UNIVERSITÀ DEGLI STUDI DI BERGAMO
Department of Economics

WORKING PAPERS

Fractional Brownian Motion to Generate Multistage Scenario Trees

Gioele Barbano, Vittorio Moriggia and Sebastiano Vitali

November 2024 - WP N. 29 Year 2024

Working papers – Department of Economics
n. 29

Fractional Brownian Motion to Generate Multistage Scenario Trees

Gioele Barbano, Vittorio Moriggia and Sebastiano Vitali

Università degli Studi di Bergamo
2024

**Fractional Brownian Motion to Generate Multistage Scenario Trees /
Barbano Gioele, Moriggia Vittorio, Vitali Sebastiano.
Bergamo: Università degli Studi di Bergamo, 2024.
Working papers of Department of Economics, n. 29
ISSN: 2974-5586
DOI: 10.13122/WPEconomics_29**

Progetto grafico: Servizi Editoriali – Università degli Studi di Bergamo
Università degli Studi di Bergamo
via Salvecchio, 19
24129 Bergamo
Cod. Fiscale 80004350163
P. IVA 01612800167

https://aisberg.unibg.it/handle/10446/288609

Fractional Brownian Motion to Generate Multistage Scenario Trees

Gioele Barbano[1], Vittorio Moriggia[2], and Sebastiano Vitali[2,*]

[1]Vienna Graduate School of Finance (VGSF), WU Vienna, Welthandelsplatz 1, Vienna, 1020, Austria
[2]Department of Economics, University of Bergamo, Via dei Caniana 2, Bergamo, 24127, BG, Italy
[*]Corresponding author: sebastiano.vitali@unibg.it

Abstract

In financial markets, asset prices are not trend-free. Today's movements can, at least slightly, affect tomorrow's direction of the asset returns. Unlike the classical Geometric Brownian Motion, the Fractional Brownian Motion captures trend percistency in asset returns' dynamics. Specifically, the parameter that defines the trend percistency is the so-called Hurst exponent. We aim to introduce a novel approach for scenario generation in multistage stochastic optimization problems based on Fractional Brownian Motion. In particular, we propose three algorithms to populate a non-recombining multistage scenario tree in which the uncertain parameters follow Fractional Brownian Motions. We show several numerical and graphical results to highlight the performance of the three algorithms with multiple values of the Hurst exponent. This approach can help to offer a more realistic framework for portfolio management in uncertain conditions.

Keywords: Scenario Generation; Stochastic Optimization; Fractional Brownian Motion.

1 Introduction

The allocation of resources in a portfolio of assets has been a subject of extensive research and debate in the field of economics and finance. Financial institutions, banks, pension funds, and insurance companies regularly face the problem to manage their economic resources. Often, these decisions aim to reduce or at least mitigate financial risks. This fundamental consideration finds its roots in the pioneering work of Harry M. Markowitz, the "father" of portfolio theory, who developed the Modern Portfolio Theory in the 1950s, with his doctoral thesis at the University of Chicago, see Markowitz [1952]. Within this framework, modelling of asset returns is a crucial element in providing a reliable decision support tool. Since Markowitz [1952], several portfolio optimization models have been proposed to allow financial institutions to build up an optimal portfolio taking into account a key element in finance: uncertainty, which can be seen as information evolving over different time stages. Uncertainty is a key concept that we constantly face in real-world decision-making problems. A mathematical methodology that provides a valid tool for decisions under conditions of uncertainty is the so-called Stochastic Optimization (or Stochastic Programming), a branch of Applied Mathematics and Operations Research born in 1950s thanks to the contribution of Dantzig [1955], Beale [1955] and Charnes and Cooper [1959]. Among the several optimization models proposed to answer the issue of how to allocate financial resources by minimizing risk, the work of Rockafellar and Uryasev [2000] represents a seminal work in optimization methods for finance, as they proposed a novel approach to optimizing or hedging a portfolio of financial instruments by including in the objective function the Conditional Value-at-Risk (*CVaR*).

The question that arises in these types of problems is how to generate scenarios to accurately represent the underlying stochastic model, considering the correlation over time among the random parameters and the generation of a "good" number of scenarios to reflect as accurately as possible the random processes modeled. In recent decades, many authors have tried to deepen the understanding of market dynamics to improve investment decision-making. Some key milestones in the analysis of market behavior include the works of Fama [1970] and Black and Scholes [1973]. These studies, with many others, established the idea that financial markets are driven by the well-known random walk hypothesis. Consequently, the fundamental process often assumed to represent the movements of asset prices is the *Brownian motion* (BM). However, financial markets have repeatedly proved that this assumption is unrealistic, and models based on it have failed to capture the complexity of real-world market behavior.

A generalization of the BM was already studied in Kolmogorov [1940] that considered Gaussian processes with the property of self-similarity. This idea has been later applied to financial processes by Mandelbrot and Van Ness [1968] who formally introduced the *Fractional Brownian Motion* (FBM) as a generalization of the BM. In this way, the idea of *self-similarity* has been converted in *trend persistency* to transform a process without memory, as the BM, in a random process in which some pattern could be identified. Unlike the classical BM, the FBM aims at capturing the proper trend of financial instrument price movements. Such a feature is crucial to understanding financial markets and it is interesting to study if it influences or not the optimal allocation of a portfolio selection problem or if the optimal allocation is somehow indifferent to the trend persistency of the available assets. Despite the concept of persistency was initially developed in hydraulics thanks to the contribution on Hurst [1951], FBM has started to gain significant success also in the financial sector over the past few decades, where numerous studies have focused on the simulation of markets and financial instruments. Notable works include those by Peters [1994], who proposes the Fractal Market Hypothesis by simulating the market's behaviors with FBM, Øksendal [2003] and Hu and Øksendal [2003], who discuss stochastic calculus of FBM and, replicating a portfolio of an European option, demonstrate that the Itô fractional Black & Scholes market with respect to FBM has no arbitrage and is complete, and Corazza and Malliaris [2005], who presented an extended study on the fractal behaviour in foreign currency markets.

In this work, we propose a new technique for generating a scenario tree that evolves over a multistage setting and adopts an FBM instead of the classic BM. In this way, we can construct paths that capture and represent the trend persistency of the asset returns. Indeed, the main characteristic of the FBM is the capability to represent trend persistency, while with BM only mean and covariance can be replicated. We propose an innovative algorithm and show that the statistics of the scenario returns well reproduce the behavior of the historical time series.

The paper is structured as follows. In Section 2 we recall the formulation of the FBM. Section 3 represents the core of our work, in which we propose three innovative scenario generation algorithms to populate a multistage scenario tree with asset returns that follow FBM process. In Section 4 we show some illustrative examples and the numerical results. Section 5 concludes the paper.

2 Fractional Brownian Motion

A *fractional Brownian motion* (FBM), also known as fractal Brownian motion, is a generalization of the ordinary Brownian motion. It is a non-linear stochastic process and is often referred as biased random walk. Like the ordinary Brownian motion, it is a Gaussian stochastic process. However, it deviates from ordinary Brownian motion in a fundamental way: it does not exhibit the property of independent future increments. This deviation makes FBM well-suited for describing various processes, particularly in the natural sciences. As a matter of fact, unlike ordinary Brownian motion, where only the current value is relevant for predicting the future, FBM takes into account the information contained in its history and how the present has evolved from the past, implying that the Markov property does not hold.

In particular, by considering the position $X(t)$ of a Brownian particle as a random function of time t, and a normalized independent Gaussian random process $\{\varepsilon\}$, the increments in the position of the Brownian particle are given by

$$X(t) - X(t_0) \sim \varepsilon \mid t - t_0 \mid^H \tag{1}$$

for any two times t_0 and t with $t_0 < t$ and $H = 0.5$. By generalising H so that it can range from 0 to 1, we define the increments positions of a fractional Brownian particle as stated by Feder [1988]: "One finds the position $X(t)$ given the position $X(t_0)$ by choosing a random number ε from a Gaussian distribution, multiplying it by the time increment $\mid t - t_0 \mid^H$ and adding the result to the given position $X(t_0)$". These increments are commonly referred to as the *fractional Gaussian noise* (FGN), and we will provide a detailed analysis of them in next sections.

Let's now formally introduce the process. FBM was originally introduced by Kolmogorov [1940], even though he did not use the name "fractional Brownian motion" but "Wiener spiral". Subsequently, it was studied by Mandelbrot and Van Ness [1968] who provided the following definition through a stochastic integral.

