Fractional Brownian motion

Fractional Brownian motion (fBM) is a natural generalization of standard Brownian motion. A continuous-time process X is said to be **Gaussian** if for any $t_1 < ... < t_n$, $X_{t_1}, ..., X_{t_n}$ has a **multivariate Normal distribution**, and a zero-mean Gaussian process is uniquely characterized by its covariance function $R(s,t) := \mathbb{E}(X_s X_t)$, since the joint density of a n-dimensional Gaussian random vector $\mathbf{X} = (X_1, ..., X_n)$ (if $\mathbb{E}(X_i) = 0$ for all $i = 1, ..N$) is given by

$$p(\mathbf{x}) \;=\; p(x_1, .., x_n) \;=\; (2\pi)^{-n/2} \mathrm{det}(\Sigma)^{-\frac{1}{2}} e^{-\frac{1}{2}\mathbf{x}^T \mathbf{\Sigma^{-1}} \mathbf{x}} \tag{1}$$

where $\mathbf{x} = (x_1, .., x_n)$ and $\mathbf{\Sigma}$ is the matrix with i,jth element $\Sigma_{ij} = \mathbb{E}(X_{t_i} X_{t_j})$, and det denotes the determinant of a matrix.

$R(s,t)$ also uniquely defines the law of X on $\{t_1, ..., t_n\}$, since it determines the covariance matrix of $(X_{t_1}, ..., X_{t_n})$ with (i,j)th element $R_{ij} := R(t_i, t_j)$, and for a zero-mean Gaussian vector, we only need its covariance to describe its joint pdf. Specifically, the joint density of a n-dimensional Gaussian random vector $\mathbf{X} = (X_1, ..., X_n)$ (with zero mean vector) is given by

$$p(\mathbf{x}) \;=\; p(x_1, .., x_n) \;=\; (2\pi)^{-n/2} \mathrm{det}(\Sigma)^{-\frac{1}{2}} \, e^{-\frac{1}{2}\mathbf{x}^T \mathbf{\Sigma^{-1}} \mathbf{x}}$$

where $\mathbf{\Sigma}$ is the matrix with i,jth element $\Sigma_{ij} = \mathbb{E}(X_i X_j)$, and det denotes the **determinant** of a matrix (if $n = 1$ this reduces to the density of a 1d $N(0, \sigma^2)$ random variable, i.e. $p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\frac{x^2}{\sigma^2}}$).

We can **sample** a Gaussian process X at $(t_1, ..., t_n)$ as $\mathbf{X} = \mathbf{CZ}$, where Z is a column vector $Z_1, .., Z_n$ of standard $N(0,1)$ random variables and C is the unique lower triangular $n \times n$ matrix such that $\mathbf{CC}^\top = \mathbf{\Sigma}$ which is known as the **Cholesky decomposition** of Σ. This means that $C_{ij} = 0$ if $i < j$, so the matrix looks like

$$C \;=\; \begin{bmatrix} C_{11} & 0 & .. & 0 \\ C_{21} & C_{22} & ... & 0 \\ ... & & & \\ C_{n1} & C_{n2} & ... & C_{nn} \end{bmatrix}$$

(see also e.g. FM06). For any (non-random) constant vector $\mathbf{x} = (x_1, ..., x_n)$, $\mathbb{E}((\sum_{i=1}^n x_i X_{t_i})^2) = \sum_{i=1}^n \sum_{j=1}^n R(t_i, t_j) x_i x_j = \mathbf{x}^\top \mathbf{\Sigma} \, \mathbf{x} \geq \mathbf{0}$, and $\mathbb{E}((\sum_{i=1}^n x_i X_{t_i})^2) > 0$ if we make the natural assumption that $X_{t_1}, ..., X_{t_n}$ are **linearly independent**, so Σ is **positive definite**, which means it has **positive determinant** and is **invertible** by standard results in linear algebra. If we don't have linear independence it means $\mathrm{Corr}(X_{t_i}, X_{t_j}) = 1$ for some i, j, we just remove some of the X_{t_i}'s until this is no longer the case. The Cholesky method gives the correct covariance because

