma_1(y) = \sqrt{a(y-b)^2 + c}, \quad \sigma_2(y) = \sqrt{C + \tanh(y)}. \tag{4.1}$$

with $a, c, > 0$, $C > 1$, $b \in \mathbb{R}$ fixed constants. Here are the numerical values chosen for the simulation:

- $a = 0.384, \quad b = 0.095, \quad c = 0.0025$
- $C = 1.2$.

4.1 Max error

We study here the error

$$\text{[Max error]} \qquad \sup_{t\in[0,T]} \mathbb{E}[|X_t - \overline{X}_t^N|^2]^{1/2}. \tag{4.2}$$

Naturally, the approximation error of a numerical scheme is computed with respect to the exact solution. Unfortunately, we do not have a closed-form expression for the elephant process (Z). Therefore, it is necessary to find an alternative method to visualize the convergence of the scheme.

The idea is as follows: compute the numerical approximation difference (i.e., the L^2-error) between the schemes $(X_{t_k}^N)_{k\in\{0,\dots,N\}}$ and $(X_{\text{Const}\cdot t_k}^{\text{Const}\cdot N})_{k\in\{0,\dots,N\}}$, where Const is a small integer relative to N (we take Const = 2 for this example).

We know that there exist constants C and C' such that:

$$\max_{k\in\{0,\dots,N\}} \mathbb{E}\left[|X_{t_k} - \overline{X}_{t_k}^N|^2\right] \le \frac{C}{N}.$$

$$\max_{k\in\{0,\dots,N\}} \mathbb{E}\left[|X_{t_{2k}} - \overline{X}_{t_{2k}}^{2N}|^2\right] \le \frac{C'}{2N}.$$

Therefore, intuitively, at the discretization point t_k, the L^2-norm between $\overline{X}_{t_{2k}}^{2N}$ and $\overline{X}_{t_k}^N$ should be of the same order as the largest error between the two schemes, that is: $\sqrt{\frac{C''}{N}}$.

Mathematically:

$$\max_{k\in\{0,\dots,N\}} \mathbb{E}\left[|\overline{X}_{t_k}^N - \overline{X}_{t_{2k}}^{2N}|^2\right] \le \frac{C''}{N}$$

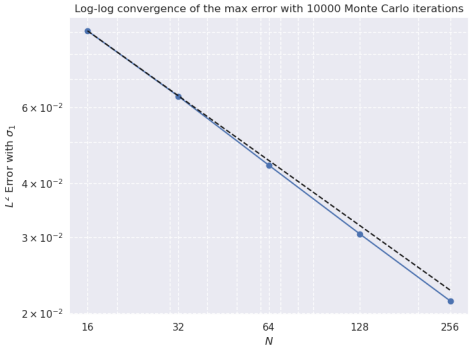
Now, suppose that, for any $t_k = kT/N$ with $k = 0, \dots, N$,

$$\|\overline{X}_{t_k}^N - \overline{X}_{t_k}^{2N}\|_2 \le C\sqrt{\frac{T}{N}} = C_T\sqrt{\frac{1}{N}}, \tag{4.3}$$

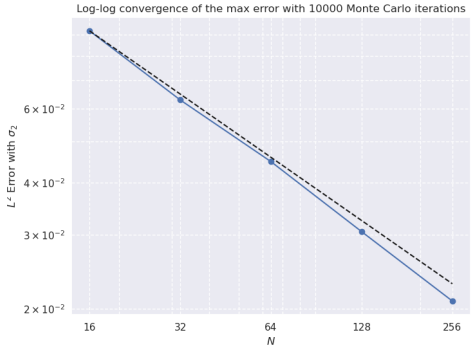
then, since $\overline{X}_{t_k}^N \to X_{t_k}$ as $N \to +\infty$, we get

$$\|\overline{X}_{t_k}^N - X_{t_k}\| \le \sum_{l=0}^{+\infty} \|\overline{X}_{t_l}^{2^l N} - \overline{X}_{t_l}^{2^{l+1} N}\|_2 \le C_T \sum_{l=0}^{+\infty} \frac{1}{2^{l/2}\sqrt{N}} = \frac{C_T}{\sqrt{N}} \sum_{l=0}^{+\infty} \frac{1}{2^{l/2}} = \frac{C_T}{\sqrt{N}} \frac{1}{1-2^{-1/2}}.$$

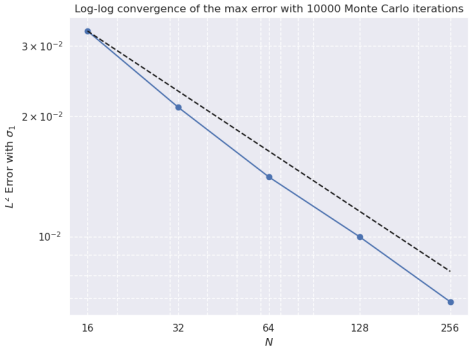
In other words, if we have a 1/2 strong order convergence for (4.3), then we have 1/2 strong order convergence for the strong error of (4.2) as well.



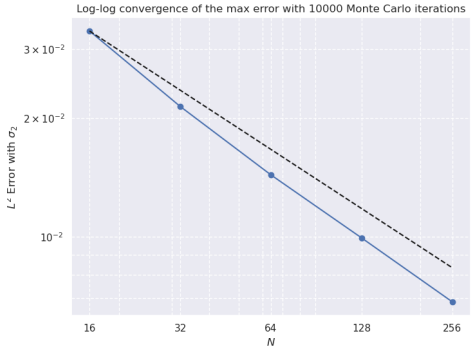
(a) Elephant process with slope≈ 0.5335 and σ_1



(b) Elephant process with slope≈ 0.4886 and σ_2

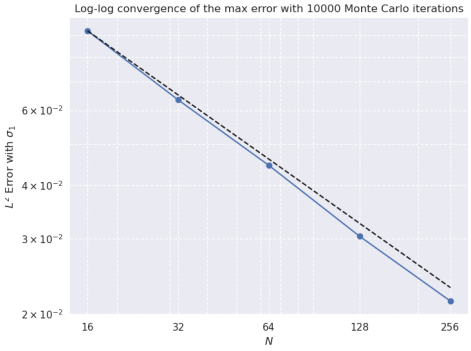


(c) Goldfish process with slope≈ 0.5655 and σ_1

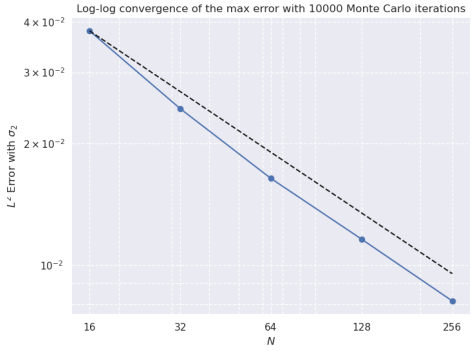


(d) Goldfish process with slope≈ 0.5610 and σ_2

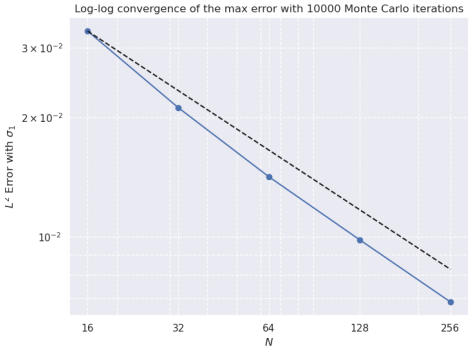
Figure 3: Strong order convergence in log-log for $H = 0.1$ and for the two examples of the σ function in Eq. (4.1)



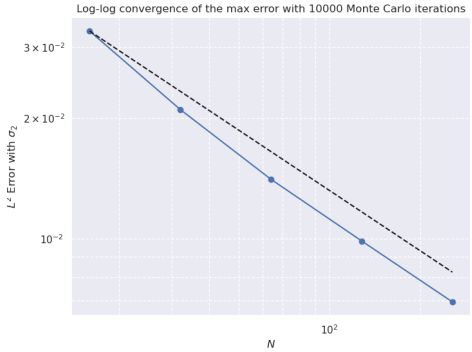
(a) Elephant process with slope≈ 0.5288 and σ_1



(b) Elephant process with slope≈ 0.5718 and σ_2



(c) Goldfish process with slope≈ 0.5609 and σ_1



(d) Goldfish process with slope≈ 0.5771 and σ_1

Figure 4: Strong order convergence in log-log for $H = 0.4$ and for two examples of σ function in Eq. (4.1)

4.2 End-point error

We study here the error

$$\text{[End Point error]} \qquad e_T := \mathbb{E}[|X_T - \overline{X}_T^N|^2]^{1/2}. \tag{4.4}$$

We will now establish an algorithm for studying the end-point error more precisely. The goal is to simulate

$$\overline{X}_T^{2n} \text{ and } \overline{X}_T^n,$$

and to compare the error

$$\|\overline{X}_T^{2n} - \overline{X}_T^n\|_2^2 \approx \frac{1}{N_{mc}} \sum_{i=1}^{N_{mc}} |X_T^{2n,i} - X_T^{n,i}|^2$$

for a given n and a large integer N_{mc}.

Observe that in this case we have to simulate the terminal value $\overline{X}_T^N$ given by

$$\overline{X}_T^N = \xi^0 + \sum_{l=0}^{N-1} \int_{t_l}^{t_{l+1}} K(T-s)ds \cdot b(\overline{Y}_l^N) + \sum_{l=0}^{N-1} \int_{t_l}^{t_{l+1}} K(T-s)dW_s \cdot \sigma(\overline{Y}_l^N),$$

and where

$$\overline{Y}_l^N = \overline{Y}_l^N + \xi^0(\varphi(t_l) - \varphi(t_{l-1}) + b(\overline{Y}_{l-1}^N)\delta t + \sigma(\overline{Y}_l^N)\Delta W_l.$$

Then, we only need to simulate N vectors G_l $(l = 0, \dots, N-1)$ such that

$$G_l = \begin{pmatrix} \Delta W_{l+1} \\ \int_{t_l}^{t_{l+1}} K(T-s)dW_s \end{pmatrix}.$$

Let $\sigma_1^2 = T/N = \delta t$, $\sigma_2^2 = \int_{t_l}^{t_{l+1}} K(T-s)^2 dW_s$ and $\rho_l = \frac{\int_{t_l}^{t_{l+1}} K(T-s)^2 dW_s}{\sqrt{\sigma_1^2}\sqrt{\sigma_2^2}}$, then the covariance matrix Σ_l reads

$$\Sigma_l = \begin{pmatrix} \sigma_1^2 & \rho_l\sigma_1\sigma_2 \\ \rho_l\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix} = L_l L_l^T,$$

with

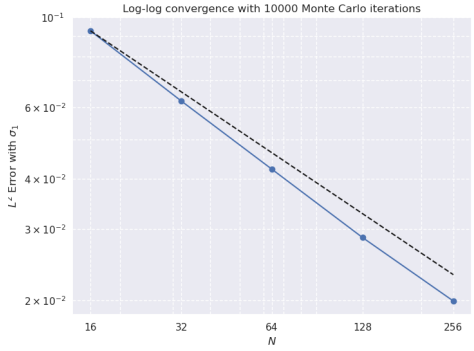
$$L_l = \begin{pmatrix} \sigma_1 & 0 \\ \rho_l\sigma_1 & \sigma_2\sqrt{1-\rho_l^2} \end{pmatrix}.$$

