Simulation of Multifractional Brownian Motion

Grace CHAN[1] and Andrew T. A. WOOD[2]

[1] Department of Statistics, University of New South Wales, Sydney, NSW, 2052, Australia.

[2] Department of Mathematical Sciences, University of Bath, Bath, BA2 7AY, UK.

Abstract. Multifractional Brownian motion (MFBm) is a generalization of Fractional Brownian motion (FBm) in which the Hurst parameter varies with time in a prescribed manner. In this paper we investigate an approximate method for simulating realizations of MFBm on a finite grid.

Keywords. Fractal dimension; Hurst parameter; kriging; long-range dependence; sample path roughness; stationary increments.

1 Introduction

1.1 Review of FBm

FBm is a zero-mean self-similar Gaussian process $\{B_H(t)\}_{t\geq 0}$ with covariance function given by

$$cov\{B_H(s), B_H(t)\} = \frac{1}{2}\left\{|s|^{2H} + |t|^{2H} - |s-t|^{2H}\right\} \tag{1}$$

where $H \in (0,1)$ is known as the *Hurst parameter*. Note that $B_H(0) \equiv 0$ and $var\{B_H(t)\} = t^{2H}$. In the special case $H = 1/2$, FBm reduces to ordinary Brownian motion. An important property of FBm, which follows from (1), is that it has *stationary increments*.

FBm has proved to be of considerable interest in diverse fields such as Hydrology, Signal Processing and Financial Mathematics. Basic properties of FBm are given by Mandelbrot and Van Ness (1968) and Falconer (1990), and multivariate generalizations are considered by Adler (1981).

The Hurst parameter H plays a dual rôle: on the one hand, it determines the *roughness* of the sample path; and on the other, it determines the *long-range dependence* properties of the process. With regard to sample path roughness, H determines the fractal (or Hausdorff) dimension, D, of the (graph of the) sample path through the following simple formula: $D = 2 - H$. See Adler (1981) and Falconer (1990) for details.

For future reference, we now mention two distinct integral representations of FBm:

$$B_H(t) = C_1(H) \left\{ \int_{-\infty}^{t} (t-s)^{H-(1/2)} dW(s) - \int_{-\infty}^{0} (-s)^{H-(1/2)} dW(s) \right\} \tag{2}$$

and

$$B_H(t) = C_2(H) \int_{-\infty}^{\infty} \{1 - \cos(st) - \sin(st)\} |s|^{-H-(1/2)} dW(s) \tag{3}$$

where $C_1(H)$ and $C_2(H)$ are chosen so that $var\{B_H(t)\} = |t|^{2H}$, and W is a pure *white noise* process, i.e. Brownian motion on $(-\infty, \infty)$. For (2), see Mandelbrot and Van Ness (1968, p.423). Representation (3) is (essentially) due to Hunt; see Mandelbrot and Van Ness (1968, p. 435) for closely related formulae.

1.2 Simulation of FBm

Suppose that we wish to simulate $\{B_H(t)\}$ at locations on a grid $\delta, 2\delta, \ldots, n\delta$, where the grid width δ is open to choice, and n, the number of grid points, is very large. Then, writing $t_j = j\delta$, we define the increments

$$Y_j = \delta^{-H} \{B_H(t_j) - B_H(t_{j-1})\} \quad (j = 1, \ldots, n) \tag{4}$$

and proceed as follows:

Step 1: Simulate $Y_1, \ldots, Y_n$;

Step 2: For each j, calculate $B_H(t_j) = \delta^H(Y_1 + \ldots + Y_j)$.

It follows from (1) that the Y-sequence is stationary, and that the covariance between Y_j and Y_k is given by (9) with $H_u = H_v = H$. So Step 1 may be performed efficiently using the *circulant embedding* approach for simulating stationary Gaussian processes, with prescribed covariance function, on a finite grid. This procedure, which is theoretically exact for many covariance functions, can cope with very large n; see Chan and Wood (1997a).