$$\mathbb{E}(\mathbf{XX}^\top) \;=\; \mathbf{C}\mathbb{E}(\mathbf{ZZ}^\top)\mathbf{C}^\top \;=\; \mathbf{CC}^\top \;=\; \mathbf{\Sigma}$$

where we have used that

$$\mathbf{ZZ}^\top \;=\; \begin{bmatrix} Z_1^2 & Z_1 Z_2 & .. & 0 \\ Z_2 Z_1 & Z_2^2 & ... & 0 \\ ... & & & \\ Z_n Z_1 & ... & ... & Z_n^2 \end{bmatrix}$$

whose expectation is the identity matrix.

A zero-mean Gaussian process B_t^H is called standard **fractional Brownian motion** (fBM) with **Hurst exponent** $H \in (0,1)$ if it has covariance function

$$R_H(s,t) \;=\; \mathbb{E}(B_t^H B_s^H) - \mathbb{E}(B_t^H)\mathbb{E}(B_s^H) \;=\; \mathbb{E}(B_t^H B_s^H) \;=\; \frac{1}{2}(|t|^{2H} + |s|^{2H} - |t-s|^{2H}) \tag{2}$$

for $s, t \in \mathbb{R}$ (B^H can be defined for all $t \in \mathbb{R}$ or just $t \in [0, \infty)$.

- For $H = \frac{1}{2}$ and $0 \leq s \leq t$, we see that $R_H(s,t) = \frac{1}{2}(t + s - (t - s)) = s$, so we see that for $H = \frac{1}{2}$, $R_H(s,t) = \min(s,t)$, i.e. when $H = \frac{1}{2}$, fBM is just a standard Brownian motion.

- For $H \neq \frac{1}{2}$, B^H is not a martingale or a Markov process (recall a process M is a martingale if $\mathbb{E}(M_t|\mathcal{F}_s) = M_s$ for $t \geq s$ and M is a Markov process if $\mathbb{E}(f(M_t)|\mathcal{F}_s) = \mathbb{E}(f(M_t)|M_s)$). When $H \in (0, \frac{1}{2})$, B^H is **rougher** than standard BM, and when $H \in (\frac{1}{2}, 1)$, B^H is smoother than standard BM (see simulations in Figure 1 below); more specifically $|B_t^H - B_s^H| \leq c_1(\omega)|t - s|^{H - \varepsilon}$ a.s. for any $\varepsilon \in [0, H)$ where $c_1(\omega)$ is a (in general random) constant depending on B^H itself (this comes partly from the **Kolmogorov Continuity Theorem**, see below for details, and Chapter 2 in FM04 for proof using the Borel-Cantelli lemma).

We now prove some basic fundamental properties of fBM:

- $R(as,at) = a^{2H}R(s,t)$, so $X_{a(.)} \sim a^H X_{(.)}$ (i.e. both processes have the same joint distribution at $(t_1,...,t_n)$), so the process X is said to be **self-similar**, and in particular $B^H_{at} \sim a^H B^H_t$.
- From (2), we see that

$$\mathbb{E}((B^H_t - B^H_s)^2) \;=\; \mathbb{E}((B^H_t)^2) + \mathbb{E}((B^H_s)^2) - 2\mathbb{E}(B^H_s B^H_t) \;=\; t^{2H} + s^{2H} - (t^{2H} + s^{2H} - |t-s|^{2H}) \\ = |t-s|^{2H}$$