The method consists in simulating at the same time $\overline{X}_T^{2N}$ and $\overline{X}_T^N$ and using the following relations:

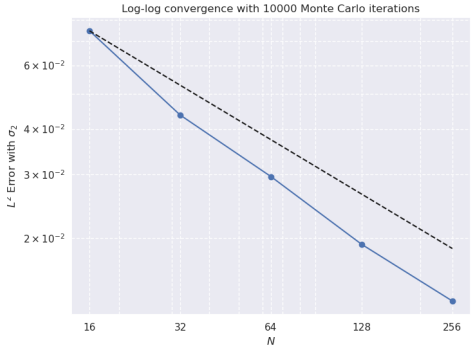
$$\Delta W_{l+1}^N = \Delta W_{2l+1}^{2N} + \Delta W_{2(l+1)}^{2N} \Leftrightarrow G_l^N[1] = G_{2l+1}^N[1] + G_{2(l+1)}^N[1].$$

We can see with Chasles relation and dyadic properties that

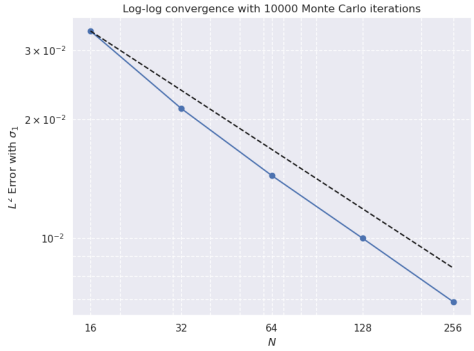
$$G_l^N[2] = G_{2l+1}^N[2] + G_{2(l+1)}^N[2].$$



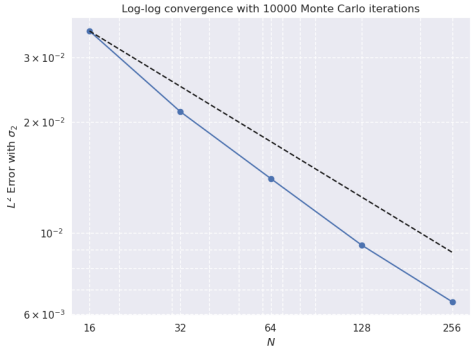
(a) Elephant process with slope = 0.5070 and σ_1



(b) Elephant process with slope ≈ 0.5452 and σ_2

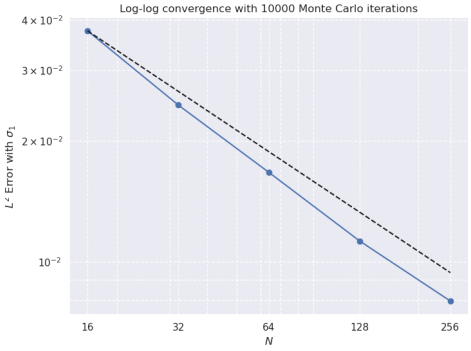


(c) Goldfish process with slope ≈ 0.5341 and σ_1

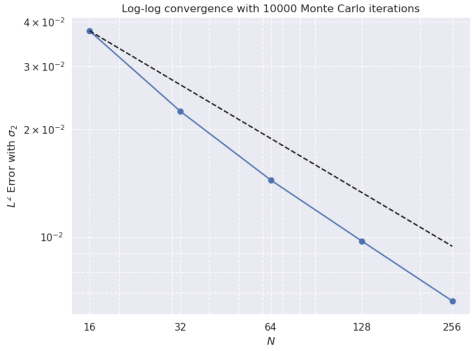


(d) Goldfish process with slope ≈ 0.6114 and σ_2

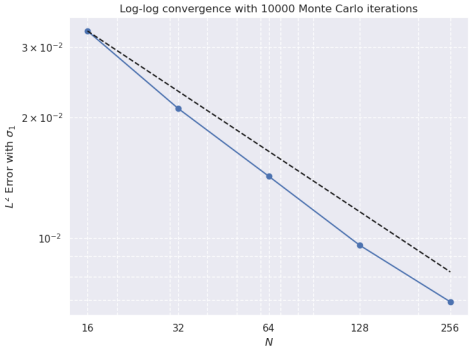
Figure 5: Strong order convergence in log-log for $H = 0.1$ and for two examples of σ function in Eq. (4.1)



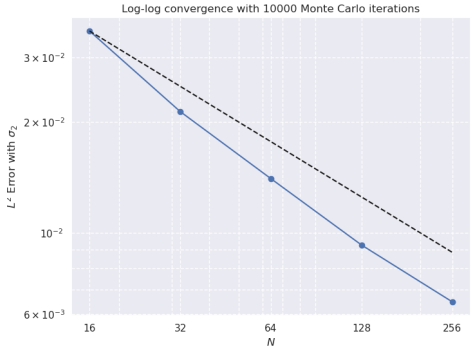
(a) Elephant process with slope ≈ 0.5639 and σ_1



(b) Elephant process with slope ≈ 0.6255 and σ_2



(c) Goldfish process with slope ≈ 0.5644 and σ_1



(d) Goldfish process with slope ≈ 0.6114 and σ_2

Figure 6: Strong order convergence in log-log for $H = 0.4$ and for two examples of σ function in Eq. (4.1)

4.3 Conclusion

This numerical experiment shows that, in the case of a fractional kernel, we retrieve the 1/2 order of convergence of our scheme. In our example, the slope is even strictly higher than 1/2; we obtain a better slope for the endpoint error, which is logical given that we can use a simpler algorithm to simulate our error. All this part and the previous plots will be added to the paper [10] to provide a numerical illustration of the method. This also serves as a good point to confirm the theoretical result.

The result is quite impressive compared to the traditional order depending on the Hurst parameter H: in fact, for stochastic Volterra equations, the order of a given scheme is typically H (for Euler scheme) or a multiple of H (for higher method). This new framework has a scheme with a strong convergence order of the error of 1/2, regardless of the value of H.

5 Hölder continuity of the elephant process

In this part, we are willing to numerically illustrate the γ–Hölder ($\gamma \in (0, H)$) continuity property of the elephant process X.

The classical **pointwise definition** of γ-Hölder continuity for a process X states that there exists a constant $C > 0$ such that, a.s., for all s, t,

$$|X_t - X_s| \leq C|t - s|^\gamma, \tag{5.1}$$

or in other words,

$$\mathbb{P}\left(\omega \in \Omega, \sup_{0<t-s<h(\omega)} \frac{|X_t(\omega) - X_s(\omega)|}{|t - s|^\gamma} \leq \delta\right) = 1, \tag{5.2}$$

with h a positive r.v and $\delta \geq 0$ This means that each sample path of X is γ-Hölder continuous.

However, in the stochastic setting, it is often more practical to work with a **moment condition** on the increments:

$$\exists C_p > 0, \quad \forall s, t, \quad \mathbb{E}\big[|X_t - X_s|^p\big] \leq C_p|t - s|^{\gamma p}, \tag{5.3}$$

for some $p \geq 1$.

- The pointwise Hölder condition controls the increments **pathwise** (almost surely).
- The moment condition controls the increments **on average** (in expectation).

Moment condition on increments:

There exist constants $C > 0$, $\alpha > 0$, and $\beta > 0$ such that for all $s, t \in [0, T]$,

$$\mathbb{E}\big[|X_t - X_s|^\alpha\big] \leq C|t - s|^{\beta+1}. \tag{5.4}$$

Consequence (Kolmogorov continuity theorem):

Under this condition, there exists a version of X whose trajectories are almost surely γ-Hölder continuous for any

$$\gamma \in \left(0, \frac{\beta}{\alpha}\right).$$

In other words, almost surely,

$$\sup_{s \neq t} \frac{|X_t - X_s|}{|t - s|^\gamma} < +\infty. \tag{5.5}$$

Remark:

The +1 in the exponent $\beta + 1$ is related to the temporal dimension (here 1) and integrability requirements needed to apply the theorem.

5.1 Example: the Brownian motion

For standard Brownian motion B, we know:

$$\mathbb{E}[|B_t - B_s|^p] = C_p|t-s|^{p/2}, \tag{5.6}$$

where $C_p = \mathbb{E}[|Z|^p]$ and $Z \sim \mathcal{N}(0,1)$ is standard normal, as $B_t - B_s \sim \sqrt{|t-s|}Z$ in law.
This corresponds to the moment condition with $\gamma = \frac{1}{2}$.
Hence, Brownian motion has almost surely γ-Hölder continuous paths for any $\gamma < \frac{1}{2}$.
This illustrates how the moment condition naturally arises from the statistical properties of increments and leads to the almost sure Hölder regularity of stochastic processes like Brownian motion.

5.2 Numerical method

We want to study the Hölder character of the trajectory of X, the elephant process, using the moment condition :

$$\mathbb{E}[|X_t - X_s|^p] \leq C|t-s|^{pH} \Rightarrow X \text{ admits a modification } \tilde{X} \text{ which has a.s } H- \text{ Hölder paths} \tag{5.7}$$

Given a subdivision $\pi = \{t_0 = 0 < ... < t_N = T\} \subset [0,T]$, the goal is to evaluate the quantity

$$\frac{\mathbb{E}[|X_{t_{i+1}^N} - X_{t_i^N}|^p]}{\left(\frac{T}{N}\right)^{pH}},\ p \geq 1,\ i = 0, N-1. \tag{5.8}$$

We replace X with $\overline{X}$, our approximation process, and we estimate $\mathbb{E}[|\overline{X}_{t_{i+1}^N} - \overline{X}_{t_i^N}|^p]$ using traditional Monte-Carlo simulation:

$$\frac{\mathbb{E}[|\overline{X}_{t_{i+1}^N} - \overline{X}_{t_i^N}|^p]}{(T/N)^{pH}} \approx \frac{1}{m \cdot (T/N)^{pH}} \sum_{k=1}^{m} |\overline{X}^{(k)}_{t_{i+1}^N} - \overline{X}^{(k)}_{t_i^N}|^p. \tag{5.9}$$

If the Hölder property holds, the ratio will converge to some positive constant that should not depend on N. Our result are gather in the following tables.

N	$H_{num} = H_{den} = 0.1$	$H_{num} = H_{den} = 0.4$	$H_{num} = 0.1$, $H_{den} = 0.4$	$H_{num} = 0.4$, $H_{den} = 0.1$
250	1.663	0.660	8.680	0.126
500	1.655	0.639	10.712	0.105
750	1.701	0.663	12.219	0.091
1000	1.650	0.670	13.213	0.085
1500	1.692	0.658	15.030	0.077
2000	1.637	0.676	16.313	0.067
3000	1.629	0.686	17.866	0.060
5000	1.664	0.659	20.479	0.053
10000	1.645	0.675	25.525	0.042

Table 3: Values of the average ratio (with $p = 1$) vs N according to $H_{num} = H_{denom}$ or $H_{num} \neq H_{denom}$.

N	$H_{num} = H_{den} = 0.1$	$H_{num} = H_{den} = 0.4$	$H_{num} = 0.1$, $H_{den} = 0.4$	$H_{num} = 0.4$, $H_{den} = 0.1$
250	4.221	0.790	126.060	0.029
500	4.675	0.779	194.897	0.017
750	4.807	0.731	234.280	0.014
1000	4.633	0.732	274.605	0.013
1500	4.285	0.735	372.800	0.009
2000	4.428	0.864	422.030	0.008
3000	4.317	0.807	547.766	0.007
5000	4.551	0.685	711.009	0.005
10000	4.518	0.794	1036.610	0.003

Table 4: Values of the average ratio (with $p = 2$) vs N according to $H_{num} = H_{denom}$ or $H_{num} \neq H_{denom}$.

