
Fractional Brownian Motions in Financial Models and Their Monte Carlo Simulation

Masaaki Kijima and Chun Ming Tam

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/53568>

1. Introduction

Fractional Brownian motion (fBm) was first introduced within a Hilbert space framework by Kolmogorov [1], and further studied and coined the name ‘fractional Brownian motion’ in the 1968 paper by Mandelbrot and Van Ness [2]. It has been widely used in various scientific fields, most notably in hydrology as first suggested in [3]. It also plays an important role in communication technology by enriching the queuing theory in terms of simulating real network traffic.

In recent years, it has been steadily gaining attention in the area of finance, as it appears that the traditional stochastic volatility model driven by ordinary Brownian motion implies geometric time-decay of the volatility smile, which is much faster than in what we see in real market data (which has a hyperbolic decay). Such stylized feature is called the *volatility persistence*, see [4] for detail discussion about the statistical test for the existence of long-range memory (author has also proposed a robust extension of the R/S statistics for that particular purpose). Several modeling approaches have been suggested capturing this persistence in conditional variance either via a unit-root or long memory process. In order to keep the pricing-framework largely intact, it is more interesting to study the long memory process, and fBm has a particular good match due to its similarity to the ordinary Brownian motion and its Gaussian properties.

In this chapter, we will outline several approaches to simulate fractional Brownian motion with $H > 1/2$, including the exact methods and approximate methods, where the Hurst Index H is a parameter used in literature to generalize Brownian motion into fractional Brownian motion, first made popular by Benoit Mandelbrot, which we will give a detail definition in Definition 1.1. We also provide a brief introduction of the truncated fractional Brownian motion (long-memory model in continuous time) as proposed in [5,6].

1.1. Financial motivation and backgrounds

Numerous empirical studies have pointed out that, in options markets, the implied volatility back-out from the Black-Scholes equation displays volatility skews or smiles; the smile effect, which is well known to practitioners, refers to the U-shape price distortion on the implied volatility surface.

In Hull and White [7] and Scott [8], they have proposed this feature of volatility to be captured by stochastic regime, known as the *stochastic volatility model*:

$$\begin{cases} \frac{dS(t)}{S(t)} = \mu(t, S_t)dt + \sigma(t)dB^1(t) \\ d(\ln \sigma(t)) = k(\theta - \ln(\sigma(t)))dt + \nu dB^2(t) \end{cases} \quad (1)$$

Here, $S(t)$ is the asset price and $\sigma(t)$ is the instantaneous volatility at time t , and $\{B^1(t), B^2(t)\}$ are ordinary standard Brownian motions. Hull and White [7] have shown that, the price of European option at time t of exercise date T is the conditional expectation of the Black Scholes option pricing formula where the constant volatility from the original formula is replaced by the quadratic average over the period $[t, T]$:

$$\sigma_{t,T}^2 = \frac{1}{T-t} \int_t^T \sigma^2(u) du \quad (2)$$

Such models are successful at capturing the symmetric smile and skewness of the implied volatility by imposing relations between the driving Brownian motions in (1) (symmetric smile is explained by independence while skewness can be explained by linear correlation).

Due to the temporal aggregation effect evident in (2), however, the smile effect deteriorates along with time-to-maturity since the temporal aggregation gradually erases the conditional heteroskedasticity; in the standard stochastic volatility setup, this particular decay is much faster than what is observed in market data. The phenomenon of slow decaying volatility smile is known as the *volatility persistence* (long-range dependence of volatility process). This phenomenon is particularly poignant for high frequency data, for which the conditional variance process displays near unit-root behavior.

Furthermore, we emphasize the existence of such phenomenon collaborated by large quantities of researches, pointing out that the conditional volatility of asset returns also displays long range dependence: [9-12] have discussed extensively the evidence of such phenomenon in empirical data. Bayesian estimation in [13] of stochastic volatility models shows similar patterns of persistence.

Motivated by this inadequacy, long-memory process was deemed more appropriate enrichment for this purpose. Hence, fractional Brownian motion is a prime candidate among all long-memory process given its tractability and similarity with the ordinary Brownian motion: both the fractional Brownian motion and ordinary Brownian motion are self-similar

with similar Gaussian structure. For discussions of estimation and evidence of the long-range dependence in conditional variance of asset returns, the reader is referred to [10] and section 3.1 of [14].

Now, we provide a formal definition of fractional Brownian motion (fBm). We adopt the definition as given in [15].

Definition 1.1 Let $H \in (0,1)$ be a constant. A fractional Brownian motion $\{B^H(t)\}_{t \geq 0}$ with Hurst index H is a continuous and centered Gaussian process with covariance function

$$\mathbb{E}[B^H(t)B^H(s)] = \frac{1}{2}(t^{2H} + s^{2H} - |t-s|^{2H}) \tag{3}$$

In particular, for $H = \frac{1}{2}$, it reduces to the ordinary Brownian motion with $\mathbb{E}[B^H(t)B^H(s)] = \min(t, s)$.

From equation (3) we have the following properties:

1. $B^H(0) = 0$ and $E[B^H(t)] = 0, \forall t \geq 0$.
2. $B^H(\cdot)$ has stationary increment: $B^H(t+s) - B^H(s)$ has the same distribution as $B^H(t)$ for any $s, t \geq 0$.
3. $B^H(\cdot)$ is self-similar, meaning that $B^H(Tt)$ has the same distribution law as $(T)^H B^H(t)$.
4. $B^H(\cdot)$ is a Gaussian process with the variance $E[B^H(t)^2] = t^{2H}, \forall t \geq 0$.
5. $B^H(\cdot)$ has continuous trajectories.

Fractional Brownian motions are divided into three very different categories: $H < \frac{1}{2}$, $H = \frac{1}{2}$, $H > \frac{1}{2}$. This is of particular importance because there is a deterministic difference between the case of $H < \frac{1}{2}$ and $H > \frac{1}{2}$, as we introduce the mathematical notion of long-range dependence.

Definition 1.2 (Long-range dependence) A stationary sequence $\{X_n\}_{n \in \mathbb{N}}$ exhibits long-range dependence if the autocovariance function $\gamma(n) := cov(X_k, X_{k+n})$ satisfies

$$\lim_{n \rightarrow \infty} \frac{\gamma(n)}{cn^{-\alpha}} = 1 \tag{4}$$

This can be written as $(n) \sim |n|^{-\alpha}$,

In this case, for some constants c and $\alpha \in (0,1)$, the dependence between X_k and X_{k+n} decays slowly as $n \rightarrow \infty$ and

$$\sum_{n=1}^{\infty} \gamma(n) = \infty$$

If we set $X_k = B^H(k) - B^H(k-1)$ and $X_{k+n} = B^H(k+n) - B^H(k+n-1)$ and apply equation (3), we have

$$\gamma_H(n) = \frac{1}{2}[(n+1)^{2H} + (n-1)^{2H} - 2n^{2H}]$$

where $\gamma_H(n) = \text{cov}(B^H(k) - B^H(k-1), B^H(k+n) - B^H(k+n-1))$. In particular,

$$\lim_{n \rightarrow \infty} \frac{\gamma_H(n)}{H(2H-1)n^{2H-2}} = 1$$

Therefore, we conclude that: for $H > 1/2$, $\sum_{n=1}^{\infty} \gamma_H(n) = \infty$, and for $H < 1/2$, $\sum_{n=1}^{\infty} |\gamma_H(n)| < \infty$. Hence, only in the case of $H > 1/2$, fractional Brownian motions display long-memory dependence.

As pointed out in [16], large lags difference between $\gamma(\cdot)$ may be difficult to estimate in practice, so that models with long-range dependence are often formulated in terms of self-similarity. Self-similarity allows us to extrapolate across time scales and deduce long time behavior from short time behavior, which is more readily observed.

Because we are interested in capturing the long-memory phenomenon observed in financial markets, the rest of this chapter will only concern the case of $H > 1/2$.

1.2. Stochastic integral representation

In the original paper [2], Mandelbrot and van Ness represent the fBm in stochastic integral with respect to the ordinary Brownian motion:

$$B^H(t) = \frac{1}{\Gamma(H + \frac{1}{2})} \left(\int_{-\infty}^0 [(t-s)^{H-\frac{1}{2}} - (-s)^{H-\frac{1}{2}}] dB(s) + \int_0^t (t-s)^{H-\frac{1}{2}} dB(s) \right) \quad (5)$$

where $\Gamma(\cdot)$ is the gamma function.

They have also included an alternative representation of the fBm which is the basis of the model in [5,6]:

$$\hat{B}^H(t) = \int_0^t \frac{(t-s)^{H-\frac{1}{2}}}{\Gamma(H + \frac{1}{2})} dB(s) \quad (6)$$

This version of fBm is 'truncated' in the sense that the integration from negative infinity to zero in equation (5) is truncated. We will refer to the model (6) as the 'truncated fractional Brownian motion' in the rest of this chapter. As pointed out in [2], the representation (6) was first proposed by Paul Lévy to define the fBm by the Riemann-Liouville fractional integral, while the original integral in equation (5) is the Weyl fractional integral.

The definition (6) of fBm is in general not asymptotically covariance-stationary, even though it retains self-similarity. For further technical discussion and rigorous definition of the truncated fractional Brownian motion, we refer the reader to [5].

Given the difference, if one employs analytical tools such as Malliavin calculus should be applied with care and note the differences between the two versions of fBM. [17] offers an in-depth discussion of the theoretical differences.

2. Fractional brownian motions in financial models

We first look at several examples that utilize the fractional Brownian motions in the realm of financial modeling.

2.1. Asset price model

In the previous section, we mention that the motivation of fBMs in finance models is to capture the long-range dependence in the volatility process. However, it is worth discussing the possibility of applying it to the asset process itself.

In practice, it is considered that an asset process driven by fBm will result in arbitrage. The idea is that, since for $H \neq 1/2$, $B^H(t)$ is not a semimartingale in continuous time, the asset process described by $B^H(t)$ violates the NFLVR (no free lunch with vanishing risk), a weaker version of arbitrage, and thus doesn't admit an equivalent martingale measure according to Theorem 7.2 of [18]. Such findings and construction of arbitrage can be found in Rogers [19].

In contrast, Cheridito [20] has given multiple classes of trading strategies that allow various level of arbitrages (NFLVR, arbitrage in the classical sense and strong arbitrage) under fBm driven assets, and shown that the arbitrages are all eliminated if the inter-transaction time is not zero, i.e., the classes of strategies become arbitrage-free. Such assumption is reasonable in practice, given the physical limit and non-zero transaction cost. For more information on arbitrage theory and its generalization, the reader is referred to [18-20].

