Exact Sampling for the Maximum of Infinite Memory Gaussian Processes

Jose Blanchet, Lin Chen, and Jing Dong

Abstract We develop an exact sampling algorithm for the all-time maximum of negative-drifted Gaussian processes with general covariance structures. In particular, our algorithm can handle non-Markovian processes even with long-range dependence. Our development combines a milestone-event construction with rare-event simulation techniques. This allows us to find a random time beyond which the running time maximum will never be reached again. The complexity of the algorithm is random but has finite moments of all orders. We also test the performance of the algorithm numerically.

1 Introduction

It is a pleasure to contribute this paper in honor of Professor Pierre L'Ecuyer, whose many contributions to stochastic simulation have significantly influenced this area. In this paper, we propose and study an exact simulation algorithm for the infinite-horizon maximum of general Gaussian processes. What makes our contribution novel relative to other results in the literature is the non-Markovian, even long-range dependence structure of the processes we study. *Long-range dependence* is an important phenomenon that arises in many applications. An early work studying such a phenomenon dates back to the 1950's when Hurst [30] studied the flow of water in Nile river. Since then, evidence of long-range dependence has been found in economics [37], internet traffic [8, 32], linguistics [3], etc. (see [24, 38] for comprehensive reviews). More generally, Gaussian processes have been one of

Jose Blanchet
Stanford University, 475 Via Ortega, Stanford, CA 94305, e-mail: jose.blanchet@stanford.edu

Lin Chen
Columbia University, 3022 Broadway, New York, NY 10027 e-mail: lc3110@columbia.edu

Jing Dong
Columbia University, 3022 Broadway, New York, NY 10027 e-mail: jing.dong@gsb.columbia.edu

the main modeling tools to capture various non-Markovian features [7, 36, 38]. For example, in contrast to long-range dependence, they are also used to model a phenomenon associated with very rough paths that has rapid short-term fluctuations. This phenomenon is found in volatility of high frequency data [26, 6]. Due to the complicated dependences, limited analytical results are available for sample-path quantities (such as the maxima) of these processes. So, simulation has been a key numerical tool for analysis. The all-time maximum of Gaussian processes arises in communication applications [1], risk management, and analysis of queues [5].

Let $\mathcal{S} = \{S_n : n \in \mathbb{Z}^+\}$ be a discrete-time Gaussian process. We assume $S_0 \equiv 0$, $\mathbb{E}[S_n] = -n\mu$ for some $\mu > 0$, i.e., the process has a negative drift, and

$$\mathrm{Var}(S_n) = \sigma^2 n^{2H} + o(n^{2H}) \text{ for } H \in (0,1) \text{ and } \sigma^2 > 0. \tag{1}$$

Note that $\mathrm{Var}(S_n)$ can grow sublinearly ($H < 1/2$) or superlinearly ($H > 1/2$) in n.

The Gaussian process $\mathcal{S}$ defined above can be fairly general. Let $X_n = S_n - S_{n-1}$ denote the increment of the Gaussian process. In the special case where $\{X_n : n \in \mathbb{Z}^+\}$ is a stationary process with mean $-\mu$ and variance σ^2, we have

$$\mathrm{Var}(S_n) = \sum_{i=1}^{n}\sum_{j=1}^{n} \mathrm{Cov}(X_i, X_j) = \sigma^2 \left(n + 2\sum_{i=1}^{n-1}(n-i)\rho_i \right),$$

where $\rho_i := \mathrm{Corr}(X_n, X_{n+i})$ for $n \in \mathbb{Z}^+$. When $\rho_i = i^{-\alpha}$ for $\alpha \in (0,1)$, $\mathrm{Var}(S_n) = Cn^{2-\alpha} + o(n^{2-\alpha})$ for some $C > 0$. In this case, $H > 1/2$ and $\mathcal{S}$ has long-range dependence. In general, a stationary Gaussian process is said to have long-range dependence or infinite memory if $\lim_{n\to\infty} \mathrm{Var}(S_n)/n = \infty$ [28].

A classic stationary Gaussian process that can be used to model both long-range dependence or rapid short-term fluctuations is fractional Brownian motion (fBM). Let $\boldsymbol{B}^H = \{B^H(t) : t \in \mathbb{R}^+\}$ be an fBM with Hurst index $H \in (0,1)$. $\boldsymbol{B}^H$ is a self-similar Gaussian process with $\mathbb{E}[B^H(t)] = 0$ and covariance function

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

When $H > 1/2$, $\boldsymbol{B}^H$ is a non-Markovian Gaussian process with long-range dependence; when $H = 1/2$, $\boldsymbol{B}^H$ is a standard Brownian motion; when $H < 1/2$, $\boldsymbol{B}^H$ is a non-Markovian Gaussian process with very rough paths (even more rough than Brownian motion). We define an embedded discrete-time process $S_n = -n\mu + \sigma B^H(n)$, which is often referred to as fractional Gaussian noise. Then, $\mathbb{E}[S_n] = -n\mu$ and $\mathrm{Var}(S_n) = \sigma^2 n^{2H}$, which satisfies (1).

In this paper, we are interested in estimating the all-time maximum of a Gaussian process, $\mathcal{S}$, with negative drift, whose variances satisfy (1). The all-time maximum is defined as

$$M_\infty = \max_{n\geq 0} S_n.$$

We develop an algorithm to draw samples from the exact distribution of M_∞. The termination time of the algorithm is random but has finite moments of all orders.

Sampling the all-time maximum of a stochastic process is in general a challenging task. Based on the definition of M_∞, naive simulation would require one to generate an infinite sequence $\{S_n : n \geq 0\}$, which is infeasible. Our algorithm combines the construction of a sequence of stopping times with rare-event simulation techniques. It allows us to identify a non-stopping time T, which is defined as as the time beyond which the running-time maximum $M_T = \max_{0 \leq n \leq T} S_n$ will never be reached again, i.e., $S_n < M_T$ for $n > T$. Then, $M_\infty = M_T$.

Our development builds on three key components 1) a milestone event construction, which is used to define the sequence of stopping times; 2) a rare-event simulation technique for Gaussian processes called Target Bridge Sampling (TBS); and 3) a sandwiching construction to sample Bernoulli random variables whose probability of success cannot be evaluated exactly. These techniques are not new and have found important applications in the simulation literature. However, how to apply these techniques jointly to solve our simulation problem is highly non-trivial.

The idea of milestone events (or record breakers) constructs a sequence of stopping times (corresponding to event times) in a way that enables us to translate the infinite-horizon simulation problem to finding the last finite stopping time in the sequence (see Section 2 for more details). The strategy was first developed in [13] and has seen applications in many subsequent works [10, 9]. The key to successful implementation of this idea is to construct a proper sequence of milestone events which involves understanding the large deviation behavior of the underlying stochastic process. In this work, we are able to apply this idea by analysing the tail behavior of the Gaussian processes.

The milestone events are defined in such a way that it becomes increasingly harder to reach them along the sequence, i.e., the probability that the stopping time is infinity is increasing. To efficiently sample the trajectories leading to the milestone events, we utilize a change of measure induced by TBS. TBS was developed in [12]. It is an importance sampling algorithm not based on exponential tilting, which allows us to circumvent the challenge of tracking the most likely path under non-Markovian structures. In [12], it is applied to estimate $\mathbb{P}(M_\infty > b)$ for large values of b, which is different from our task here: drawing exact samples from M_∞. In addition, the implementation in [12] involves a truncation which induces a bias in the estimator. We circumvent the truncation using a sandwiching construction.

The sandwiching construction is also known as the series method [17]. In our application of the method, we want to generate a Bernoulli random variable whose probability of success p cannot be evaluated exactly. The idea is to approximate p from above and below by sequences of refined bounds. Then, we can sequentially update the bounds to determine whether U is less than p, where U is a Uniform random variable (see Section 3 for more details). When applying the method, the key is to come up with a carefully-designed sequence of bounds. As we will explain in Section 3, due to the general covariance structure, constructing the sequence of upper bounds can be quite involved.

Literature review. There is a growing amount of literature on generating discrete-time Gaussian processes with complex time correlation. The papers [22, 4] develop efficient exact simulation algorithms for Gaussian process with long-range dependence. They come up with techniques to reduce the computational complexity associated with the high-dimensional covariance matrices. These techniques can be applied in our algorithm as well when generating the underlying Gaussian processes. Efficient simulation of fBM and related processes has also attracted continuous attention from the literature [33, 35, 31, 20, 16] (see also [18] for a review).

As explain above, all-time maximum is in general difficult to simulate exactly. For random walks with independent increments, the paper [25] is among the first to use rare-event simulation techniques to draw exact samples of the maximum when the increments are light-tailed distributed. Later extensions include heavy-tailed increments [14] and the maximum over a nonlinear boundary [11]. For Gaussian processes with general covariance structures, rare-event simulation techniques have been applied to estimate the tail probability of M_∞, i.e., $\mathbb{P}(M_\infty > b)$ for large values of b [29, 15, 21, 2, 12]. The paper [19] investigates this tail behavior analytically. Comparing to previous works, substantial new developments are required to draw exact samples from M_∞. Meanwhile, the existing rare-event simulation techniques can be applied as an important intermediate step in our development. The papers [23, 34] develop exact simulation algorithms for max-stable fields at a finite collection of locations, which requires generating the maximum of a Gaussian random field.