This shows numerically that our process is indeed γ–Hölder with $\gamma \in (0, H)$.

6 Conclusion

On the numerical side, with particular reference to simulations via Euler-type schemes, the proposed approach is very promising. Indeed, we prove a remarkable strong error convergence rate of order $(\gamma \wedge 1/2)$, with $\gamma \in (0, 1)$ (depending on the regularity of the drift and volatility coefficients), for the non-Markovian process X in the case of a fractional convolution kernel parametrized by the Hurst coefficient H, with $H \in (0, 1/2)$. In this case the simulated trajectories display a rough behavior, yet the corresponding Euler-type simulation scheme has a strong rate of convergence of order $1/2$, i.e., independent of H. They represents a crucial improvement with respect to what is found in the literature on standard Volterra processes, where the proposed schemes show a convergence error rate typically equal to H.

On a personal level, I truly enjoyed this internship, as it provided me with my first real research experience over three months. I had to find answers independently, which allowed me to learn a great deal about Markovian and non-Markovian processes, particularly in the simulation of stochastic processes and Stochastic Differential Equations (SDEs).

I also learned how to write clear, reproducible, and well-documented code to effectively communicate my results, as I was solely responsible for the coding aspect of the project. Additionally, I had the opportunity to collaborate for one week with Gilles Pagès, who visited Padova to work with my tutors on their joint research. This week was intense but highly rewarding, as I actively participated in discussions and contributed meaningfully to the project.

My work has been included in the paper [10] (see the latest version), and I continue to collaborate with my tutors to further develop and publish additional work on this topic.

7 Appendix

This appendix is an introduction to Markov processes and the study of a famous non Markovian process : the fractional Brownian motion.

This part is mostly based on [29] and [27] with some additional references for the fractional Brownian motion.

7.1 Markov processes

7.1.1 Introduction and examples

Let $d \geq 1$, the dimension constant and $(\Omega, \mathcal{F}, \mathbb{P})$ a complete probability space. We denote $L_m = [L^m(\Omega, \mathcal{F}, \mathbb{P})]^d$, with $m \geq 2$ a fixed integer. Let $X = (X_i)_{i\in\mathbb{N}}$ denote a discrete stochastic process in L_m such that $\forall i \in \mathbb{N},\ X_i \in L_m$ and $X_0 = 0$.
Let $(G_i)_{i\in\mathbb{N}}$ be a discrete filtration($\forall i < j,\ \mathcal{G}_i \subset \mathcal{G}_j \subset \mathcal{F}$). We will assume that X_i is $\mathcal{G}_i-$ measurable, for all $i > 0$, which amounts to take X adapted to the filtration $(G_i)_{i\in\mathbb{N}}$.
Markov processes are processes that retain no memory of where it have been in the past: only the current state of a Markov process can influence where it will go next. The simplest example of Markov process is the random walk in one dimension $d = 1$.
Let $\xi_i, i = 1...N$ be independent, identically distributed mean zero and variance 1 random variables. X is called a one-dimension random walk if

$$X_N = \sum_{i=1}^{N} \xi_i, \quad X_0 = 0. \tag{7.1}$$

Let $(i_n)_n$ a sequence of integers, then for X verify

$$\forall (n, m) \in \mathbb{N}^2,\ \mathbb{P}(X_{n+m} = i_{n+m} | X_1 = i_1, ..., X_n = i_m) = \mathbb{P}(X_{n+m} = i_{n+m} | X_n = i_n). \tag{7.2}$$

In words, the probability that the random walk will be at in+m at time $n + m$ depends only on its current value (at time n) and not on how it got there. The random walk is an example of a discrete time Markov chain. We will say that a stochastic process Z with state space $S = \mathbb{Z}$ is a discrete time Markov chain provided that the Markov property (7.2) is satisfied. Before continuing, we introduce the definition of a *continuous stochastic process* with the associated notations.

Definition 7.1 - Stochastic Process.
Let T be an ordered set, $(\Omega, \mathcal{F}, \mathbb{P})$ a probability space and $(E, \mathcal{G})$ a measurable space. A stochastic process is a collection of random variables $X = \{X_t, t \in T\} = (X_t)_{t\in T}$ where, for each fixed $t \in T$, X_t is a random variable from $(\Omega, \mathcal{F}, P)$ to $(E, \mathcal{G})$:

$$X_t : \omega \in \Omega \to X_t(\omega) \in E, \tag{7.3}$$

Ω is known as the **sample space**, where E is the **state space** of the stochastic process X.

Remark 7.1.
The set T can be either discrete, for example the set of positive integers $\mathbb{Z}^+$, or continuous, $T = \mathbb{R}^+$. The state space E will usually be $\mathbb{R}^d$ equipped with the σ-algebra of Borel sets. A stochastic process X may be viewed as a function of both $t \in T$ and $\omega \in \Omega$. For a fixed sample point $\omega \in \Omega$, the function $X_t(\omega) : T \to E$ is called a (realization, trajectory) of the process X.

Consider now a continuous-time stochastic process X_t with state space $S = \mathbb{Z}$ and denote by $\{X_s, s \le t\}$ the collection of values of the stochastic process up to time t. We will say that X_t is a Markov process provided that

$$\mathbb{P}(X_{t+h} = i_{t+h} | \{X_s, s \le t\}) = \mathbb{P}(X_{t+h} = i_{t+h} | X_t), \tag{7.4}$$

for all $h > 0$. A continuous-time, discrete state space Markov process is called a continuous-time Markov chain. A standard example of a continuous-time Markov chain is the Poisson process of rate λ with

$$P(N_{t+h} = j | N_t = i) = \begin{cases} 0 & \text{if } j < i, \\ \frac{e^{-\lambda s}(\lambda s)^{j-i}}{(j-i)!} & \text{if } j > i. \end{cases} \tag{7.5}$$

Similarly, we can define a continuous-time Markov process with state space $\mathbb{R}$, as a stochastic process whose future depends on its present state and not on how it got there:

$$\mathbb{P}(X_{t+h} \in \Gamma | \{X_s, s \le t\}) = \mathbb{P}(X_{t+h} \in \Gamma | X_t) \tag{7.6}$$

for all Borel sets Γ. We can consider continuous-time Markov processes for which a conditional probability density exists:

$$\mathbb{P}(X_{t+h} \in \Gamma | X_t = x) = \int_\Gamma p(y, t+h | x, t)\, dy. \tag{7.7}$$

Example 7.1.
The Brownian motion is a Markov process with conditional probability density given by the following formula

$$\mathbb{P}(W_{t+h} \in \Gamma | W_t = x) = \int_\Gamma \frac{1}{\sqrt{2\pi h}} \exp\left(-\frac{|x-y|^2}{2h}\right) dy. \tag{7.8}$$

The Markov property of Brownian motion follows from the fact that it has independent increments.

Example 7.2.
The stationary Ornstein-Uhlenbeck process $V_t = e^{-t} W(e^{2t})$ is a Markov process with conditional probability density

$$p(y, t | x, s) = \frac{1}{\sqrt{2\pi(1 - e^{-2(t-s)})}} \exp\left(-\frac{|y - xe^{-(t-s)}|^2}{2(1 - e^{-2(t-s)})}\right). \tag{7.9}$$

7.1.2 Definition

In order to give the definition of a Markov process we need to use the conditional expectation of the stochastic process conditioned on all past values. Let us consider a stochastic process $X = (X_t)_{t\geq 0}$ and state space $(\mathbb{R}^d, \mathcal{B})$, where $\mathcal{B}$ denotes the Borel σ-algebra. We define the σ-algebra generated by $\{X_t, t \in \mathbb{R}^+\}$, denoted by $\sigma(X_t, t \in \mathbb{R}^+)$, to be the smallest σ-algebra such that the family of mappings $\{X_t, t \in \mathbb{R}^+\}$ is a stochastic process with sample space $(\Omega, \sigma(X_t, t \in \mathbb{R}^+))$ and state space $(\mathbb{R}^d, \mathcal{B})$. In other words, the σ-algebra generated by X_t is the smallest σ-algebra such that X_t is a measurable function (random variable) with respect to it.

We define now a filtration on $(\Omega, \mathcal{F})$ to be a nondecreasing family $\{\mathcal{F}_t, t \in \mathbb{R}^+\}$ of sub-σ-algebras of $\mathcal{F}$:

$$\mathcal{F}_s \subseteq \mathcal{F}_t \subseteq \mathcal{F} \text{ for } s \leq t.$$

We set $\mathcal{F}_\infty = \sigma(\bigcup_{t\geq 0} \mathcal{F}_t)$. The filtration generated by our stochastic process X is:

$$\mathcal{F}_t^X := \sigma(X_s; s \leq t).$$

Now we are ready to give the definition of a Markov process.

Definition 7.2 - Markov Process.
Let X_t be a stochastic process defined on a probability space $(\Omega, \mathcal{F}, \mu)$ with values in $\mathbb{R}^d$ and let $\mathcal{F}_t^X$ be the filtration generated by $\{X_t, t \in \mathbb{R}^+\}$. Then $\{X, t \in \mathbb{R}^+\}$ is a Markov process provided that

$$\mathbb{P}(X_t \in \Gamma | \mathcal{F}_s^X) = \mathbb{P}(X_t \in \Gamma | X_s) \tag{7.10}$$

for all $t, s \in T$ with $t > s$, and $\Gamma \in \mathcal{B}(\mathbb{R}^d)$.

Remark 7.2.
We remark that the filtration $\mathcal{F}_t^X$ is generated by events of the form $\{\omega | X_{t_1} \in \Gamma_1, X_{t_2} \in \Gamma_2, \ldots, X_{t_n} \in \Gamma_n\}$, with $0 \leq t_1 < t_2 < \cdots < t_n \leq t$ and $\Gamma_i \in \mathcal{B}(\mathbb{R}^d)$. The definition of a Markov process is thus equivalent to the hierarchy of equations

$$\mathbb{P}(X_t \in \Gamma | X_{t_1}, X_{t_2}, \ldots, X_{t_n}) = \mathbb{P}(X_t \in \Gamma | X_{t_n}) \quad \text{a.s.}$$

for $n > 1$ and $0 \leq t_1 < t_2 < \cdots < t_n \leq t$ with $\Gamma \in \mathcal{B}(E)$.

7.2 Stationary process

We first need to introduce the notion of *stationary process* and *self-similar process* then the notion of H-**sssi** process.

Definition 7.3 - Stationary Process.
A stochastic process $\{X_t\}_{t\geq 0}$ is said to be a stationary process if any collection $\{X_{t_1}, X_{t_2}, \ldots, X_{t_n}\}$ has the same distribution as $\{X_{t_1+\tau}, X_{t_2+\tau}, \ldots, X_{t_n+\tau}\}$, for each $\tau \geq 0$. That is,

$$\{X_{t_1}, X_{t_2}, \ldots, X_{t_n}\} \overset{d}{=} \{X_{t_1+\tau}, X_{t_2+\tau}, \ldots, X_{t_n+\tau}\}. \tag{7.11}$$

The above equation (7.11) expresses the time translational invariance of the finite dimensional distributions of a stationary process. Let X be a stationary process, then the following elementary properties hold:

- Varying t, all the random variables X_t have the same law; i.e., $X_{t_1} \overset{d}{=} X_{t_2}$ for any $t_1, t_2 \geq 0$.
- All the moments, if they exist, are constant in time.
- The distribution of X_{t_1} and X_{t_2} depends only on the difference $\tau = t_2 - t_1$ (time lag).

