

# Numerical Explorations for Fast Spectrum of Fractional Gaussian Noise

Martin Mächler  
ETH Zurich

June 2011 (L<sup>A</sup>T<sub>E</sub>X<sup>ed</sup> August 30, 2024)

---

## Abstract

The package **longmemo** ....  
Paxson (1997) ... ..

*Keywords:* Euler-Maclaurin Formula, Fractional Gaussian Noise, Spectrum.

---

## 1. .. intro ..

The spectral density of fractional Gaussian noise (“fGn”) with Hurst parameter  $H \in (0, 1)$  is (Beran (1986, 1994))

$$f_H(\lambda) = \mathcal{A}(\lambda, H) \left( |\lambda|^{-2H-1} + \mathcal{B}(\lambda, H) \right), \quad (1)$$

for  $\lambda \in [-\pi, \pi]$ , where  $\mathcal{A}(\lambda, H) = 2 \sin(\pi H) \Gamma(2H + 1) (1 - \cos \lambda)$ , and

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

For the Whittle estimator of  $H$  and also other purposes, its advantageous to be able to evaluate  $f_H(\lambda_i)$  efficiently for a whole vector of  $\lambda_i$ , typically Fourier frequencies  $\lambda_i = 2\pi i/n$ , for  $i = 1, 2, \dots, \lfloor (n-1)/2 \rfloor$ . Such evaluation is problematic because of the infinite sum for  $\mathcal{B}(\lambda, H)$  in (2).

Traditionally, e.g., already in Appendix.... of Beran (1994), the infinite sum  $\sum_{j=1}^{\infty}$  had been replaced by  $\sum_j^{200}$  — which was still not very efficient and not extremely accurate. In our R package **longmemo**, we now provide the function `B.specFGN( $\lambda, H$ )` to compute  $\mathcal{B}(\lambda, H)$ , using several ways to compute the infinite sum approximately, e.g., for  $H = 0.75$  and  $n = 500$ , i.e., at 250 Fourier frequencies,

```
> require("longmemo")
> fr <- .ffreq(500)
> B.1 <- B.specFGN(fr, H = 0.75, nsum = 200, k.approx=NA)
> B.xct <- B.specFGN(fr, H = 0.75, nsum = 10000, k.approx=NA)
> all.equal(B.xct, B.1)
```

```
[1] "Mean relative difference: 0.0001243095"
```

which means that the 200 term approximation is accurate to 4 decimal digits for  $H = .75$  but the accuracy is smaller for smaller  $H$ .