Definition 1. *Let $0 < H < 1$, and let b_0 be an arbitrary real number. We call the following random function $B_H(t)$ the reduced fractional Brownian motion with parameter H and starting value b_0 at time $t = 0$. For $t > 0$, $B_H(t)$ is defined by*

$$B_H(0) = b_0$$

and

$$B_H(t) - B_H(0) = \frac{1}{\Gamma(H+1/2)} \left\{ \int_{-\infty}^{0} [(t-s)^{H-1/2} - (-s)^{H-1/2}] dB(s) + \int_{0}^{t} (t-s)^{H-1/2} dB(s) \right\} \tag{2}$$

where Γ represents the Gamma function $\Gamma(\alpha) := \int_0^\infty x^{\alpha-1} exp(-x)dx$, the real number H is the Hurst exponent and the integrator $B(t)$ denotes the ordinary Brownian motion. Definition 1 states that the random function B_H at time t depends on all previous increments $dB(t-1)$ of an ordinary Gaussian random process $B(t)$ with mean and variance equal to 0 and 1, respectively.

Equation 2, representing the distance between increments, describes the so-called stochastic integral with respect to usual Brownian motion, whose interpretation is not an easy task. Since this is not the aim of the thesis, for details the reader is referred to Karatzas and Shreve [1996]. For simulation purposes, Equation 2 can be rewritten in a simpler manner to understand:

$$B_H(t) - B_H(t-1) = \frac{n^{-H}}{\Gamma(H+1/2)} \left[\sum_{i=1}^{nt} i^{H-1/2} r_{(1+n(M+1)-i)} + \sum_{i=1}^{n(M-1)} \left((n+1)^{H-1/2} - i^{H-1/2} r_{(1+n(M-1+t)-i)} \right) \right] \tag{3}$$

where r is equal to a series of M Gaussian random variables.

Equation 3 is the discrete form of Equation 2. Indeed, it replaces the integral with summations, without change the meaning. Moreover, this equation represents the moving average over a finite range of random Gaussian M values, weighted by a power law function of Hurst exponent. Following Feder [1988] and Peters [1994], let's derive the expected value and variance of fractional Brownian increments. Let $B_H(t)$ be the particle position (instead of $X(t)$ of Equation 1), a FBM has

$$\mathbb{E}[B_H(t_{i+1}) - B_H(t_i)] = 0 \tag{4}$$

and

$$Var[B_H(t_{i+1}) - B_H(t_i)] = |t_{i+1} - t_i|^{2H} \tag{5}$$

As a consequence of Equation 4 and Equation 5, it becomes even more evident that Brownian motion is merely a special case of the more general FBM, occurring only when $H = 0.5$. It is worth noting that FBM has infinitely long-run correlations. Past increments are correlated with future increments. In particular, given the increment $B_H(t_1) - B_H(t_0)$ from time t_0 to time t_1, the probability to have an increment $B_H(t_2) - B_H(t_1)$ averaged over the distribution of the past increments is equal to

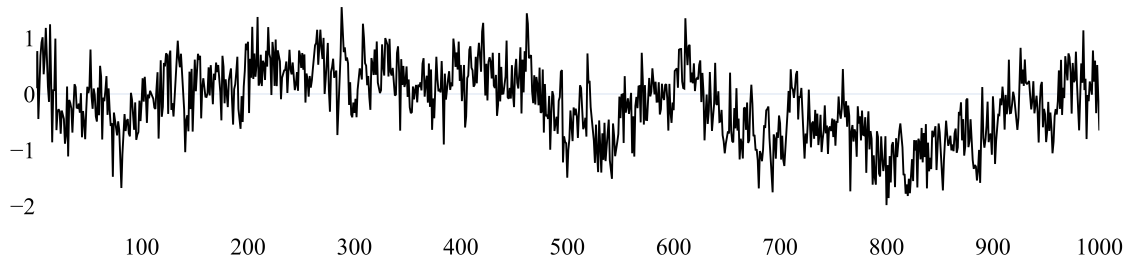
$$[B_H(t_1) - B_H(t_0)][B_H(t_2) - B_H(t_1)]. \tag{6}$$

By setting $B_H(t_1) = 0$ and by normalizing with the variance of B_H, the correlation function of future increments $B_H(t_2)$ with past increments $B_H(t_0)$ is given by

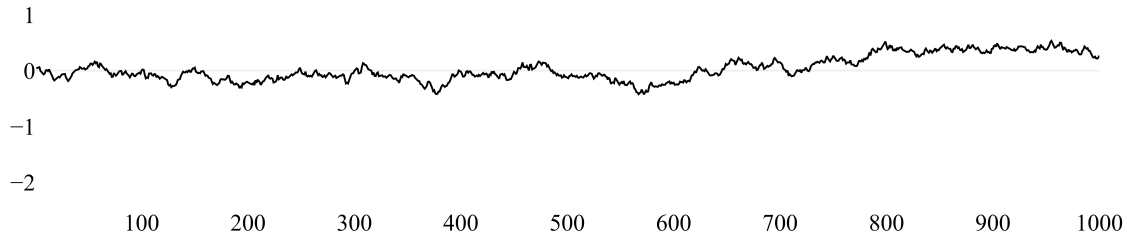
$$C(t) = \frac{B_H(t_0) B_H(t_2)}{B_H(t_2)^2} = 2(2^{2H-1} - 1). \tag{7}$$

The FBMs corresponding to $0 < H < 1/2$, $1/2 < H < 1$, and $H = 1/2$, respectively, differ in many significant ways. Let's see in details the three cases:

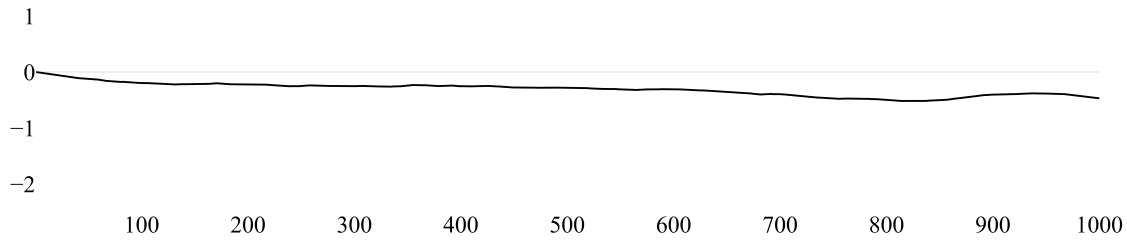
- if $1/2 < H < 1$, then $C(t)$ is positive and the process is defined to be persistent. It means that we have infinite long-run correlations and a long-memory effect occurs over multiple time scales;
- if $0 < H < 1/2$, then $C(t)$ is negative and the process is defined to be anti-persistent. It means that a reversal effect occurs over multiple time scales. An increasing trend in the past is more likely to be followed by a decreasing trend in the future and viceversa;
- if $H = 1/2$, then $C(t)$ is equal to zero and there is not a long-memory effect. The process becomes the ordinary Brownian motion with independent increments.



(a) Anti-persistent FBM, $H = 0.1$



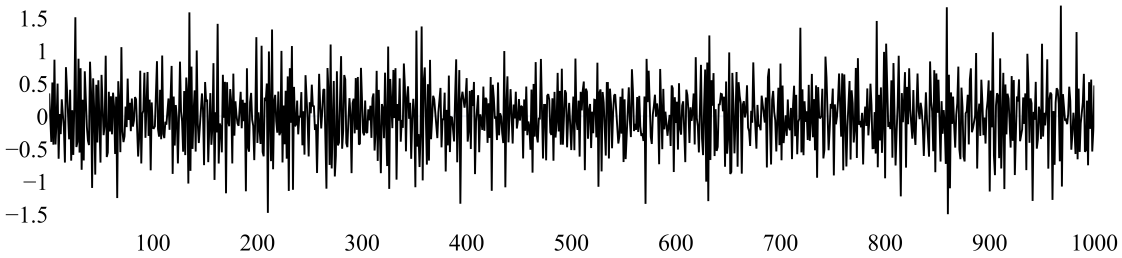
(b) Classic BM, $H = 0.5$



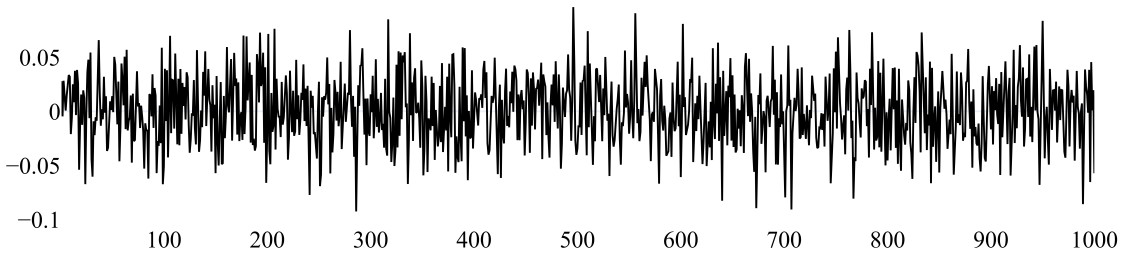
(c) Persistent FBM, $H = 0.9$

Figure 1: FBMs with different values of the Hurst exponent H and different trend percistencies.