2.2. Stochastic volatility model

As we have mentioned in the introductory section, the main motivation for fBm is to capture the volatility persistence, the stylized feature observed in empirical data. There are several prominent models involving a volatility process with fBm. Here, we just outline several of them.

2.2.1. Long-memory continuous model

In [5, 6], Comte and Renault consider the following stochastic volatility model driven by fBm:

$$\begin{cases} \frac{dS(t)}{S(t)} = rdt + \sigma(t)dB(t) \\ \sigma(t) = \sigma_0 e^{x(t)} \\ dx(t) = -kx(t)dt + v d\hat{B}^H(t) \end{cases} \quad (7)$$

where the log-volatility term follows a fractional-OU process driven by the truncated fBm (6). The asset price process follows a geometric Brownian motion with volatility persistence.

Although Mandelbrot [2] deemed it as signifying the origin too heavily, the model (7) is easier and more robust than the ordinary fBMs from the perspective of simulation. A simulation example is explored in Section 5.

2.2.2. Affine fractional stochastic volatility model

In [21], Comte et al. assume that the squared-volatility process follows

$$\sigma^2(t) = \theta + X^{(\alpha)}(t) \quad (8)$$

where $X^{(\alpha)}(t)$ is defined by the fractional integral:

$$X^{(\alpha)}(t) = \int_{-\infty}^t \frac{(t-s)^{H-1/2}}{\Gamma(H+1/2)} x(s) ds \quad (9)$$

Similar treatment of the fractional integration is outlined in [5]. The affine structure in (8) is similar to the one originally studied by Duffie et al. [22].

The affine structure is adopted for the extra tractability, and thus better suited for practical option pricing and hedging than the original idea (7). In fact, Comte et al. [21] have shown that this model can better depict the difference between the short and long memory properties in the resulting option prices.

2.2.3. Martingale expansion

Fukasawa [23] adopts and expands the asymptotic expansion technique first proposed by Yoshida [24] of European option prices around the Black-Scholes equation by means of perturbation technique and partial Malliavin calculus. It is shown that the logarithm of the asset process can be expressed as

$$\ln S_t = Z_t = rt - \frac{1}{2} \int_0^t g(Y_s^n)^2 ds + \int_0^t g(Y_s^n) [\theta dW_s' + \sqrt{1 - \theta^2} dW_s]$$

with

$$Y_s^n = y + \epsilon_n W_s^H, \quad W_t^H = \int_0^t K_H(t, s) dW_s'$$

Here, r is the riskless rate, $\theta \in (-1, 1)$, $y \in \mathbb{R}$ is a constant, (W, W') is a 2-dimensional standard (independent) Brownian motion, $\epsilon_n \rightarrow 0$ is a deterministic sequence for $n \rightarrow \infty$, and $g(\cdot)$ is the stochastic volatility process which is an adapted process for the minimal filtration.

Note that W_t^H is a fractional Brownian motion with Hurst index H , where $K_H^t(t, s)$ is the kernel of the stochastic integral representation over a finite interval of Brownian motion. According to [15], pertaining to our interest, for the case of $H > 1/2$, the kernel has the following expression:

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

where

$$c_H = \left\{ \frac{H(2H-1)}{\beta(2-2H, H-1/2)} \right\}^{1/2}$$

Then, according to Corollary (2.6) of Fukasawa [23], the implied volatility can be expressed as

$$\sigma \left\{ 1 - \frac{\epsilon_n}{2} \rho_{13} d_2 \right\} + o(\epsilon_n) = a T^{H-1/2} \log(K/S) + \sigma + b T^{H+1/2} + o(\epsilon_n) \tag{10}$$

where d_2 is the typical argument in the $N(d_2)$ of the Black-Scholes formula, and

$$\rho_{13} = \frac{2\theta g'(y)c_H T^H}{g(y)}, \quad \sigma = g(y), \quad a = \frac{\theta g'(y)c_H}{\sigma} \epsilon_n, \quad b = -a \left(r - \frac{\sigma^2}{2} \right) \tag{11}$$

Equation (10) can be seen as an approximation ϵ_n is the model parameter calibrated from market data. This shape of this approximation remains largely stationary, to check the accuracy of the approximation, we have no means but performing Monte Carlo simulation of fBm.

3. Simulation with exact methods

First, we look at methods that completely capture the covariance structure and true realization of the fractional Brownian motion (fBm) or fractional Gaussian noise (fGn). Any method described in this section has their starting point at the covariance matrix. The approximate scheme we see later is merely numerically close to the value of fBm (or fGn) or asymptotically coincides with it. The collection of algorithm in this section is not meant to be exhaustive. For more algorithm and discussion, see [25].

3.1. Hosking method

The Hosking method utilizes the well-known conditional distribution of the multivariate Gaussian distribution on a recursive scheme to generate samples based on the explicit cova-

riance structure. This method generates a general stationary Gaussian process with given covariance structure, not limited to generating fBms.

More specifically, this algorithm generates an fGn sequence $\{Z_k\}$ and fBm is recovered by accumulative sum. That is, the distribution of Z_{n+1} conditioned on the previous realization Z_n, \dots, Z_1, Z_0 can be explicitly computed.

Denote $\gamma(k)$ as the autocovariance function of the zero-mean process:

$$\gamma(k) := \mathbb{E}(X_n X_{n+k})$$

where we assume for convenience that $\gamma(0)=1$. For $n, k=0,1, 2 \dots$, we have the following recursive relationship for the $(n+1) \times (n+1)$ autocovariance matrix $\Gamma(n) = \{\gamma(i-j)\}_{i,j=0,\dots,n}$:

$$\Gamma(n+1) = \begin{pmatrix} 1 & c(n)' \\ c(n) & \Gamma(n) \end{pmatrix} \tag{12}$$

$$= \begin{pmatrix} \Gamma(n) & F(n)c(n) \\ c(n)'F(n) & 1 \end{pmatrix} \tag{13}$$

where $c(n)$ is the $(n+1)$ -column vector with elements $c(n)_k = \gamma(k+1)$, $k=0, \dots, n$ and $F(n) = (1(i=n-j))_{i,j=0,\dots,n}$ is the $(n+1) \times (n+1)$ 'mirrored' identity matrix

$$F(n) = \begin{pmatrix} 0 & \dots & 0 & 1 \\ 0 & \dots & 1 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Theorem 3.1 (Multivariate Gaussian distribution) Any multivariate Gaussian random vector z can be partitioned into z_1 and z_2 with the mean vector and covariance matrix with the corresponding partition:

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \tag{14}$$

The distribution of z_1 conditioned on $z_2=a$ is a multivariate normal $(z_1 | z_2=a) \sim N(\bar{\mu}, \bar{\Sigma})$ with

$$\bar{\mu} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(a - \mu_2) \tag{15}$$

$$\bar{\Sigma} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \tag{16}$$

If we substitute equation (12) into the partition in (14) with $\Sigma_{11}=1$, $\mu=0$, we have the following expression for the conditional distribution:

$$\mu_{n+1} = \mathbb{E}(Z_{n+1} | Z_n, \dots, Z_0) = c(n)' \Gamma(n)^{-1} \begin{pmatrix} Z_n \\ \vdots \\ Z_1 \\ Z_0 \end{pmatrix} \quad (17)$$

$$\sigma_{n+1}^2 = \text{Var}(Z_{n+1} | Z_n, \dots, Z_0) = 1 - c(n)' \Gamma(n)^{-1} c(n) \quad (18)$$

With $Z_0 \sim N(0,1)$, subsequently X_{n+1} for $n=0,1, \dots$ can be generated.

Taking the inverse of $\Gamma(\cdot)$ at every step is computationally expensive; the algorithm proposed by Hosking [26] computes the inverse $\Gamma(n+1)^{-1}$ recursively. The next result is due to Dieker [25].

Proposition 3.1 (Hosking algorithm for simulating fGn) Define $d(n) = \Gamma(n)^{-1} c(n)$, and applying the blockwise method of inversion on equation (13):

$$\Gamma(n+1) = \frac{1}{\sigma_n^2} \begin{pmatrix} \sigma_n^2 \Gamma(n)^{-1} + F(n) d(n) d(n)' F(n) & -F(n) d(n) \\ -d(n)' F(n) & 1 \end{pmatrix} \quad (19)$$

where σ_{n+1}^2 satisfies the recursion

$$\sigma_{n+1}^2 = \sigma_n^2 - \frac{(\gamma^{(n+1)} - \tau_{n+1})^2}{\sigma_n^2} \quad (20)$$

with $\tau_n := d(n)' F(n) c(n) = c(n)' F(n) d(n)$. Also, the recursion for $d(n+1) = \Gamma(n+1)^{-1} c(n+1)$ is obtained as

$$d(n+1) = \begin{pmatrix} d(n) - \phi_n F(n) d(n) \\ \phi_n \end{pmatrix}$$

where

$$\phi_n = \frac{\gamma^{(n+2)} - \tau_n}{\sigma_n^2}$$

With $\mu_1 = \gamma(1) Z_0$, $\sigma_1^2 = 1 - \gamma(1)^2$, $\tau_0 = \gamma(1)^2$, μ_{n+1} , σ_{n+1}^2 , τ_{n+1} can be readily computed, and fractional Brownian motion is recovered by the cumulative summation.

This algorithm is also applicable to non-stationary processes (see [27] for details). Even though this algorithm is very simple and easy to understand and sample paths can be generated on-the-fly, the complexity of this algorithm is of $O(N^2)$ and computational (as well as memory) expense of this algorithm grows at a prohibitive speed.

3.2. Cholesky method

Given that we are dealing with the covariance structure in matrix form, it is natural to go with the Cholesky decomposition: decomposing the covariance matrix into the product of a lower triangular matrix and its conjugate-transpose $\Gamma(n)=L(n)L(n)^*$. If the covariance matrix is proven to be positive-definite (the situation will be addressed in the next subsection), $L(n)$ will have real entries and $\Gamma(n)=L(n)L(n)'$.