Therefore, the covariance function $\gamma(t_1, t_2) = \gamma(t_2 - t_1)$ depends only on $\tau = t_2 - t_1$. We write

$$\gamma(\tau) = \mathbb{E}[(X_t - \mu)(X_{t-\tau} - \mu)] = \mathrm{Cov}(X_t, X_{t-\tau}),$$

where $\mu = \mathbb{E}[X(t)]$ and $\gamma(\tau)$ indicates the covariance coefficient at the lag τ.

Definition 7.4 - Stationary Increment Process.
A stochastic process $\{X_t\}_{t\geq 0}$ is said to be a stationary increment process, shortly **si**, if for any $h > 0$:

$$\{X_{t+h} - X_h\}_{t\geq 0} \overset{d}{=} \{X_t - X_0\}_{t\geq 0}.$$

Remark 7.3.
If X is an **si**-process, then, for any $\delta t > 0$, the increment process $\{Y_{\delta t}(t)\}_{t\geq 0} = \{\Delta_{\delta t} X_t\}_{t\geq 0} = \{X_{t+\delta t} - X_t\}_{t\geq 0}$ is a stationary process.

Proposition 7.1 - Gaussian case.
A Gaussian process is stationary if for any $t \in \mathbb{R}^+$, $\mathbb{E}[X_t] = \mu \in \mathbb{R}$ and the covariance function satisfies

$$\forall (t_1, t_2) \in \mathbb{R}^2,\ \gamma(t_1, t_2) = \gamma(t_2 - t_1). \tag{7.12}$$

7.3 Self-similar processes

Definition 7.5 - Self-similar processes.
A real valued stochastic process $X = \{X_t\}_{t\geq 0}$ is said to be *self-similar* with index $H > 0$, shortly H-**ss**, if for any $a > 0$:

$$\{X_{at}\}_{t\geq 0} \overset{\mathcal{L}}{=} \{a^H X_t\}_{t\geq 0}, \tag{7.13}$$

We observe that the transformation scales differently "space" and "time", one often prefers using the word self-affine process. The index H is said to be Hurst's exponent or scaling exponent of the process. In the same way, it is possible to define self-similarity for processes $\{X_t\}_{t\in\mathbb{R}}$.

Remark 7.4.
Observe that, if X is an H-**ss** process, then all the finite-dimensional distributions of X in $[0, \infty[$ are completely determined by the distribution in any finite real interval.

Proposition 7.2 - Non-degenerate process.
Let $X = \{X_t\}_{t\geq 0}$ be a non-degenerate stationary process (X is degenerate if $\forall t \geq 0, X_t = 0$, a.s.), then X cannot be a H-**ss** process.

We have the following proposition that relates stationary and self-similar processes:

Proposition 7.3 - Stationary and self-similar.
Let $\{X_t\}_{t\geq 0}$ a H-**ss** process, then the process

$$Y(t) = e^{-tH} X(e^t),\ t \geq 0, \tag{7.14}$$

is stationary. Moreover, if $Y = \{Y_t\}_{t\geq 0}$ is a stationary process, then

$$X(t) = t^H Y(\log t),\ t \geq 0, \tag{7.15}$$

is an H-**ss** process.

7.4 H-sssi processes

Definition 7.6 - H-sssi processes.
Let $(\Omega, \mathcal{F}, \mathcal{F}_t, P)$ be a filtered probability space. A stochastic process $X = \{X_t\}_{t\in I}$, $\mathcal{F}_t$-adapted, which is H-ss with stationary increments, is said to be an H-sssi process with exponent H.

Example 7.3.
A example of H-**sssi** process is the standard Brownian motion with $H = 1/2$.

Suppose now that X is a L^2 process to ensure that the variance of X_t for all $t \geq 0$ is finite. It a regular assumption regarding the many properties of such process, we highlight some of them

1. $X_0 = 0$ almost surely.

   Indeed, for any $a > 0$, one has:

   $$X(0) = X(a0) \overset{\mathcal{L}}{=} a^H X(0).$$

   Then, it follows that $\sigma(X(0))$ is degenerate, that is it is made up of only null-events or measure one events with respect to P. From (Prop. 2.6) follows that $X(0)$ is trivial, i.e. constant almost everywhere. But, if we set $X(0) = c$, then:

   $$\mathbb{E}(X(0)) = c = \mathbb{E}(X(a0)) = a^H c \Leftrightarrow c = 0.$$

2. If $H \neq 1$, then for any $t \geq 0$, $\mathbb{E}(X_t) = 0$.

   Indeed: $\mathbb{E}(X(2t)) = 2^H \mathbb{E}(X(t))$. On the other hand, from the stationary increments property and from property 1:

   $$2^H \mathbb{E}(X(t)) = \mathbb{E}(X(2t)) = \mathbb{E}(\underbrace{X(2t) - X(t)}_{=X(t), \text{in law}}) + \mathbb{E}(X(t)) = 2\mathbb{E}(X(t)) \Leftrightarrow \mathbb{E}(X(t)) = 0.$$

3. As $X(t+h) - X(h) \overset{\mathcal{L}}{=} X(t) - X(0)$ then $X(t-t) - X(-t) \overset{\mathcal{L}}{=} X(t) - X(0)$ One has:

   $$X(-t) \overset{\mathcal{L}}{=} -X(t),$$

   it follows from the first property and the stationarity of the increments:

   $$X(-t) \overset{a.s.}{=} X(-t) - X(0) \overset{\mathcal{L}}{=} X(0) - X(t) \overset{a.s.}{=} -X(t).$$

   The above property allows us to extend the definition of an H-sssi process to the whole real line: $\{X_t\}_{t\in\mathbb{R}}$.

4. Let $\sigma^2 = \mathbb{E}(X_1^2)$. Then,

   $$\mathbb{E}(X_t^2) = |t|^{2H}\sigma^2. \tag{7.16}$$

   Indeed, from the third property and the self-similarity:

   $$\mathbb{E}(X(t)^2) = \mathbb{E}(X^2(|t|\text{sign}(t))) = |t|^{2H}\mathbb{E}(X^2(\text{sign}(t))) = |t|^{2H}\mathbb{E}(X_1^2) = |t|^{2H}\sigma^2.$$

5. The covariance function of an H-sssi process X, with $\mathbb{E}(X_1^2) = \sigma^2$, turns out to be:

$$\gamma_H(s,t) = \mathrm{Cov}(X_s, X_t) = \mathbb{E}(X_s X_t) = \frac{\sigma^2}{2}\left(|t|^{2H} + |s|^{2H} - |t-s|^{2H}\right). \tag{7.17}$$

   As from the fourth property and the stationarity of the increments:

$$\mathbb{E}(X_s X_t) = \frac{1}{2}\left(\mathbb{E}X_s^2 + \mathbb{E}X_t^2 - \mathbb{E}(X_t - X_s)^2\right).$$

6. If $X = \{X_t\}_{t\in I}$ is an H-sssi process, then one must have $H \leq 1$.

   The constraint of the scaling exponent follows directly from the stationarity of the increments:

$$2^H \mathbb{E}|X_1| = \mathbb{E}|X_2| = \mathbb{E}|X_2 - X_1 + X_1| \leq \mathbb{E}|X_2 - X_1| + \mathbb{E}|X_1| = 2\mathbb{E}|X_1|,$$

   therefore, $2^H \leq 2 \Leftrightarrow H \leq 1$.

Remark 7.5.
The case $H = 1$ corresponds a.s. to $X_t = tX_1$. Indeed, on the $L^2(\Omega, P)$ norm:

$$\mathbb{E}(X_t - tX_1)^2 = \mathbb{E}(X_t^2 + t^2X_1^2 - 2tX_tX_1) = \sigma^2(2t^2 - 2t^2) = 0.$$

Remark 7.6.
The Brownian motion is a Gaussian H-**sssi** process with $H = 1/2$, so that its properties follow from those of a generic H-**sssi** process. In particular here we focus on the independence of the increments, which is due to (7.17) together with the definition of Gaussian process. We observe that for a generic H-**sssi** process this is no longer true. Indeed, let X_t be an H-**sssi** process with $H \leq 1$ and take:

$$X_{t_1} - X_{t_2}, X_{t_3} - X_{t_4}, \quad 0 \leq t_1 < t_2 \leq t_3 < t_4,$$

then, using (7.17), one has:

$$\begin{aligned}\mathbb{E}(X_{t_1} - X_{t_2})(X_{t_3} - X_{t_4}) &= \gamma_H(t_1,t_3) - \gamma_H(t_1,t_4) - \gamma_H(t_2,t_3) + \gamma_H(t_2,t_4),\\ &= \frac{\sigma^2}{2}\left(|t_1 - t_4|^{2H} + |t_2 - t_3|^{2H} - |t_1 - t_3|^{2H} - |t_2 - t_4|^{2H}\right),\end{aligned} \tag{7.18}$$

which is always zero if $H = \frac{1}{2}$, but is non-zero if $H \neq \frac{1}{2}$.

The above remark states that a generic adapted H-**sssi** process X is not a martingale provided that $H \neq \frac{1}{2}$.

7.5 Fractional Brownian motion

This part is mostly based on the several books : [23], [28], [21] and courses [27], [29].

7.5.1 Definition

We now introduce the *fractional Brownian motion* which is an extension of the traditional Brownian motion. Take $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t\geq 0}, \mathbb{P})$ be a probability space and we will consider processes on $\mathbb{R}^+$, but for a H-**sssi** process, we saw with property 3. that we can extend to $\mathbb{R}$.

Definition 7.7 - Fractional Brownian motion.
Let $B_H = \{B_H(t)\}_{t\geq 0}$, be a H-**sssi** Gaussian stochastic process with $H \in]0,1[$, defined on Ω adapted to $\{\mathcal{F}_t\}_t$, with $\sigma^2 = \mathbb{E}[B_H(1)^2]$. A such process is called *fractional Brownian motion* (shortly fBm) of order H.

- **Remark 3.20.** Let $\{B_H(t)\}_{t\geq 0}$, $0 < H < 1$, be a fractional Brownian motion, then:
  - If $H = \frac{1}{2}$, $\{B_{1/2}(t)\}_{t\geq 0}$ is the ordinary Brownian motion, with autocovariance function:

$$\gamma(s,t) = \begin{cases} \sigma^2 \min(|s|,|t|), & \text{if } \frac{|s|}{s} = \frac{|t|}{t}, \\ 0, & \text{otherwise.} \end{cases} \tag{7.19}$$

  - The case $H = 1$ won't be considered because it corresponds to the trivial case of a line with random slope (Remark 7.5):

$$B_1(t) = tB_1(1).$$

  In the following, we implicitly assume that the filtration is the natural filtration of the fBm.

- **Remark 3.21.** If one has $I = \mathbb{R}$ one speaks about two-sided fBm. On the other hand, if $I = \mathbb{R}^+$ one speaks about one-sided fBm.

Proposition 7.4 - Covariance function of the fBm.
By using (7.17), the covariance function of a fBm is

$$\mathbb{E}(B_H(t)B_H(s)) = \frac{\sigma^2}{2}\left(s^{2H} + t^{2H} - |t-s|^{2H}\right), \tag{7.20}$$

with $\sigma^2 = \mathbb{E}[B_H(1)^2]$.

7.5.2 Fractional Brownian motion characterization

The following proposition characterizes the fBm and gives a useful criterion which allows to recognize whether a given process is a fractional Brownian motion.