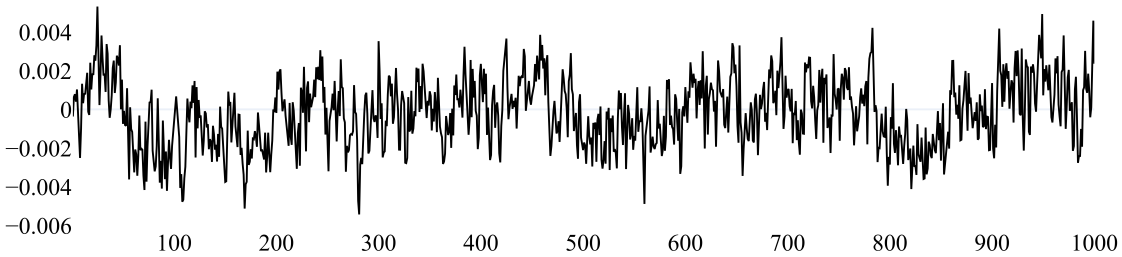
Figure 1 provides graphical representations of the distinctions between FBMs while allowing for variations in the parameter $H = 0.1, 0.5, 0.9$. Despite all three simulated motions share $\mu = 0$ and $\sigma = 1$, barring approximations, noticeable differences emerge. The central graph depicts the standard financial model characterized by a completely random pattern. The lower graph represents a persistent process that closely follows the mean value with minimal deviation. Conversely, the upper graph exhibits a fragmented pattern, which is indicative of an anti-persistent effect. The above characteristics of FBM express the presence of long-range dependence during the motion. In fact, the expected value of FBM at a given moment depends on all the values assumed by the motion in the earlier instants. Figure 1 further illustrates the distinctions between FBM while allowing for variations in the parameter $H = 0.1, 0.5, 0.9$. Although all three simulated motions have $\mu = 0$ and $\sigma = 1$, barring approximations, the differences are clear. In particular, the middle chart represents the standard financial model characterized by a completely random pattern. The bottom chart, instead, represents a persistent process that closely follows the mean value without deviating too much from it. Then, the top chart represents the opposite case. Here, the fragmented pattern is evident, leading to an anti-persistent effect. Learning from Mandelbrot and Hudson [2008], the three processes just shown in Figure 1, are the sums of their increments, which we have already defined as FGN. Figure 2 provides an illustrative representation of them. As before, the middle chart represents the increments of a classical Brownian motion. Thinking about stock returns, we can consider them as independent from each other. In the bottom chart, when H is large, the resulting price trends are broad. Finally, in the top chart, using Mandelbrot's words, the action is furious but still constrained.



(a) Anti-persistent FGN, $H = 0.1$



(b) Gaussian noises, $H = 0.5$



(c) Persistent FGN, $H = 0.9$

Figure 2: Fractional Gaussian noises with different values of the Hurst exponent H and different trend percistencies.

3 Scenario generation

3.1 Single-stage FBM scenario generation

Financial prices must be modeled with a proper process, capable of capturing the specific properties. It would be tempting to suppose that stock prices follow the FBM. In factm, it would not take into account the independence of the expected return required by investors from stock concerning the stock price, see Hull [2022]. To address this issue, financial mathematics commonly employs the Geometric Fractional Brownian Motion (GFBM), which, similar to FBM compared to BM, is a generalization of the more common GBM.

Let $S(t)$ be the stock price at time t, the expected drift rate in $S(t)$ should be assumed to be $\mu S(t)$ for some constant parameter μ. This means that, by considering a short interval of time $\Delta t = t_{i+1} - t_i$ for any $i = 1, ..., T$, the expected increase in $S(t)$ is $\mu S(t)\Delta t$, where the parameter μ is the expected rate of return on the stock. Then, let's assume that the uncertainty in returns in a short period Δt, represented by the standard deviation, should be proportional to the stock price. Therefore, according to Hull [2022], the model can be written as

$$dS(t) = \mu S(t)dt + \sigma S(t)dB_H(t) \tag{8}$$

or

$$\frac{dS(t)}{S(t)} = \mu dt + \sigma dB_H(t).$$

Equation 8 represents the GFBM, whose special case $H = 0.5$ is referred to GBM. According to Hu and Øksendal [2003], the solution to Equation 8 for any arbitrary initial value S_0 is given as follows:

$$S(t) = S_0 e^{\mu t - \frac{1}{2}\sigma^2 t^{2H} + \sigma B_H(t)}, \quad t \geq 0. \tag{9}$$

The scenario generation technique based on this approach is well-known when the scenarios have the so-called fan structure. However, at the best of our knowledge, a scenario generation technique that uses GFBM to build a non-recombining scenario tree with a given branching structure has not been proposed yet. In the next section, we suggest an algorithm to populate a non-recombining multistage scenario tree with a GFBM.

3.2 Multistage FBM scenario generation algorithms

To solve a multistage stochastic problem it is fundamental to have a reliable multistage stochastic tree that represents the future evolution of the uncertain parameters. Firstly, we simply simulated the evolution of assets by generating a series of FGN for each of them. Since we are dealing with more than one asset, it's more convenient to work within a matrix framework. Let's denote by $t = 1, \ldots, T$ the number of points to generate and by $i = 1, \ldots, n$ the specific asset. The generic price $P_i(t)$ of asset i at time t is given by:

$$P_i(t) = P_{i,0} \exp\left(\mu_i - \frac{1}{2}\sigma_i^2 + \sigma_i B_{Hi}(t)\right), \quad t \geq 0. \tag{10}$$

Then, given that the price at time t depends on the price at time $t-1$ (which will be known at time t), for computational purposes, the implemented formula results in the following equation

$$P_i(t) = P_{i,t-1} \exp\left(\mu_i - \frac{1}{2}\sigma_i^2 + \sigma_i X_{Hi}(t)\right) \tag{11}$$

where $X_{Hi}(t)$ represents the FGN of asset i at time t. This formula ensures that the price at time t is generated by taking into account the information evolution from the price at time $t-1$ to time t. Finally, since we are interested in simulating logarithmic returns, Equation 11 becomes:

$$r_i(t) = \mu_i - \frac{1}{2}\sigma_i^2 + \sigma_i X_{Hi}(t) \tag{12}$$

Several simulation methods exist to generate FGNs and, consequently, FBMs (see, for example, Dieker [2004]). We adopt the fbmwoodchan function, cf. Wood and Chan [1994], which is part of the FracLab Tool of Matlab, to generate a sequence of increments that follow the GFBM. In Section 4, we show a sample of the scenarios that we are able to produce with the code implemented in Python.

However, considering that we are generating scenarios for a portfolio of assets, Equation 10 does not account for the diversification principle. To address this issue, we multiply the matrix of FGN by the

correlation matrix of historical logarithmic returns after applying the Cholesky factorization to the latter. Specifically, the Cholesky method is an algorithm that decomposes a symmetric, positive definite matrix $A_{n\times n}$ into the product of a lower-triangular matrix L and its conjugate transpose, denoted as L^*:

$$A = LL^*$$

This process enables us to obtain a matrix of FGN whose correlation matrix matches that of historical logarithmic returns. Thus, we implement Algorithm 1, whose idea is relatively simple: we generate a series of FGN for each asset based on GFBM.

Algorithm 1 FBM-based multistage scenario tree generation, full tree approach

Input:
- numAsset ▷ Number of assets involved in the problem
- H ▷ Hurst exponent, with $H \in [0,1]$
- numOFnodes ▷ Number of nodes of the scenario tree
- ρ ▷ Desired correlation structure

Output:
- increments ▷ FGNs associated with a multistage scenario tree

Using:
- StdNorm(x) ▷ Normalizes the columns of matrix x
- ApplyCorr(x,ρ) ▷ Applies the correlation structure ρ to the columns of matrix x
- WoodChan(x,H) ▷ Generates a FBM with size x and Hurst exponent H

for k = 1 **to** numAsset **do** ▷ Initialization of nodes in the first stage
  increments[:][k] ← WoodChan(numOFnodes,H) ▷ Generates a sample of FGNs
end for
increments ← StdNorm(increments) ▷ Normalizes each column of increment matrix
increments ← ApplyCorr(increments,ρ) ▷ Makes the columns of increment matrix correlated according to ρ

Algorithm 2 FBM-based multistage scenario tree generation, node-by-node approach

Input:
- numAsset ▷ Number of assets involved in the problem
- H ▷ Hurst exponent, with $H \in [0,1]$
- numOFnodes ▷ Number of nodes of the scenario tree
- ρ ▷ Desired correlation structure
- ar_1,ar_2 ▷ Initial values for the real part of a complex random vector
- ai_1,ai_2 ▷ Initial values for the imaginary part of a complex random vector