Suppose that in matrix form the $(n+1) \times (n+1)$ product is given by

$$\begin{pmatrix} \gamma(0) & \gamma(1) & \gamma(2) & \dots & \gamma(n) \\ \gamma(1) & \gamma(0) & \gamma(1) & \dots & \gamma(n-1) \\ \gamma(2) & \gamma(1) & \gamma(0) & \dots & \gamma(n-2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \gamma(n) & \gamma(n-1) & \gamma(n-2) & \dots & \gamma(0) \end{pmatrix} = \begin{pmatrix} l_{00} & 0 & 0 & \dots & 0 \\ l_{10} & l_{11} & 0 & \dots & 0 \\ l_{20} & l_{21} & l_{22} & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ l_{n0} & l_{n1} & l_{n2} & \dots & l_{nn} \end{pmatrix} \times \begin{pmatrix} l_{00} & l_{10} & l_{20} & \dots & l_{n0} \\ 0 & l_{11} & l_{21} & \dots & l_{n1} \\ 0 & 0 & l_{22} & \dots & l_{n2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & l_{nn} \end{pmatrix}$$

It is easy to see that $l_{00}^2 = \gamma(0)$ and that $l_{10}l_{00} = \gamma(1)$ and $l_{10}^2 + l_{11}^2 = \gamma(0)$ on $i=1$ (2nd row). For $i \geq 1$, the entries of the lower triangular matrix can be determined by

$$l_{i,0} = \frac{\gamma(i)}{l_{0,0}}$$

$$l_{i,j} = \frac{1}{l_{j,j}} \left(\gamma(i-j) - \sum_{k=0}^{j-1} l_{i,k}^{j-1} l_{j,k} \right), \quad 0 < j \leq n$$

$$l_{i,i}^2 = \gamma(0) - \sum_{k=0}^{i-1} l_{i,k}^2$$

Given independent, identically distributed (i.i.d.) standard normal random variables $(V_i)_{i=0, \dots, n+1}$, the fGn sequence is generated by

$$Z_{n+1} = \sum_{k=0}^{n+1} l_{n+1,k} V_k$$

Or in matrix form, we have $Z(n)=L(n)V(n)$. If $\Gamma(n)$ is assumed to be positive-definite, the non-negativity of $l_{i,i}^2$ is guaranteed and $L(n)$ is guaranteed to be real. The covariance structure of the process is captured, since

$$Cov(Z(n)) = Cov(L(n)V(n)) = L(n)Cov(V(n))L(n)' = L(n)L(n)' = \Gamma(n) \tag{21}$$

Even though the Cholesky method is easy to understand and implement, the computation time is $O(N^3)$, which renders this scheme extremely uneconomical in practice. To resolve this problem, we will proceed to another exact method. The idea is similar to retain the same relation as equation (21), but with a different decomposition.

3.3. Fast fourier transform method

As we have seen from the last section, using the Cholesky decomposition seems to be the most straightforward idea to simulate Gaussian process with a given covariance structure;

but, it also is the most rudimentary and thus slow. In order to improve upon the speed, the idea of utilizing the fast Fourier transform (FFT) was proposed by Davies and Harte [28] and further generalized by Dietrich and Newsam [29].

Similar to the idea before, this method tries to find a decomposition of the covariance matrix as $\Gamma = GG'$ and the sample is generated by $y = Gx$ for given standard normal random variable x . Then, on the given covariance structure, we have

$$Cov(y) = Cov(Gx) = G Cov(x)G' = GG' = \Gamma$$

The idea is to 'embed' the original covariance matrix a circulant matrix in order to carry out the FFT. Before we outline the idea, we shall give out some detail of the linkage between Fourier transform and the circulant matrix.

Definition 3.1 (Circulant matrix) Circulant matrix is a special case of the Toeplitz matrix where each row vector is shifted to the right (the last element is shifted back to the beginning of the row). In matrix form, an n -by- n circulant matrix can be written as

$$C = \begin{pmatrix} c_0 & c_{n-1} & c_{n-2} & \dots & c_1 \\ c_1 & c_0 & c_{n-1} & \dots & c_2 \\ c_2 & c_1 & c_0 & \dots & c_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{n-1} & c_{n-2} & \dots & c_1 & c_0 \end{pmatrix}$$

Remark: As one can see, the first row/column completely describes the whole matrix, and it can be put more succinctly in the following form:

$$c_{j,k} = c_{j-k(\text{mod } n)}, \text{ where } 0 \leq j, k \leq n - 1$$

Note that the indices range from 0 to $n-1$ instead of the usual convention that ranges from 1 to n .

Definition 3.2 (Generating circulant matrix) We define an n -by- n generating circulant matrix by

$$G = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

By a simple calculation, we can see that the 'square' of the generating circulant matrix is given by

$$G^2 = \begin{pmatrix} 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

From the point of view of row and column operation of the matrix, this can be seen as each row of the matrix being shifted one element forward, where the bumped element is replaced to the end of the row (it can also be thought of as the whole row is shifted down and the bumped row is placed back on top, but this is irrelevant to our interest). Arbitrary power can be deduced accordingly; this operation has a cycle of n iterations.

The generating circulant matrix is served as our building block. Looking back at our original circulant matrix, we have a corresponding polynomial

$$p(x) = c_0 + c_1x + c_2x^2 + \dots + c_{n-1}x^{n-1} \tag{22}$$

Then, the original circulant matrix C can be expressed as

$$C = \begin{pmatrix} c_0 & c_{n-1} & c_{n-2} & \dots & c_2 & c_1 \\ c_1 & c_0 & c_{n-1} & \dots & c_3 & c_2 \\ c_2 & c_1 & c_0 & \dots & \vdots & c_3 \\ c_3 & c_2 & c_1 & \dots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & c_{n-1} \\ c_{n-1} & c_{n-2} & c_{n-3} & \dots & c_1 & c_0 \end{pmatrix}$$

$$C = p(G) = c_01 + c_1G + c_2G^2 + \dots + c_{n-1}G^{n-1} \tag{23}$$

This can be verified by doing the row-operation of arbitrary power on G as shown above. It can be shown that each operation is one-element sub-diagonal compared to the previous power.

Definition 3.3 (Fourier matrix) The Fourier matrix is introduced as

$$F = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 & 1 \\ 1 & \xi & \xi^2 & \dots & \xi^{n-2} & \xi^{n-1} \\ 1 & \xi^2 & \xi^{2 \times 2} & \dots & \vdots & \xi^{n-2} \\ 1 & \xi^3 & \xi^{3 \times 2} & \dots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \xi^2 \\ 1 & \xi^{n-1} & \xi^{n-2} & \dots & \xi^2 & \xi \end{pmatrix}$$

Here, we define the n -th unity root as $\omega = e^{2\pi i \frac{1}{n}}$, and $\xi = \bar{\omega} = e^{-2\pi i \frac{1}{n}}$ is the conjugate of the unity root.

The Fourier matrix can be defined using the positive argument ω instead of ξ . Also, as we will see later, some definition includes the normalizing scalar $\frac{1}{\sqrt{n}}$ (or $\frac{1}{n}$). This is analogous to the continuous counterpart of the Fourier integral definition $F(f) = \int_{-\infty}^{\infty} x(t)e^{-2\pi ift} dt$ or $\int_{-\infty}^{\infty} x(t)e^{2\pi ift} dt$, as long as the duality is upheld by the opposite sign in the exponential power in the inverse Fourier transform. This duality will be restated in the diagonalization representation of the circulant matrix later.

Proposition 3.2 If $\frac{1}{\sqrt{n}}$ normalizes the Fourier matrix, then $\frac{1}{\sqrt{n}}F$ is a unitary matrix. It is symmetric (i.e., $F^T = F$), and the inverse of the Fourier matrix is given by

$$F^{-1} = \left(\frac{\sqrt{n}}{\sqrt{n}} F^{-1} \right) = \frac{1}{\sqrt{n}} \left(\frac{1}{\sqrt{n}} F \right)^{-1} = \frac{1}{\sqrt{n}} \left(\frac{1}{\sqrt{n}} \bar{F}^T \right) = \frac{1}{n} \bar{F} = \frac{1}{n} \begin{pmatrix} 1 & 1 & 1 & \dots & 1 & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{n-2} & \omega^{n-1} \\ 1 & \omega^2 & \omega^{2 \times 2} & \dots & \vdots & \omega^{n-2} \\ 1 & \omega^3 & \omega^{3 \times 2} & \dots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \omega^2 \\ 1 & \omega^{n-1} & \omega^{n-2} & \dots & \omega^2 & \omega \end{pmatrix}$$

Proposition 3.3 If we multiply the Fourier matrix with the generating circulant matrix, we have

$$FG = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 & 1 \\ 1 & \xi & \xi^2 & \dots & \xi^{n-2} & \xi^{n-1} \\ 1 & \xi^2 & \xi^{2 \times 2} & \dots & \vdots & \xi^{n-2} \\ 1 & \xi^3 & \xi^{3 \times 2} & \ddots & \vdots & \xi^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \xi^2 \\ 1 & \xi^{n-1} & \xi^{n-2} & \dots & \xi^2 & \xi \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & \vdots & 0 \\ 0 & 0 & 1 & \ddots & \vdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & 0 \\ 0 & 0 & 0 & \dots & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & \dots & 1 & 1 & 1 \\ \xi & \xi^2 & \dots & \xi^{n-2} & \xi^{n-1} & 1 \\ \xi^2 & \xi^{2 \times 2} & \dots & \vdots & \xi^{n-2} & 1 \\ \xi^3 & \xi^{3 \times 2} & \ddots & \vdots & \vdots & 1 \\ \vdots & \vdots & \ddots & \vdots & \xi^2 & 1 \\ \xi^{n-1} & \xi^{n-2} & \dots & \xi^2 & \xi & 1 \end{pmatrix}$$

This is the same as shifting (rotating) the first column to the back of the matrix, and is also equivalent to multiplying the first row with ξ^0 , the 2nd row with ξ^1 , etc. In matrix operation, it can be seen as

$$FG = \begin{pmatrix} \xi^0 & 0 & 0 & \dots & 0 & 0 \\ 0 & \xi^1 & 0 & \dots & 0 & 0 \\ 0 & 0 & \xi^2 & \dots & \vdots & 0 \\ 0 & 0 & 0 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & 0 \\ 0 & 0 & 0 & \dots & 0 & \xi^{n-1} \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & \dots & 1 & 1 \\ 1 & \xi & \xi^2 & \dots & \xi^{n-2} & \xi^{n-1} \\ 1 & \xi^2 & \xi^{2 \times 2} & \dots & \vdots & \xi^{n-2} \\ 1 & \xi^3 & \xi^{3 \times 2} & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \xi^2 \\ 1 & \xi^{n-1} & \xi^{n-2} & \dots & \xi^2 & \xi \end{pmatrix} = \Lambda F$$

where Λ is the diagonal matrix with the k -th diagonal $\Lambda_k = \xi^k$, for $0 \leq k \leq n - 1$. It follows that

$$FGF^{-1} = \Lambda \tag{24}$$

That is, the Fourier matrix diagonalizes the generating circulant matrix with eigenvalues $\{\xi^k\}_{0 \leq k \leq n-1}$.