Proposition 7.5 - fBm characterization.
Let $X = \{X_t\}_{t\geq 0}$ be a stochastic process, defined on the probability space $(\Omega, \mathcal{F}, P)$, such that:

1. $P(X_0 = 0) = 1$.
2. X is a zero-mean Gaussian process such that, for any $t \geq 0$, $\mathbb{E}(X_t^2) = \sigma^2 t^\alpha$ for some $\sigma > 0$ and $0 < \alpha < 2$.
3. X is a si-process.

Then, $\{X_t\}_{t\geq 0}$ is a (one-sided) fractional Brownian motion of order $H = \alpha/2$.

Proof. Since X is a zero-mean Gaussian process, its finite-dimensional distributions are completely characterized by its autocovariance function. Given that, for any $t > 0$:

$$\mathbb{E}(X_t^2) = \sigma^2 |t|^\alpha$$

and X has stationary increments, it follows that the autocovariance function is given by (7.17), which is the autocovariance of a fBm with $H = \alpha/2$. □

The corollary below follows immediately:

Corollary 7.1 - Equivalent statements.
Let $X = \{X_t\}_{t\geq 0}$ be a stochastic process defined on the probability space $(\Omega, \mathcal{F}, P)$. Let $0 < H < 1$ and $\sigma^2 = \mathbb{E}(X_1^2)$. The following statements are equivalent:

1. X is an H-**sssi** Gaussian process.
2. X is a (one-sided) fractional Brownian motion with scaling exponent H.
3. X is Gaussian with zero mean and covariance function given by (7.20).

Remark 7.7.
Fractional Brownian motion is non-Markovian provided that $H \neq 1/2$.

7.5.3 Fractional Brownian motion trajectories

Proposition 7.6 - Continuity.
For any $\omega \in \Omega$, the function

$$t \to B_H(t, \omega), \tag{7.21}$$

is continuous.

Proof. With the stationarity of the increment :

$$B_H(t_2) - B_H(t_1) \overset{\mathcal{L}}{=} B_H(t_2 - t_1) - B_H(0)$$

As $B_H(0) = 0$, we obtain :

$$B_H(t_2) - B_H(t_1) \overset{\mathcal{L}}{=} B_H(t_2 - t_1),$$

then ,

$$B_H(t_2 - t_1) \overset{\mathcal{L}}{=} (t_2 - t_1)^H B_H(1),$$

which leads to

$$\mathbb{E}\left[|(t_2 - t_1)^H B_H(1)|^p\right] = (t_2 - t_1)^{pH} \mathbb{E}\left[|B_H(1)|^p\right].$$

Choose $p > \frac{1}{H}$; thus, for any $t_1, t_2 \geq 0$, using the self-similarity and the stationarity of the increments, one has:

$$\mathbb{E}\,|B_H(t_2) - B_H(t_1)|^p = \mathbb{E}\,|B_H(1)|^p\,|t_2 - t_1|^{pH},$$

which, by using the Kolomogorov continuity theorem, completes the proof. □

Proposition 7.7 - Non-differentiability.
For any $\omega \in \Omega$, the function

$$t \to B_H(t, \omega), \tag{7.22}$$

is not differentiable in the L^2-norm.

Proof. Indeed, for any $t_1, t_2 \geq 0$:

$$\mathbb{E}\left(\frac{B_H(t_2) - B_H(t_1)}{t_2 - t_1}\right) = \sigma^2 |t_2 - t_1|^{2H-2} \to +\infty, \text{ as } t_1 \to t_2. \tag{7.23}$$

□

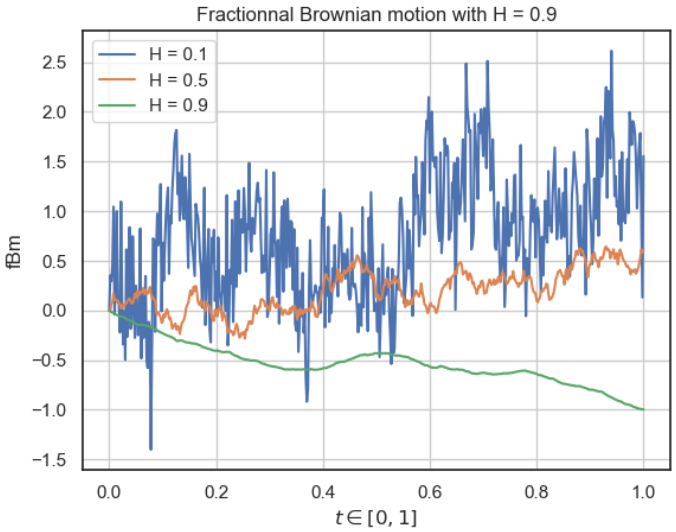


Figure 7: Trajectories of the fBm for some values of H

Here are some trajectories for different values of H, when $H = 0.5$, it's the standard Brownian motion. For $H > 0.5$, the trajectory is smoother and more regular.
For $H < 0.5$, the trajectory is very irregular with large variations.

7.5.4 Discrete time fractional Gaussien noise

Consider a time grid $I_{\delta t} = \{t_k = k\delta t | k \in \mathbb{N}\}$ with $\delta t > 0$.Then, we can define the discrete time process $\{B_H(t_k)\}_{k\in\mathbb{N}}$ defined as the restriction on $I_{\delta t}$ of the continuous process B_H. We will denote $B_H(k) := B_H(t_k)$.

Definition 7.8 - Fractional Gaussian noise.
The discrete time increment process of B_H, namely process Z such that

$$Z_k = \Delta B_H(k) = B_H(k+1) - B_H(k), \tag{7.24}$$

is called discrete time *fractional Gaussian noise* (fGn) of order H.

The following properties for Z :

1. Z is stationary.
2. For any $k \geq 1$, $\mathbb{E}[Z_k] = 0$.
3. For any $k \geq 1$, $\mathbb{E}[Z_k^2] = \sigma^2 = \mathbb{E}[B_H(1)]^2$.
4. The covariance function is

$$\gamma(k) = \mathbb{E}[Z_t Z_{t+k}] = \frac{\sigma^2}{2}(|k+1|^{2H} - 2|k|^{2H} + |k-1|^{2H}) = \frac{\sigma^2}{2}\Delta^2|k|^2, \tag{7.25}$$

with Δ^2 the second operator finite differencing operator : $\Delta^2 X_k = X_{k+1} - 2X_k + X_{k-1}$.

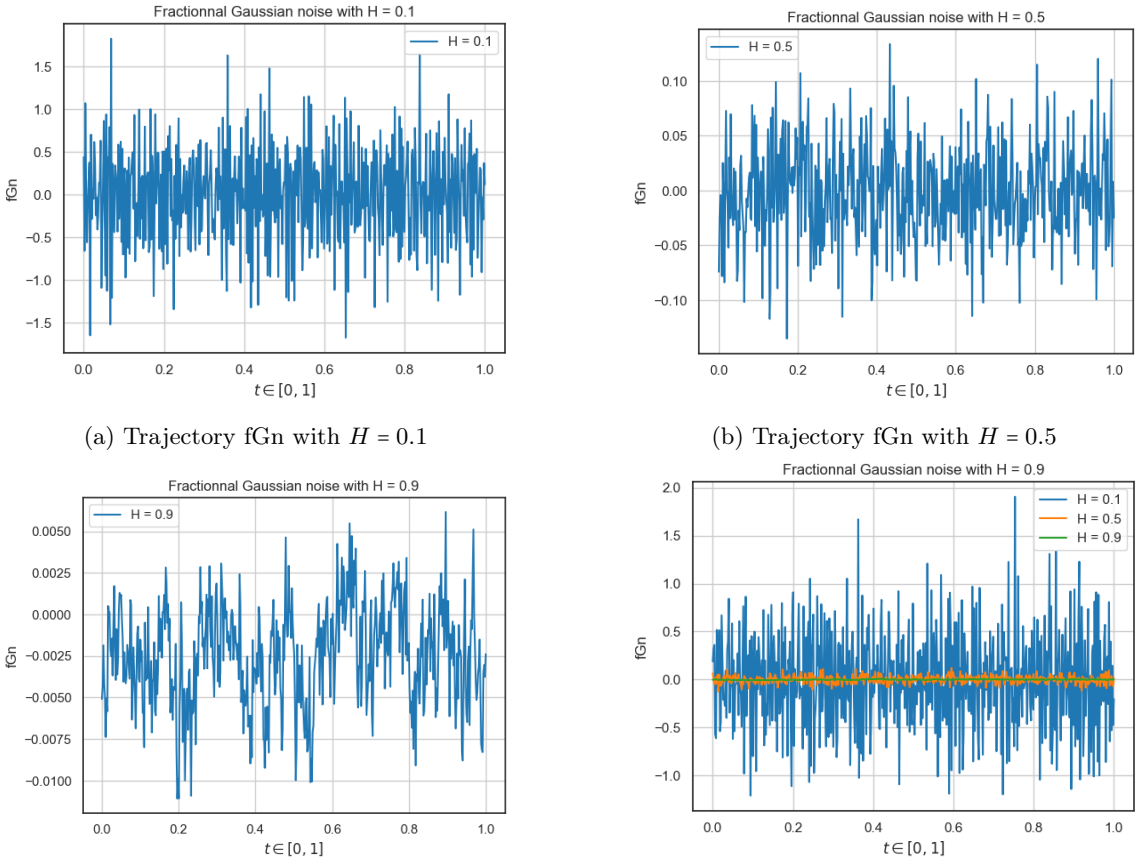
With also have some interesting properties about the covariance function.

Proposition 7.8 - Covariance function .
Let $k \neq 0$. Therefore,

1. $\gamma(k) = 0$, if $H = 1/2$ (purely random region),
2. $\gamma(k) < 0$, if $H \in]0, 1/2[$,
3. $\gamma(k) > 0$, if $H \in]1/2, 1[$.

Proposition 7.9 - Asymptotic equivalent.
If $H \neq 1/2$, then :

$$\gamma(k) \underset{k\to+\infty}{\sim} \sigma^2 H(2H-1)|k|^{2H-2}. \tag{7.26}$$



(a) Trajectory fGn with $H = 0.1$

(b) Trajectory fGn with $H = 0.5$

(c) Trajectory fGn with $H = 0.9$

(d) Superposition of trajectories of fGn

Figure 8: Fractionnal Gaussian noise

- **Trajectory with** $H = 0.1$ (Figure 8a):
  - When the Hurst index H is low (e.g., $H = 0.1$), the trajectory of fractional Gaussian noise exhibits anti-persistent behavior.
  - This means that increases in the time series are likely to be followed by decreases, and vice versa, leading to a more erratic and choppy appearance.
  - The plot shows rapid fluctuations around the mean, indicating a lack of long-term memory in the process.
- **Trajectory with** $H = 0.5$ (Figure 8b):
  - When $H = 0.5$, the fractional Gaussian noise reduces to standard Gaussian white noise.
  - This trajectory exhibits no memory, meaning each point in the time series is independent of the others.

  - The plot shows a random walk-like behavior with no apparent trends or patterns, reflecting the absence of long-term correlations.

- **Trajectory with** $H = 0.9$ (Figure 8c):
  - When the Hurst index H is high (e.g., $H = 0.9$), the trajectory exhibits persistent behavior.
  - This means that increases in the time series are likely to be followed by further increases, and decreases by further decreases, leading to smoother and more trend-like trajectories.
  - The plot shows a more structured and less erratic behavior, indicating the presence of long-term memory in the process.