Output:
- increments ▷ FGNs associated with a multistage scenario tree

Using:
- StdNorm(x) ▷ Normalizes the columns of matrix x
- ApplyCorr(x,ρ) ▷ Applies the correlation structure ρ to the columns of matrix x
- WoodChan(x,H,ar_1,ai_1,ar_2,ai_2) ▷ Generates a FBM with size x and Hurst exponent H
- Random(x) ▷ Create a shape-x array with random standard normal samples
- Anc(x) ▷ Returns the ancestor node of node x

for k = 1 **to** numAsset **do** ▷ Initialization of nodes in the first stage
  ar_rand ← Random(numOFnodes) ▷ Generate a random vector for the real part
  ai_rand ← Random(numOFnodes) ▷ Generate a random vector for the imaginary part
  ▷ Generates an initial sample of FGNs by specifying initial values
  initsample[1][k] ← WoodChan(1000,H,0,0,ar_rand[1], ai_rand[1])
  for n = 1 **to** numOFnodes **do**
    ▷ Generates a sample of FGNs by starting from specified initial values
    inctrements[k][n] ← WoodChan(1000,H,ar_rand[Anc[n]],ai_rand[Anc[n]],ar_rand[n], ai_rand[n])
  end for
end for
increments ← StdNorm(increments) ▷ Normalizes each column of increment matrix
increments ← ApplyCorr(increments,ρ) ▷ Makes the columns of increment matrix correlated according to ρ

When the uncertainty is modeled through a GBM, the way to populate the scenario tree nodes, i.e. to generate the sequence of nodes that corresponds to the discretization of the stochastic process is basically the same as in the single-stage approach because the stochastic increments are assumed to be independent from one stage to the next. However, the GFBM relies on the hypothesis that the increments are not independent and the stochastic path shows some kind of persistency in its increments. For this purpose, we develop Algorithm 2, which generates FGN paths node-by-node. The idea here is to generate scenarios

by starting from values recorded in previous nodes, by not breaking the trend. Specifically, we modified the fbmwoodchan function to take specified initial values as inputs. This allows us to generate scenarios at each node that begin from the value positioned in the corresponding parent node. However, this method lacks some key features of FBM. While it does not break the trend, it only considers a single increment to generate the next one, disregarding any information from the previous series. As a result, it can produce values that may appear almost random, which contrasts significantly with the underlying concept of FBM.

With this in mind, we finally propose Algorithm 3, which overcomes the drawback of the previous two algorithms to populate the nodes of a scenario tree. The main idea of the new algorithm is:

1. to assign to nodes in the first stage a value v_n randomly extract from a large fractional Gaussian noise sample, let's call it F_0
2. to generate, for each children node $n+$ of a given parent node n, a new fractional Gaussian noise sample F_{n+};
3. to find, within F_{n+}, the position i of the closest value to the value v_n assigned to the parent node;
4. to assign to the children node $n+$ the value v_{n+} that is in the subsequent position, i.e. $v_{n+} = F_{n+}[i+1]$

In this way, we are able to generate multiple subsequent increments (the values v_{n+} in the children nodes $n+$) for a given increment (the values v_n in the parent node n), conserving the similarity as much as possible, even if the GFBM generates only one subsequent increment. Increasing the dimension of F_{n+} helps to find a *closest value* in step 3 that is really close enough. In Section 4, we show a sample of the scenarios that we are able to produce with the code implemented in Python.

Algorithm 3 FBM-based multistage scenario tree generation, closest parent approach

```
Input:
    numAsset                                  ▷ Number of assets involved in the problem
    H                                         ▷ Hurst exponent, with H ∈ [0,1]
    nodeSTG1                                  ▷ Number of nodes in the first stage
    numOFnodes                                ▷ Number of nodes of the scenario tree
    ρ                                         ▷ Desired correlation structure

Output:
    increments                                ▷ FGNs associated with a multistage scenario tree

Using:
    Pick(x,y)                                 ▷ Randomly extracts y elements from vector x
    StdNorm(x)                                ▷ Normalizes the columns of matrix x
    ApplyCorr(x,ρ)                            ▷ Applies the correlation structure ρ to the columns of matrix x
    WoodChan(x,H)                             ▷ Generates a FBM with size x and Hurst exponent H
    ArgMin(x)                                 ▷ Returns the index of the minimum value of a vector x
    Abs(x)                                    ▷ Returns the absolute value of number x
    Anc(x)                                    ▷ Returns the ancestor node of node x

for k = 1 to numAsset do                      ▷ Initialization of nodes in the first stage
    initsample ← WoodChan(1000,H)             ▷ Generates an initial sample of FGNs
    for n = 1 to nodeSTG1 do
        increments[k][n] ← Pick(initsample, 1)  ▷ Randomly assigns a FGN to nodes in first stage
    end for
    for n = nodeSTG1 + 1 to numOFnodes do
        sample ← WoodChan(100,H)              ▷ Generates a new sample of FGNs
                                              ▷ Finds the position of the closest FGN to ancestor node value of n
        idx ← ArgMin(Abs(sample - increments[k][Anc(n)]))
        increments[k][n] ← sample[idx+1]      ▷ Assigns value of child node indexed by idx to the matrix of increments
    end for
end for
increments ← StdNorm(increments)              ▷ Normalizes each column of increment matrix
increments ← ApplyCorr(increments,ρ)          ▷ Makes the columns of increment matrix correlated according to ρ
```

4 Empirical results

In this section, we provide extended results to highlight the quality of the scenarios generated with all algorithms presented in previous section, for a set of 10 assets that correspond to 10 ETFs as reported in Table 1. These assets represent a typical asset universe for a portfolio selection problem since they span from liquidity to bond and equity. For each asset, we consider the historical time series of the daily prices

Table 1: Asset list

Asset Class	ETF Name
Liquidity	LYXOR Euro Overnight Return UCITS ETF
WorldGovBond	FTSE World Gov. Bond Index - Total Return
EuroCorpBond	Bloomberg EuroAgg Corp. Total Return Index Value Unhedged
EuroGovBond	Bloomberg EuroAgg Gov. Total Return Index Value Unhedged
EmMarkGovBond	J.P. Morgan Gov. Bond Index Emerging Markets Global Core
EmMarkEquity	MSCI EM U$ - Price Index
WorldEquity	MSCI World Universal $ - Price Index
WorldEquityESG	MSCI World ESG Universal $ - Price Index
EuroEquity	Euro STOXX 50 - Price Index
EuroEquityESG	Euro STOXX 50 ESG E - Price Index

from March 2012 to November 2023. From these series, we compute the annual returns with a daily rolling procedure. The statistics of the historical time series reported hereafter are referred to the annual returns of the assets.

All the results discussed in this section are generated using a standard laptop equipped with an Intel(R) Core(TM) i5-1035G1 CPU running at 1.19 GHz and 8 GB of RAM, operating on the Windows 11 platform. The chosen main development tool is Python 3.10. Algorithm 1 is quite fast for generating many scenarios over 10 stages, while, given the higher complexity, Algorithm 2 and Algorithm 3 are a little bit slower. In particular, Algorithm 3 takes approximately 0.7 seconds to generate a tree with 20 scenarios evolving over 10 stages (branching 5-2-2-1-1-1-1-1-1-1 and 175 nodes), approximately 10 seconds to generate a tree with 1,000 scenarios (branching 5-5-5-2-2-2-1-1-1-1 and 5,905 nodes), and approximately 100 seconds to generate a tree with 10,000 scenarios (branching 10-10-5-5-2-2-1-1-1-1 and 58,110 nodes). As expected, the computing time increases linearly with the number of nodes.

4.1 Data analysis

To evaluate the quality of generated scenarios, we compare the statistics of the historical series with their statistics computed over the nodes of the scenario trees. We perform this analysis for three different levels of H (0.1, 0.5, 0.9) and all assets. To make the statistics more robust, we increase the number of scenarios to 1000 with branching 5-5-5-2-2-2-1-1-1-1. Once the scenario tree has been generated, for each asset we compute the average return, the volatility of the returns, their V@R, and their CV@R. The V@R and the CV@R are computed on the empirical return distributions with $\alpha = 0.05$. Since all algorithms yield very similar results, in Table 2 we show only the statistics of the returns obtained with Algorithm 3. As expected, the annual returns generated with the FBM are aligned with the ones generated with the classical GBM. Indeed, all the statistics are almost equal for the three values of H. Only the CV@R deviations of all the statistics are slightly increasing with the value of H, but the magnitude of these differences is almost negligible. We can conclude that for all the algorithms no significant differences are observed based on H, and therefore it is not possible to distinguish between the different algorithms by evaluating the statistics of annual returns over the scenario tree.