Theorem 3.2 The circulant matrix is decomposable by the Fourier matrix, i.e. $C = F^{-1}\Lambda F$ with eigenvalue matrix $\Lambda = \{p(\xi^k)\}_{k=0 \dots n-1}$. Also, with equation (23), the diagonalization of C can be written as

$$\begin{aligned} FCF^{-1} &= F(c_01 + c_1G + c_2G^2 + \dots + c_{n-1}G^{n-1})F^{-1} \\ &= c_01 + c_1(FGF^{-1}) + c_2(FGF^{-1})^2 + \dots + c_{n-1}(FGF^{-1})^{n-1} \\ &= \begin{pmatrix} p(1) & 0 & 0 & \dots & 0 & 0 \\ 0 & p(\xi) & 0 & \dots & 0 & 0 \\ 0 & 0 & p(\xi^2) & \dots & \vdots & 0 \\ 0 & 0 & 0 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & 0 \\ 0 & 0 & 0 & \dots & 0 & p(\xi^{n-1}) \end{pmatrix} \end{aligned}$$

Note that $(FGF^{-1})^2 = FGF^{-1}FGF^{-1} = FGGF^{-1} = FG^2F^{-1}$. The other powers can be deduced iteratively.

This theorem gives us the fundamental theoretical framework to build up the FFT exact simulation of fBms. The basic idea of the simulation is to embed the covariance matrix into a bigger circulant matrix to carry out the discrete Fourier transform as outlined above (with technique of FFT). Such technique is called Circulant Embedding Method (CEM), and is outlined in Dietrich and Newsam [29] and Perrin et al. [30].

Suppose that we need sample size of N (N should be a power of 2, i.e. $N = 2^s$ for some $g \in \mathbb{N}$ for the sake of convenience when facilitating FFT). Generate the N -by- N covariance matrix Γ with entries $\Gamma_{j,k} = \gamma(|j - k|)$, where γ is the covariance function given in the definition of fractional Gaussian noise (fGn), by

$$\Gamma = \begin{pmatrix} \gamma(0) & \gamma(1) & \dots & \gamma(N-1) \\ \gamma(1) & \gamma(0) & \dots & \gamma(N-2) \\ \vdots & \vdots & \ddots & \vdots \\ \gamma(N-1) & \gamma(N-2) & \dots & \gamma(0) \end{pmatrix}$$

The technique to simulate fGn with FFT is called the Circulant Embedding Method (CEM), generalized by Davies and Harte [28], and consists of embedding this covariance matrix into a bigger M -by- M (with $M = 2N$) circulant covariance matrix C such as

$$C = \begin{pmatrix} \gamma(0) & \gamma(1) & \dots & \gamma(N-1) & 0 & \gamma(N-1) & \gamma(N-2) & \dots & \gamma(2) & \gamma(1) \\ \gamma(1) & \gamma(0) & \dots & \gamma(N-2) & \gamma(N-1) & 0 & \gamma(N-1) & \dots & \gamma(3) & \gamma(2) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \gamma(N-1) & \gamma(N-2) & \dots & \gamma(0) & \gamma(1) & \gamma(2) & \gamma(3) & \dots & \gamma(N-1) & 0 \\ 0 & \gamma(N-1) & \dots & \gamma(1) & \gamma(0) & \gamma(1) & \gamma(2) & \dots & \gamma(N-2) & \gamma(N-1) \\ \gamma(N-1) & 0 & \dots & \gamma(2) & \gamma(1) & \gamma(0) & \gamma(1) & \dots & \gamma(N-3) & \gamma(N-2) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \gamma(1) & \gamma(2) & \dots & 0 & \gamma(N-1) & \gamma(N-2) & \gamma(N-3) & \dots & \gamma(1) & \gamma(0) \end{pmatrix}$$

where the covariance matrix is embedded on the top left hand corner. It is sufficient to point out that

$$C_{0,k} = \begin{cases} \gamma(k), & k=0, \dots, N-1 \\ \gamma(2N-k), & k=N+2, \dots, 2N-1 \end{cases}$$

Remark: As Perrin et al. [30] have pointed out, the size M can be $M \geq 2(N-1)$, and the case $M=2(N-1)$ is minimal embedding. For any other choice of M , the choice of $C_{0,N}, \dots, C_{0,M-N+1}$ is arbitrary and can be conveniently chosen as long as the symmetry of the matrix is upheld; more zeros can be padded if M is bigger to make C circulant. For the rest of the chapter, we will concern ourselves with the case $M=2N$.

From Theorem 3.2, we know that, given any circulant matrix, it can be decomposed as $C=Q\Lambda Q^*$, where

$$(Q)_{j,k} = \frac{1}{\sqrt{2N}} \exp\left(-2\pi i \frac{jk}{2N}\right), \text{ for } j, k=0, \dots, 2N-1 \tag{25}$$

The matrix Λ is the diagonal matrix with eigenvalues

$$\lambda_k = \sum_{j=0}^{2N-1} c_{0,j} \exp\left(2\pi i \frac{jk}{2N}\right), \text{ for } j, k=0, \dots, 2N-1 \tag{26}$$

This differs slightly from the previous definition, but similar to the continuous counterpart; the sign of the exponential power in the Fourier transform is just conventional difference. The approach is identical as long as the duality is maintained. That is, if written in the form of $C=Q\Lambda Q^*$, the sign of the exponential power of the component in Q and Λ should be opposite. In the case of the previous theorem where $C=F^{-1}\Lambda F$, it is easy to check that F^{-1} and $\Lambda(\xi)$ indeed have the opposite sign in the exponential power.

It should be noted that C is not guaranteed to be positive-definite. Davies and Harte [28] suggest setting zero every negative value that may appear in Λ . In Perrin et al. [30], they prove that the circulant covariance matrix for fGn is always non-negative definite, so the approach is feasible without any modification. The reader is referred to Dietrich and Newsam [29] and Wood and Chan [31] for more detail on dealing with this issue.

Assuming that C is positive definite and symmetric, the eigenvalues are positive and real. The ‘square root’ of C is readily formed, $S = Q\Lambda^{1/2}Q^*$, where $\Lambda^{1/2}$ is the diagonal matrix with eigenvalues $1, \sqrt{\lambda_1}, \dots, \sqrt{\lambda_{2N-1}}$. It is easy to check that $SS^* = SS' = C$. So, S has the desired properties we look for.

Theorem 3.3 (Simulation of fGn with FFT) The simulation of the sample path of fGn, we are going to simulate $y = SV$, consists of the following steps:

1. Compute the eigenvalues $\{\lambda_k\}_{k=0, \dots, 2N-1}$ from equation (26) by means of FFT. This will reduce the computational time from $O(N^2)$ to $O(N \log N)$.
2. Calculate $W = Q^*V$.
3. Generate two standard normal random variables $W_0 = V_0^{(1)}$ and $W_N = V_N^{(1)}$
4. For $1 \leq j < N$, generate two standard normal random variables $V_j^{(1)}$ and $V_j^{(2)}$ and let

$$W_j = \frac{1}{\sqrt{2}}(V_j^{(1)} + iV_j^{(2)})$$

$$W_{2N-j} = \frac{1}{\sqrt{2}}(V_j^{(1)} - iV_j^{(2)})$$

1. Compute $Z = Q\Lambda^{1/2}W$. This can be seen as another Fourier transform of the vector $\Lambda^{1/2}W$:
2. $Z_k = \frac{1}{\sqrt{2N}} \sum_{j=0}^{2N-1} \sqrt{\lambda_j} W_j \exp\left(-2\pi i \frac{jk}{2N}\right)$
3. It is identical to carry out FFT on the following sequence:

$$w_j = \begin{cases} \sqrt{\frac{\lambda_0}{2N}} V_0^{(1)} & j = 0; \\ \sqrt{\frac{\lambda_j}{4N}} (V_j^{(1)} + iV_j^{(2)}) & j = 1, \dots, N-1; \\ \sqrt{\frac{\lambda_N}{2N}} V_N^{(1)} & j = N; \\ \sqrt{\frac{\lambda_j}{4N}} (V_{2N-j}^{(1)} - iV_{2N-j}^{(2)}) & j = N+1, \dots, 2N-1; \end{cases}$$

Due to the symmetric nature of the sequence, the Fourier sum of $\{w_j\} = \{Z_k\}_{k=0}^{2N-1}$ is real. The first N samples have the desired covariance structure. But, since the 2nd half of samples ($N \dots 2N-1$) are not independent of the first N samples, this sample cannot be used.

1. Recover fBm from the recursive relationship:
2. $B^H(0) = 0; B^H(i) = B^H(i-1) + Z_{i-1}$ for $1 \leq i \leq N$

4. Approximate methods

As we have seen in the previous section, the exact methods all take on the covariance structure matrix as starting point and try to reproduce the covariance structure by different decomposition. That can be time and resource consuming, so rather it is preferable to have approximation of the fractional Brownian motion that permits robust simulation.

In this section, we will start with the Mandelbrot representation due to historical reason and move onto several other methods that provide us with better understanding of the process and increasing robustness.

4.1. Mandelbrot representation

Recalling from Section 2.3, fractional Brownian motion permits a stochastic integral representation. To approximate equation (5), it is natural to truncate the lower limit from negative infinity to some point, say at $-b$:

$$\tilde{B}^H(n) = C_H \left(\sum_{k=-b}^0 [(n-k)^{H-\frac{1}{2}} - (-k)^{H-\frac{1}{2}}] B_1(k) + \sum_{k=0}^n (n-k)^{H-\frac{1}{2}} B_2(k) \right) \quad (27)$$

Note that the C_H is not the same constant term as in equation (5), because one has to recalculate the normalizing factor due to the truncation. As pointed out in [25], the recommended choice of b is $N^{3/2}$. Even though this is a straightforward way to generate fractional Brownian motion, it is rather inefficient. It is included in this section for the sake of completeness.

4.2. Euler hypergeometric integral

fBm permits the stochastic integral form involving the Euler hypergeometric integral:

$$B^H(t) = \int_0^t K_H(t, s) dB(s) \quad (28)$$

where $B(s)$ is the standard Brownian motion and

$$K_H(t, s) = \frac{(t-s)^{H-\frac{1}{2}}}{\Gamma(H+\frac{1}{2})} F_{2,1} \left(H - \frac{1}{2}; \frac{1}{2} - H; H + \frac{1}{2}; 1 - \frac{t}{s} \right) \quad (29)$$

is the hypergeometric function, which can be readily computed by most mathematical packages. By discretizing (28), at each time index t_j , we have

$$B^H(t_j) = \frac{1}{T} \sum_{i=0}^{j-1} \left(\int_{t_i}^{t_{i+1}} K_H(t_j, s) ds \right) \delta B_i \quad (30)$$

where $\delta B_i = \sqrt{\frac{T}{n}} \Delta B_i$ and ΔB_i is drawn according to the standard normal distribution. This means that $\{\delta B_i\}_{i=1 \dots n}$ is the increments of a scaled Brownian motion on $[0, T]$ with quadratic variation T . The inner integral can be computed by the Gaussian quadrature efficiently.