- **Superposition of Trajectories** (Figure 8d):
  - The superposition of trajectories with different Hurst indices ($H = 0.1$, $H = 0.5$, and $H = 0.9$) provides a comparative visualization of how the Hurst index affects the behavior of fractional Gaussian noise.
  - Trajectories with lower H values show more rapid fluctuations and less structure, while those with higher H values exhibit smoother and more persistent trends.
  - This comparison highlights the impact of the Hurst index on the memory and correlation structure of the time series.

7.5.5 Integral representation of the fractional Brownian motion

Let $\{B(t), t \in \mathbb{R}\}$ be a Brownian motion defined on the probability space $(\Omega, \mathcal{F}, \mathcal{F}_t, P)$. We introduce the following notation: $x^+ = \max(x, 0)$. The proposition below provides a first representation of fractional Brownian motion in terms of stochastic integrals of Brownian motion. Mandelbrot and Van Ness obtained this representation in [22].

Proposition 7.10 - Moving average representation.
Let $\{B_H(t), t \geq 0\}$ be a standard one-sided fractional Brownian motion with $0 < H < 1$. Then, for any $t \geq 0$

$$B_H(t) = \int_{\mathbb{R}} f_t(x) dB(x) = \frac{1}{C_1(H)} \int_{\mathbb{R}} \left((t-x)_+^{H-\frac{1}{2}} - (-x)_+^{H-\frac{1}{2}} \right) dB(x), \tag{7.27}$$

where:

$$C_1(H) = \left(\int_0^\infty \left((1+x)^{H-\frac{1}{2}} - x^{H-\frac{1}{2}} \right)^2 dx + \frac{1}{2H} \right)^{1/2} \tag{7.28}$$

$$= \frac{\Gamma(H+\frac{1}{2})}{\left(\Gamma(2H+1)\sin(\pi H)\right)^{1/2}}. \tag{7.29}$$

This representation (7.27) is called moving average representation.

Remark 7.8.
The definition can write with another form

$$B_H(t) = \frac{1}{C_1(H)} \int_{-\infty}^{t} ((t-u)^{H-1/2} - (-u)_+^{H-1/2}) dB(u) \tag{7.30}$$

We observe that for $H = \frac{1}{2}$ one obtains,

$$B(t) = \int_0^t dB_x, \quad t \geq 0, \tag{7.31}$$

that is a standard Bm (see Exa. 4.2).

The function $f_t(x)$ is called representation kernel. Moreover, because the integration is carried up to time $t \geq 0$, representation (5.1) is called causal.

We can provide several fBm representations. In fact,

Proposition 7.11 - Integral representation of the fBm.
For any couple of real numbers a, b, the stochastic integral:

$$\int_{\mathbb{R}} \left[a \left((t-x)_+^{H-\frac{1}{2}} - (-x)_+^{H-\frac{1}{2}} \right) + b \left((t-x)_-^{H-\frac{1}{2}} - (-x)_-^{H-\frac{1}{2}} \right) \right] dB(x), \tag{7.32}$$

where $x_- = -\min(x, 0)$, defines (up to a multiplicative constant) a fractional Brownian motion representation.

Remark 7.9.
Setting $a = \frac{1}{C_1(H)}$ and $b = 0$, we recover the moving average representation.

When $a = b = 1$, (7.32) reduces to the so called well-balanced representation:

$$B_H(t) = \int_{\mathbb{R}} \left(|t-x|^{H-\frac{1}{2}} - |x|^{H-\frac{1}{2}} \right) dB(x). \tag{7.33}$$

The most common way of representing the fBm is with this form

$$B_H(t) = \int_{\mathbb{R}} K_H(t,x) dB(x) = \frac{1}{\Gamma(H+1/2)} \int_{\mathbb{R}} \left((t-x)_+^{H-\frac{1}{2}} - (-x)_+^{H-\frac{1}{2}} \right) dB(x), \tag{7.34}$$

7.6 Simulation of the fractional Brownian motion

This part is based on [13] and[16] .

7.6.1 Simulation of Gaussian vectors

Take $\mathbf{X} = (X_1, ..., X_n) \in (L_2)^n$ a non-degenerate Gaussian vector such that

$$\begin{cases} \mathbb{E}[X] = \mu \in \mathbb{R}^n, \\ \Sigma = (\mathrm{Cov}(X_i, X_j))_{i,j=1...n}, \end{cases} \tag{7.35}$$

with Σ a semi-definite matrix. Now we use Cholesky decomposition for semi-definite matrix.

Proposition 7.12 - Cholesky decomposition.
For $S \in S_n^{++}(\mathbb{R})$ (semi-definite matrix), there exists a lower triangular matrix. L such that $S = LL^T$.

We recall the pseudo-code algorithm for this decomposition :

Algorithm 1 Cholesky Decomposition

```
function CHOLESKY(A)
    L ← zeros(n, n)                          ▷ Initialize matrix L with zeros
    n ← number of rows in A
    for i ← 1 to n do
        for j ← 1 to i do
            sum ← Σ_{k=1}^{j-1} L[i,k] · L[j,k]
            if i = j then
                L[i,j] ← √(A[i,i] − sum)
            else
                L[i,j] ← (A[i,j] − sum)/L[j,j]
            end if
        end for
    end for
    return L
end function
```

It already exists a *python* function in the library *numpy* with the extension *.linalg* : *numpy.linalg.cholesky*, see Numpy documentation.

To compute numerically the m samples of Gaussian vector we use this result.

Proposition 7.13 - Linear transformation.
Let $A \in GL_n(\mathbb{R})$ a inversible matrix, $b \in \mathbb{R}^n$ and $Z_1, ..., Z_n$ a iid standard normal random variables. In particular, $\mathbf{Z}$ is a Gaussian vector and the vector

$$\mathbf{X} = b + A\mathbf{Z}, \tag{7.36}$$

is a vector Gaussian of mean $b = \mathbb{E}[\mathbf{X}]$ and of covariance matrix $\Sigma = AA^T$.

It follows this algorithm to simulate m samples of size n of Gaussian vectors.

Algorithm 2 Generate Multivariate Gaussian Samples

Require: A positive definite covariance matrix $\Sigma \in \mathbb{R}^{n \times n}$
Require: A mean vector $\mu \in \mathbb{R}^n$
Require: An integer $m \in \mathbb{N}$ representing the number of samples
Ensure: m samples of the multivariate Gaussian with mean μ and covariance Σ
function MultivariateGaussianSamples(Σ, μ, m)
    Compute L such that $\Sigma = LL^T$ ▷ Cholesky decomposition
    Initialize an empty list S to store samples
    for $i \leftarrow 1$ to m **do**
        Simulate n independent samples from the standard normal distribution: $Z = (Z_1, \ldots, Z_n)$
        Compute the sample: $X = \mu + LZ$
        Append X to S
    end for
    return S
end function

Remark 7.10.
Since fBm is a centered Gaussian process with a known covariance matrix, we start by simulating a centered Gaussian vector. Then, we perform a Cholesky decomposition of the covariance matrix to find its square root. By multiplying this root matrix by the centered Gaussian vector, we obtain a Gaussian vector whose correlations correspond to those of fBm.

Remark 7.11.
This method is probably the most intuitive one but as a high cost of order $O(n^3) + mO(n^2)$, the first term is due to the Cholesky decomposition and the second one is from the matrix multiplication $L\mathbf{Z}$.

7.6.2 Simulation of stationnary Gaussian vector

In this section, we are going to discuss an algorithm that allows simulating paths from certain stationary Gaussian processes with a cost of the order $m \cdot O(n \log n)$. The algorithm is based on a matrix decomposition $S = AA^T$ (where S is embedded in Σ) that can be computed more efficiently. But more importantly, it also allows evaluating the matrix multiplication AZ more efficiently.
Let $\{t_0, t_1, \dots, t_n\} \subset \mathbb{R}$ such that $t_k = t_0 + k\delta t$ for $k = 0, 1, \dots, n$. Denote by Σ the covariance matrix of the Gaussian vector $\mathbf{X} \coloneqq (X_{t_0}, \dots, X_{t_n})$, where $\mathbf{X}$ is a stationary Gaussian process with covariance function C. Define $c_k \coloneqq C(t_0, t_k)$ for $k = 0, 1, \dots, n$. Then the covariance matrix is of the form

$$\Sigma = \begin{pmatrix} c_0 & c_1 & \cdots & c_n \\ c_1 & c_0 & \cdots & c_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ c_n & c_{n-1} & \cdots & c_0 \end{pmatrix}. \tag{7.37}$$

A matrix of this form is called *Toeplitz*. This form is not yet what we need to obtain a more efficient decomposition. Consider the following matrix that is obtained from reflecting and shifting the first row of Σ:

$$\Pi = \begin{pmatrix} c_0 & c_1 & \cdots & c_{n-1} & c_n & c_{n-1} & \cdots & c_2 & c_1 \\ c_1 & c_0 & \cdots & c_n & c_{n-1} & c_n & \cdots & c_3 & c_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{n-1} & c_n & \cdots & c_0 & c_1 & c_2 & \cdots & c_{n-1} & c_n \\ c_n & c_{n-1} & \cdots & c_1 & c_0 & c_1 & \cdots & c_{n-2} & c_{n-1} \\ c_{n-1} & c_n & \cdots & c_2 & c_1 & c_0 & \cdots & c_{n-3} & c_{n-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ c_1 & c_2 & \cdots & c_n & c_{n-1} & c_{n-2} & \cdots & c_1 & c_0 \end{pmatrix} \in \mathbb{R}^{2n \times 2n}. \tag{7.38}$$

This matrix satisfies the property that the ith row can be obtained by periodically shifting the first row i-entries towards the right. Such matrices are called *circulant*. Clearly, our original covariance matrix Σ is embedded in the upper-left block of Π.
We define

$$\forall l < m,\ \Sigma_{l,m} = \begin{pmatrix} c_l & c_{l+1} & \cdots & c_m \\ c_{l+1} & c_l & \cdots & c_{m-1} \\ \vdots & \vdots & \ddots & \vdots \\ c_m & c_{m-1} & \cdots & c_l \end{pmatrix} \in \mathbb{R}^{(m-l+1)\times(m-l+1)}. \tag{7.39}$$

With $\Sigma_{0,n} = \Sigma$. Similarly, we define the matrix $\Delta_{l,m}$ as

$$\forall l < m,\ \Delta_{l,m} = \begin{pmatrix} c_m & \cdots & c_l \\ \vdots & \ddots & \vdots \\ c_l & \cdots & c_m \end{pmatrix} \in \mathbb{R}^{(m-l+1)\times(m-l+1)}, \tag{7.40}$$

We those new notations we have

$$\Pi = \begin{pmatrix} & & & c_{n-1} & \dots & c_1 \\ & \Sigma_{0,n} & & & \Sigma_{1,n} & \\ & & & c_1 & \dots & c_{n-1} \\ c_{n-1} & & c_1 & & & \\ \vdots & \Delta_{2,n} & \vdots & & \Sigma_{0,n-2} & \\ c_1 & & c_{n-1} & & & \end{pmatrix} \in \mathbb{R}^{2n \times 2n}. \tag{7.41}$$

We can study the matrix Π and end up with this following result.