Notations. Throughout the paper, we use $\Phi(x)$ to denote the cumulative distribution function (cdf) of standard Gaussian distribution, and denote $\bar{\Phi}(x) := 1 - \Phi(x)$ as the tail cdf. Let $\mathcal{S}_n = (S_1, \ldots, S_n)^\top$. Recall that $\mathcal{S}$ (without the subscript) denotes the whole process. We denote $\mathbf{\Sigma}_n$ as the covariance function of $\mathcal{S}_n$. In particular, $\gamma_{ij} = \mathrm{Cov}(S_i, S_j)$ is the (i, j)-th entry of $\mathbf{\Sigma}_n$. Let $\boldsymbol{U}_{nk} = (\gamma_{1k}, \ldots, \gamma_{nk})^\top$. We also define $\tilde{\mathcal{S}}_n = (S_1 + \mu, S_2 + 2\mu, \ldots, S_n + n\mu)^\top$ as the mean-zero counterpart of $\mathcal{S}_n$. Let $M_n = \max_{1 \le l \le n} S_l$, i.e., the running-time maximum. Last, let $\mathbb{P}_n(\cdot) = \mathbb{P}(\cdot|\mathcal{S}_n)$.

2 Basic strategy

In this section, we introduce the main idea of our algorithm. We start by introducing the milestone-event construction, which allows us to decompose the problem into generating a sequence of downward-crossing and upward-crossing events. We then discuss how to generate these events sequentially.

2.1 Milestone events

An upward-crossing event is an event where the Gaussian process reaches a new maximum. Our goal is to find these upward-crossing events sequentially until we find the last one. To achieve this algorithmically, we also need to define some downward-crossing events.

Given $\mathcal{S}_n$, let

$$q(n) = \sum_{k=n+1}^{\infty} \mathbb{P}_n \left(S_k > M_n\right), \tag{2}$$

which provides an upper bound for the probability of having an upward-crossing event beyond n. We define a sequence of downward-crossing and upward-crossing event times as follows. Let $\tau_0^+ \equiv 0$. For $k \geq 1$, if $\tau_{k-1}^+ < \infty$, define

$$\tau_k^- := \widetilde{\min}\{n > \tau_{k-1}^+ : q(n) < 3/4\}, \quad \tau_k^+ := \inf\left\{n > \tau_k^- : S_n > M_{\tau_k^-}\right\},$$

where $\widetilde{\min}$ is a notation we introduce to denote the caveat that we only require τ_k^- to be a time at which $q(n) < 3/4$. It does not need to be the first time at which this happens. This indicates that τ_k^- is not uniquely defined, which gives us some flexibility when designing the algorithm. For example, we can define τ_k^- as the first time after τ_{k-1}^+ where our algorithm can "detect" that $q(n) < 3/4$.

Note that the downward-crossing event time τ_k^- is defined in such a way that the chance of having an upward-crossing event after τ_k^- will be upper bounded by $3/4$, i.e., $\mathbb{P}_{\tau_k^-}(\tau_k^+ < \infty) \leq q(\tau_k^-) < 3/4$. This indicates that the upward-crossing event only happens a finite number of times. In particular, let $K = \sup\{k : \tau_k^+ < \infty\}$, i.e., the number of upward-crossing events. Then, K is stochastically bounded by a Geometric random variable with probability of success $1/4$.

Once we find K, $M_\infty = \max_{0 \leq n \leq \tau_{K+1}^-} S_n$. Thus, by generating the Gaussian process $\mathcal{S}$ up to the random time τ_{K+1}^-, we are able to recover the all-time maximum of $\mathcal{S}$. We also remark that finding K or τ_{K+1}^- is not straightforward since τ_{K+1}^- is not a stopping time, i.e., finding K requires knowing information beyond $\mathcal{S}_{\tau_K^+}$.

The next lemma shows that the downward-crossing events are well-defined.

Lemma 1 *For any fixed $a \in (0,1)$, there exists a random integer L, such that for any $n > L$, $q(n) < a$. Moreover, for any $\eta > 0$, $\mathbb{E}[L^\eta] < \infty$.*

The proof of Lemma 1 and all the subsequent lemmas are delayed to the appendix.

The next theorem shows that τ_{K+1}^- is well defined. In particular, even though τ_{K+1}^- is random, it has finite moments of all orders.

Theorem 1 *For any $\eta > 0$, $\mathbb{E}[(\tau_{K+1}^-)^\eta] < \infty$.*

The proof of Theorem 1 is based on a construction argument and is provided in Section 4.2.1 as part of the complexity analysis of Algorithm 2.

2.2 Main algorithm

Based on the milestone event construction, we next explain how to find these events sequentially.

Downward-crossing events. Finding the downward-crossing events can be down under the nominal measure. The main difficulty is to check whether $q(n) < 3/4$, because $q(n)$ involves an infinite sum which cannot be calculated exactly. To overcome the difficulty, we use a sandwiching construction. We derive a sequence of upper and lower bounds, $\{\mathcal{U}(\ell)\}_{\ell\geq 1}$ and $\{\mathcal{L}(\ell)\}_{\ell\geq 1}$ satisfying

$$\mathcal{L}(1) \leq \mathcal{L}(2) \leq \cdots \leq q(n) \leq \cdots \leq \mathcal{U}(2) \leq \mathcal{U}(1)$$

and $\lim_{\ell\to\infty}\mathcal{U}(\ell) = \lim_{\ell\to\infty}\mathcal{L}(\ell) = q(n)$. Then, we can sequentially refine the bounds until the stopping criterion — $\mathcal{U}(\ell) < 3/4$ or $\mathcal{L}(\ell) > 3/4$ — is met to determine whether $q(n) < 3/4$. The technical and algorithmic details are provided in Section 3 and Algorithm 3.

Upward-crossing events. Sampling the upward-crossing event is more challenging. In particular, since $\mathbb{P}(\tau_k^+ < \infty | \mathcal{S}_{\tau_k^-}, \tau_k^-) < 1$, given τ_k^- and $\mathcal{S}_{\tau_k^-}$, if we generate the Gaussian process under the nominal measure until $S_n > M_{\tau_k^-}$, we may never be able to find τ_k^+, i.e., the algorithm can take an infinite amount of time. To overcome this challenge, we employ a rare-event simulation technique for Gaussian processes called TBS [12]. We comment that there are many candidate rare-event simulation techniques. We choose TBS because it is especially well-suited for Gaussian processes with general covariance structures. The implementation contains two key components: a change-of-measure and an acceptance-rejection step.

First, given $\mathcal{S}_n$, we introduce a new measure $\mathbb{Q}_n$ under which the upward-crossing event happens with probability 1. In particular, given $\mathcal{S}_n$, define

$$\kappa_n := \inf\{k > n : S_k > M_n\}.$$

Then, the new measure is defined through the following likelihood ratio

$$\frac{d\mathbb{P}_n}{d\mathbb{Q}_n} 1\{\kappa_n < \infty\} = \frac{\sum_{\ell=n+1}^{\infty} \mathbb{P}_n(S_\ell > M_n)}{\sum_{m=\kappa_n}^{\infty} \mathbb{P}_{\kappa_n}(S_m > M_n)}. \tag{3}$$

If we simulate $S_{n+1}, ..., S_{\kappa_n}$ under $\mathbb{Q}_n$, we have a proposed upward-crossing path. To algorithmically achieve this, we use TBS defined in Algorithm1. The idea is to first sample a target upward-crossing time, and then use Gaussian bridge to sample the process conditional on the upward-crossing event at the target time.

Algorithm 1 Target Bridge Sampling (Given $\mathcal{S}_n$)

1. Sample $N(n) > n$ with probability mass function

$$f_n(m) = \frac{\mathbb{P}_n(S_m > M_n)}{\sum_{k=n+1}^{\infty} \mathbb{P}_n(S_k > M_n)} = \frac{\mathbb{P}_n(S_m > M_n)}{q(n)}, \text{ for } m = n+1, \ldots. \tag{4}$$

2. Given $N(n) = m$ and $\mathcal{S}_n$, sample S_m conditional on $S_m \geq M_n$.
3. Conditional on S_m and $\mathcal{S}_n$, sample $S_{n+1}, \ldots, S_{m-1}$. Calculate $\kappa_n = \min\{k > n : S_k > M_n\}$.
4. Output $S_{n+1}, \ldots, S_{\kappa_n}$.

Lemma 2 verifies that the probability measure induced by TBS is our target measure $\mathbb{Q}_n$. We comment that there remains an implementation challenge — how to sample $N(n)$, since $q(n)$ cannot be evaluated exactly. We address this challenge using a sandwiching construction in Algorithm 4 in Section 3.

Lemma 2 *The probability measure induced by TBS satisfies* (3).