4.3. Construction by correlated random walk

This particular algorithm proposed in [32] of constructing fBm relies on the process of correlated random walks and summation over generated paths. This is similar to the generation of ordinary Brownian method through summation of the sample paths of normal random walk, which is related to the central limit theorem.

Definition 4.1 (Correlated Random Walk) For any $p \in [0,1]$, denote X_n^p as the correlated random walk with persistence index p . It consists of a jump on each time-step with jump size of either +1 or -1 such that:

- $X_0^p = 0, P(X_1^p = -1) = \frac{1}{2}, P(X_1^p = +1) = \frac{1}{2}$
- $\forall n \geq 1, \epsilon_n^p \equiv X_n^p - X_{n-1}^p$ which equals either +1 or -1
- $\forall n \geq 1, P(\epsilon_{n+1}^p = \epsilon_n^p \mid \sigma(X_k^p, 0 \leq k \leq n)) = p$

Theorem 4.1 For any $m \geq 1, n \geq 0$, we have

$$\mathbb{E}[\epsilon_m^p \epsilon_{m+n}^p] = (2p - 1)^n \tag{31}$$

In order to add additional randomness into the correlated random walks, we replace the constant persistence index p with a random variable μ , and we denote the resulting correlated random walk as X_n^μ . Or, to put it more formally, denote by P^p the law of X^p for a given persistence index p . Now, consider a probability measure μ on $[0,1]$, which we call the corresponding probability law P^μ , the annealed law of the correlated walk associated to μ . Note that $P^\mu := \int_0^1 P^p d\mu(p)$.

Proposition 4.1 For all $m \geq 1, n \geq 0$, we have

$$\mathbb{E}[\epsilon_m^\mu \epsilon_{m+n}^\mu] = \int_0^1 (2p - 1)^n d\mu(p) \tag{32}$$

The next result is due to Enriquez [32]. The proof is based on Lemma 5.1 of Taqqu [33].

Theorem 4.2 For $\frac{1}{2} < H < 1$, denote by μ^H the probability on $[\frac{1}{2}, 1]$ with density $(1 - H)2^{3-2H}(1 - p)^{1-2H}$. Let $(X^{\mu^H, i})_{i \geq 1}$ be a sequence of independent processes with probability law P^{μ^H} . Then,

$$\mathbb{L}^D \lim_{N \rightarrow \infty} \mathbb{L} \lim_{N \rightarrow \infty} c_H \frac{X^{\mu^H, 1} + \dots + X^{\mu^H, M}}{N^H \sqrt{M}} = B_H(t) \tag{33}$$

where $c_H = \sqrt{\frac{H(2H - 1)}{\Gamma(3 - 2H)}}$, \mathbb{L} stands for the convergence in law, and \mathbb{L}^D means the convergence in the sense of weak convergence in the Skorohod topology on $D[0,1]$, the space of cadlag functions on $[0,1]$. Here, $[\cdot]$ is the floor function and rounds the argument to the closest integer, M is the number of trajectories of correlated random walks and N is number of time steps.

Remark: For $H=1/2$, there is a similar expression (see [32]). The order of limit in equation (33) is important, because if reversed, the sum would result in 0. Theorem 4.2 is mostly for theoretical construction.

In [32], the above theorem is further simplified from double summations into a single summation by applying Berry-Essen bound: As long as $M(N)$ is of order $O(N^{2-2H})$,

$$\mathbb{L}^D \lim_{N \rightarrow \infty} c_H \frac{X^{\mu^H, 1} + \dots + X^{\mu^H, M(N)}}{N^H \sqrt{M(N)}} = B_H(t) \tag{34}$$

In practice, any probability measure with moment equivalent to $\frac{1}{n^{2-2H} L(n)}$, where L is a slowly varying function, will be used. This could be shown by Karamata's theorem, for which further elaboration is found in [33]. In [32], three families of equivalent measures are provided, and specifically the 2nd family of the measures $(\mu'_{H,k})_{k>0}$ is most appropriate for

simulation purpose: For $H > \frac{1}{2}$, $\mu'_{H,k}$ has the density of $1 - \frac{(1 - U^{\frac{1}{k}})^{\frac{1}{2-2H}}}{2}$ with the corresponding normalizing factor $c'_{H,k} = \frac{c_H}{\sqrt{k}}$. U is a standard uniform random variable. The error given by the Berry-Essen bound for $H > \frac{1}{2}$ is given by

$$0.65 \times D_H N^{1-H} / \sqrt{kM} \tag{35}$$

where $D_H = \sqrt{\frac{6(2H - 1)}{(H + 1)(2H + 1)}}$. Here, k serves as a control variable of order $k(N) = o(N)$, and the error term contains $\frac{1}{\sqrt{k}}$ which can be seen as a way to restrict error with the price of distur-

tion of the covariance relation in X_N , though it is advisable to keep $k \leq 1$. For more discussion on the choice of k , we refer to Section 4 of [32].

Theorem 4.3 (Simulation of fBm with correlated random walk) Simulating fBm with correlated random walk for the case of $H > \frac{1}{2}$ consists of the following steps:

1. Calculate $M(N)$ by the tolerable error level from equation (35). Calculate $\lfloor NT \rfloor$, where $\lfloor \cdot \rfloor$ is the floor function, and create time-index $\{t_j\} : \{1, 2, \dots, \lfloor NT \rfloor\}$.
2. Simulate M independent copies of $\{\mu_{H,k}^j\}_{j=1 \dots M} = 1 - \frac{(1 - U^{\frac{1}{k}})^{\frac{1}{2-2H}}}{2}$ for M trajectories.
3. Simulate M copies of correlated random walks:
 - If $t_j = 1$, $\epsilon_1^j = 2 * \text{Bernoulli}(\frac{1}{2}) - 1$, $X_1^j = \epsilon_1^j$
 - If $t_j > 1$, $\epsilon_{t_j}^j = \epsilon_{t_{j-1}}^j * (2 * \text{Bernoulli}(\mu_{H,k}^j) - 1)$, $X_{t_j}^j = X_{t_{j-1}}^j + \epsilon_{t_j}^j$
4. At each t_j , calculate
 - $$B^H(t_j) = c_H \cdot \frac{X_{\lfloor N t_j \rfloor}^{\mu_{H,k}^1} + \dots + X_{\lfloor N t_j \rfloor}^{\mu_{H,k}^M}}{N^H \sqrt{M}}$$

Remark: For any given time-horizon T , it is easier to simulate the path of $B^H(1)$ with given resolution N and scale it to arrive at $B^H(T) = T^H B^H(1)$.

This algorithm is interesting from a theoretical point of view, since it gives us a construction of fractional Brownian motion reflecting its ordinary Brownian motion counterpart with the help of central limit theorem. But, in practice, it might not be fast enough for simulation purpose that requires large number of simulated paths.

4.4. Conditional random midpoint displacement

This algorithm is put forth by Norros et al. [34] and uses the similar approach to compute the conditional distribution of the fractional Gaussian noises as we have seen in the Hosking algorithm in Section 3.1. The difference is that this algorithm does not completely capture the covariance of all the sample points, instead it chooses certain number of points of the generated samples and uses a different ordering to do conditioning on (recall that, in the Hosking method, the conditioning is done by chronological order).

Again we are interested in the stationary fractional Gaussian noises and will back out the fractional Brownian motion on the time interval $[0, 1]$, which can be scaled back according to self-similarity relationship. We will first outline the idea of bisection method, which will be expanded into the conditional mid-point replacement scheme later on.

4.4.1. Bisection scheme and basic algorithm

In this section we adopt the notation in Norros et al. [34]: $Z(t)$ is the fractional Brownian motion, and $X_{i,j}$ is the fractional Gaussian noise of a certain interval j in a given level i .

The idea is to simulate $Z(t)$ on the interval $[0,1]$. First, given $Z(0)=0$ and $Z(1)$ with the standard normal distribution of $N(0,1)$, we compute the conditional distribution of $\{Z(\frac{1}{2}) \mid Z(0), Z(1)\} \sim N(\frac{1}{2}Z(1), 2^{-2H} - \frac{1}{4})$. The bisection involves the indices i and j , where i indicates the 'level' and j for the 'position'. Let

$$X_{i,j} = Z(j \cdot 2^{-i}) - Z((j-1) \cdot 2^{-i}), \text{ for } i=0,1, 2, \dots, j=1, \dots, 2^i$$

It is easy to see that, at any given level i , the interval $[0,1]$ is divided into 2^i sub-intervals.

If we denote $(i-1)$ th level as the 'mother-level', it will be divided twice finer in the next level. So given any interval on the mother level, it is easy to observe the relationship

$$X_{i,2j-1} + X_{i,2j} = X_{i-1,j} \tag{36}$$

Because of equation (36), it is enough to just generate $X_{i,j}$ for odd number j . So, let us proceed from left to right, assuming that the sample points $X_{i,1}, \dots, X_{i,2k}$ have already been generated ($k \in \{0,1, \dots, 2^{i-1} - 1\}$). For the point $X_{i,2k+1}$, we have

$$X_{i,2k+1} = e(i, k) [X_{i, \max(2k-m+1, 1)}, \dots, X_{i, 2k}, X_{i-1, k+1}, \dots, X_{i-1, \min(k+n, 2^{i-1})}] + \sqrt{v(i, k)} U_{i,k} \tag{37}$$

where $U_{i,k}$ are i.i.d. standard Gaussian random variables $i=0,1, \dots; k=0,1, \dots, 2^{(i-1)} - 1$. Equation (37) can be rewritten as

$$X_{i,2k+1} = e(i, k) [X_{i, \max(2k-m+1, 1)}, \dots, X_{i, 2k}, X_{i-1, k+1}, \dots, X_{i-1, \min(k+n, 2^{i-1})}] + \sqrt{v(i, k)} U_{i,k} = \mathbb{E}[X_{i,2k+1} \mid X_{i, \max(2k-m+1, 1)}, \dots, X_{i, 2k}, X_{i-1, k+1}, \dots, X_{i-1, \min(k+n, 2^{i-1})}] \tag{38}$$

$$v(i, k) = \text{Var}[X_{i,2k+1} \mid X_{i, \max(2k-m+1, 1)}, \dots, X_{i, 2k}, X_{i-1, k+1}, \dots, X_{i-1, \min(k+n, 2^{i-1})}] \tag{39}$$

As mentioned before, this scheme conditions on a fixed number of past samples instead of the whole past, where the two integers ($m \geq 0, n \geq 1$) indicate the number of the intervals the expectation and variance are conditioned on, and is called RMD(m, n). Looking at (38) and (39) " $X_{i, \max(2k-m+1, 1)}, \dots, X_{i, 2k}$ " indicates that there are at most m neighboring increments to the left of the interval in question ($X_{i,2k+1}$), and " $X_{i-1, k+1}, \dots, X_{i-1, \min(k+n, 2^{i-1})}$ " indicates that there are at most n neighboring increments to the right of the interval.