  so $B^H_t - B^H_s \sim N(0, |t-s|^{2H})$; thus B^H has **stationary increments**.
- There exists a function $k(s,t)$ such that B^H can be realized as $B^H_t = \int_0^t k(s,t)dB_s$ where B is standard Brownian motion, and $k(s,t) \sim const.(t-s)^{H-\frac{1}{2}}$ as $s \nearrow t$, so k blows up as $s \nearrow t$ when $H \in (0, \frac{1}{2})$. The lower triangular Cholesky matrix approximates the function k such that $B^H_t = \int_0^t k(s,t)dB_s$ where B is a standard Brownian motion **if you use the same Z vector to generate B and B^H as you should do for this Task in Part 2**. In particular, for $0 < t < u$ we have the **conditional decomposition**:

$$B^H_u \;=\; \int_0^t k(s,u)dB_s + \int_t^u k(s,u)dB_s\,.$$

  The two expressions on the right hand side are **independent**, and conditioned on B up to time t, B^H has conditional distribution which is $N(\int_0^t k(s,u)dB_s, \int_t^u k(s,u)^2 ds)$. In this sense we see that the process B^H has memory. Since $\int_0^t k(s,u)dB_s \neq B_t$ when $H \neq \frac{1}{2}$ and not just a simple function of B_t, we see that B^H is not a martingale, nor is Markov.
- $\mathbb{E}((Z^H_t)^2) = \mathbb{E}((B^H_t)^2)$. A commonly used simpler version of this process is the **Riemann-Liouville** process $Z^H_t = \sqrt{2H}\int_0^t (t-s)^{H-\frac{1}{2}}dB_s$, which is also self-similar, but no longer has stationary increments. Note that $Z^H_t \sim B^H_t$, but B^H and Z do not have the same covariance function and Z does not have stationary increments.
- If we set $X_n = B^H_n - B^H_{n-1}$; then X_n is a discrete-time Gaussian process; in fact from the stationary increments property above we know that $X_k \sim N(0,1)$ for all k i.e. (X_k) is a sequence of $N(0,1)$ random variables which are not independent of each other. Thus X is a discrete-time stationary process, and X is known as **fractional Gaussian noise**(fGn)); then $\rho_n = \mathbb{E}(X_{k+n}X_k)$ depends only on n (not k) and has **autocovariance** function

$$\begin{aligned}\rho(n) \;:=\; \mathbb{E}(X_{k+n}X_k) \;&=\; \mathbb{E}((B^H_{k+n} - B^H_{k+n-1})(B^H_k - B^H_{k-1})) \\ &=\; R_H(k+n,k) + R_H(k+n-1,k-1) - R_H(k+n,k-1) - R_H(k+n-1,k) \\ &=\; \frac{1}{2}[(n+1)^{2H} - n^{2H} - (n^{2H} - (n-1)^{2H})] \;\sim\; const. \times n^{2H-2} \qquad (n \to \infty)\end{aligned}$$

  and thus (by convexity of the function $g(n) := n^{2H}$), we see that $\mathbb{E}(X_{k+n}X_k) > 0$ if $H \in (\frac{1}{2}, 1)$ (which we call **persistent**) and $\mathbb{E}(X_{k+n}X_k) < 0$ for $H \in (0, \frac{1}{2})$ (which we call **anti-persistent**). Loosely speaking, for $H > \frac{1}{2}$, if B^H was increasing in the past, it is more likely to increase in the future, and vice versa. Similarly for $H < \frac{1}{2}$, if B^H was increasing in the past, it is more likely to decrease in the future, and vice versa.
-

We now recall the **Kolmogorov continuity theorem**:

Theorem 0.1 *Let $\alpha, \varepsilon, c > 0$ and X be a random process which satisfies*

$$\mathbb{E}(|X_t - X_s|^\alpha) \;\leq\; c|t-s|^{1+\varepsilon}\,.$$

*Then X is γ-**Hölder continuous** for all $\gamma \in [0, \frac{\varepsilon}{\alpha})$.*