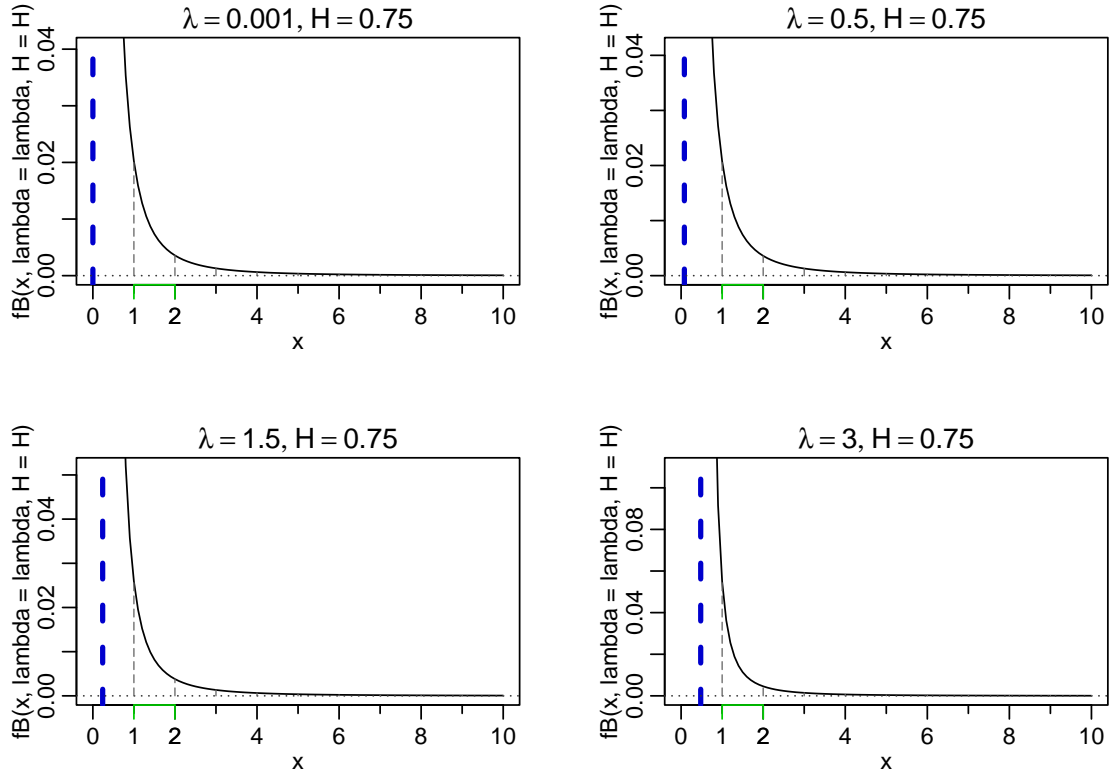
For this reason, Paxson (1997) derived formulas for fast and stilly quite accurate approximations of  $\mathcal{B}(\lambda, H)$ , noting that  $\mathcal{B}(\lambda, H) = \sum_{j=1}^{\infty} f(j; \lambda, H)$  for

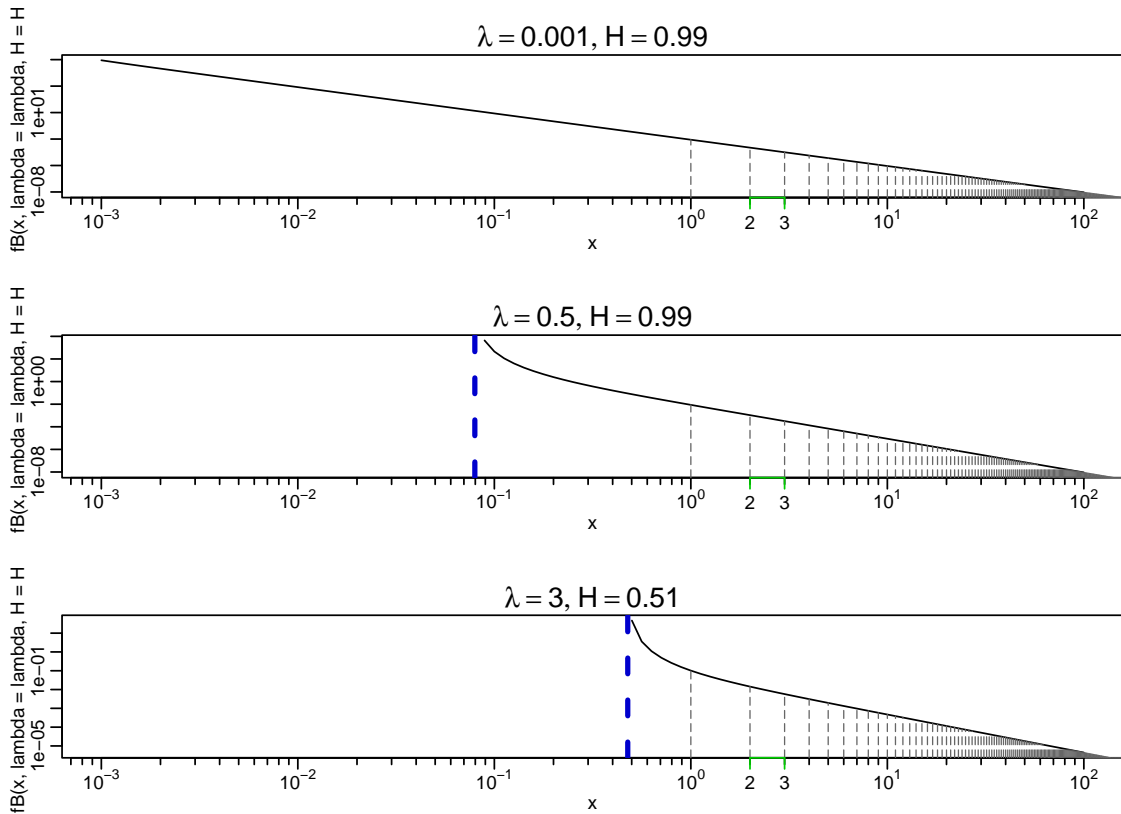
$$f(x; \lambda, H) = (2\pi x + \lambda)^{-(2H+1)} + (2\pi x - \lambda)^{-(2H+1)}, \quad (3)$$