Proposition 7.14 - Eigenvalues and eigenvector of Π.
Define the complex number $\omega_n = e^{-\pi i/n}$ and the vector $q_k = \left(\frac{1}{\sqrt{2n}}\omega_n^{jk}\right)_{o\leq j\leq 2n-1}$ for $k = 0, ..., 2n-1$. Then for any $k = 0, ..., 2n-1$, q_k is an eigenvector of Π with eigenvalue $\lambda_k = \sum_{j=0}^{2n-1}\gamma_j\omega_n^{jk}$. With $\gamma = (c_0, c_1, ..., c_{n-1}, c_n, c_{n-1}, ..., c_1)$.

Proof. Let $k = 0...2n-1$ fixed. As the first line vector of the matrix Π is γ we easily verify that

$$(\Pi q_k)_1 = (\lambda_k q_k)_1. \tag{7.42}$$

We now focus on the line between $l = 1...n-1$ which corresponds to the submatrix:

$$\tilde{\Pi} = \begin{pmatrix} c_1 & c_0 & \cdots & c_n & c_{n-1} & c_n & \cdots & c_3 & c_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{n-1} & c_n & \cdots & c_0 & c_1 & c_2 & \cdots & c_{n-1} & c_n \end{pmatrix} \in \mathbb{R}^{(n-1)\times(2n-1)}$$

Let $l = 1...n-1$, the l−th line is

$$(\tilde{\Pi}q_k)_l = (\underbrace{c_l\cdots c_0}_{l+1 \text{ terms}} \quad \underbrace{c_1\cdots c_{n-l}}_{n-l \text{ terms}} \quad \underbrace{c_{n-l+1}\cdots c_{n-1}}_{l-1 \text{ terms}} \quad \underbrace{c_n\cdots c_{l+1}}_{n-l \text{ terms}})q_k =: L_l q_k. \tag{7.43}$$

We want to verify that

$$(\tilde{\Pi}q_k)_l = (\lambda_k q_k)_l = \lambda_k\omega_n^{lk}. \tag{7.44}$$

We verify that the vector L_l is a vector of dimension $(l+1)+(n-l)+(l-1)+(n-l) = 2n$. This leads to the following expression

$$\sqrt{2n}(\tilde{\Pi}q_k)_l = \sum_{j=0}^{l} c_{l-j}\omega_n^{jk} + \sum_{j=1}^{n-l} c_j\omega_n^{(l+j)k} + \sum_{j=1}^{l-1} c_{n-l+j}\omega_n^{(n+j)k} + \sum_{j=0}^{n-(l+1)} c_{n-j}\omega_n^{(n+j+l)k} := I_1 + I_2 + I_3 + I_4. \tag{7.45}$$

Each term corresponds to the previous decomposition : $I_1 \Leftrightarrow c_l....c_0$, $(l+1)$ terms and so on. One can easily establish the following properties

$$\forall l, m = 0...2n-1,\ \omega_n^{(l+m)k} = \omega_n^{lk}\omega_n^{mk} \tag{7.46}$$

$$\forall j = 0...n,\ \gamma_j = c_j \tag{7.47}$$

$$\forall j = n+1...2n-1,\ \gamma_j = c_{2n-j} \tag{7.48}$$

Before continuing, it is better to compute the other side which is

$$\begin{aligned} \sqrt{2n}(\lambda_k q_k)_l &= \left(\sum_{j=0}^{2n-1}\gamma_j\omega_n^{jk}\right)\omega_n^{lk} = \sum_{j=0}^{n}\gamma_j\omega_n^{(j+l)k} + \sum_{j=n+1}^{2n-1}\gamma_j\omega_n^{(j+l)k}, \quad (7.46) \\ &= \sum_{j=0}^{n} c_j\omega_n^{(j+l)k} + \sum_{j=n+1}^{2n-1} c_{2n-j}\omega_n^{(j+l)k} := S_1 + S_2, \quad (7.47), (7.48) \end{aligned} \tag{7.49}$$

Lets do some change of index and redevelopment, we obtain the following expressions

$$I_2 = \omega_n^{lk} \sum_{j=1}^{n-l} c_j \omega_n^{jk}, \tag{7.50}$$

$$I_3 = \omega_n^{lk} \sum_{p=n-l+1}^{n-1} c_p \omega_n^{pk}, \quad (p = n - l + j) \tag{7.51}$$

$$I_1 = \sum_{p=2n-l}^{2n} c_{2n-p} \omega_n^{(l-2n+p)k}, \quad (l - j = 2n - p) \tag{7.52}$$

$$I_4 = \sum_{p=n}^{2n-(l+1)} c_{2n-p} \omega_n^{(p+l)k}, \quad (n - j = 2n - p) \tag{7.53}$$

We directly obtain from (7.52) and (7.50) that

$$I_2 + I_3 = \omega_n^{lk} \sum_{j=1}^{n-1} c_j \omega_n^{jk}. \tag{7.54}$$

Recall that $\forall \theta \in \mathbb{R},\ \omega_n^{(\theta+2n)k} = \exp(-ik\pi(\theta + 2n)/n) = \exp(ik\pi\theta/n) = \omega_n^{k\theta}$, then

$$I_1 + I_4 = \sum_{p=n}^{2n} c_{2n-p} \omega_n^{(p+l)k} = c_0 \underbrace{\omega_n^{(2n+l)k}}_{=\omega_n^{lk}} + c_n \omega_n^{(n+l)k} + S_2. \quad (7.46), (7.49) \tag{7.55}$$

And then,

$$\begin{aligned} (I_2 + I_3) + (I_1 + I_4) &= \omega_n^{lk} \sum_{j=1}^{n-1} c_j \omega_n^{jk} + c_0 \omega_n^{lk} + c_n \omega_n^{(n+l)k} + S_2, \\ &= \omega_n^{lk} \sum_{j=0}^{n+1} c_j \omega_n^{jk} + S_2 = S_1 + S_2. \quad (7.49) \end{aligned} \tag{7.56}$$

This last show that

$$\forall l = 1...n-1,\ (\tilde{\Pi} q_k)_l = (\lambda_k q_k)_l. \tag{7.57}$$

The method is similar for the rest of the lines ($l \geq n$) and can be easily do following the previous demonstration. □

We move on with another proposition in order to properly establish our algorithm.

Proposition 7.15 - Unitary matrix.
The matrix $Q = (\frac{1}{\sqrt{2n}} \omega_n^{jk})_{0 \leq j,k \leq 2n-1}$ is unitary, i.e., $QQ^* = I_{2n-1}$. Q^* here is the transpose conjugate of Q.

Proof. Define the k-th columns of Q by q_k. Q is unitary if is columns form an orthonormal basis. Let $k, l = 0...2n-1$ fixed. Then, we compute the inner product of q_k and q_l

$$\langle q_k, q_l \rangle_{\mathbb{C}^{2n-1}} = \frac{1}{2n} \sum_{m=0}^{2n-1} \omega_n^{mk} \overline{\omega_n^{lk}} = \frac{1}{2n} \sum_{m=0}^{2n-1} \omega_n^{(m-l)k}. \tag{7.58}$$

If $k = l$, $\langle q_l, q_k \rangle = \|q_k\| = 1$ and for $k \neq l$, $\langle q_l, q_k \rangle = 0$, as the ω_n is a root of unity. We can also notice that

$$l \neq k,\ \langle q_l, q_k \rangle = \sum_{m=0}^{2n-1} (e^{-\pi i(k-l)/n})^m = \frac{1 - (e^{-\pi i(k-l)/n})^{2n}}{1 - e^{-\pi i(k-l)/n}} = 0. \tag{7.59}$$

As $(e^{-\pi i(k-l)/n})^{2n} = e^{-2\pi i(k-l)} = 0$. It shows that $\langle q_l, q_k \rangle = \delta_{k,l}$ and the basis $(q_0, ..., q_{2n-1})$ is orthonormal, that implies Q unitary. □

Proposition 7.16 - Covariance matrix.
Define $A = Q\text{diag}(\sqrt{\lambda_1}, ..., \sqrt{\lambda_{2n-1}})Q^*$, the matrix A is real and such that $\Pi = AA^T$.

Proof. Let $k = 0, ..., 2n-1$ denote $N = 2n$, the (m,n)-th entry of A is given by:

$$A_{mn} = \sum_{k=0}^{N-1} Q_{mk}\sqrt{\lambda_k}Q^*_{kn}$$

We leads to

$$A_{mn} = \frac{1}{N}\sum_{k=0}^{N-1} \omega^{(m-n)k}\sqrt{\lambda_k}$$

Then notice that $\lambda_k = \overline{\lambda_{N-k}}$ as $\omega_n^{(N-k)l} = \omega_n^{-kl} = \overline{\omega_n^{kl}}$. For each term in the sum involving k, there is a corresponding term involving $N-k$, we have:

$$\omega^{(m-n)k}\sqrt{\lambda_k} + \overline{\omega^{(m-n)k}\sqrt{\lambda_k}} = 2\Re\left(\omega^{(m-n)k}\sqrt{\lambda_k}\right).$$

Thus, each pair of terms in the sum contributes a real number, and the entire sum A_{mn}.By explicitly computing the entries A_{mn} and considering the complex conjugate pairs of eigenvalues and exponential terms, we see that the imaginary parts cancel out. Therefore, the matrix A is real.

Now, compute AA^T, with $\Lambda = \text{diag}(\sqrt{\lambda_1}, ..., \sqrt{\lambda_{2n-1}})$:

$$AA^T = (Q\sqrt{\Lambda}Q^*)(Q\sqrt{\Lambda}Q^*) = Q\sqrt{\Lambda}(Q^*Q)\sqrt{\Lambda}Q^*$$

Since Q is unitary, $Q^*Q = I$:

$$AA^T = Q\sqrt{\Lambda}I\sqrt{\Lambda}Q^* = Q\Lambda Q^*$$

Given that Λ contains the eigenvalues of Π, and $\Pi = Q\Lambda Q^*$, we have:

$$AA^T = Q\Lambda Q^* = \Pi$$

□

Know that we have showed that $\Pi = AA^T$ with A a real matrix, Π is semi-definite and we can consider it as a covariance matrix. Π can be expressed as an orthogonal decomposition such that

$$\Pi = Q\text{diag}(\lambda_0, ..., \lambda_{2n-1})Q^T. \tag{7.60}$$

If the eigenvalues are real and non-negative then Π is semi-definite and then it can be a covariance matrix.