Second, we apply an acceptance-rejection step. In particular, given the proposed path $S_{n+1},...,S_{\kappa_n}$, we sample a Bernoulli random variable I with probability of success

$$p(\kappa_n) = \left(\sum_{m=\kappa_n}^{\infty} \mathbb{P}_{\kappa_n}(S_m > M_n)\right) = \left(1 + \sum_{m=\kappa_n+1}^{\infty} \mathbb{P}_{\kappa_n}(S_m > M_n)\right)^{-1} \leq 1. \tag{5}$$

If the Bernoulli is a success, i.e., $I = 1$, the proposed path is accepted and it is the path leading to the next upward-crossing event as verified by the following lemma:

Lemma 3 *Given $\mathcal{S}_n$, for $S_{n+1},...,S_{\kappa_n}$ generated under $\mathbb{Q}_n$, let I denote a Bernoulli random variable with probability of success $p(\kappa_n)$. Then*

$$\mathbb{Q}_n(I = 1) = \frac{\mathbb{P}_n(\kappa_n < \infty)}{\sum_{l=n+1}^{\infty} \mathbb{P}_n(S_l > M_n)} = \frac{\mathbb{P}_n(\kappa_n < \infty)}{q(n)}$$

and $\mathbb{Q}_n((S_{n+1},...,S_k) \in \cdot, \kappa_n = k | I = 1) = \mathbb{P}_n((S_{n+1},...,S_k) \in \cdot, \kappa_n = k | \kappa_n < \infty)$.

We note from Lemma 3 that $\mathbb{Q}_n(I = 1) = \mathbb{P}_n(\kappa_n < \infty)/q(n) > \mathbb{P}_n(\kappa_n < \infty)$. Thus, given $\tau_k^- = n$ and $\mathcal{S}_n$, we generate another independent Bernoulli random variable J with probability of success $q(n)$. Note that

$$\mathbb{Q}_n(J = 1, I = 1) = q(n)\frac{\mathbb{P}_n(\kappa_n < \infty)}{q(n)} = \mathbb{P}_n(\kappa_n < \infty) = \mathbb{P}_n(\tau_k^+ < \infty),$$

$$\begin{aligned} &\mathbb{Q}_n((S_{n+1},...,S_k) \in \cdot, \kappa_n = k | I = 1, J = 1) \\ &= \mathbb{Q}_n((S_{n+1},...,S_k) \in \cdot, \kappa_n = k | I = 1) \text{ by independence} \\ &= \mathbb{P}_n((S_{n+1},...,S_k) \in \cdot, \kappa_n = k | \kappa_n < \infty) \text{ by Lemma 3.} \end{aligned} \tag{6}$$

Meanwhile, we also note that

$$\begin{aligned} \mathbb{Q}_n(J = 0) + \mathbb{Q}_n(J = 1, I = 0) &= 1 - q(n) + q(n)\left(1 - \frac{\mathbb{P}_n(\kappa_n < \infty)}{q(n)}\right) \\ &= \mathbb{P}_n(\kappa_n = \infty) = \mathbb{P}_n(\tau_k^+ = \infty). \end{aligned} \tag{7}$$

This indicates that to determine whether $\tau_k^+ < \infty$, we first sample J. If $J = 0$, we can claim that $\tau_k^+ = \infty$. If $J = 1$, we further apply $\mathbb{Q}_n$ to sample a proposed path and sample I. If $I = 1$, we accept the proposed path as the path leading to the next upward-crossing event. If $I = 0$, we can claim that $\tau_k^+ = \infty$. We remark that sampling J and I is not straightforward, as $q(n)$ and $p(\kappa_n)$ can not be evaluated exactly. We will explain how to do that using a sandwiching construction in Algorithms 4 and 5 in Section 3.

We conclude the section by summarizing the ideas discussed above and presenting the main simulation algorithm — Algorithm 2.

Algorithm 2 Simulating the all-time maximum of $\mathcal{S}$

Step 0: Initialization.

1. Set $k = 0$, $\tau_k^+ = 0$, $n = 0$, and $S_0 = 0$.

Step 1: Downward-crossing event.

1. Sample S_{n+1} conditional on $\mathcal{S}_n$.
2. Call Algorithm 3 to sample $W \in \{0, 1\}$. If $W = 0$, set $n = n + 1$ and go back to **Step 1.1**. If $W = 1$, go to **Step 1.3**.
3. Set $n = \tau_k^-$ and $M_n = \max_{1 \le l \le n} S_l$.

Step 2: Upward-crossing event.

1. Call Algorithm 4 to sample $J \sim \text{Bernoulli}(q(n))$. If $J = 0$, go to **Step 3**. If $J = 1$, Algorithm 4 also outputs a random time $N \sim f_n(\cdot)$.
2. Given N, sample S_N according to $\mathbb{P}_n(S_N \in \cdot | S_N > M_n)$.
3. Conditional on S_N and $\mathcal{S}_n$, sample $S_{n+1}, \dots, S_{N-1}$. Calculate $\kappa_n = \min\{l \ge n : S_l > M_n\}$.
4. Call Algorithm 5 to sample I ~Bernoulli($p(\kappa_n)$). If $I = 1$, set $k = k + 1$, $\tau_k^+ = \kappa_n$, $n = \tau_k^+$ and go to **Step 1**. If $I = 0$, go to **Step 3**.

Step 3: Output.

1. Output $M_\infty = M_n$. (We can also output $\mathcal{S}_n$)

Theorem 2 *The output of Algorithm 2 follows the same distribution as M_∞.*

Before we prove Theorem 2, we need to introduce the details of the intermediate steps in Algorithm 2. Thus, the proof is delayed to Section 4.1.

3 Intermediate Steps in Algorithm 2

In this section, we present the details of the intermediate steps in Algorithm 2. In particular, we introduce Algorithms 3 – 5.

The fundamental challenge in these algorithms is that we need to compare a number, say u, to a probability p that cannot be evaluated exactly. We resolve this challenge by deriving a sequence of upper and lower bounds for p.

Given $\mathcal{S}_n$, recall from (2) that $q(n) = \sum_{k=n+1}^{\infty} \mathbb{P}_n(S_k > M_n)$. We define the lower bounds by simply truncating the infinite sum to a finite number of terms:

$$q(n, \ell) := \sum_{k=n+1}^{\ell} \mathbb{P}_n(S_k > M_n)$$

The upper bounds are more challenging to construct. Define

$$h(\ell) = \frac{8\sigma^2}{(1-H)\mu^2} \exp\left(-\frac{\mu^2}{16\sigma^2}\ell^{2-2H}\right), \tag{8}$$

$$B(n) = \max\left\{\left(\frac{2\sigma^2 n^H \|\mathbf{\Sigma}_n^{-1}\|_1 \|\tilde{S}_n\|_1}{\mu}\right)^{\frac{1}{1-H}}, \left(\frac{2\sigma^2}{\pi\mu^2}\right)^{\frac{1}{2(1-H)}}, \left(\frac{2H-1}{1-H}\frac{16\sigma^2}{\mu^2}\right)^2, n+1\right\}. \tag{9}$$

Lemma 4 *For any* $\eta > 0$, $\mathbb{E}[B(n)^\eta] < \infty$. *For any* $\ell \geq B(n)$,

$$q(n,\ell) < q(n) \leq q(n,\ell) + h(\ell).$$

In addition, $q(n,\ell) \leq q(n,\ell+1) \leq \cdots \leq q(n) \leq \cdots \leq q(n,\ell+1) + h(\ell+1) \leq q(n,\ell) + h(\ell)$ *and* $\lim_{\ell\to\infty} q(n,\ell) = \lim_{\ell\to\infty} q(n,\ell) + h(\ell) = q(n)$.

Based on Lemma 4, we have constructed a proper sequence of lower and upper bounds for $q(n)$. These bounds allow us to check whether we have reached a downward-crossing event in Algorithm 3 and to sample J ~Bernoulli($q(n)$) in Algorithm 4.

Algorithm 3 Given n and $\mathcal{S}_n$, output W where $W = 1$ implies $q(n) < 3/4$.

---

0. Calculate $B(n)$ and set $\ell = \lceil B(n) \rceil$.
1. Sample U_0 ~Uniform$[1/2, 3/4]$.
2. Calculate $\mathcal{L}(\ell) = q(n,\ell)$ and $\mathcal{U}(\ell) = q(n,\ell) + h(\ell)$.
3. If $U_0 \leq \mathcal{L}(\ell)$, set $W = 0$, and go to **Step 4**; if $U_0 \geq \mathcal{U}(\ell)$, set $W = 1$ and go to **Step 4**; otherwise, set $\ell = \ell + 1$ and go to **Step 2**.

4 Output W.

In Algorithm 3, we do not compare $q(n)$ to $3/4$ directly. Instead, we compare $q(n)$ to a uniform random variable on $[1/2, 3/4]$: U_0. In this case, $W = 1$ implies that $U_0 > q(n)$, which further implies that $q(n) < 3/4$, i.e., the criteria for the downward-crossing event is met.

Algorithm 4 Given n and $\mathcal{S}_n$, sample J ~Bernoulli($q(n)$).