2 Multifractional Brownian motion

In some real datasets there is evidence that the roughness of the sample path varies with location. In such cases, a single number H (or D) may not provide an adequate global description of the roughness of the sample path and there is motivation for developing models which allow for varying roughness. Lévy-Véhel (1995) has considered such datasets in Image Analysis and Signal Processing contexts, and these led him to consider a generalization of FBm which he calls Multifractional Fractional Brownian motion (MFBm). A key feature of MFBm is that sample path roughness is described by a function $H(t)$ rather than just a single number. It makes sense to define the *local* fractal dimension of the sample path as $D(t) = 2 - H(t)$.

Lévy-Véhel's (1995) construction of MFBm is as follows: given a continuous function $H(t)$ strictly bounded by 0 and 1, define $X(H(t), t) \equiv B_{H(t)}(t)$ for $t \geq 0$ using (2). However, for present purposes, Lévy-Véhel's construction is inconvenient because the covariance of $B_{H_1}(t_1)$ and $B_{H_2}(t_2)$, obtained via (2), is rather complex. It turns out that, if we use (3) rather than (2), then the covariance of $B_{H_1}(t_1)$ and $B_{H_2}(t_2)$ is much simpler, and is given by

$$\frac{g(H_1, H_2)}{2}\{|t_1|^{H_1+H_2} + |t_2|^{H_1+H_2} - |t_1 - t_2|^{H_1+H_2}\} \tag{5}$$

where $g(H_1, H_2) = \{I(H_1)I(H_2)\}^{-1/2} I\{(H_1 + H_2)/2\}$ and

$$I(H) = \begin{cases} \dfrac{\Gamma(1-2H)}{H} \sin\left\{\dfrac{\pi}{2}(1-2H)\right\} & H \in (0, 1/2), \\ \pi & H = 1/2, \\ \dfrac{\Gamma\{2(1-H)\}}{H(2H-1)} \sin\left\{\dfrac{\pi}{2}(2H-1)\right\} & H \in (1/2, 1). \end{cases} \tag{6}$$

The derivation of (5) follows directly from (3). We omit the details, except to mention that (6) is obtained using formulae on p.422 of Gradshteyn and Ryzhik (1965). Note that when $H_1 = H_2$, $g(H_1, H_2) \equiv 1$. It would be interesting to know whether the differences between the MFBm's constructed via (2) and (3) are of any significance in statistical applications.

3 Simulation of MFBm

3.1 The Algorithm

Suppose that we are given a (deterministic) time-varying Hurst function $H(t)$ and that we wish to simulate an MFBm $X(H(t), t) \equiv B_{H(t)}(t)$ at locations on a finite grid. Choose $0 < H_1 < \ldots H_m < 1$, for example

$$H_u = u/(m+1) \quad (u = 1, \ldots, m). \tag{7}$$

These H_u's will be Hurst parameters for m correlated FBms whose joint covariance structure is determined by (5). Let δ, n and the t_j's be as in subsection 1.2 and, bearing (3) in mind, define increments

$$Y_{j,u} = \delta^{-H_u} \{B_{H_u}(t_j) - B_{H_u}(t_{j-1})\} \quad (j = 1, \ldots, n;\ u = 1, \ldots, m) \tag{8}$$

and write $W_j = (Y_{j,1}, \ldots, Y_{j,m})^T$. Our simulation algorithm may be specified in outline as follows:

Step 1 Simulate $\{Y_{j,u} : j = 1, \ldots, n;\ u = 1, \ldots, m\}$.

Step 2 For each j and u, calculate $B_{H_u}(t_j) = \delta^{H_u}(Y_{1,u} + \ldots + Y_{j,u})$.

Step 3 For each j, "predict" $X(H(t_j), t_j)$ using some form of kriging based on the "observations" $\{B_{H_u}(t_j) : j = 1, \ldots, n;\ u = 1, \ldots, m\}$.