However, the value of H is not expected to influence the value of the annual return in each single node: taking into account the trend should have a significant impact when we consider the whole path from the root node to the leaves of the tree. Therefore, we compute the cumulative returns over the 10 stages and, again, we compute the four statistics of these cumulative returns. Now, the difference among the three scenario generation algorithms becomes clear. Specifically, Table 3 reports the statistics of the cumulative returns generated by Algorithm 1 which appear somewhat confusing. There is no identifiable effect of different H values on both reward and risk sides, as we cannot determine whether low or high levels of H lead to higher or lower levels of average returns and risk. For example, by looking at EmMarkEquity, an higher level of H leads to a decrease of the average return and an increase of risk. However, for the asset EuroEquityESG we find the opposite results: by increasing H, we get higher levels of the average return and lower levels of risk. Table 4 reports the statistics of the cumulative returns generated by Algorithm 2 and we notice the same characteristics highlighted for Algorithm 1 in Table 3. Therefore, the improvements made in Algorithm 2 do not have the capability to efficiently represent the assets' trends. In Table 5, we show the statistics of the cumulative returns generated with Algorithm 3. Finally, the impact of an anti-persistent trend versus a persistent trend appears evident. With an anti-persistent trend ($H = 0.1$) the returns somehow counterbalance each other along the scenarios and the cumulative return is less volatile and also much less risky than the classic GBM ($H = 0.5$). Indeed, for all assets the average return is smaller, the

Table 2: Statistics of the historical annual returns and of the annual returns generated to populate the scenario trees for all considered assets and for three values of H, generating with Algorithm 3 and identical to the results obtained with Algorithm 1 and Algorithm 2.

		Statistics			
Asset class	Series	Average Return	Volatility	V@R	CV@R
Liquidity	History	-0.23%	0.62%	-0.73%	-0.74%
	Scenarios $H = 0.1$	-0.22%	0.62%	-1.25%	-1.51%
	Scenarios $H = 0.5$	-0.22%	0.62%	-1.25%	-1.51%
	Scenarios $H = 0.9$	-0.22%	0.62%	-1.25%	-1.51%
WorldGovBond	History	1.25%	5.27%	-10.38%	-12.43%
	Scenarios $H = 0.1$	1.25%	5.27%	-7.42%	-9.58%
	Scenarios $H = 0.5$	1.25%	5.27%	-7.42%	-9.62%
	Scenarios $H = 0.9$	1.25%	5.27%	-7.41%	-9.64%
EuroCorpBond	History	1.57%	5.25%	-10.99%	-13.23%
	Scenarios $H = 0.1$	1.57%	5.25%	-7.05%	-9.23%
	Scenarios $H = 0.5$	1.57%	5.25%	-7.06%	-9.25%
	Scenarios $H = 0.9$	1.57%	5.25%	-7.06%	-9.27%
EuroGovBond	History	1.38%	6.77%	-14.45%	-16.33%
	Scenarios $H = 0.1$	1.38%	6.77%	-9.76%	-12.57%
	Scenarios $H = 0.5$	1.38%	6.78%	-9.77%	-12.58%
	Scenarios $H = 0.9$	1.38%	6.78%	-9.77%	-12.61%
EmMarkGovBond	History	-0.50%	9.85%	-17.58%	-19.83%
	Scenarios $H = 0.1$	-0.50%	9.85%	-16.70%	-20.80%
	Scenarios $H = 0.5$	-0.50%	9.86%	-16.71%	-20.81%
	Scenarios $H = 0.9$	-0.50%	9.85%	-16.70%	-20.83%
EmMarkEquity	History	1.35%	17.76%	-24.74%	-27.76%
	Scenarios $H = 0.1$	1.35%	17.75%	-27.83%	-35.23%
	Scenarios $H = 0.5$	1.35%	17.75%	-27.83%	-35.21%
	Scenarios $H = 0.9$	1.35%	17.75%	-27.83%	-35.33%
WorldEquity	History	8.71%	13.78%	-14.04%	-17.29%
	Scenarios $H = 0.1$	8.71%	13.78%	-13.93%	-19.70%
	Scenarios $H = 0.5$	8.71%	13.79%	-13.97%	-19.74%
	Scenarios $H = 0.9$	8.71%	13.91%	-13.99%	-19.74%
WorldEquityESG	History	8.84%	13.81%	-13.99%	-17.78%
	Scenarios $H = 0.1$	8.84%	13.81%	-13.87%	-19.63%
	Scenarios $H = 0.5$	8.84%	13.82%	-13.88%	-19.68%
	Scenarios $H = 0.9$	8.84%	13.83%	-13.91%	-19.69%
EuroEquity	History	6.49%	14.55%	-16.65%	-19.18%
	Scenarios $H = 0.1$	6.48%	14.55%	-17.44%	-23.51%
	Scenarios $H = 0.5$	6.48%	14.58%	-17.47%	-23.58%
	Scenarios $H = 0.9$	6.48%	14.56%	-17.49%	-23.56%
EuroEquityESG	History	6.97%	14.39%	-15.67%	-18.35%
	Scenarios $H = 0.1$	6.97%	14.39%	-16.68%	-22.69%
	Scenarios $H = 0.5$	6.97%	14.41%	-16.68%	-22.74%
	Scenarios $H = 0.9$	6.97%	14.40%	-16.73%	-22.75%

standard deviation is also smaller, and the V@R and CV@R represent a cumulative behavior that induces lower losses: in some cases (see WorldEquity and WorldEquityESG), the worst scenarios still give a very positive return. On the other hand, with a persistent trend ($H = 0.9$) the cumulative returns are much more dispersed and risky than the GBM, but also the average return is higher. In particular, average return and standard deviation are higher than GBM for all cases, and V@R and CV@R highlight a cumulative effect that can induce huge losses. Now, for all assets, these two risk measures are deeply negative meaning that even assets with a high average return could produce huge losses in the long run.

Table 3: Statistics of the cumulative returns of the scenario trees for all considered assets and for three values of H, based on Algorithm 1.

		Statistics			
Asset class	Hurst Coeff	Average Return	Volatility	V@R	CV@R
Liquidity	$H = 0.1$	-2.28%	1.75%	-5.27%	-6.07%
	$H = 0.5$	-2.00%	1.55%	-4.18%	-4.91%
	$H = 0.9$	-1.44%	1.13%	-3.06%	-3.23%
EmMarkEquity	$H = 0.1$	10.98%	51.43%	-52.21%	-62.12%
	$H = 0.5$	4.50%	40.74%	-62.81%	-65.88%
	$H = 0.9$	-4.29%	37.13%	-62.50%	-65.34%
WorldEquity	$H = 0.1$	130.24%	82.71%	24.58%	23.40%
	$H = 0.5$	118.03%	62.20%	22.72%	19.69%
	$H = 0.9$	107.10%	40.02%	38.58%	29.43%
WorldEquityESG	$H = 0.1$	133.59%	85.21%	29.90%	24.66%
	$H = 0.5$	120.50%	63.00%	27.52%	21.75%
	$H = 0.9$	108.43%	39.07%	41.45%	32.61%
EuroEquity	$H = 0.1$	87.31%	78.49%	-4.30%	-4.72%
	$H = 0.5$	90.12%	79.74%	4.45%	-7.39%
	$H = 0.9$	100.14%	79.16%	13.54%	-7.60%
EuroEquityESG	$H = 0.1$	96.58%	80.90%	-1.61%	-2.07%
	$H = 0.5$	98.69%	80.24%	9.11%	0.19%
	$H = 0.9$	107.66%	78.27%	19.70%	0.06%
WorldGovBond	$H = 0.1$	14.51%	17.84%	-10.83%	-13.34%
	$H = 0.5$	15.44%	19.84%	-9.57%	-18.01%
	$H = 0.9$	15.63%	15.77%	-7.53%	-12.91%
EuroCorpBond	$H = 0.1$	18.18%	20.14%	-5.19%	-11.52%
	$H = 0.5$	19.05%	20.57%	-3.10%	-11.56%
	$H = 0.9$	20.58%	16.00%	-1.43%	-7.87%
EuroGovBond	$H = 0.1$	16.90%	25.76%	-22.83%	-23.50%
	$H = 0.5$	18.74%	28.49%	-15.36%	-28.42%
	$H = 0.9$	20.18%	23.03%	-8.50%	-20.73%
EmMarkGovBond	$H = 0.1$	-5.21%	28.32%	-43.82%	-51.09%
	$H = 0.5$	-5.84%	31.04%	-51.69%	-56.52%
	$H = 0.9$	-6.24%	29.23%	-50.22%	-51.74%

Table 4: Statistics of the cumulative returns of the scenario trees for all considered assets and for three values of H, based on Algorithm 2.