---

0. Calculate $B(n)$ and set $\ell = \lceil B(n) \rceil$.
1. Sample U ~Uniform[0, 1].
2. If $U \leq q(n,\ell)$, set $J = 1$ and $\ell = \min\{h \geq n+1 : q(n, h-1) < U \leq q(n,h)\}$, and go to **Step 5**; otherwise, set $\ell = \ell + 1$ and go to **Step 3**.
3. Calculate $\mathcal{L}(\ell) = q(n,\ell)$ and $\mathcal{U}(\ell) = q(n,\ell) + h(\ell)$.
4. If $U \leq \mathcal{L}(\ell)$, set $J = 1$, $N = \ell$, and go to **Step 5**; if $U \geq \mathcal{U}(\ell)$, set $J = 0$ and go to **Step 5**; otherwise, set $\ell = \ell + 1$ and go to **Step 3**.
5. If $J = 1$, output J and $N = \ell$; otherwise, output J

In Algorithm 4, when $J = 1$, it also outputs $N = \ell$. To see this, note that

$$\mathbb{P}_n(N=\ell|J=1)=\frac{\mathbb{P}_n(\mathcal{L}(\ell-1)<U<\mathcal{L}(\ell))}{P(U<q(n))}=\frac{\mathbb{P}_n(S_\ell>M_n)}{\sum_{k=n+1}^{\infty}\mathbb{P}_n(S_k>M_n)}=f_n(\ell). \tag{10}$$

Thus, as a byproduct of Algorithm 4, we also get a sample of $N\sim f_n(\cdot)$.

Lastly, given κ_n and $\mathcal{S}_{\kappa_n}$, we develop an algorithm to sample Bernoulli$(p(\kappa_n))$ (Algorithm 5). Recall from (5) that $p(\kappa_n)=(\sum_{\ell=\kappa_n}^{\infty}\mathbb{P}_{\kappa_n}(S_\ell>M_n))^{-1}$. Define

$$\tilde{q}(n,\ell):=\sum_{i=k+1}^{\ell}\mathbb{P}_{\kappa_n}(S_i>M_n).$$

Then, we have the following analog of Lemma 4.

Lemma 5 *For $\ell\geq B(\kappa_n)$, where $B(k)$ is defined in* (9)*,*

$$(1+\tilde{q}(n,\ell)+h(\ell))^{-1}\leq p(\kappa_n)\leq(1+\tilde{q}(n,\ell))^{-1},$$

where $h(\ell)$ is defined in (8)*. In addition, $(1+\tilde{q}(n,\ell)+h(\ell))^{-1}\leq(1+\tilde{q}(n,\ell+1)+h(\ell+1))^{-1}\leq\cdots\leq p(\kappa_n)\leq\cdots\leq(1+\tilde{q}(n,\ell+1))^{-1}\leq(1+\tilde{q}(n,\ell))^{-1}$ and $\lim_{\ell\to\infty}(1+\tilde{q}(n,\ell))^{-1}=\lim_{\ell\to\infty}(1+\tilde{q}(n,\ell)+h(\ell))^{-1}=p(\kappa_n)$.*

Then, Algorithm 5 follows based on the sequence of bounds in Lemma 5.

Algorithm 5 Given κ_n and $\mathcal{S}_{\kappa_n}$, sample I ~Bernoulli$(p(\kappa_n))$.

0. Calculate $B(\kappa_n)$ and set $\ell=\lceil B(\kappa_n)\rceil$.
1. Sample U ~Uniform[0, 1].
2. Calculate $\mathcal{L}(\ell)=(1+\tilde{q}(n,\ell)+h(\ell))^{-1}$ and $\mathcal{U}(\ell)=(1+\tilde{q}(n,\ell))^{-1}$.
3. If $U\leq\mathcal{L}(\ell)$, set $I=1$ and go to **Step 4**; if $U\geq\mathcal{U}(\ell)$, set $I=0$ and go to **Step 4**; otherwise, set $\ell=\ell+1$ and go to **Step 2**.
4. Output I

4 Analysis of Algorithm 2

In this section, we provide detailed analysis to verify the correctness and complexity of Algorithm 2. In particular, we provide the proof of Theorems 1 and 2.

4.1 Output analysis (Proof of Theorem 2)

Proof In Step 1 of Algorithm 2, we simulate $\mathcal{S}$ under the nominal measure.

For Step 2, we first verify the output of Algorithms 4. Because $\lim_{\ell\to\infty}q(n,\ell)+h(\ell)=q(n)$ by Lemma 4, we have $\mathbb{P}_n(J=1)=\mathbb{P}_n(U<q(n))=q(n)$. In addition, from (10), we have $\mathbb{P}_n(N=\ell|J=1)=f_n(\ell)$. Therefore, in Step 2.1, if $J=1$,

$N \sim f_n(\cdot)$. Then, Steps 2.1 - 2.3 constitute the TBS procedure, i.e., $S_{n+1}, \ldots, S_{\kappa_n}$ in Step 2.3 is a sample path drawn under $\mathbb{Q}_n$. We next verify the output of Algorithm 5. Since $\lim_{\ell\to\infty}(1 + \tilde{q}(n, \ell) + h(\ell))^{-1} = p(\kappa_n)$ by Lemma 5, $\mathbb{Q}_n(\ell = 1) = \mathbb{Q}_n(U < p(\kappa_n)) = p(\kappa_n)$. Therefore, Step 2.4 is the acceptance-rejection step. From (6),

$$\mathbb{Q}_n((S_{n+1}, ..., S_k) \in \cdot, \kappa_n = k | I = 1, J = 1) = \mathbb{P}_n((S_{n+1}, ..., S_k) \in \cdot, \kappa_n = k | \kappa_n < \infty).$$

Thus, if we accept the path, it is is the path leading to the next upward-crossing event. Meanwhile, from (7), $\mathbb{Q}_n(J = 0) + \mathbb{Q}_n(J = 1, I = 0) = \mathbb{P}_n(\tau_k^+ = \infty)$. Thus, if we go to Step 3, there is no more record breakers, i.e., $n = \kappa_{K+1}^-$. □

4.2 Complexity analysis

In this section, we conduct detailed complexity analysis of Algorithm 2. Note that the computational cost of Algorithm 2 is random due the random length of the sample path, i.e., τ_{K+1}^-, and the random number of iterations in Algorithms 3 – 5. We will show that these random quantities have finite moments of all orders.

Let $C(i)$ denote the computational complexity of generating the i-th sample of M_∞ from Algorithm 2, $M_\infty(i)$. Let $\mathcal{N}(c)$ denote the total number of $M_\infty(i)$'s generated with a computational budget of c, i.e., $\mathcal{N}(c) := \max\{n \geq 0 : C(1) + \cdots + C(n) \leq c\}$. Then, if $\mathbb{E}[C(1)] < \infty$, which is the case in our setting, and $\mathrm{Var}(M_\infty(1)) < \infty$, we achieve the canonical $\sqrt{c}$ rate of convergence [27], i.e.,

$$\sqrt{c}\left(\sum_{i=1}^{\mathcal{N}(c)} M_\infty(i) - \mathbb{E}[M_\infty(1)]\right) \Rightarrow \sqrt{\mathrm{e}[C(1)]\mathrm{Var}(M_\infty(1))}\boldsymbol{N}(0,1) \text{ as } c \to \infty,$$

where $\boldsymbol{N}(0,1)$ denote a Gaussian distribution with mean 0 and variance 1. It is also important to study how the complexity, e.g., $\mathbb{E}[C(1)]$, depends on the covariance structure. We investigate this through numerical experiments in Section 5.

4.2.1 Proof of Theorem 1

Before we prove Theorem 1, we first introduce an auxiliary lemma.

Lemma 6 *Given $\mathcal{S}_n$, for $N(n)$ and J generated in Step 2.1 in Algorithm 2, we have for any $\eta > 0$, $\mathbb{E}[N(n)^\eta 1\{J = 1\} | \mathcal{S}_n] < \infty$.*

Proof of Theorem 1 Let $\tilde{T} = \tau_{K+1}^-$, which is the length of the Gaussian process generated in Algorithm 2.

First, let G denote number of times we visit Step 1 in Algorithm 2, which is the same as the number of times we visit Step 2. We note that when we visit Step 2.1 in Algorithm 2, if $J = 0$, the algorithm terminates. By the construction of downward-crossing event, $q(n) < 3/4$ when in Step 2.1. Thus, G is stochastically upper bounded by a Geometric random variable with probability of success $1/4$.

Second, we study the number of elements of the Gaussian process generated in Step 1. If we set $a = 1/2$ in Lemma 1, then there exists a random time, $\tilde{L}$, which has finite moments of all orders, such that for $n > \tilde{L}$, $q(n) < 1/2$. When $q(n) < 1/2$, $W = 1$ in Algorithm 3 with probability 1. This suggests that when $n > \tilde{L}$, every time we go to Step 1, we only generate one more point of the Gaussian process, i.e., $\tau_{k+1}^- = \tau_k^+ + 1$. Thus, the number of elements of the Gaussian process we generate in Step 1 is upper bounded by $\tilde{L} + G$, which has finite moments of all orders.

Third, we study the number of elements of the Gaussian process we generate in Step 2 in Algorithm 2. When we visit Step 2 at $\tau_k^- = n$, if $J = 1$ in Step 2.1, we generate a path leading to the next upward-crossing event. The number of elements of the Gaussian process we generate is upper bounded by $N(n) - n$. By Lemma 6, $N(n)$ has finite moments of all orders. Then, the number of elements of the Gaussian process we generate in Step 2 is upper bounded by $\sum_{k=1}^{G} N(\kappa_k^-) - \kappa_k^-$, which has finite moments of all orders. □

4.2.2 Complexity of Algorithms 3 – 5

Based on the sandwiching construction, in Algorithms 3 – 5 we compare a uniform random variable with an unknown probability p using the iteratively updated bounds for p. We present the analysis for Algorithm 4 next.