From above, we know that

$$B^H_t - B^H_s \;\sim\; N(0,(t-s)^{2H}) \;\sim\; (t-s)^H Z$$

where $Z \sim N(0,1)$, so $\mathbb{E}(|B^H_t - B^H_s|^q) = \mathbb{E}(|Z|^q)(t-s)^{qH}$. Then applying the Kolmogorov continuity theorem with $\alpha = q$ and $1 + \varepsilon = qH$, we see that B^H is γ-Hölder continuous for all $0 < \gamma < \frac{\varepsilon}{\alpha} = \frac{qH-1}{q}$ for any

$q > 1/H$; but $\frac{qH-1}{q} \nearrow H$ as $q \to \infty$, so we can make the stronger statement that B^H is γ-Hölder continuous for all $0 < \gamma < H$. Note the theorem does not tell us that B^H isn't smooth, but in chapter 2 of FM14 (see `https://nms.kcl.ac.uk/martin.forde/Chap2.pdf)\,we`will be able to make the more precise statement that fBM is H-Hölder continuous but not $H+\varepsilon$-Hölder continuous.

In Task 2, we consider a sample path of the process $X = \nu B^H$, so we can set $\Sigma_{ij} = \nu^2 \mathbb{E}(B^H_{i/n} B^H_{j/n}) = \mathbb{E}(X_{i/n} X_{j/n})$. You need to numerically maximize the log of the likelihood function in (1) over H and ν (note MATLAB and Python minimize not maximize so one has to minimize minus the log likelihood function to get a maximizer).

For first task in Part 2, you need to use the **self-similarity** of fBM to derive equality of the form

$$\frac{1}{n}\mathbb{E}(\sum_{i=1}^{n} |B^H_{i\Delta} - B^H_{(i-1)\Delta}|^q) \quad = \quad \Delta^p \, \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}(|(B^H_i - B^H_{i-1})|^q) \quad = \quad f(T,n,q,H)$$

for some power p and some function f (also using that $B^H_i - B^H_{i-1} \sim N(0,1)$ for all i) which you need to determine, expressed in terms $K_q = \mathbb{E}(|Z|^q)$ where $Z \sim N(0,1)$. An **estimator** $\hat{H}_n$ for H is said to be biased if $\mathbb{E}(\hat{H}_n) \neq H$ (or unbiased otherwise). Use **Jensen's inequality** to determine whether $\hat{H}_n$ is biased or not in the first task.

Since $B^H_t = 0$, do not include $t = 0$ in your $\Sigma_{i,j}$ matrix, or else you will get a zero determinant for Σ. For Cholesky, do not include $t = 0$ in your set of time points $t_1, ..., t_n$. For the Monte Carlo task in Part 2, you can estimate the call price with

$$\text{CallPrice} \quad \approx \quad \frac{1}{M}\sum_{i=1}^{M}(S^i_T - K)^+$$

where S^i is the ith sample path and M is the total number of Monte Carlo paths (or better use **antithetic variables** leads to four different call price estimates using $(W,B), (-W,B), (W,-B), (-W,-B)$ since we have two Brownian motions (and then take the average of all four estimates for each i) or (better) use the **Renault-Touzi formula**:

$$\text{CallPrice} \quad = \quad \mathbb{E}(\max(S_T - K, 0)) \quad = \quad \mathbb{E}(C^{BS}(\tilde{S}_0, K, \sigma = 1, \tau = (1-\rho^2)\int_0^T \sigma_u^2 du, r = 0)) \tag{3}$$

with antithetic variables (just B and $-B$ here) (see chapter 1 of FM14 for proof of where this comes from) where $\sigma_t = \sqrt{V_t}$, $C^{BS}(S,K,\sigma,\tau,r)$ is the usual Black-Scholes (Black-Scholes) formula from e.g. FM02, and $\log \tilde{S}_0 := \log S_0 - \frac{1}{2}\rho^2 \int_0^t \sigma_u^2 du + \rho \int_0^t \sigma_u dB_u$, i.e. we use the BS formula with a *random* maturity and (when $\rho \neq 0$) a random initial stock price $\tilde{S}_0$ (proof of why this formula works is in Chap1 document for FM14 `https://nms.kcl.ac.uk/martin.forde/Chap1.pdf).` This formula means we only have to simulate the second Brownian motion B and not the first Brownian motion W to estimate the call price, which reduces the **sample variance** of our call price estimate (again see FM14 for proog of this); one should also use antithetic variables as well, i.e. use the mirror image of B and then take the average of the two MC approximations for the call price).