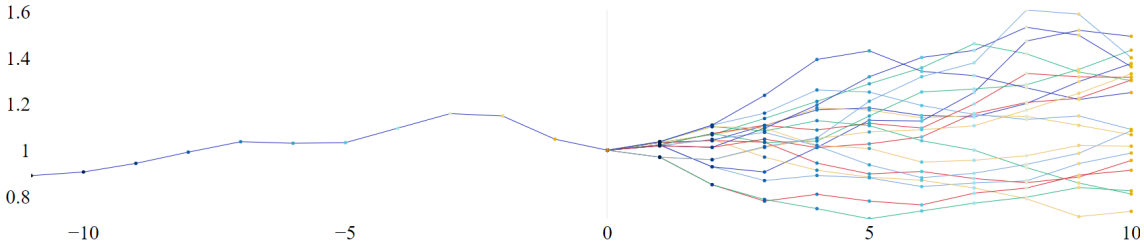
		Statistics			
Asset class	Hurst Coeff	Average Return	Volatility	V@R	CV@R
Liquidity	$H = 0.1$	-1.35%	2.76%	-5.37%	-6.01%
	$H = 0.5$	-1.26%	2.80%	-5.29%	-5.96%
	$H = 0.9$	-1.21%	2.58%	-4.83%	-5.47%
EmMarkEquity	$H = 0.1$	11.34%	91.04%	-66.09%	-66.80%
	$H = 0.5$	4.39%	86.65%	-68.40%	-69.57%
	$H = 0.9$	-11.82%	63.62%	-68.33%	-68.73%
WorldEquity	$H = 0.1$	130.37%	161.30%	-12.11%	-14.52%
	$H = 0.5$	120.78%	158.54%	-17.95%	-18.79%
	$H = 0.9$	97.65%	122.16%	-15.47%	-17.50%
WorldEquityESG	$H = 0.1$	134.53%	165.02%	-11.38%	-14.61%
	$H = 0.5$	125.13%	162.31%	-17.15%	-18.83%
	$H = 0.9$	101.96%	125.33%	-15.36%	-16.45%
EuroEquity	$H = 0.1$	84.99%	142.74%	-39.77%	-51.27%
	$H = 0.5$	77.98%	139.22%	-41.02%	-54.25%
	$H = 0.9$	59.84%	107.54%	-33.85%	-50.69%
EuroEquityESG	$H = 0.1$	93.80%	147.98%	-35.63%	-49.75%
	$H = 0.5$	86.96%	144.66%	-36.55%	-52.78%
	$H = 0.9$	68.89%	112.24%	-28.36%	-48.87%
WorldGovBond	$H = 0.1$	13.45%	21.95%	-20.47%	-27.56%
	$H = 0.5$	13.70%	22.51%	-20.59%	-29.25%
	$H = 0.9$	13.68%	20.64%	-16.89%	-27.97%
EuroCorpBond	$H = 0.1$	15.66%	27.17%	-20.79%	-21.92%
	$H = 0.5$	14.82%	27.39%	-22.64%	-22.94%
	$H = 0.9$	13.07%	23.94%	-20.34%	-21.65%
EuroGovBond	$H = 0.1$	12.43%	28.37%	-26.89%	-35.08%
	$H = 0.5$	11.99%	28.77%	-27.45%	-37.29%
	$H = 0.9$	10.79%	25.81%	-24.07%	-36.20%
EmMarkGovBond	$H = 0.1$	-4.62%	40.95%	-67.07%	-68.96%
	$H = 0.5$	-6.29%	41.03%	-67.80%	-70.20%
	$H = 0.9$	-10.41%	34.98%	-64.00%	-67.37%

Table 5: Statistics of the cumulative returns of all considered assets and for three values of H, based on Algorithm 3.

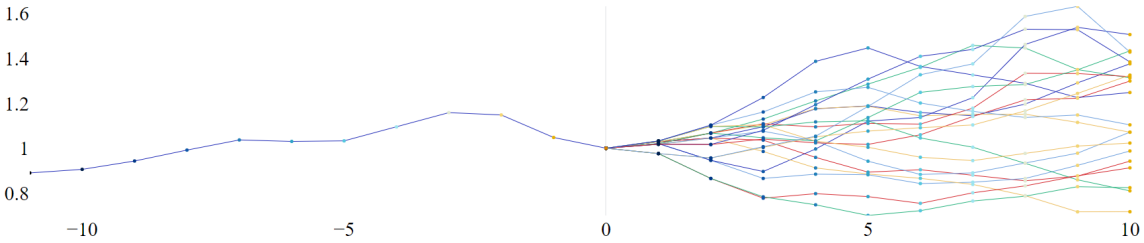
		Statistics			
Asset class	Hurst Coeff	Average Return	Volatility	V@R	CV@R
Liquidity	$H = 0.1$	-2.21%	1.25%	-4.26%	-4.77%
	$H = 0.5$	-2.21%	1.82%	-5.17%	-5.90%
	$H = 0.9$	-2.19%	2.96%	-6.99%	-8.16%
WorldGovBond	$H = 0.1$	12.13%	12.06%	-6.62%	-10.74%
	$H = 0.5$	13.02%	17.73%	-13.92%	-19.40%
	$H = 0.9$	15.05%	29.33%	-27.08%	-34.44%
EuroCorpBond	$H = 0.1$	15.82%	12.36%	-3.36%	-7.49%
	$H = 0.5$	16.88%	18.18%	-10.73%	-16.44%
	$H = 0.9$	19.09%	30.21%	-24.28%	-31.95%
EuroGovBond	$H = 0.1$	13.04%	15.68%	-10.87%	-15.92%
	$H = 0.5$	14.57%	23.15%	-19.62%	-26.17%
	$H = 0.9$	18.23%	38.95%	-35.06%	-43.48%
EmMarkGovBond	$H = 0.1$	-7.49%	19.03%	-35.44%	-40.82%
	$H = 0.5$	-4.76%	28.90%	-45.00%	-51.68%
	$H = 0.9$	2.53%	50.86%	-59.72%	-67.33%
EmMarkEquity	$H = 0.1$	4.82%	40.37%	-48.44%	-56.30%
	$H = 0.5$	14.83%	64.31%	-60.73%	-69.03%
	$H = 0.9$	43.97%	133.97%	-78.44%	-85.54%
WorldEquity	$H = 0.1$	119.81%	59.32%	36.11%	21.64%
	$H = 0.5$	131.37%	90.68%	11.53%	-5.77%
	$H = 0.9$	160.30%	169.69%	-26.72%	-44.42%
WorldEquityESG	$H = 0.1$	122.44%	60.10%	37.65%	22.96%
	$H = 0.5$	134.11%	91.89%	12.75%	-4.76%
	$H = 0.9$	163.43%	172.09%	-26.08%	-43.92%
EuroEquity	$H = 0.1$	78.41%	52.16%	5.62%	-6.54%
	$H = 0.5$	87.81%	79.44%	-14.72%	-28.81%
	$H = 0.9$	116.01%	152.73%	-45.83%	-59.90%
EuroEquityESG	$H = 0.1$	86.90%	53.69%	11.85%	-0.90%
	$H = 0.5$	96.41%	81.61%	-9.59%	-24.23%
	$H = 0.9$	124.90%	156.51%	-42.16%	-56.92%

4.2 *Graphical analysis*

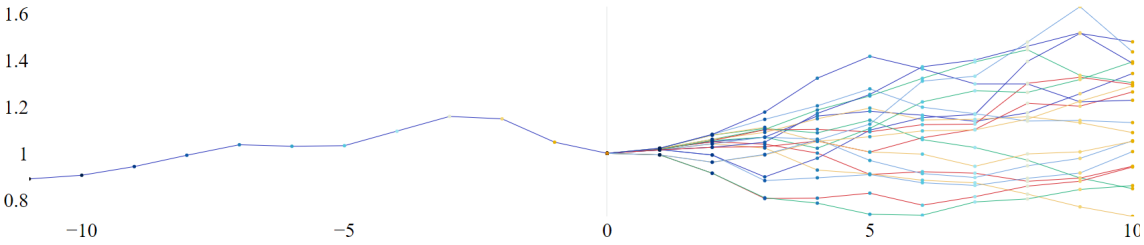
We now present the graphical results of the proposed algorithms. As we highlighted in Tables 3 and 4, the scenarios generated with Algorithm 1 and Algorithm 2 do not show a trend percistency as indicated by the value of H. Indeed, as a representative example, in Figure 3 we present the scenarios for the asset World-GovBond generated with Algorithm 2 assuming three different levels of H (0.1, 0.5, 0.9). For illustrative purpose, we limit the number of scenarios to 20. We observe that they look almost indistinguishable with



(a) *H=0.1*



(b) *H=0.5*



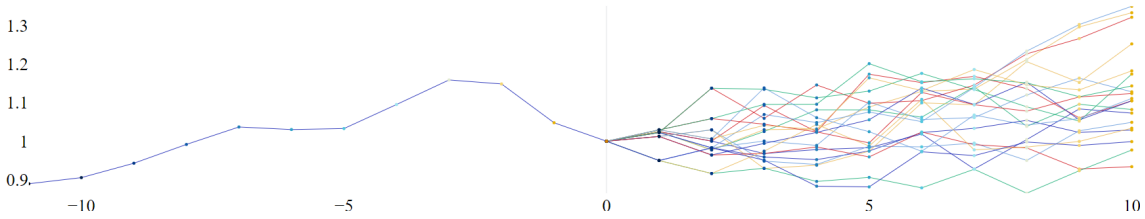
(c) *H=0.9*

Figure 3: Historical path and scenario tree with 20 scenarios generated with Algorithm 2 for asset World-GovBond