Using (5), it is seen that $\{W_j : j = 1, 2, \ldots\}$ is a stationary vector-valued Gaussian sequence with the covariance of $Y_{j,u}$ and $Y_{k,v}$ given by

$$\frac{g(H_u, H_v)}{2}\left\{|j-k-1|^{H_u+H_v} + |j-k+1|^{H_u+H_v} - 2|j-k|^{H_u+H_v}\right\}. \tag{9}$$

Thus Step 1 can be performed efficiently using the circulant embedding approach for stationary *vector-valued* Gaussian processes; see Chan and Wood (1997b) and also `http://www.maths.unsw.edu.au/~grace` for details of this algorithm. Since Step 2 is a matter of simple arithmetic, this leaves Step 3, which we now discuss.

3.2 Kriging Neighbourhoods

The kriging in Step 3 is performed as follows: at each location $(H(t_j), t_j)$, we specify a set of neighbours $N_j = \{(v, k)\}$ and then predict $X(H(t_j), t_j)$ by

$$X(H(t_j), t_j) = \sum_{(v,k)\in N_j} \gamma_{v,k}^{(j)} B_{H_v}(t_k) \tag{10}$$

We shall use the two types of N_j specified in Figure 1 ("small" and "large").

If we adopt a Mean Squared Error (MSE) criterion then, using (5) to obtain all covariances below, the optimal choice of $\gamma^{(j)} = (\gamma_{v,k}^{(j)} : (v, k) \in N_j)$ is given by

$$\hat{\gamma}^{(j)} = \{cov(Z_j)\}^{-1} cov\left(Z_j, B_{H(t_j)}(t_j)\right) = A_j^{-1} B_j \quad \text{(say)} \tag{11}$$

where $Z_j = (B_{H_v}(t_k) : (v, k) \in N_j)$; and the optimal MSE is given by

$$MSE_j = var\left\{B_{H(t_j)}(t_j)\right\} - B_j^T A_j^{-1} B_j \quad (j = 1, \ldots, n). \tag{12}$$

If $(H(t_j), t_j)$ is close to the boundary of the simulation region then some of the $B_{H_v}(t_k)$ at neighbouring locations may not have been simulated. In such cases, we simply use those neighbouring locations which are available, with the obvious modifications to (10)–(12).

Figure 1: Kriging neighbourhoods (X=target, o=neighbour)

Small

	t_{j-1}	t_j	t_{j+1}
H_{u+1}	o	o	o
$H(t_j)$		X	
H_u	o	o	o

Large

	t_{j-2}	t_{j-1}	t_j	t_{j+1}	t_{j+2}
H_{u+2}	o	o	o	o	o
H_{u+1}	o	o	o	o	o
$H(t_j)$			X		
H_u	o	o	o	o	o
H_{u-1}	o	o	o	o	o

4 Numerical Results

We now describe numerical results obtained using the simulation algorithm presented in the previous section. Three smooth functions were considered for $H(t)$:

linear: $H(t) = t$;

logistic: $H(t) = 0.3 + 0.3/[1 + \exp\{-100(t - 0.7)\}]$;

periodic: $H(t) = 0.5 + 0.49\sin(4\pi t)$.

We only report the results with $\delta = 1/n$, $n = 1000$, $m = 8$ and the H_u's chosen according to (7); other cases were broadly similar.

In Figure 2, each function on the left is a single realization of an MFBm, obtained using the small kriging neighbourhoods (see Figure 1); and each curve on the right is the corresponding local dimension function $D(t) = 2 - H(t)$. Observe that Figure 2 displays exactly the behaviour we would expect: the larger (smaller) the value of $D(t)$ on the right, the rougher (smoother) the sample path on the left.

In Figure 3, the MSE obtained from (12) is displayed in the linear and periodic cases. Note that, in both cases, the MSE is very small unless $H(t)$ is "out of range" (i.e. unless either $H(t) < H_1 = 1/9$ or $H(t) > H_m = 8/9$). In the logistic case, $H(t)$ was always "within range" and the MSE was uniformly close to zero. In particular, the largest value of MSE_j in (12) in the logistic case was 3.2×10^{-5} (5×10^{-5}) when the large (small) kriging neighbourhoods were used. These MSE results suggest two broad conclusions: first, the importance of choosing H_1 and H_m so that $H(t)$ stays within range (i.e. $H_1 \leq H(t) \leq H_m$