Given $\mathcal{S}_n$, let $\Theta - n$ denote the number of iterations in Algorithm 4:

$$\Theta := \inf\{l > n : U \le q(n,l) \text{ or } U \ge q(n,l) + h(l)\},$$

where U ~Uniform$[0,1]$. Then, for any $l > B(n)$,

$$\mathbb{P}_n(\Theta > l) = \mathbb{P}_n(q(n,l) < U < q(n,l) + h(l)) = h(l).$$

We first note that for any $\eta > 0$, $\sum_{l=1}^{\infty} l^{\eta} h(l) < \infty$. Next, from Lemma 4, $\mathbb{E}[B(n)^{\eta}] < \infty$. Thus, Θ has finite moments of all orders.

Similarly, we can show that the number of iterations in Algorithm 3 and 5 also has finite moments of all orders. Note that for the sandwiching construction in Algorithm 5, we have $(1 + q_b(n,l))^{-1} - (1 + q_b(n,l) + h(l))^{-1} \le h(l)$.

5 Numerical experiments and final discussions

In this section, we implement our algorithm and test its performance based on fractional Gaussian noise. This complements our complexity analysis in Section 4.2. Consider $S_n = -n\mu + B^H(n)$ where B^H is an fBM with Hurst index $H \in (0,1)$. We set $\mu = 1$ and use three different values of H: $H = 0.3, 0.5$, and 0.7, corresponding to the cases where X_n's are negatively , un-, and positively correlated, respectively.

In Figure 1, we show the distribution of M_∞ based on 10^3 independent copies of it, i.e. we repeat Algorithm 2 10^3 times for each model. On the left panel, we plot the histogram of M_∞. We note that there is a very high probability that $M_\infty = 0$. On the right panel, we plot the histogram of M_∞ conditional on $M_\infty > 0$. We observe that as H increases, the tail of the distribution of M_∞ becomes heavier. In particular, as H increases, M_∞ is more likely to take very large values.

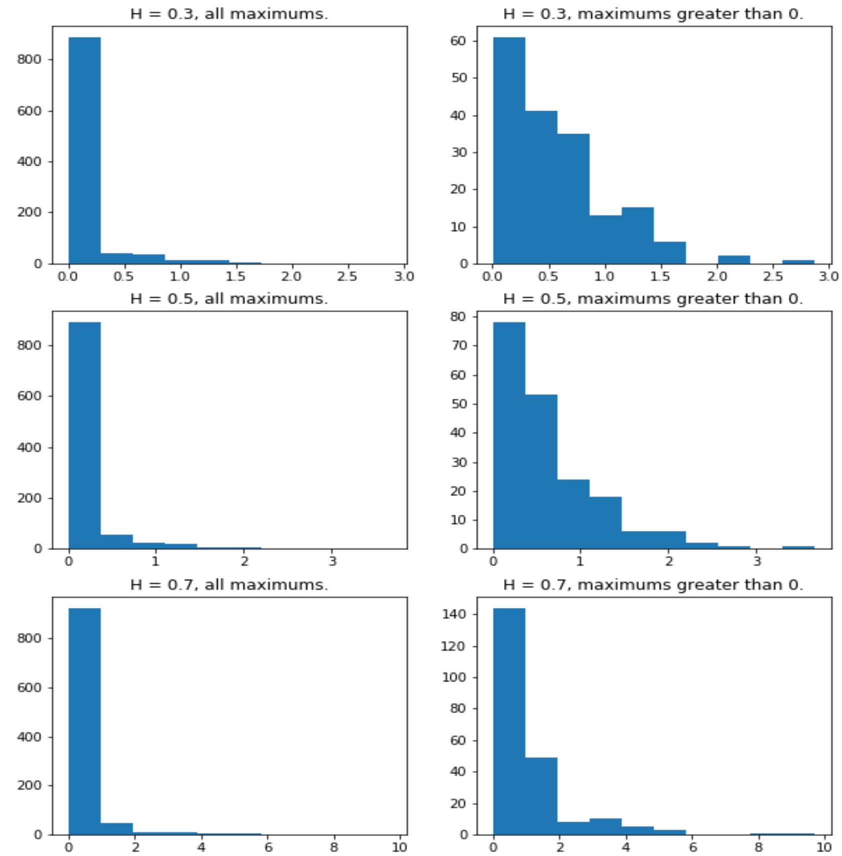

Fig. 1 Histograms of M_∞ where $S_n = -n + B^H(n)$ and $H = 0.3, 0.5$, or 0.7.

We next look at the complexity of our algorithm. We analyze two quantities: 1) the length of the sample path generated in Algorithm 2, which we denote as $\tilde{T}$ and 2) the number of iterations in Algorithm 4, which we denote as Θ_m. We note that even though $\tilde{T}$ is a natural measure of the complexity, in actual implementations, the most time-consuming part is the intermediate step – Algorithm 4, i.e., sampling a Bernoulli random variable with probability of success $q(n)$. In Figure 2, we plot the histogram of $\tilde{T}$ (left panel) and Θ_m (right panel) based on 10^3 independent replications of Algorithm 2. We observe that as H increases, $\tilde{T}$ tends to take larger values. More importantly, as H increases, the tail of the distribution of Θ_m becomes heavier very rapidly. For example, when $H = 0.3$, Θ_m tends to take very small values, i.e., less than 14. However, when $H = 0.7$, the distribution of Θ_m has an extremely heavy tail. In our sample, Θ_m can be as large as 5×10^5.

Based on the numerical experiments, we note that as H increase, the computational complexity increases. Indeed, when $H \geq 0.9$ in our fractional Gaussian noise example, Θ_m in Algorithm 4 can be too large to handle computationally, e.g, 10^{17}. Note that for $\ell > B(n)$,

$$\mathbb{P}_n(\Theta_m > \ell) = h(\ell) = \frac{8\sigma^2}{(1-H)\mu^2} \exp\left(-\frac{\mu^2}{16\sigma^2}\ell^{2-2H}\right),$$

which can decay very slowly when H is close to 1. Even $B(n)$ (defined in (9)) can be very large in this case. How to sample M_∞ when H is close to 1 in a computationally efficient way would require fundamentally new developments, which is an interesting future research direction.

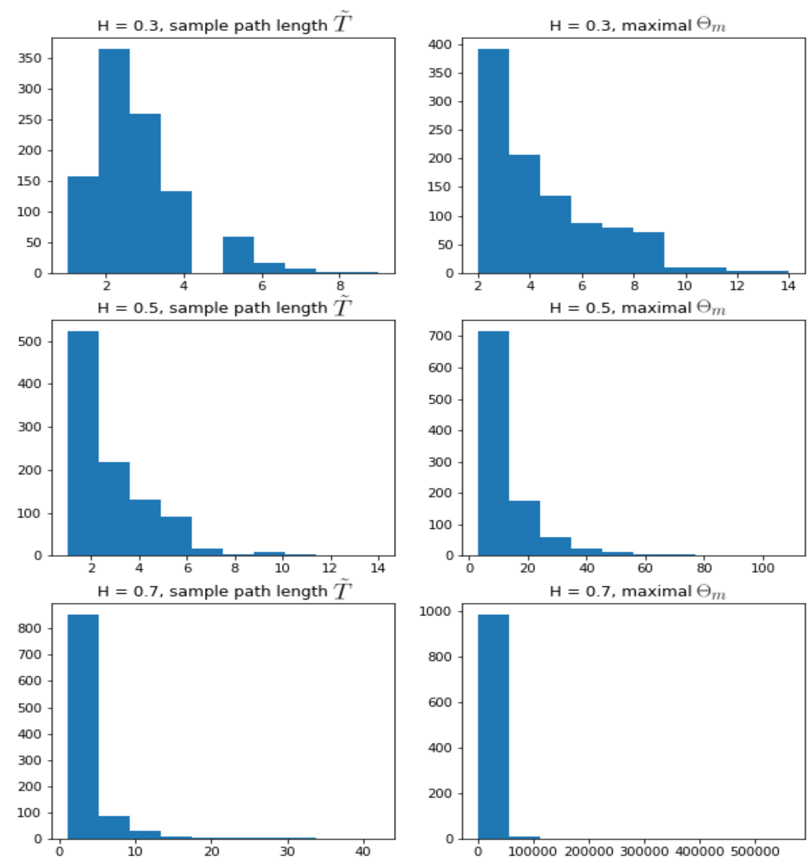

Fig. 2 Histograms of $\tilde{T}$ and Θ_m. $S_n = -n + B^H(n)$ and $H = 0.3, 0.5$, or 0.7.