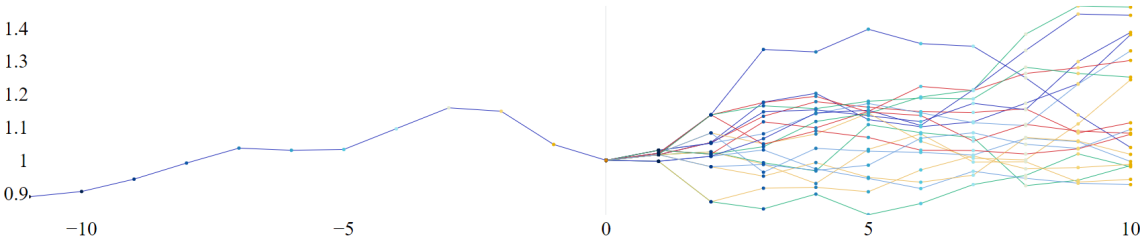
respect to the value of H. The scenarios of the other assets show the same inconsistency. Therefore, we focus on Algorithm 3 that in Table 5 already showed its capability to produce cumulative returns with the desired trend percistency. Observing Figure 4, we can appreciate the impact of H on the scenario trees generated with Algorithm 3. Indeed, it is possible to distinguish well the different path behaviors, and to notice that:

- when $H = 0.1$ the trends are anti-persistent meaning that if a return in a given node is positive, there is a higher probability that in the next node, the return is negative, and vice-versa;
- when $H = 0.5$ the trend is not considered and we observe the classic GBM random walk;
- when $H = 0.9$ the trends are highly persistent meaning that if a return in a given node is positive, there is a higher probability that in the next node the return is still positive, and vice-versa.

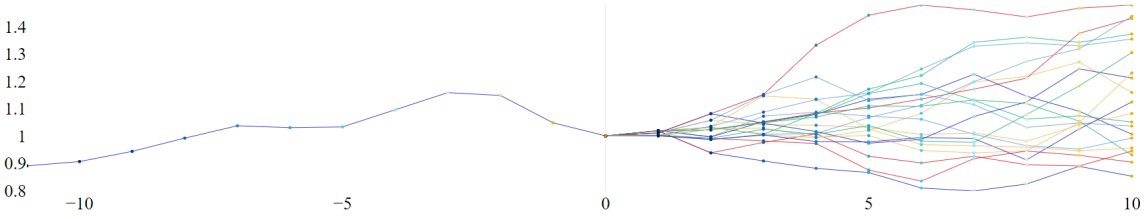
Examples of other assets that confirm the same graphical evidences are in Figures 5, 6, and 7.



(a) *H=0.1*

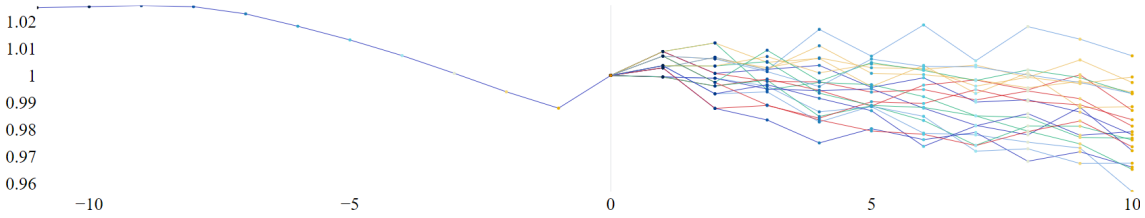


(b) *H=0.5*

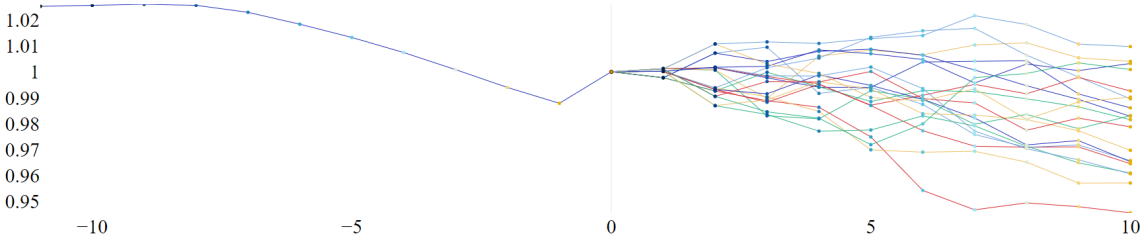


(c) *H=0.9*

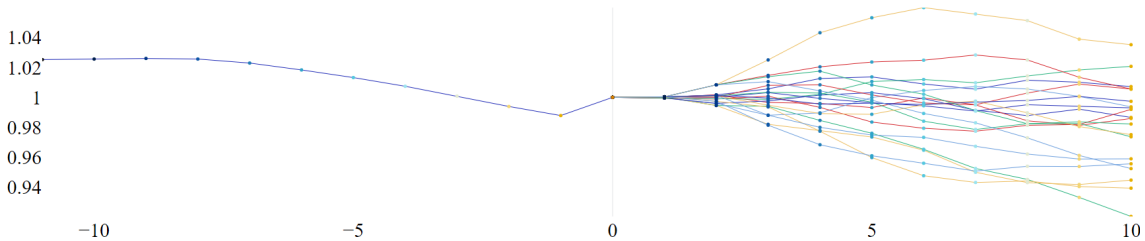
Figure 4: Historical path and scenario tree with 20 scenarios generated with Algorithm 3 for asset World-GovBond



(a) *H=0.1*

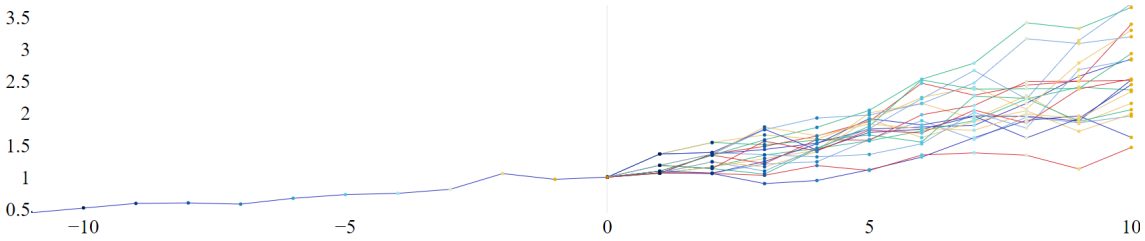


(b) *H=0.5*

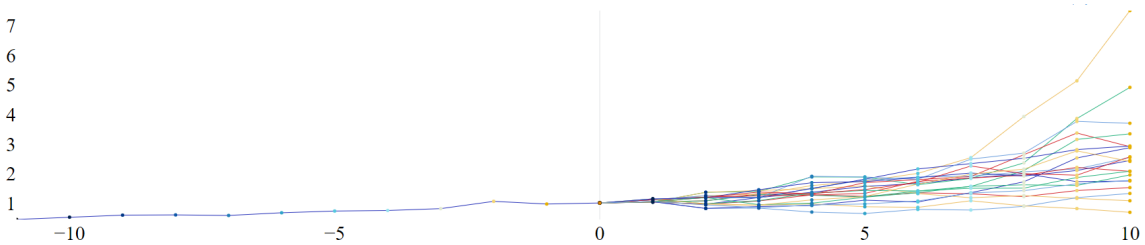


(c) *H=0.9*

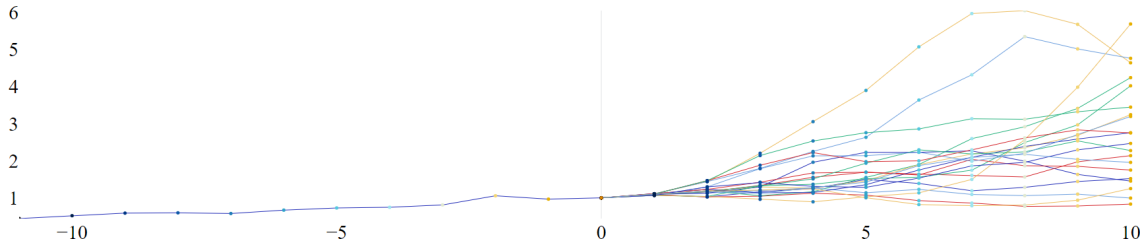
Figure 5: Historical path and scenario tree with 20 scenarios generated with Algorithm 3 for asset Liquidity