Figure 2: Three examples of Multifractional Brownian motion

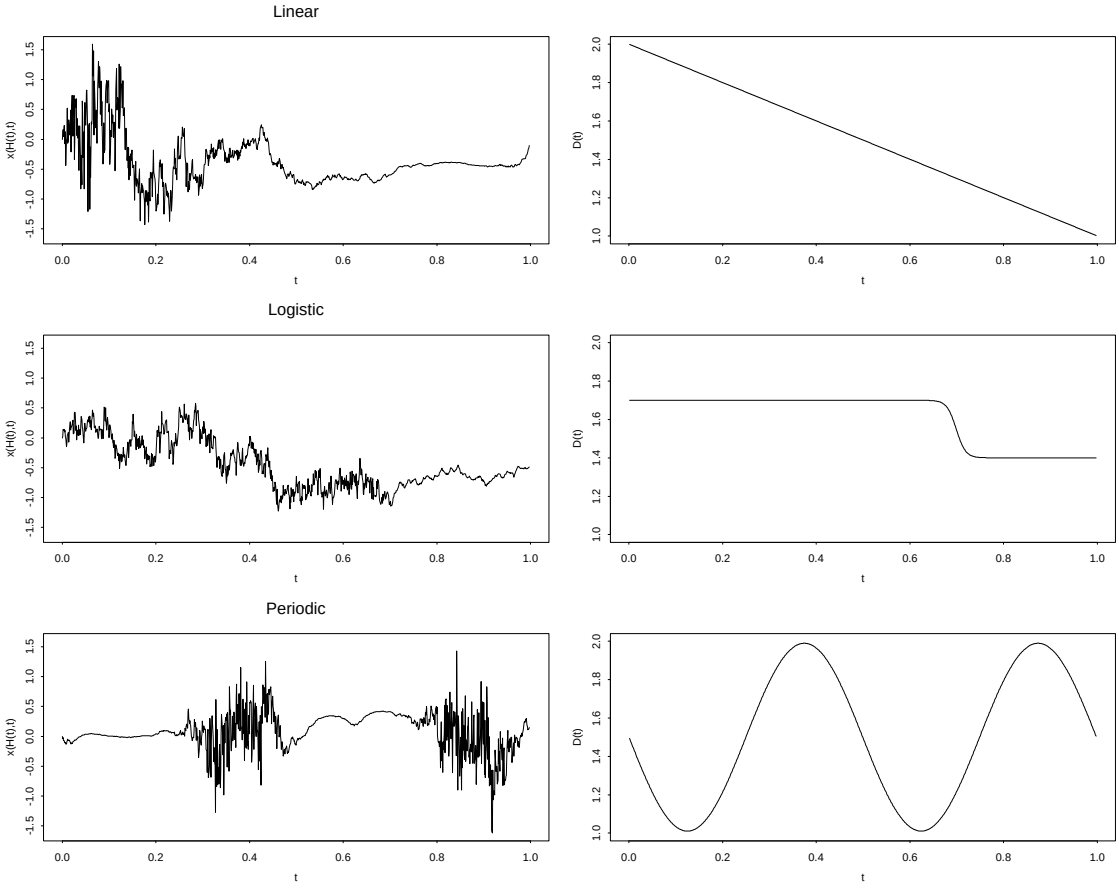

Figure 3: Prediction mean squared errors

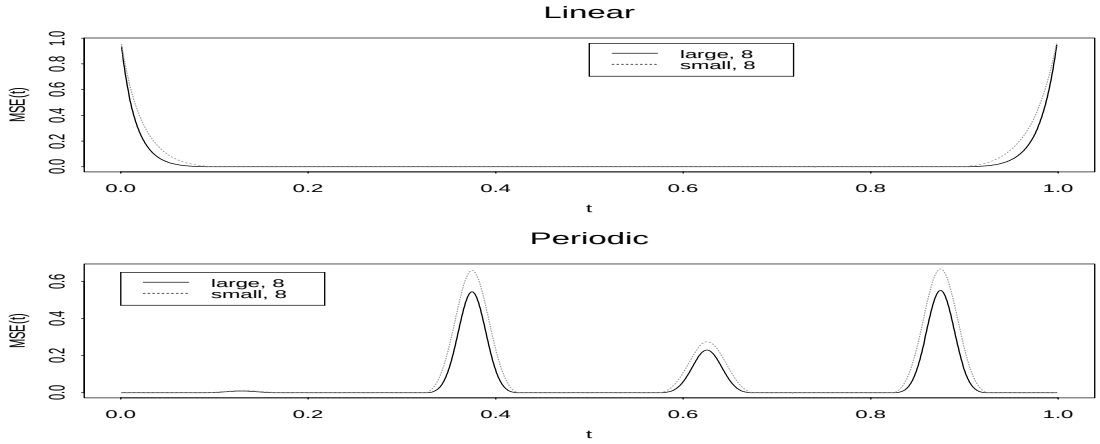

Table 1: CPU Timings (in seconds)

m	n	neighbourhood	$H(t)$	T_1	T_2	T_3	min:sec
8	1000	small	logistic	3.169	0.554	0.554	1:54
				(0.217)	(0.006)	(0.002)	
8	1000	large	logistic	3.261	0.349	9.799	16:58
				(0.249)	(0.013)	(0.140)	
8	500	small	logistic	3.202	0.517	0.279	1:23
				(0.251)	(0.006)	(0.002)	
8	500	large	logistic	3.165	0.535	4.933	9:10
				(0.256)	(0.005)	(0.014)	
8	1000	large	periodic	3.322	0.372	5.886	10:29
				(0.255)	(0.007)	(0.022)	
8	1000	large	linear	3.396	0.392	7.727	13:35
				(0.272)	(0.005)	(0.027)	

for all t); and second, that the gain in distributional accuracy achieved by using the large instead of small kriging neighbourhoods is rather modest.

In Table 1 we compare CPU timings for the small and large kriging neighbourhoods. Each entry in the table is the mean of 20 runs. The quantities T_1, T_2 and T_3 are defined as follows: T_1 is the "start up" time for Step 1 (i.e. it is only incurred once); T_2 is the (additional) time per realization of the Y–array in Step 1 and Step 2; and T_3 is the time needed to perform Step 3. The last column is the approximated CPU time in minutes and seconds for simulating 100 realization, which is equal to

$$T_1 + (T_2 + T_3) \times 100.$$