We similarly approximate (3) with

$$\frac{1}{M}\sum_{i=1}^{M} C^{BS}(\tilde{S}^i_0, K, 1, (1-\rho^2)\int_0^T V^i_u du, 0)$$

where V^i is the i'th Monte Carlo path for V (note we are ignoring antithetic paths here). Then (since the MC paths are i.i.d), the variance of this expression is

$$\frac{1}{M} C^{BS}(\tilde{S}^i_0, K, 1, (1-\rho^2)\int_0^T V^i_u du, 0)$$

so the **sample standard deviation** of the MC call price estimate scales like $\frac{1}{\sqrt{M}}$, which we can also use to estimate confidence intervals for the call price.

It is very important to check the convergence of a Monte Carlo estimate for a call price by varying the Number of time steps (with a fixed random seed) and the Number of paths (without fixing, since convergence is typically quite slow and bias large for rough models with e.g $H \leq .05$.

Recall the definition of **implied volatility**: $\hat{\sigma}$ is the unique volatility value $\hat{\sigma}$ such that Call Price $= C^{BS}(S,K,\hat{\sigma},T,r)$, where in our case here Call Price will be the Monte Carll price appoximation.

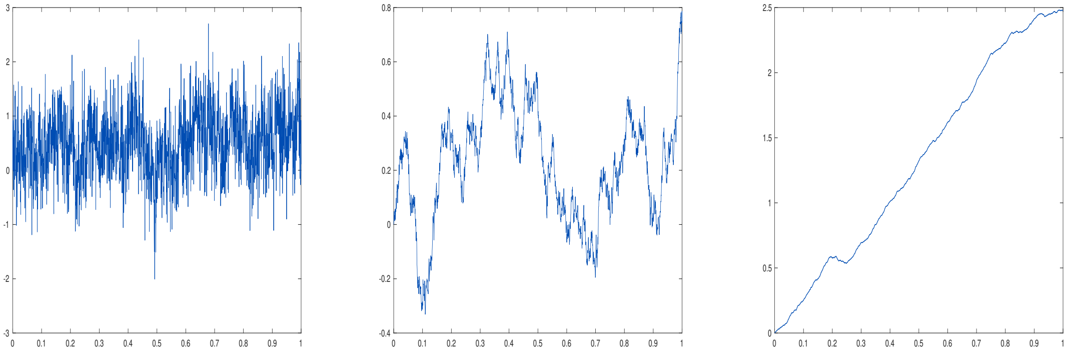

Figure 1: (i) Here we have plotted a Monte Carlo simulation of B^H using the Cholesky method for $H = .05$, $H = .5$ and $H = 0.9$

If N is total number of points in sample path, then N/m is the number of V points we will estimate (and the number of time windows), and each V point is estimating by adding up m equidistant sums of squares of log returns for each time window.

For estimating V with sums of squares of log returns, refer to chapter 3 of FM14 Lecture notes (see `https://nms.kcl.ac.uk/ma`

One important numerical trick to be aware of for the MLE computation: sometimes computing $\log(\det(\Sigma))$ will give NaN if the covar matrix is too big (i.e. number of data points N in the sample path is too large), but can get round this by multiplying covariance matrix Σ by a constant $c_1 > 1$ (e.g 2, 4 or whatever makes it work), so $\det(c_1\Sigma) = c_1^N \det(\Sigma)$ (basic linear algebra result), hence

$$\log\det(\Sigma) \quad = \quad \log\det(c_1\Sigma) - N\log c_1$$

so the expression on the right hand side is what one should code to compute $\log\det(\Sigma)$.