(a) *H=0.1*

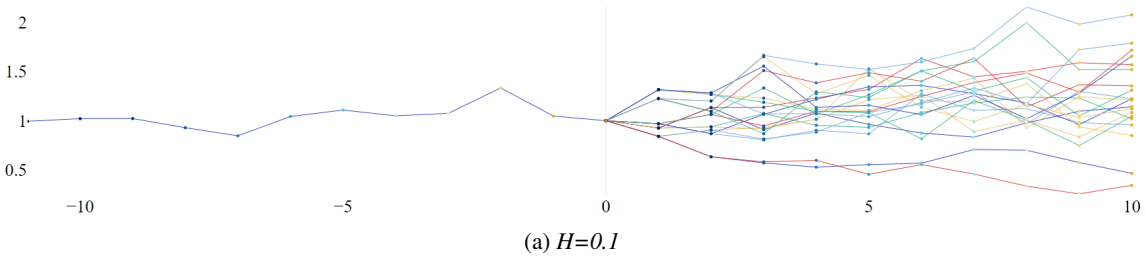


(b) *H=0.5*

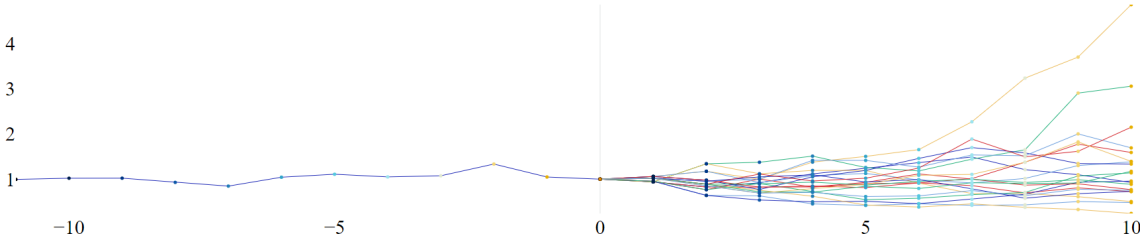


(c) *H=0.9*

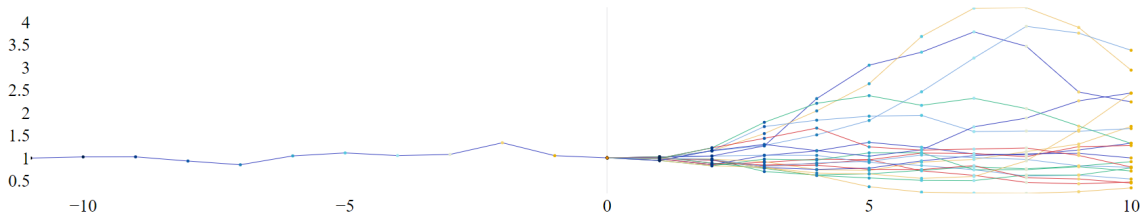
Figure 6: Historical path and scenario tree with 20 scenarios generated with Algorithm 3 for asset WorldEquityESG



(a) *H=0.1*



(b) *H=0.5*



(c) *H=0.9*

Figure 7: Historical path and scenario tree with 20 scenarios generated with Algorithm 3 for asset EmMarkEquity

5 Conclusion

In this paper, we focused on the generation of scenarios based on the FBM to populate a multistage scenario tree. The FBM represents a generalization of the ordinary BM. This extension allows us to account for the long-memory effect or, conversely, the reversal effect, in time series. This behavior has been studied especially for financial time series. For this reason, we proposed three different algorithms that adopt FBM to generate scenario trees representing the evolution of the financial returns of a set of assets. In particular, we consider an asset universe of 10 ETF, each representing peculiar performance characteristics. Comparing the statistics of the cumulative returns of the scenario trees that evolve with an horizon of 10 years, we highlighted the differences among the three algorithms and we showed that the third is capable to represent effectively the trend percistency and, thus, to capture the real market behavior when the Hurst coefficient is greater than 0.5. Indeed, when we assumed a persistent trend, the cumulative returns allowed the asset to reach a relevant average return, but remarking a potential huge losses on the worse scenarios.

To conclude, we believe that the adoption of FBM in stochastic optimization problem could help to give a better representation of the uncertain parameters. This is particularly true when these quantities regard financial returns, but it could be applied in many other fields where the movement of the random variables show some trend with a clear persistency. Of course, after the generation of a scenario tree with the FBM, a natural extension of this work would be to test how much the use of uncertain parameters with a persistent or an anti-persistent trend affects the optimal solution of a stochastic optimization model.

Acknowledgments

The work of Sebastiano Vitali was supported by MIUR-ex60% 2023 and 2024 sci.resp. Sebastiano Vitali. The work of Vittorio Moriggia was supported by MIUR-ex60% 2024 sci.resp. Vittorio Moriggia.

Declarations

Conflict of interest: No potential conflict of interest was reported by the authors.
Consent for publication: The authors have consented to the submission of the work to the journal.

References

E. M. L. Beale. On Minimizing a Convex Function Subject to Linear Inequalities. *Journal of the Royal Statistical Society: Series B (Methodological)*, 17(2):173–184, 1955. ISSN 00359246. doi: 10.1111/j.2517-6161.1955.tb00191.x.

F. Black and M. Scholes. The Pricing of Options and Corporate Liabilities. *Journal of Political Economy*, 81(3):637–654, 1973. doi: 10.1086/260062.

A. Charnes and W. W. Cooper. Chance-Constrained Programming. *Management Science*, 6(1):73–79, 1959. ISSN 0025-1909, 1526-5501. doi: 10.1287/mnsc.6.1.73.

M. Corazza and A. G. Malliaris. Multifractality in foreign currency markets. *Economic Uncertainty, Instabilities And Asset Bubbles: Selected Essays*, 6(2):151, 2005.

G. B. Dantzig. Linear Programming under Uncertainty. *Management Science*, 1(3-4):197–206, 1955. ISSN 0025-1909, 1526-5501. doi: 10.1287/mnsc.1.3-4.197.

T. Dieker. *Simulation of Fractional Brownian Motion*. Doctoral dissertation, Masters Thesis, University of Twente, Department of Mathematical Sciences, The Netherlands, 2004.

E. F. Fama. Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance*, 25(2):383, 1970. ISSN 00221082. doi: 10.2307/2325486.

J. Feder. *Fractals*. Physics of Solids and Liquids. Plenum Press, New York, 1988. ISBN 978-0-306-42851-7.

Y. Hu and B. Øksendal. Fractional White Noise Calculus and Applications to Finance. *Infinite Dimensional Analysis, Quantum Probability and Related Topics*, 06(01):1–32, 2003. ISSN 0219-0257, 1793-6306. doi: 10.1142/S0219025703001110.

J. Hull. *Options, Futures, and Other Derivatives*. Pearson, Harlow, England, eleventh edition, global edition edition, 2022. ISBN 978-1-292-41065-4.

H. E. Hurst. Long-Term Storage Capacity of Reservoirs. *Transactions of the American Society of Civil Engineers*, 116(1):770–799, 1951. ISSN 0066-0604, 2690-4071. doi: 10.1061/TACEAT.0006518.

I. Karatzas and S. E. Shreve. *Brownian Motion and Stochastic Calculus*. Number 113 in Graduate Texts in Mathematics. Springer, New York, 2nd ed edition, 1996. ISBN 978-0-387-97655-6 978-3-540-97655-4.

A. N. Kolmogorov. Wienersche spiralen und einige andere interessante kurven in hilbertscen raum. *Acad. Sci. URSS (NS)*, 26:115–118, 1940.

B. B. Mandelbrot and R. L. Hudson. *The (Mis)Behaviour of Markets: A Fractal View of Risk, Ruin, and Reward*. Profile, London, pbk. ed edition, 2008. ISBN 978-1-84668-262-9.

B. B. Mandelbrot and J. W. Van Ness. Fractional Brownian Motions, Fractional Noises and Applications. *SIAM Review*, 10(4):422–437, 1968. ISSN 0036-1445, 1095-7200. doi: 10.1137/1010093.

H. Markowitz. Portfolio Selection. *The Journal of Finance*, 7(1):77, 1952. ISSN 00221082. doi: 10.2307/2975974.

B. Øksendal. *Fractional Brownian Motion in Finance*. Matematisk Institutt, Universitetet i Oslo, 2003.

E. E. Peters. *Fractal Market Analysis: Applying Chaos Theory to Investment and Economics*. Wiley Finance Editions. J. Wiley & Sons, New York, 1994. ISBN 978-0-471-58524-4.

R. T. Rockafellar and S. Uryasev. Optimization of conditional value-at-risk. *The Journal of Risk*, 2(3): 21–41, 2000. ISSN 14651211. doi: 10.21314/JOR.2000.038.

A. T. A. Wood and G. Chan. Simulation of Stationary Gaussian Processes in [0,1] d. *Journal of Computational and Graphical Statistics*, 3(4):409, 1994. ISSN 10618600. doi: 10.2307/1390903.