Denote by Γ_{ik} the covariance matrix with $X_{i,2k+1}$ as the first entry, and $X_{i, \max(2k-m+1, 1)}, \dots, X_{i, 2k}, X_{i-1, k+1}, \dots, X_{i-1, \min(k+n, 2^{i-1})}$ as the rest of the entries. Then, we have

$$\Gamma_{ik} = \text{Cov}([X_{i,2k+1}, X_{i,\max(2k-m+1)}, \dots, X_{i,2k}, X_{i-1,k+1}, \dots, X_{i-1,\min(k+n,2^i)}])$$

where

$$\text{Cov}([x_1, x_2, \dots, x_n]) = \begin{pmatrix} \text{Cov}(x_1, x_1) & \text{Cov}(x_1, x_2) & \dots & \text{Cov}(x_1, x_n) \\ \text{Cov}(x_2, x_1) & \text{Cov}(x_2, x_2) & \dots & \text{Cov}(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ \text{Cov}(x_n, x_1) & \text{Cov}(x_n, x_2) & \dots & \text{Cov}(x_n, x_n) \end{pmatrix}$$

Similar to (14)-(16), we can partition Γ_{ik} as

$$\Gamma_{ik} = \begin{pmatrix} \text{Var}(X_{i,2k+1}) & \Gamma_{ik}^{(1,2)} \\ \Gamma_{ik}^{(2,1)} & \Gamma_{ik}^{(2,2)} \end{pmatrix}$$

Note that $\Gamma_{ik}^{(1,2)} = (\Gamma_{ik}^{(2,1)})'$. Hence, we have

$$e(i, k) = \Gamma_{ik}^{(1,2)} (\Gamma_{ik}^{(2,2)})^{-1} \tag{40}$$

$$v(i, k) = \frac{|\Gamma_{ik}^{(1,2)}|}{|\Gamma_{ik}^{(2,2)}|} \tag{41}$$

By the stationarity of the increment of Z and by self-similarity, $e(i, k)$ is independent of i and k when $2k \geq m$ and $k \leq 2^{i-1} - n$ (meaning that the sequence is not truncated by $\max(\cdot)$ and $\min(\cdot)$), and it only depends on i only when $2^i < m + 2n$.

4.4.2. On-the-fly RMD(m,n) generation

Norros et al. [34] further propose that, instead of having the previous level ($i-1$) completely generated first, partition and conditioning can be done ‘on-the-fly’, meaning that we can have multiple unfinished levels going at the same time. Unlike the previous RMD(m,n) scheme, the level here is defined differently.

First define the ‘resolution’ by δ , as the smallest interval that we will be dealing with in this scheme. Note that this is different from the previous sub-section where at the i -th level $\delta = 2^{-i}$, which can be bisected finer indefinitely. In the on-the-fly RMD scheme, the minimum interval length is defined as δ .

At each level i , the interval $[0, 2^i \delta]$ is split finely into interval with length of δ and $X_{i,k}$ samples are generated on each point until all points within the interval are filled. Then, the trace is expanded to the next level $i + 1$, the interval $[0, 2^{i+1} \delta]$, and this procedure can be considered as ‘enlargement’. So, instead of having a pre-determined time-horizon and zooming in with twice-finer resolution in the original RMD scheme, the new RMD scheme has a pre-determined resolution and expand twice-fold the horizon at each level.

Within the same interval $[0, 2^i \delta]$, the following rules are applied for the inner intervals:

1. Must have $n-1$ (or all) right-neighboring intervals (of which, have the same length as the mother-interval).
2. Also, there should be m (or all) left-neighboring intervals (of which, have the same length as the interval being considered).
3. If there are intervals that satisfy both the conditions above, choose the one as left as possible.
4. When all intervals are filled out, expand to the next level by ‘enlargement’.

Here we use

$$Y_{i,j} = Z(j \cdot 2^i \delta) - Z((j-1) \cdot 2^i \delta), \quad i=0,1, \dots, j=1,2, \dots \quad (42)$$

instead of $X_{i,j}$ to avoid confusion with the original RMD scheme notation.

Similar to the ordinary RMD, we have the following equations from the conditional distribution of multivariate Gaussian processes: The enlargement stage is defined by

$$Y_{i,1} = \mathbb{E}[Y_{i,1} | Y_{i-1,1}] + \sqrt{\text{Var}[Y_{i,1} | Y_{i-1,1}]} U_{i,1} = e(i, 1) Y_{i-1,1} + \sqrt{v(i, 1)} U_{i,1} \quad (43)$$

Inner interval points are constructed similar to the ordinary RMD scheme (the right-neighboring intervals are of level $i+1$ instead of $i-1$) as

$$Y_{i,2k+1} = e(i, k) [Y_{i, \max(2k-m+1, 1)}, \dots, Y_{i, 2k}, Y_{i+1, k+1}, \dots, Y_{i+1, \min(k+n, N_{i+1})}]' + \sqrt{v(i, k)} U_{i,k} \quad (44)$$

where N_{i+1} is the last generated increment on the $(i+1)$ -th level. The order of splitting for the on-the-fly scheme is given in Figure 1, where its ordinary RMD counterpart’s splitting order is also given.

Norros et al. [34] have done an extensive comparison between on-the-fly RMD schemes in terms of accuracy and robustness compared to the FFT and aggregate methods. On-the-fly RMD and FFT are significantly faster than the aggregate method, and on-the-fly RMD can generate samples with no fixed time-horizon, while for FFT the whole trace has to be generated before it can be used. So, RMD seems superior in terms of flexibility.

4.5. Spectral method

In this subsection, we will look into the spectral method of approximating the fractional Gaussian noises, which has the origin from spectral analysis in physics: A time-domain can be transformed into a frequency-domain without loss of information through Fourier transform. With the typical Fourier-time series, the original input is deterministic and transformed into the spectral density that represents the magnitude of different frequencies in the frequency domain. It is possible to extend this approach to analyzing stochastic processes.

Though it is impossible to study all realization, it is possible to analyze in a probabilistic/distribution sense by observing the expected frequency information contained in the autocovariance function.

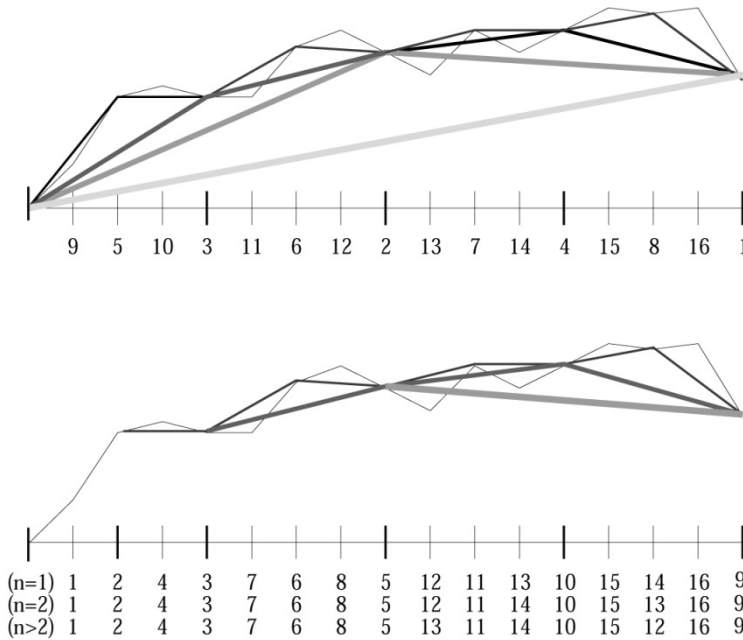


Figure 1. The order of splits by the ordinary RMD(m,n) scheme (top) and the on-the-fly RMD(m,n) scheme (bottom). Note that, for the on-the-fly scheme, the order changes according to the choice of n.

Spectral density is computed for frequencies, $-\pi \leq \lambda \leq \pi$, as

$$f(\lambda) = \sum_{j=-\infty}^{\infty} \gamma(j) \exp(ij\lambda) \tag{45}$$

The $\gamma(\cdot)$ here is the autocovariance function, which can be recovered by the inverse formula

$$\gamma(j) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(\lambda) \exp(-ij\lambda) \tag{46}$$

The spectral density of the fGn can be approximated according to [25] and [35] as

$$f(\lambda) = 2 \sin(\pi H) \Gamma(2H + 1) (1 - \cos \lambda) [|\lambda|^{-2H-1} + B(\lambda, H)] \tag{47}$$

where

$$B(\lambda, H) = \sum_{j=1}^{\infty} \{(2\pi j + \lambda)^{-2H-1} + (2\pi j - \lambda)^{-2H-1}\}$$

Note that the domain is only $-\pi \leq \lambda \leq \pi$, since any frequency higher would only correspond to amplitude between our desired sample points.

The problem with the above expression is that there is no known form for $B(\lambda, H)$, Paxson [35] proposes the following scheme for $B(\lambda, H)$:

$$B(\lambda, H) \cong \tilde{B}_3(\lambda, H) = a_1^d + b_1^d + a_2^d + b_2^d + a_3^d + b_3^d + \frac{a_3^{d'} + b_3^{d'} + a_4^{d'} + b_4^{d'}}{8H\pi}$$

where

$$d = -2H - 1, d' = -2H, a_k = 2k\pi + \lambda, b_k = 2k\pi - \lambda$$

Moreover, with the help of the Whittle estimator, Paxson [35] shows that

$$\tilde{B}_3(\lambda, H)'' = [1.0002 - 0.000134\lambda] (\tilde{B}_3(\lambda, H) - 2^{-7.65H-7.4})$$

gives a very robust and unbiased approximation for the $B(\lambda, H)$. See Appendix A of [35] for a detailed comparison and justification of this approximation.