Appendix. Proof of the technical lemmas

Proof of Lemma 1 Note that S_k conditional on $\mathcal{S}_n$ is still a Gaussian random variable with conditional mean

$$\mu_n(k) = \mathbb{E}[S_k|\mathcal{S}_n] = -k\mu + \boldsymbol{U}_{nk}^\top \boldsymbol{\Sigma}_n^{-1}\tilde{\mathcal{S}}_n,$$

and conditional variance

$$\sigma_n(k)^2 = \mathrm{Var}[S_k|\mathcal{S}_n] = \sigma^2 k^{2H} - \boldsymbol{U}_{nk}^\top \boldsymbol{\Sigma}_n^{-1}\boldsymbol{U}_{nk}.$$

The proof of the lemma is divided into three steps. We first establish bounds for the conditional mean $\mu_n(k)$. Let $\tilde{\mu}_n(k) = \boldsymbol{U}_{nk}^\top \boldsymbol{\Sigma}_n^{-1}\tilde{\mathcal{S}}_n$. As $\tilde{\mu}_n(k)$ is a linear combination of $\tilde{\mathcal{S}}_n$, it follows a Normal distribution with mean 0 and variance $\boldsymbol{U}_{nk}^\top \boldsymbol{\Sigma}_n^{-1}\boldsymbol{U}_{nk}$. By the law of total variance, $\boldsymbol{U}_{nk}^\top \boldsymbol{\Sigma}_n^{-1}\boldsymbol{U}_{nk} < \sigma^2 k^{2H}$. In this case, for any fixed $\delta \in (0,\mu)$,

$$\mathbb{P}(\tilde{\mu}_n(k) > \delta k) \le \mathbb{P}\left(\frac{\tilde{\mu}_n(k)}{\sqrt{\boldsymbol{U}_{nk}^\top \boldsymbol{\Sigma}_n^{-1}\boldsymbol{U}_{nk}}} > \frac{\delta k}{\sigma k^H}\right) = \bar{\Phi}\left(\frac{\delta}{\sigma}k^{1-H}\right). \tag{11}$$

Then,

$$\sum_{n=1}^{\infty}\sum_{k=n}^{\infty}\mathbb{P}(\tilde{\mu}_n(k) > \delta k) = \sum_{k=1}^{\infty}\sum_{n=1}^{k}\mathbb{P}(\tilde{\mu}_n(k) > \delta k) \le \sum_{k=1}^{\infty} k\bar{\Phi}\left(\frac{\delta}{\sigma}k^{1-H}\right) < \infty.$$

By Borel-Cantelli Lemma, there exits a random number $L_0 \ge n$, which is finite almost surely, such that when $k > L_0$, $\tilde{\mu}_n(k) \le \delta k$, which further implies that $\mu_n(k) \le -(\mu-\delta)k$.

We next establish bounds for $\sum_{n=1}^\infty q(n)$. For $k > L_0$, we have $\mu_n(k) \le -(\mu-\delta)k$ and $\sigma_n(k)^2 \le \sigma^2 k^{2H}$. Thus, for any $b \ge 0$,

$$\mathbb{P}_n(S_k > b) \le \mathbb{P}_n\left(\frac{S_k - \mu_n(k)}{\sigma_n(k)} > \frac{b + (\mu-\delta)k}{\sigma k^H}\right) \le \bar{\Phi}\left(\frac{\mu-\delta}{\sigma}k^{1-H}\right). \tag{12}$$

Based on the analysis above, let $b = \max_{1\le l\le n} S_l$. We decompose $\sum_{n=1}^\infty q(n)$ into three parts:

$$\sum_{n=1}^{\infty}\sum_{k=n}^{\infty}\mathbb{P}_n(S_k > b) \le \underbrace{\sum_{n=1}^{L_0}\sum_{k=n}^{L_0}\mathbb{P}_n(S_k > b)}_{\text{(I)}} + \underbrace{\sum_{n=1}^{L_0}\sum_{k=L_0}^{\infty}\mathbb{P}_n(S_k > b)}_{\text{(II)}} + \underbrace{\sum_{n=L_0}^{\infty}\sum_{k=n}^{\infty}\mathbb{P}_n(S_k > b)}_{\text{(III)}}.$$

Part (I) only involves a finite number of terms. For part (II), from (12), we have

$$(\mathrm{II}) \leq L_0 \sum_{k=L_0}^{\infty} \bar{\Phi}\left(\frac{\mu-\delta}{\sigma} k^{1-H}\right) < \infty.$$

Similarly, for part (III), from (12), we have

$$(\mathrm{III}) = \sum_{k=L_0}^{\infty} \sum_{n=L_0}^{k} \mathbb{P}_n(S_k > b) \leq \sum_{k=L_0}^{\infty} (k-L_0)\bar{\Phi}\left(\frac{\mu-\delta}{\sigma} k^{1-H}\right) < \infty.$$

Putting parts (I)–(III) together, we have $\sum_{n=1}^{\infty} q(n) < \infty$. By Borell-Cantelli Lemma, there exits L, which is finite almost surely, such that for any $n > L$, $q(n) < a$.

Lastly, we show that $\mathbb{E}[L^{\eta}] < \infty$ for any $\eta > 0$. Let L_1 denote a large enough constant, such that $\sum_{k=L_1}^{\infty} \bar{\Phi}\left(\frac{\mu-\delta}{\sigma} k^{1-H}\right) < a$. Then, $L \leq \max\{L_0, L_1\}$. Thus, to prove $\mathbb{E}[L^{\eta}] < \infty$, we only need to show that $\mathbb{E}[L_0^{\eta}] < \infty$. Define $\mathcal{A}_n = \bigcup_{k=n}^{\infty}\{\tilde{\mu}_n(k) > \delta k\}$. Then $L_0^{\eta} \leq \sum_{n=1}^{\infty} 1\{\mathcal{A}_n\} n^{\eta}$, and

$$\begin{aligned}
\mathbb{E}[L_0^{\eta}] \leq \mathbb{E}\left[\sum_{n=1}^{\infty} 1\{\mathcal{A}_n\} n^{\eta}\right] &= \sum_{n=1}^{\infty}\sum_{k=n}^{\infty} \mathbb{P}(\tilde{\mu}_n(k) > \delta k) n^{\eta} \\
&= \sum_{k=1}^{\infty}\sum_{n=1}^{k} n^{\eta}\mathbb{P}(\tilde{\mu}_n(k) > \delta k) \leq \sum_{k=1}^{\infty} k^{\eta}\bar{\Phi}\left(\frac{\delta}{\sigma} k^{1-H}\right) < \infty,
\end{aligned}$$

where the last inequality follows from (11). □

Proof of Lemma 2 With a little abuse of notation, we denote $\mathbb{Q}_n$ as the measure induced by the TBS procedure. First note that

$$\begin{aligned}
&\mathbb{Q}_n((S_{n+1},...,S_k) \in \cdot, \kappa_n = k) = \sum_{m=n+1}^{\infty} f_n(m)\mathbb{P}_n((S_{n+1},...,S_k) \in \cdot, \kappa_n = k | S_m > b) \\
&= \sum_{m=n+1}^{\infty} f_n(m) \frac{\mathbb{P}_n((S_{n+1},...,S_k) \in \cdot, \kappa_n = k, S_m > b)}{\mathbb{P}_n(S_m > b)} \\
&= \sum_{m=n+1}^{\infty} \frac{\mathbb{P}_n((S_{n+1},...,S_k) \in \cdot, \kappa_n = k, S_m > b)}{\sum_{\ell=n+1}^{\infty} \mathbb{P}_n(S_{\ell} > b)} \\
&= \sum_{m=n+1}^{\infty} \mathbb{P}_n((S_{n+1},...,S_k) \in \cdot, \kappa_n = k) \frac{\mathbb{P}_n(S_m > b | (S_{n+1},...,S_k) \in \cdot, \kappa_n = k)}{\sum_{\ell=n+1}^{\infty} \mathbb{P}_n(S_{\ell} > b)} \\
&= \mathbb{P}_n((S_{n+1},...,S_k) \in \cdot, \kappa_n = k) \frac{\sum_{m=k}^{\infty} \mathbb{P}_n(S_m > b | (S_{n+1},...,S_k) \in \cdot, \tau(b) = k)}{\sum_{\ell=n+1}^{\infty} \mathbb{P}_n(S_{\ell} > b)}.
\end{aligned}$$

Thus, $\frac{d\mathbb{P}_n}{d\mathbb{Q}_n}(S_{n+1},...,S_{\kappa_n}, \kappa_n < \infty) = \frac{\sum_{\ell=n+1}^{\infty} \mathbb{P}_n(S_{\ell} > b)}{\sum_{m=\kappa_n}^{\infty} \mathbb{P}_{\kappa_n}(S_m > b)}$. □

Proof of Lemma 3 Let $\mathbb{E}_{\mathbb{Q}}$ denote the expectation under measure $\mathbb{Q}$. Suppose $M_n = b$. First note that by Lemma 2,

$$
\begin{aligned}
\mathbb{Q}_n(I=1) &= \mathbb{E}_{\mathbb{Q}_n}\left[\Big(\sum_{\ell=\kappa_n}^{\infty}\mathbb{P}_{\kappa_n}(S_\ell>b)\Big)^{-1}\right] \\
&= \mathbb{E}_{\mathbb{P}_n}\left[\frac{1}{\sum_{\ell=\kappa_n}^{\infty}\mathbb{P}_{\kappa_n}(S_\ell>b)}\frac{\sum_{\ell=\kappa_n}^{\infty}\mathbb{P}_{\kappa_n}(S_\ell>b)}{\sum_{\ell=n+1}^{\infty}\mathbb{P}_n(S_\ell>b)}1\{\kappa_n<\infty\}\right] \\
&= \mathbb{E}_{\mathbb{P}_n}\left[\Big(\sum_{\ell=n+1}^{\infty}\mathbb{P}_n(S_\ell>b)\Big)^{-1}1\{\kappa_n<\infty\}\right] = \frac{\mathbb{P}_n(\kappa_n<\infty)}{\sum_{\ell=n+1}^{\infty}\mathbb{P}_n(S_\ell>b)}
\end{aligned} \tag{13}
$$