and the fact that  $\sum_{j=1}^{\infty} f(j)$  is a Riemann sum approximation of  $\int_0^{\infty} f(x) dx$  or  $\int_1^{\infty} f(x) dx$ .

```
> fB <- function(x, lambda, H) {
  u <- 2 * pi * x
  h <- -(2 * H + 1)
  (u + lambda)^h + (u - lambda)^h
}
```

Now its clear that  $f(x)$  cannot be computed (or “is infinite”) at  $x = 0$ , and more specifically,  $f(x)$  tends to  $\infty$  when  $x \rightarrow \frac{\lambda}{2\pi}$ , as in the second term of  $f$ ,  $2\pi x - \lambda$  only remains positive when  $2\pi x > \lambda$ . This is always fulfilled for  $x \in \{1, 2, \dots\}$ , as  $\lambda < \pi$ , but is problematic when considering  $\int_0^b f(x) dx$  as above. Some illustrations of the function  $f(x; \lambda, H)$  and its “pole” at  $\frac{\lambda}{2\pi}$ :





So, very clearly, Paxson’s first formula, using  $\int_0^1 f(x) dx$  is not feasible, as  $f(x)$  is *not* defined (or defined as  $\infty$ ) for  $x \leq \lambda/(2\pi)$ .

However, his generalized formula, “(7), p. 15”,

$$\sum_{i=1}^{\infty} f_i \approx \sum_{j=1}^k f_j + \frac{1}{2} \int_k^{k+1} f(x) dx + \int_{k+1}^{\infty} f(x) dx, \tag{4}$$

clearly *is* usable for  $k \geq 1$  (but not for  $k = 0$ , contrary to what he suggests). Indeed, with `B.specFGN( $\lambda, H, k.approx$ )`, we now provide the result of applying approximation (4) to the infinite sum for  $\mathcal{B}(\lambda, H)$  in (2).

Paxson ended the  $k = 3$  approximation which he further improved considerably, empirically, by numerical comparison (and least squares fitting) with the “accurate” formula using `nsum = 10’000` terms. In the following section, we propose another improvement over Paxson’s original idea:

## 2. Better approximations using the Euler–Maclaurin formula

Copied straight from [http://en.wikipedia.org/wiki/Euler-Maclaurin\\_formula](http://en.wikipedia.org/wiki/Euler-Maclaurin_formula) :

If  $n$  is a natural number and  $f(x)$  is a smooth, i.e., sufficiently often differentiable function defined for all real numbers  $x$  between 0 and  $n$ , then the integral

$$I = \int_0^n f(x) dx \tag{5}$$

can be approximated by the sum (or vice versa)

$$S = \frac{1}{2}f(0) + f(1) + \cdots + f(n-1) + \frac{1}{2}f(n)$$

(see trapezoidal rule). The Euler–Maclaurin formula provides expressions for the difference between the sum and the integral in terms of the higher derivatives  $f^{(k)}$  at the end points of the interval 0 and  $n$ . Explicitly, for any natural number  $p$ , we have

$$S - I = \sum_{k=2}^p \frac{B_k}{k!} \left( f^{(k-1)}(n) - f^{(k-1)}(0) \right) + R$$

where  $B_1 = -1/2$ ,  $B_2 = 1/6$ ,  $B_3 = 0$ ,  $B_4 = -1/30$ ,  $B_6 = 1/42$ ,  $B_8 = -1/30$ , ... are the Bernoulli numbers, and  $R$  is an error term which is normally small for suitable values of  $p$ . (The formula is often written with the subscript taking only even values, since the odd Bernoulli numbers are zero except for  $B_1$ .)