With the approximation scheme for the spectral density at hand, we can now look at the spectral analysis of a stationary discrete-time Gaussian process (fractional Gaussian noise; fGn) $X = \{X_n : n=0, \dots, N-1\}$, which can be represented in terms of the spectral density $f(\lambda)$ as

$$X_n = \int_0^\pi \sqrt{\frac{f(\lambda)}{\pi}} \cos(n\lambda) d B_1(\lambda) - \int_0^\pi \sqrt{\frac{f(\lambda)}{\pi}} \sin(n\lambda) d B_2(\lambda)$$

where B_1 and B_2 are independent standard Brownian motions and the equality is understood in terms of distribution. Define $\xi_n(\lambda) = \sqrt{\frac{f(\lambda)}{\pi}} \cos(n\lambda)$ and fix some integer l . After setting $t_k = \frac{\pi k}{l}$ for $k=0, \dots, l-1$, we can approximate it by a simple function $\xi_n^{(l)}$ defined on $[0, \pi]$ for $0 \leq n \leq N-1$ by

$$\xi_n^{(l)}(\lambda) = \sqrt{\frac{f(t_1)}{\pi}} \cos(nt_1) 1_{(0)}(\lambda) + \sum_{k=0}^{l-1} \sqrt{\frac{f(t_{k+1})}{\pi}} \cos(nt_{k+1}) 1_{(t_k, t_{k+1})}(\lambda) \tag{48}$$

which is similar to the typical construction of stochastic integral.

Define the sine counterpart as $\theta_n^{(l)}(\lambda)$, and then integrate both $\xi_n^{(l)}(\lambda)$ and $\theta_n^{(l)}(\lambda)$ with respect to $d B_1(\lambda)$ and $d B_2(\lambda)$ on $[0, \pi]$ to approximate X_n . Then, we have

$$\hat{X}_n^{(l)} = \sum_{k=0}^{l-1} \sqrt{\frac{f(t_{k+1})}{l}} [\cos(nt_{k+1}) U_k^{(0)} - \sin(nt_{k+1}) U_k^{(1)}] \tag{49}$$

where $U_k^{(\cdot)}$ are i.i.d. standard normal random variables. $U_k^{(0)}$ and $U_k^{(1)}$ are independent, as they are resulted from integration from the two aforementioned independent Brownian motions.

Similar to the Fourier transform approach, the fGns can be recovered by applying the FFT to the sequence of $\hat{X}_n^{(l)}$ efficiently to the following coefficient:

$$a_k = \begin{cases} 0, & k=0 \\ \frac{1}{2}(U_{k-1}^{(0)} + iU_{k-1}^{(1)}), & k=1, \dots, l-1 \\ U_{k-1}^{(0)} \sqrt{\frac{f(t_k)}{l}}, & k=l \\ \frac{1}{2}(U_{2l-k-1}^{(0)} + iU_{2l-k-1}^{(1)}), & k=l+1, \dots, 2l-1 \end{cases} \tag{50}$$

It is easy to check that the covariance structure of fGns can be recovered, with the help of product-to-sum trigonometric identity, as

$$\begin{aligned} Cov(\hat{X}_m^{(l)}, \hat{X}_n^{(l)}) &= \sum_{k=0}^{l-1} \frac{f(t_{k+1})}{l} \cos((m-n)t_{k+1}) \\ &\cong 2 \int_0^\pi \frac{f(\lambda)}{2\pi} \cos(n\lambda) d\lambda \\ &= \frac{1}{2\pi} \int_{-\pi}^\pi f(\lambda) \exp(-in\lambda) d\lambda = \gamma(n) \end{aligned}$$

Paxson [35] has also proposed another method for simulating fGns, where in [25] it was proven to be related to (50) with the case $l = N / 2$:

$$b_k = \begin{cases} 0, & k=0 \\ \sqrt{\frac{R_k f(t_k)}{N}} \exp(i\Phi_k), & k=1, \dots, N/2-1 \\ \sqrt{\frac{f(t_{N/2})}{2N}} U_{N/2}^{(0)}, & k=N/2 \\ b_k^*, & k=N/2+1, \dots, N-1 \end{cases} \tag{51}$$

Here, R_k is a vector of exponentially distributed random variables with mean 1, and Φ_k are uniformly distributed random variables on $[0, 2\pi]$ independent of R_k . This method is of order $N \log(N)$, and only one FFT is required instead of 2 times compared to the Davis-Harte FFT method. Hence, it is about 4 times faster.

Remark: The Paxson algorithm in (54) is improved by Dieker [25] to retain the normality of the sequence and its relationship with the original spectral representation.

5. A numerical example: fBm volatility model

Finally, this section provides a numerical example of Monte Carlo simulation of fBm volatility model. In Section 2.3, we have briefly mentioned the truncated fractional Brownian motion. This section outlines the stochastic volatility model explored by Comte and Renault [6]. We follow the example given in [6], with the following setup to simulate the volatility process:

$$\begin{cases} \sigma(t) = \sigma_0 e^{x(t)} \\ dx(t) = -kx(t)dt + \nu d\hat{B}^H(t) \end{cases} \tag{52}$$

where ν is the volatility factor of the log-volatility process $x(t)$. The volatility process is the exponential of an OU-process driven by the truncated fBm. Also, we assume that $x(0)=0, k>0, \frac{1}{2}<H<1$.

Solving the OU-process with integrating factor, we have

$$x(t) = \int_0^t \nu e^{-k(t-s)} d\hat{B}^H(s) \tag{53}$$

By applying the fractional calculus or using the formulas provided in [5], we can formulate $x(t)$ in another way as

$$x(t) = \int_0^t a(t-s) dB(s) \tag{54}$$

where $B(t)$ is an ordinary standard Brownian motion and

$$a(\theta) = \frac{\nu}{\Gamma(H + \frac{1}{2})} \frac{d}{dx} \int_0^\theta e^{-ku} (\theta - u)^{H-\frac{1}{2}} du = \frac{\nu}{\Gamma(H + \frac{1}{2})} (\theta^\alpha - k e^{-\theta} \int_0^\theta e^{ku} u^\alpha du) \tag{55}$$

By applying the ordinary discretization scheme to (54), we have

$$\tilde{x}(t) = \sum_{j=1}^{t_N=t} a\left(t_N - \frac{t_j - 1}{n}\right) (B(t_j) - B(t_{j-1})) \tag{56}$$

Here, the coefficient $a(\cdot)$ can be calculated by symbolic packages such as matlab and mathematics. In our case of OU-process, it is a summation of constant with incomplete gamma function and gamma function.

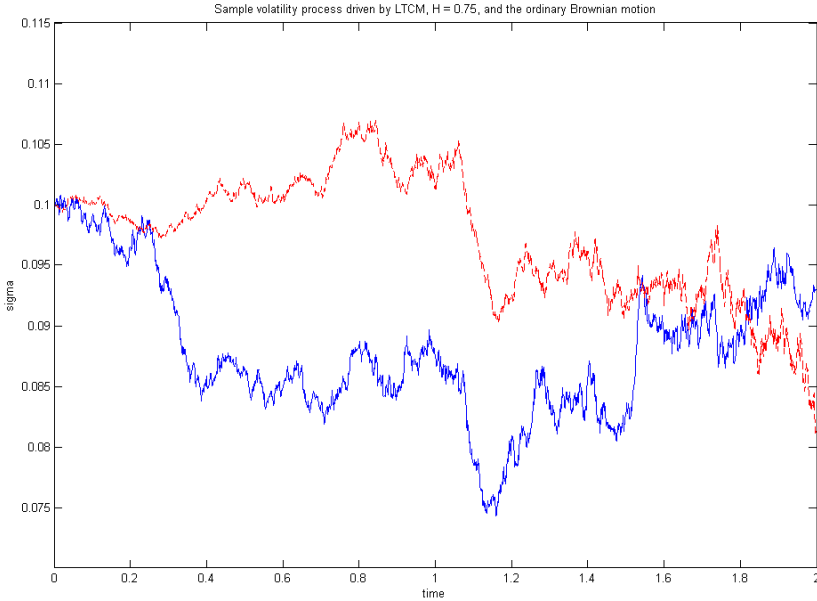


Figure 2. shows a sample path of $\sigma(t)=\exp(\bar{x}(t))$ for $k=1, \sigma_0=0.1, \nu=0.3, H=0.75, T=2$. For the purpose of comparison, a sample path of the volatility process driven by an ordinary OU-process (the dotted line): $\sigma(t)=\exp(y(t))$, where $y(t)=\int_0^t \nu e^{-k(t-s)} dB(s)$, is shown alongside, which has the same parameters except the Hurst index.

The sample path of the fractional-OU driven volatility process has shown more of a persistent trend, i.e. more prominent trend (more smooth and less reversal) compared to the ordinary-OU driven volatility process, which is what to be expected according [5]. This approach only imitate fractional Brownian motion if the time-step in discretization scheme is very small, renders it not robust enough for practical purpose. But for analytical purpose, it can be shown it is equivalent to the S-transform approach outlined in [36]. For more discussion of its statistical property and justification of its stability as compared to the original stationary version, we direct the reader to [5] and [6]. We provided this approach for the sake of completeness and readers' interest.

We have also included fractional Brownian motion simulated by the circulant-embedding method, with the same parameters as above in Figure 3. This approach is more robust since the covariance structure does not depends on the step-size. Figure 3 shows a sample path of $\sigma(t)=\exp(x(t))$, where $x(t)$ is the fBM generated by the circulant-embedding FFT with the same parameters as Figure 2: $k=1, \sigma_0=0.1, \nu=0.3, H=0.75, T=2$. In these two examples, the fractional Brownian motions are scaled, so that the variance of $B^H(T)$ equals to the ordinary Brownian motion $B^{1/2}(T)$.