Next, by Bayes rule,

$$
\mathbb{Q}_n((S_{n+1},\dots,S_{\kappa_n})\in\cdot,\kappa_n\in\cdot|I=1) = \frac{\mathbb{Q}_n(I=1|\kappa_n,\mathcal{S}_{\kappa_n})\mathbb{Q}_n((S_{n+1},\dots,S_{\kappa_n})\in\cdot,\kappa_n\in\cdot)}{\mathbb{Q}_n(I=1)}. \tag{14}
$$

As $\mathbb{Q}_n(I=1|\kappa_n,(S_{n+1},\dots,S_{\kappa_n})) = \frac{1}{\sum_{\ell=\tau(b)}^{\infty}\mathbb{P}_{\kappa_n}(S_\ell>b)}$, plugging (13) in (14), we have

$$
\begin{aligned}
&\mathbb{Q}_n((S_{n+1},\dots,S_k)\in\cdot,\kappa_n=k|I=1) \\
&= \frac{1}{\sum_{\ell=k}^{\infty}\mathbb{P}_k(S_\ell>b)}\mathbb{Q}_n((S_{n+1},\dots,S_k)\in\cdot,\kappa_n=k)\frac{\sum_{\ell=n+1}^{\infty}\mathbb{P}_n(S_\ell>b)}{\mathbb{P}_n(\kappa_n<\infty)} \\
&= \mathbb{E}_{\mathbb{Q}_n}\left[1\{(S_{n+1},\dots,S_k)\in\cdot,\kappa_n=k\}\frac{\sum_{\ell=n+1}^{\infty}\mathbb{P}_n(S_\ell>b)}{\sum_{\ell=k}^{\infty}\mathbb{P}_k(S_\ell>b)}\right]\frac{1}{\mathbb{P}_n(\kappa_n<\infty)} \\
&= \frac{\mathbb{E}_{\mathbb{P}_n}\left[1\{(S_{n+1},\dots,S_k)\in\cdot,\kappa_n=k\}\right]}{\mathbb{P}_n(\kappa_n<\infty)} \text{ by Lemma 2} \\
&= \mathbb{P}_n(S_{n+1},\dots,S_k)\in\cdot,\kappa_n=k|\kappa_n<\infty).
\end{aligned}
$$

Proof of Lemma 4 Given $\mathcal{S}_n$, suppose $M_n=b$. We also define

$$
N_1 = \left(\frac{2\sigma^2 n^H\|\boldsymbol{\Sigma}_n^{-1}\|_1\|\tilde{\boldsymbol{S}}_n\|_1}{\mu}\right)^{\frac{1}{1-H}},\ N_2 = \left(\frac{2\sigma^2}{\pi\mu^2}\right)^{\frac{1}{2(1-H)}},\ \text{and } N_3 = \left(\frac{2H-1}{1-H}\frac{16\sigma^2}{\mu^2}\right)^2.
$$

Note that for any $k>n$, S_k conditional on $\mathcal{S}_n$ is still a Gaussian random variable with conditional mean $\mu_n(k)=\mathbb{E}[S_k|\mathcal{S}_n] = -k\mu + \boldsymbol{U}_{nk}^\top\boldsymbol{\Sigma}_n^{-1}\tilde{\mathcal{S}}_n$, and conditional variance $\sigma_n(k)^2 = \mathrm{Var}[S_k|\mathcal{S}_n] = \sigma^2 k^{2H} - \boldsymbol{U}_{nk}^\top\boldsymbol{\Sigma}_n^{-1}\boldsymbol{U}_{nk}$.

We first establish the sequence of bounds. The lower bound is straightforward. For the upper bound, note that for $k\ge N_1$,

$$
\mu_n(k) \le -k\mu + \sigma^2(nk)^H\|\boldsymbol{\Sigma}_n^{-1}\tilde{S}_n\|_1 \le -k\mu + \sigma^2(nk)^H\|\boldsymbol{\Sigma}_n^{-1}\|_1\|\tilde{S}_n\|_1 \le -\frac{k\mu}{2}.
$$

Next, note that for $k\ge\max\{N_1,N_2\}$,

$$
\mathbb{P}_n(S_k>b) \le \frac{1}{\sqrt{2\pi}}\frac{\sigma_n(k)}{b-\mu_n(k)}\exp\left(-\frac{(b-\mu_n(k))^2}{\sigma_n(k)^2}\right) \le \exp\left(-\frac{\mu^2}{8\sigma^2}k^{2-2H}\right). \tag{15}
$$

To see the second inequality, note that when $k \geq N_1$, $\mu_n(k) \leq -k\mu/2$ and $\sigma_n(k) \leq \sigma k^H$. Thus, $\frac{b-\mu_n(k)}{\sigma_n(k)} \geq \frac{b+k\mu}{2\sigma k^H} \geq \frac{\mu}{2\sigma} k^{1-H}$. And for $k \geq N_2$, $\frac{1}{\sqrt{2\pi}} \left(\frac{\mu}{2\sigma} k^{1-H}\right)^{-1} \leq 1$.

Lastly, we have for $\ell \geq \max\{N_1, N_2, N_3\}$,

$$
\begin{aligned}
\sum_{k=\ell+1}^{\infty} \mathbb{P}_n(S_k > b) &\leq \sum_{k=\ell+1}^{\infty} \exp\left(-\frac{\mu^2}{8\sigma^2} k^{2-2H}\right) \quad \text{from (15) as } \ell \geq \max\{N_1, N_2\} \\
&\leq \int_{\ell}^{\infty} \exp\left(-\frac{\mu^2}{8\sigma^2} k^{2-2H}\right) \mathrm{d}k \\
&= \frac{1}{2-2H} \int_{\ell^{2-2H}}^{\infty} y^{(2H-1)/(2-2H)} \exp\left(-\frac{\mu^2}{8\sigma^2} y\right) \mathrm{d}y \\
&\leq \frac{1}{2-2H} \int_{\ell^{2-2H}}^{\infty} \exp\left(-\frac{\mu^2}{16\sigma^2} y\right) \mathrm{d}y \quad \text{as } \ell \geq N_3 \\
&\leq \frac{8\sigma^2}{(1-H)\mu^2} \exp\left(-\frac{\mu^2}{16\sigma^2} \ell^{2-2H}\right) = h(\ell).
\end{aligned}
$$

For $\mathbb{E}[B(n)^\eta]$, we first note that N_2 and N_3 are finite constants. Thus, we only need to show that $\mathbb{E}[N_1^\eta] < \infty$. For any fixed n,

$$
\begin{aligned}
\mathbb{E}[N_1^\eta] &= \mathbb{E}\left[\left(\frac{2\sigma^2 n^H \|\boldsymbol{\Sigma}_n^{-1}\|_1 \|\tilde{\boldsymbol{S}}_n\|_1}{\mu}\right)^{\frac{\eta}{1-H}}\right] = \left(\frac{2\sigma^2 n^H \|\boldsymbol{\Sigma}_n^{-1}\|_1}{\mu}\right)^{\frac{\eta}{1-H}} \mathbb{E}\left[\left(\sum_{k=1}^{n} |S_k + k\mu|\right)^{\frac{\eta}{1-H}}\right] \\
&\leq \left(\frac{2\sigma^2 n^H \|\boldsymbol{\Sigma}_n^{-1}\|_1}{\mu}\right)^{\frac{\eta}{1-H}} n^{\frac{\eta}{1-H}-1} \sum_{k=1}^{n} \mathbb{E}\left[|S_k + k\mu|^{\frac{\eta}{1-H}}\right] \\
&= \left(\frac{2\sigma^2 \|\boldsymbol{\Sigma}_n^{-1}\|_1}{\mu}\right)^{\frac{\eta}{1-H}} n^{\frac{\eta H + \eta + H - 1}{1-H}} \frac{\Gamma(\frac{\eta/(1-H)+1}{2})}{\sqrt{\pi}} (2\sigma^2)^{\frac{\eta}{2(1-H)}} \sum_{k=1}^{n} k^{\frac{\eta H}{1-H}} \\
&\leq \left(\frac{2^{3/2}\sigma^3 \|\boldsymbol{\Sigma}_n^{-1}\|_1}{\mu}\right)^{\frac{\eta}{1-H}} \frac{\Gamma(\frac{\eta/(1-H)+1}{2})}{\sqrt{\pi}} n^{\frac{2\eta H + \eta}{1-H}}.
\end{aligned}
$$