Note that

$$-B_1(f(n) + f(0)) = \frac{1}{2}(f(n) + f(0)).$$

Hence, we may also write the formula as follows:

$$\sum_{i=0}^n f(i) = \int_0^n f(x) dx - B_1(f(n) + f(0)) + \sum_{k=1}^p \frac{B_{2k}}{(2k)!} \left( f^{(2k-1)}(n) - f^{(2k-1)}(0) \right) + R. \quad (6)$$

.....

In the context of computing asymptotic expansions of sums and series, usually the most useful form of the Euler–Maclaurin formula is

$$\sum_{n=a}^b f(n) \sim \int_a^b f(x) dx + \frac{f(a) + f(b)}{2} + \sum_{k=1}^{\infty} \frac{B_{2k}}{(2k)!} \left( f^{(2k-1)}(b) - f^{(2k-1)}(a) \right),$$

where  $a$  and  $b$  are integers. Often the expansion remains valid even after taking the limits  $a \rightarrow -\infty$  or  $b \rightarrow +\infty$ , or both. In many cases the integral on the right-hand side can be evaluated in closed form in terms of elementary functions even though the sum on the left-hand side cannot.

(end of citation from Wikipedia)

— —

### 3. Session Information

> `toLatex(sessionInfo())`

- R version 4.4.1 (2024-06-14), x86\_64-pc-linux-gnu
- Locale: LC\_CTYPE=en\_US.UTF-8, LC\_NUMERIC=C, LC\_TIME=en\_US.UTF-8, LC\_COLLATE=C, LC\_MONETARY=en\_US.UTF-8, LC\_MESSAGES=C, LC\_PAPER=en\_US.UTF-8, LC\_NAME=C, LC\_ADDRESS=C, LC\_TELEPHONE=C, LC\_MEASUREMENT=en\_US.UTF-8, LC\_IDENTIFICATION=C

- Time zone: Etc/UTC
- TZcode source: system (glibc)
- Running under: Ubuntu 24.04.1 LTS
- Matrix products: default
- BLAS: /usr/lib/x86\_64-linux-gnu/openblas-pthread/libblas.so.3
- LAPACK:  
/usr/lib/x86\_64-linux-gnu/openblas-pthread/libopenblas-p-r0.3.26.so ;  
LAPACK version3.12.0
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: longmemo 1.1-3
- Loaded via a namespace (and not attached): buildtools 1.0.0, compiler 4.4.1,  
knitr 1.48, maketools 1.3.0, sfsmisc 1.1-19, sys 3.4.2, tools 4.4.1, xfun 0.47

## 4. Conclusion

## References

- Beran J (1986). *Estimation, testing and prediction for self-similar and related processes*. Ph. d. thesis, no 8074, Swiss Federal Institute of Technology (ETH).
- Beran J (1994). *Statistics for Long-Memory Processes*. Monographs on Statistics and Applied Probability 61. Chapman & Hall, N. Y.
- Paxson V (1997). “Fast, approximate synthesis of fractional Gaussian noise for generating self-similar network traffic.” *SIGCOMM Comput. Commun. Rev.*, **27**, 5–18. ISSN 0146-4833. URL <http://doi.acm.org/10.1145/269790.269792>.

### Affiliation:

Martin Mächler  
Seminar für Statistik, HG G 16  
ETH Zurich  
8092 Zurich, Switzerland  
E-mail: [maechler@stat.math.ethz.ch](mailto:maechler@stat.math.ethz.ch)  
URL: <http://stat.ethz.ch/people/maechler>