Remark 7.12.
One can notice that if $\hat{X}$ is a Gaussian vector with covariance matrix Π, then $X = \hat{X}_{1,n+1}$ (the $n+1$ first coordinates) is a Gaussian vector of covariance matrix Σ

. Note that the multiplication of a vector $x \in \mathbb{R}^{2n}$ with the matrix Q corresponds to a discrete Fourier transform, i.e.,

$$(Qx)_k = \frac{1}{\sqrt{2n}} \sum_{j=0}^{2n-1} x_j \omega_n^{jk} = \frac{1}{\sqrt{2n}} \sum_{j=0}^{2n-1} x_j e^{-2\pi i \frac{jk}{2n}} = \frac{1}{\sqrt{2n}} \mathrm{DFT}(x)_k. \tag{7.61}$$

Analogously, the multiplication with Q^* corresponds to an inverse discrete Fourier transform, i.e.,

$$(Q^*x)_k = \frac{1}{\sqrt{2n}} \sum_{j=0}^{2n-1} x_j \omega_n^{-jk} = \frac{1}{\sqrt{2n}} \sum_{j=0}^{2n-1} x_j e^{2\pi i \frac{jk}{2n}} = \sqrt{2n} \mathrm{IDFT}(x)_k. \tag{7.62}$$

These multiplications can therefore be algorithmically computed using the *fast Fourier transform* (FFT), see Numpy documentation or Scipy documentation for examples in *python* . While the standard evaluation of Q^*x would cost $O(n^2)$, using the FFT this product can be evaluated with costs $O(n \log n)$. We are finally ready to formulate an efficient algorithm for the simulation of the Gaussian process X on the grid $\{t_0 + k\delta t | k = 0..n\}$, given that the associated matrix Π is positive semi-definite.
The algorithm is

Algorithm 3 Efficient algorithm for simulate m sample of stationary Gaussian vector

Require: Mean $\mu \in \mathbb{R}$ and covariance function C of a stationary Gaussian process
Require: $\{t_0, \dots, t_n\}$ a regular time grid
Require: m the number of samples
Ensure: m sample paths of the Gaussian process on the grid $\{t_0, \dots, t_n\}$
1: Define $\gamma = (c_0, \dots, c_{n-1}, c_n, c_{n-1}, \dots, c_1) \in \mathbb{R}^{2n}$ with $c_k := C(t_0, t_k)$ for all $k = 0, \dots, 2n-1$.
2: Compute $\lambda = \mathrm{DFT}(\gamma)$ and verify that $\lambda_k > 0$ for all $k = 0, \dots, 2n-1$.
3: **for** $i = 1$ to m **do**
4: Simulate $2n$ independent samples from the standard normal distribution $Z = (Z_0, \dots, Z_{2n-1})$.
5: Compute $\hat{X} = \mathrm{DFT}\left(\mathrm{diag}\left(\sqrt{\lambda_0}, \dots, \sqrt{\lambda_{2n-1}}\right) \cdot \mathrm{IDFT}(Z)\right)$.
6: Save new sample path $\mu + \hat{X}_{0:n}$.
7: **end for**

Remark 7.13.
The whole point of the algorithm is to avoid using the Choleski decomposition and the multiplication matrix which brings high cost. We do these two steps in one operation which is

$$\hat{X} = \mathrm{DFT}\left(\mathrm{diag}\left(\sqrt{\lambda_0}, \dots, \sqrt{\lambda_{2n-1}}\right) \cdot \mathrm{IDFT}(Z)\right).$$

In fact, this line is equivalent to the line,

$$\hat{X} = AZ.$$

As

1. We compute $\mathrm{IDFT}(Z) \Leftrightarrow Q^* Z$, by (7.62).
2. Then we compute $\mathrm{diag}\left(\sqrt{\lambda_0}, \dots, \sqrt{\lambda_{2n-1}}\right) \cdot \mathrm{IDFT}(Z) \Leftrightarrow \Lambda Q^* Z$.
3. The we compute $\mathrm{DFT}\left(\mathrm{diag}\left(\sqrt{\lambda_0}, \dots, \sqrt{\lambda_{2n-1}}\right) \cdot \mathrm{IDFT}(Z)\right) \Leftrightarrow (Q\Lambda Q^*)Z = AZ$, , by (7.62).

And at the end, we only take $\hat{X}_{0:n}$ which is the $n+1$ first point of the $\hat{X}$ which represents X as Remark 7.12 explains.

7.6.3 Simulation of fractional Brownian motion

Know, we have all the tools to build an algorithm to simulate sample of fBm. We will use the discrete time fractional increment (or noise), see section 7.5.4. We start with the following result and we keep the previous notation of the section 7.5.4.

Proposition 7.17 - Zero mean process.
Let $\Delta > 0$ and define the zero mean Gaussian vector $\tilde{\mathbf{X}}$ by

$$\tilde{\mathbf{X}}_k := \Delta^H \sum_{j=1}^{k} Z_j. \tag{7.63}$$

with $Z_j = B_H(t_{j+1}) - B_h(t_j)$, with $t_j = j\Delta$. Then,

- $\mathbb{E}[\tilde{\mathbf{X}}_m \tilde{\mathbf{X}}_l] = \mathbb{E}[B_H(\Delta_m) B_H(\Delta_n)]$.
- $\tilde{\mathbf{X}}$ represent an evaluation of an fBm on the time grid $\{0.\Delta, \Delta, ..., n\Delta\}$

Based on this proposition, the algorithm is

Algorithm 4 Simulation of m samples of fractional Brownian motion

Require: H the Hurst parameter
Require: n the size of the time grid
Require: Δ the mesh of the time grid
Require: m the number of samples
Ensure: m sample paths of the fractional Brownian motion on the grid $\{0, \Delta, 2\Delta, \dots, n\Delta\}$

1: Define $c = \frac{1}{2}\left(|k+1|^{2H} + |k-1|^{2H} - 2|k|^{2H}\right)$ for $0 \le k \le n-1$.
2: Use Algorithm 3 (3) with c to simulate $(\Delta X^{(k)})_{1\le k\le m}$, where each $\Delta X^{(k)}$ is a sample of the increment process on $\{0, \Delta, 2\Delta, \dots, (n-1)\Delta\}$.
3: For each $k = 1, \dots, m$, save a new sample $X^{(k)} := \left[\Delta^H \sum_{l=1}^{j} X_{l-1}^{(k)}\right]_{0\le j\le n}$.
4: Return $(X^{(k)})_{1\le k\le m}$.

Remark 7.14.
Line 1 : of the algorithm is due to the expression of the covariance matrix, see (7.25). Once we have the right covariance matrix, we use algorithm 3 (3) and get a sample of discrete fractional time increment (line 2 :).
Then, we use Proposition 7.17 and get m evaluations of the fBm with line 3.

References

[1] E. Abi Jaber, *Volterra processes in finance*, PhD thesis, Ecole polytechnique, 2024.

[2] E. Abi Jaber, M. Larsson, and S. Pulido, *Affine volterra processes*, The Annals of Applied Probability, (2019).

[3] A. Alfonsi and A. Kebaier, *Approximation of stochastic volterra equations with kernels of completely monotone type*, Mathematics of Computation, 93 (2024), pp. 643–677.

[4] E. Alòs, O. Mazet, and D. Nualart, *Stochastic calculus with respect to gaussian processes*, The Annals of Probability, 29 (2001), pp. 766–801.

[5] E. Azmoodeh, T. Sottinen, L. Viitasaari, and A. Yazigi, *Necessary and sufficient conditions for hölder continuity of gaussian processes*, Statistics & Probability Letters, 94 (2014), pp. 230–235.

[6] O. E. Barndorff-Nielsen, *Notes on the gamma kernel*, Thiele Centre for Applied Mathematics in Natural Sciences, (2012).

[7] M. A. Berger and V. J. Mizel, *Volterra equations with itô integrals—i*, The Journal of Integral Equations, (1980), pp. 187–245.

[8] ——, *Volterra equations with itô integrals—ii*, The Journal of Integral Equations, (1980), pp. 319–337.

[9] M. Bergerhausen, D. J. Prömel, and D. Scheffels, *Neural stochastic volterra equations: learning path-dependent dynamics*, arXiv preprint arXiv:2407.19557, (2024).

[10] O. Bonesini, G. Callegaro, M. Grasselli, et al., *From elephant to goldfish (and back): memory in stochastic volterra processes*, arXiv preprint arXiv:2306.02708, (2023).

[11] H.-P. Breuer, E.-M. Laine, J. Piilo, and B. Vacchini, *Colloquium: Non-markovian dynamics in open quantum systems*, Reviews of Modern Physics, 88 (2016), p. 021002.

[12] P. Čoupek and B. Maslowski, *Stochastic evolution equations with volterra noise*, Stochastic Processes and their Applications, 127 (2017), pp. 877–900.

[13] L. Coutin and P. Carmona, *Fractional brownian motion and the markov property*, Electronic Communications in Probability, 3 (1998), p. 12.

[14] L. Coutin and L. Decreusefond, *Stochastic volterra equations with singular kernels*, in Stochastic analysis and mathematical physics, Springer, 2001, pp. 39–50.

[15] L. Decreusefond and A. S. Üstünel, *Stochastic analysis of the fractional brownian motion*, Potential analysis, 10 (1999), pp. 177–214.

[16] T. Dieker, *Simulation of fractional Brownian motion*, PhD thesis, Masters Thesis, Department of Mathematical Sciences, University of Twente . . . , 2004.

[17] G. Gripenberg, S.-O. Londen, and O. Staffans, *Volterra integral and functional equations*, no. 34, Cambridge University Press, 1990.

[18] B. Jourdain and G. Pagès, *Convex ordering for stochastic volterra equations and their euler schemes*, Finance and Stochastics, 29 (2025), pp. 1–62.

[19] G. Jumarie, *On the representation of fractional brownian motion as an integral with respect to (dt) a*, Applied Mathematics Letters, 18 (2005), pp. 739–748.

[20] M. Li, C. Huang, and Y. Hu, *Numerical methods for stochastic volterra integral equations with weakly singular kernels*, IMA Journal of Numerical Analysis, 42 (2022), pp. 2656–2683.

[21] P. Malliavin, *Stochastic analysis*, vol. 313, Springer, 2015.

[22] B. B. Mandelbrot and J. W. Van Ness, *Fractional brownian motions, fractional noises and applications*, SIAM review, 10 (1968), pp. 422–437.

[23] Y. Mishura and I. S. Mishura, *Stochastic calculus for fractional Brownian motion and related processes*, vol. 1929, Springer Science & Business Media, 2008.

[24] Y. Mishura, G. Shevchenko, and S. Shklyar, *Gaussian processes with volterra kernels*, in International Conference on Stochastic Processes and Algebraic Structures, Springer, 2019, pp. 249–276.

[25] Y. Mishura and S. Shklyar, *Gaussian volterra processes with power-type kernels. part i*, Modern Stochastics: Theory and Applications, 9 (2022), pp. 313–338.

[26] ——, *Gaussian volterra processes with power-type kernels. part ii*, Modern Stochastics: Theory and Applications, 9 (2022), pp. 431–452.

[27] A. Mura, *Non-markovian stochastic processes and their applications: from anomalous diffusion to time series analysis*, Università di Bologna, (2008).

[28] D. Nualart, *The Malliavin calculus and related topics*, vol. 1995, Springer, 2006.

[29] G. A. Pavliotis, *Stochastic processes and applications*, Texts in applied mathematics, 60 (2014).

[30] F. A. Pollock, C. Rodríguez-Rosario, T. Frauenheim, M. Paternostro, and K. Modi, *Non-markovian quantum processes: Complete framework and efficient characterization*, Physical Review A, 97 (2018), p. 012127.

[31] Q. Ren, Y. He, and J. Liu, *Balanced euler methods for the strong approximation of stochastic volterra integral equations*, Applied Mathematics Letters, (2025), p. 109613.

[32] A. Richard, X. Tan, and F. Yang, *Discrete-time simulation of stochastic volterra equations*, Stochastic Processes and their Applications, 141 (2021), pp. 109–138.

[33] J. H. Shapiro, *Volterra Adventures*, vol. 85, American Mathematical Soc., 2018.

[34] T. Sottinen and L. Viitasaari, *Stochastic analysis of gaussian processes via fredholm representation*, International journal of stochastic analysis, 2016 (2016), p. 8694365.

[35] Y. Wang, *Stochastic volterra integral equations with a parameter*, Advances in Difference Equations, 2017 (2017), p. 333.

[36] Z. Wang, *Existence and uniqueness of solutions to stochastic volterra equations with singular kernels and non-lipschitz coefficients*, Statistics & probability letters, 78 (2008), pp. 1062–1071.