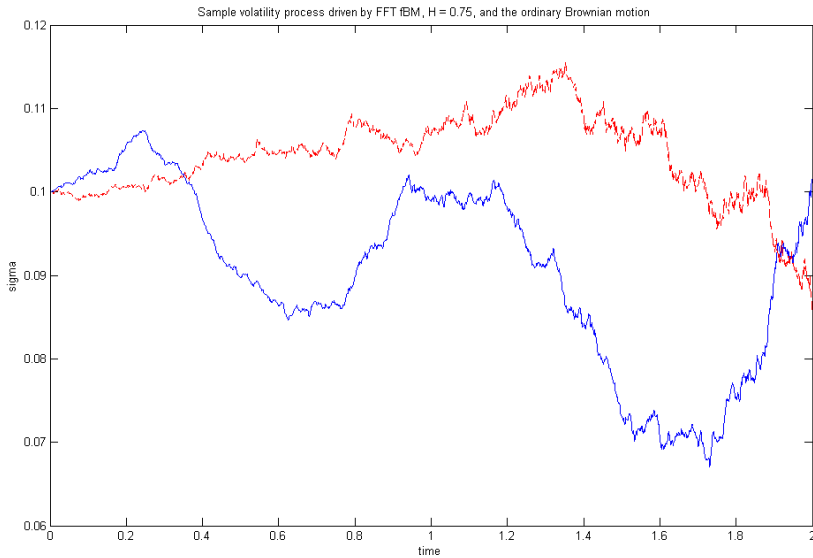


Figure 3.

6. Conclusion

Motivated by the inadequacy in capturing long-dependence feature in volatility process of the traditional stochastic volatility framework, we explore the possibility of fractional Brownian motion (fBm) in financial modeling and various schemes of the Monte Carlo simulation. Starting from the general definition, fBm can be considered as an extension of the ordinary Brownian motion with an autocovariance function that depends on both time indices instead of just the minimum between the two.

With different values of Hurst index, we can distinguish fractional Brownian motion into three different cases: $H < \frac{1}{2}$, $H = \frac{1}{2}$ and $H > \frac{1}{2}$. Since only the case of $H > \frac{1}{2}$ displays a long-dependence behavior, that is the case we are interested in. Several prominent examples of fBm in financial modeling are given.

Simulation schemes are divided into the exact schemes and approximate schemes. While the former will capture the complete structure for the whole length of sample size, the latter either approximates the value of the real realization or truncates the covariance structure for robustness.

We start with the exact scheme of Hosking algorithm that utilizes the multivariate Gaussian distribution of fractional Gaussian noises and simulates the sample points conditioned on

the previous samples. Alternatively, instead of simulating each conditioned on the past sample points, we can first construct the covariance matrix of the size of the sample we want, and proceed to find the ‘square root’ of the covariance matrix and multiply with a standard normal variable vector, for which the product vector will be the fractional Gaussian noise (fGn) with exact covariance structure as the covariance matrix. To find the ‘square root’, we first investigate the Cholesky decomposition, but the computational and memory expense is too large to be feasible in practice. In contrast, fast Fourier transform (FFT) simulation embeds the original covariance matrix in a larger circulant matrix and simulates by diagonalizing the circulant matrix into the product of eigenvalue matrix and unitary matrix. The FFT method is significantly more robust than the previous schemes.

We then look into the approximate schemes; namely the construction of fBm with correlated random walks, which can be viewed as an extension of construction of Brownian motion with ordinary random walk. This method gives us interesting insight into the true working of fBm, especially the idea of long-range dependence. This approach is not only interesting and easy to implement, but also the error can be calculated explicitly. The drawback of this approach is that the speed slows down significantly with large sample points, and the trade-off is made based on the error function. The last simulation approach we look at is the conditional random midpoint displacement (RMD) scheme, which is mathematically similar to the Hosking scheme, but with fixed number of past sample points it conditions on. The on-the-fly version of RMD scheme can indefinitely generate sample points with given resolution. Finally, we include the spectral method for approximating fBm.

Comparing all the schemes and also referring the studies done in [34], we conclude that if the time-horizon is known beforehand, the FFT/spectral schemes would be the best scheme due to the high speed and accuracy. Alternately, if samples should be generated indefinitely, the on-the-fly conditioned RMD scheme seems to offer similar level of accuracy and speed as the FFT scheme.

At last, we provided numerical examples for the truncated version of fBM proposed in [5, 6], as well as the fBM generated by FFT for comparison in robustness.

Acknowledgements

We wish to express our gratitude to Professor Alex Novikov for sharing relevant research materials with us, and Professor Bernt Øksendal and Professor Enriquez Nathanaël for interesting discussions and remarks.

Author details

Masaaki Kijima and Chun Ming Tam

Graduate School of Social Sciences, Tokyo Metropolitan University, Japan

References

- [1] Kolmogorov, A.N. (1940), 'Wienersche Spiralen und einige andere interessante Kurven im Hilbertschen Raum', *C.R. (Doklady) Acad. URSS (N.S)*, 26, 115-118.
- [2] Mandelbrot, B.B. and Van Ness, J.W. (1968), 'Fractional Brownian Motions, Fractional Noises and Applications', *SIAM Review*, 10, 422-437.
- [3] Mandelbrot, B.B. (1965), 'Une classe de processus homothétiques a soi; Application a la loi climatologique de H. E. Hurst', *C.R. Acad. Sci. Paris*, 260, 3274-3277.
- [4] Lo A.W. (1991), 'Long-Term Memory in Stock Market Price', *Econometrica*, 59, 1279-1313.
- [5] Comte, F. and Renault, E. (1996), 'Long Memory Continuous Time Models', *J. Econometrics*, 73, 101-149.
- [6] Comte, F. and Renault, E. (1998), 'Long Memory in Continuous-Time Stochastic Volatility Models', *Mathematical Finance*, 8, 291-323.
- [7] Hull, J. and White, A. (1987), 'The Pricing of Options on Assets with Stochastic Volatilities', *Rev. Finan. Studies*, 3, 281-300.
- [8] Scott, L. (1987), 'Option Pricing when the Variance Changes Randomly: Estimation and an Application', *J. Financial and Quant. Anal.*, 22, 419-438.
- [9] Ding, Z., Granger, C.W.J. and Engle, R.F. (1993), 'A Long Memory Property of Stock Market Returns and a New Model', *J. Empirical Finance*, 1, 83-106.
- [10] Crato, N. and de Lima, P. (1994), 'Long-Range Dependence in the Conditional Variance of Stock Returns', *Economics Letters*, 45, 281-285.
- [11] Baillie, R.T., Bollerslev, T. and Mikkelsen, H.O. (1996), 'Fractionally Integrated Generalized Autoregressive Conditional Heteroskedasticity', *J. Econometrics*, 74, 3-30.
- [12] Bollerslev, T. and Mikkelsen, H.O. (1996), 'Modeling and Pricing Long Memory in Stock Market Volatility', *J. Econometrics*, 73, 151-184.
- [13] Jacquier, E., Polson, N.G. and Rossi, P.E. (1994), 'Bayesian Analysis of Stochastic Volatility Models', *J. Bus. and Econ. Statistics*, 12, 371-417.
- [14] Breidt, F.J., Crato, N. and de Lima, P. (1998), 'The Detection and Estimation of Long Memory in Stochastic Volatility', *J. Econometrics*, 83, 325-348.
- [15] Biagini, F., Hu, Y., Øksendal, B. and Zhang T. (2008), *Stochastic Calculus for Fractional Brownian Motion and Applications*, Springer.
- [16] Cont, R. (2007), 'Volatility Clustering in Financial Markets: Empirical Facts and Agent-Based Models', In: Kirman, A. and Teyssiere, G. (ed.), *Long Memory in Economics*, Springer, 289-310.

- [17] Marinucci, D. and Robinson, P.M. (1999), 'Alternative Forms of Fractional Brownian Motion', *J. Statistical Planning and Inference*, 80, 111-122.
- [18] Delbaen, F. and Schachermayer, W. (1994), 'A General Version of the Fundamental Theorem of Asset Pricing', *Math. Ann.*, 300, 463-520.
- [19] Rogers, L.C.G. (1997), 'Arbitrage with Fractional Brownian Motion', *Mathematical Finance*, 7, 95-105.
- [20] Cheridito, P. (2003), 'Arbitrage in Fractional Brownian Motion Models', *Finance and Stochastics*, 7, 533-553.
- [21] Comte, F., Coutin, L. and Renault, E. (2012), 'Affine Fractional Stochastic Volatility Models', *Annals of Finance*, 8, 337-378.
- [22] Duffie, D., Pan, R. and Singleton, K. (2000), 'Transform Analysis and Asset Pricing for Affine Jump-Diffusion', *Econometrica*, 68, 1343-1376.
- [23] Fukasawa, M. (2011), 'Asymptotic Analysis for Stochastic Volatility: Martingale Expansion', *Finance and Stochastics*, 15, 635-654.
- [24] Yoshida, N. (2001), 'Malliavin Calculus and Martingale Expansion', *Bull. Sci. Math.*, 125, 431-456.
- [25] Dieker, T. (2002), *Simulation of Fractional Brownian Motion*, Master Thesis, Vrije Universiteit, Amsterdam.
- [26] Hosking, J.R.M. (1984), 'Modeling Persistence in Hydrological Time Series Using Fractional Differencing', *Water Resources Research*, 20, 1898-1908.
- [27] Brockwell, P.J. and Davis, R.A. (1987), *Time Series: Theory and Methods*, Springer, New York.
- [28] Davies, R.B. and Harte, D.S. (1987), 'Tests for Hurst Effect', *Biometrika*, 74, 95-102.
- [29] Dietrich, C.R. and Newsam, G.N. (1997), 'Fast and Exact Simulation of Stationary Gaussian Processes through Circulant Embedding of the Covariance Matrix', *SIAM J. Sci. Computing*, 18, 1088-1107.
- [30] Perrin, E., Harba, R., Jennane, R. and Iribarren, I. (2002), 'Fast and Exact Synthesis for 1-D Fractional Brownian Motion and Fractional Gaussian Noises', *IEEE Signal Processing Letters*, 9, 382-384.
- [31] Wood, A.T.A. and Chan, G. (1994), 'Simulation of Stationary Gaussian Processes in $0,1d$ ', *J. Computational and Graphical Statistics*, 3, 409-432.
- [32] Enriquez, N. (2004), 'A Simple Construction of the Fractional Brownian Motion', *Stochastic Processes and Their Applications*, 109, 203-223.
- [33] Taqqu, M. (1975), 'Weak Convergence to Fractional Brownian Motion and to the Rosenblatt Process', *Z. Wahr. Verw. Gebiete*, 31, 287-302.

- [34] Norros, I., Mannersalo, P. and Wang, J.L. (1999), 'Simulation of Fractional Brownian Motion with Conditionalized Random Midpoint Displacement', *Advances in Performance Analysis*, 2, 77-101.
- [35] Paxson, V. (1997), 'Fast, Approximate Synthesis of Fractional Gaussian Noise for Generating Self-Similar Network Traffic', *Computer Communication Review*, 27, 5-18.
- [36] Bender, C. (2003), 'An S-Transform approach to integration with respect to a fractional Brownian motion', *Bernoulli*, 9(6), p. 955-983.