Proof of Lemma 5 Given κ_n and $\mathcal{S}_{\kappa_n}$, suppose $M_n = b$. First note that

$$
\tilde{q}(n, \ell) \leq \tilde{q}(n, \ell) \leq \cdots \leq \sum_{i=\kappa_n+1}^{\infty} \mathbb{P}_{\kappa_n}(S_i > b).
$$

Next, following the proof of Lemma 4, we have for $\ell \geq B(\kappa_n)$,

$$
\tilde{q}(n, \ell) + h(\ell) \geq \tilde{q}(n, \ell+1) + h(\ell+1) \geq \cdots \geq \sum_{i=\kappa_n+1}^{\infty} \mathbb{P}_{\kappa_n}(S_i > b).
$$

Since $\mathbb{P}_k(S_k > b) = 1$, $p(k) = (1 + \sum_{i=k+1}^{\infty} \mathbb{P}_k(S_i > b))^{-1}$, and for $\ell \geq B(\kappa_n)$, $(1 + \tilde{q}(n,\ell) + h(\ell))^{-1} \leq p(\kappa_n) \leq (1 + \tilde{q}(n,\ell))^{-1}$. The rest of the results follow similarly. □

Proof of Lemma 6 We first note that in Step 2.1 in Algorithm 2, $\mathbb{P}_n(N(n) = \ell, J = 1) = \mathbb{P}_n(S_\ell > M_n)$. Next, following the same lines of analysis as the proof of Lemma 1, we have for any $\delta > 0$, there exists $L_0 > 0$ such that for $\ell > L_0$, $\mathbb{P}_n(S_\ell > M_n) \leq \bar{\Phi}\left(\frac{\mu-\delta}{\sigma}\ell^{1-H}\right)$, and for any $\eta > 0$, $\mathbb{E}[L_0^\eta] < \infty$. Then for any $\eta > 0$,

$$\mathbb{E}[N(n)^\eta|\mathcal{S}_n] = \sum_{\ell=n+1}^{\infty} \ell^\eta \mathbb{P}_n(N(n) = \ell) \leq \mathbb{E}[N_0^\eta|\mathcal{S}_n] + \mathbb{E}\left[\sum_{\ell=N_0+1}^{\infty} \ell^\eta \bar{\Phi}\left(\frac{\mu-\delta}{\sigma}\ell^{1-H}\right)\middle|\mathcal{S}_n\right].$$

Thus,

$$\mathbb{E}[N(n)^\eta] = \mathbb{E}[\mathbb{E}[N(n)^\eta|\mathcal{S}_n]] \leq \mathbb{E}[N_0^\eta] + \mathbb{E}\left[\sum_{\ell=N_0+1}^{\infty} \ell^\eta \bar{\Phi}\left(\frac{\mu-\delta}{\sigma}\ell^{1-H}\right)\right] < \infty.$$

References

1. Addie, R., Mannersalo, P., Norros, I.: Most probable paths and performance formulae for buffers with gaussian input traffic. European Transactions on Telecommunications **13**(3), 183–196 (2002)
2. Adler, R.J., Blanchet, J.H., Liu, J.: Efficient Monte Carlo for high excursions of gaussian random fields. The Annals of Applied Probability **22**(3), 1167–1214 (2012)
3. Alvarez-Lacalle, E., Dorow, B., Eckmann, J.P., Moses, E.: Hierarchical structures induce long-range dynamical correlations in written texts. Proceedings of the National Academy of Sciences **103**(21), 7956–7961 (2006)
4. Ambikasaran, S., Foreman-Mackey, D., Greengard, L., Hogg, D.W., O'Neil, M.: Fast direct methods for Gaussian processes. IEEE Transactions on Pattern Analysis and Machine Intelligence **38**(2), 252–265 (2016). DOI 10.1109/TPAMI.2015.2448083
5. Asmussen, S.: Applied Probability and Queues, 2nd edn. Springer (2003)
6. Bayer, C., Friz, P., Gatheral, J.: Pricing under rough volatility. Quantitative Finance **16**(6), 887–904 (2016)
7. Beran, J.: Statistical methods for data with long-range dependence. Statistical science pp. 404–416 (1992)
8. Beran, J., Sherman, R., Taqqu, M.S., Willinger, W.: Long-range dependence in variable-bit-rate video traffic. IEEE Transactions on communications **43**(2/3/4), 1566–1579 (1995)
9. Blanchet, J., Chen, X., Dong, J.: ε-Strong simulation for multidimensional stochastic differential equations via rough path analysis. The Annals of Applied Probability **27**(1), 275–336 (2017)
10. Blanchet, J., Dong, J.: Perfect sampling for infinite server and loss systems. Advances in Applied Probability **47**(3), 761–786 (2015)
11. Blanchet, J., Dong, J., Liu, Z.: Exact sampling of the infinite horizon maximum of a random walk over a nonlinear boundary. Journal of Applied Probability **56**(1), 116–138 (2019)
12. Blanchet, J., Li, C.: Efficient simulation for the maximum of infinite horizon discrete-time Gaussian processes. Journal of Applied Probability **48**, 467–489 (2011)

13. Blanchet, J., Sigman, K.: On exact sampling of stochastic perpetuities. Journal of Applied Probability **48**(A), 165–182 (2011)
14. Blanchet, J., Wallwater, A.: Exact sampling of stationary and time-reversed queues. ACM Transactions on Modeling and Computer Simulation **25**(4), 26 (2015)
15. Bucklew, J.A., Radeke, R.: On the Monte Carlo simulation of digital communication systems in Gaussian noise. IEEE Transactions on Communications **51**(2), 267–274 (2003)
16. Chen, Y., Dong, J., Ni, H.: epsilon-strong simulation of fractional Brownian motion and related stochastic differential equations. Mathematics of Operations Research (2021)
17. Devroye, L.: Non-Uniform Random Variate Generation. Springer-Verlag (1986)
18. Dieker, A.: Simulation of fractional Brownian motion. Ph.D. thesis, Masters Thesis, Department of Mathematical Sciences, University of Twente (2004)
19. Dieker, A.: Extremes of Gaussian processes over an infinite horizon. Stochastic Processes and their Applications **115**(2), 207–248 (2005)
20. Dieker, A.B., Mandjes, M.: On spectral simulation of fractional Brownian motion. Probability in the Engineering and Informational Sciences **17**(3), 417–434 (2003)
21. Dieker, A.B., Mandjes, M.: Fast simulation of overflow probabilities in a queue with Gaussian input. ACM Transactions on Modeling and Computer Simulation **16**(2), 119–151 (2006)
22. Dietrich, C., Newsam, G.N.: Fast and exact simulation of stationary Gaussian processes through circulant embedding of the covariance matrix. SIAM Journal on Scientific Computing **18**(4), 1088–1107 (1997)
23. Dombry, C., Engelke, S., Oesting, M.: Exact simulation of max-stable processes. Biometrika **103**(2), 303–317 (2016)
24. Doukhan, P., Oppenheim, G., Taqqu, M.: Theory and applications of long-range dependence. Springer Science & Business Media (2002)
25. Ensor, K., Glynn, P.: Simulating the maximum of a random walk. Journal of Statistical Planning and Inference **85**, 127–135 (2000)
26. Gatheral, J., Jaisson, T., Rosenbaum, M.: Volatility is rough. Quantitative finance **18**(6), 933–949 (2018)
27. Glynn, P.W., Whitt, W.: The asymptotic efficiency of simulation estimators. Operations research **40**(3), 505–520 (1992)
28. Heyde, C., Yang, Y.: On defining long-range dependence. Journal of Applied Probability **34**, 939–944 (1997)
29. Huang, C., Devetsikiotis, M., Lambadaris, I., Kaye, A.: Fast simulation of queues with long-range dependent traffic. Stochastic Models **15**(3), 429–460 (1999)
30. Hurst, H.E.: Long-term storage capacity of reservoirs. Transactions of the American society of civil engineers **116**(1), 770–799 (1951)
31. Jean-Francois, C.: Simulation and identification of the fractional Brownian motion: a bibliographical and comparative study. Journal of statistical software **5**, 1–53 (2000)
32. Karagiannis, T., Molle, M., Faloutsos, M.: Long-range dependence ten years of internet traffic modeling. IEEE internet computing **8**(5), 57–64 (2004)
33. Lau, W.C., Erramilli, A., Wang, J.L., Willinger, W.: Self-similar traffic generation: The random midpoint displacement algorithm and its properties. In: Proceedings IEEE International Conference on Communications ICC'95, vol. 1, pp. 466–472. IEEE (1995)
34. Liu, Z., Blanchet, J., Dieker, A., Mikosch, T.: On logrithmically optimal exact simulation of max-stable and related random fields on a compact set. Bernoulli **25**(4A), 2949–2981 (2019)
35. Norros, I., Mannersalo, P., Wang, J.L.: Simulation of fractional Brownian motion with conditionalized random midpoint displacement. Advances in Performance Analysis **2**(1), 77–101 (1999)
36. Robinson, P.M.: Gaussian semiparametric estimation of long range dependence. The Annals of statistics pp. 1630–1661 (1995)
37. Robinson, P.M.: Time series with long memory. Advanced Texts in Econometrics (2003)
38. Samorodnitsky, G.: Long range dependence. now Publishers Inc (2007)