The results in Table 1 show that if the small kriging neighbourhoods are used, then the proportional of total CPU time spent in Step 3 is very modest; but if the large kriging neighbourhoods are used, then this proportion is rather substantial. Bearing Table 1 and the comments in the previous paragraph in mind, and noting that the algorithm is somewhat easier to program when the small neighbourhoods are used, we suggest that the approach based on the small kriging neighbourhoods is to be preferred.

5 Conclusions

The results in Section 4 indicate that our algorithm for simulating MFBm using the small kriging neighbourhoods should have sufficient distributional accuracy for most practical purposes. However, note how important it is to ensure that $H(t)$ stays "within range" (see Section 4); this can always be arranged in practice, provided $H(t)$ is bounded away from 0 and 1. In future work, we hope to discuss statistical applications of MFBm.

References

Adler, R. (1981). *The Geometry of Random Fields.* Wiley, New York.

Chan, G. & Wood, A.T.A. (1997a). An algorithm for simulating stationary Gaussian random fields. *Applied Statistics* **46**, 171–181.

Chan, G. & Wood, A.T.A. (1997b). Simulation of stationary Gaussian vector fields. *Statistics and Computing*, submitted.

Falconer, K.J. (1990). *Fractal Geometry: Mathematical Foundations and Applications.* Wiley, New York.

Gradshteyn, I.S. & Ryzhik, I.M. (1965). *Tables of Integrals, Series and Products.* Academic Press, New York.

Lévy-Véhel, J. (1995). Fractal approaches in signal processing. *Fractals*, **3**, 755–775.

Mandelbrot, B. B. & Van Ness, J. W. (1968). Fractional brownian motions, fractal noises and applications. *SIAM Review*, **10**, 422–437.