Exact Sampling for the Maximum of Infinite Memory Gaussian Processes

Jose Blanchet\textsuperscript{a,1}, Lin Chen\textsuperscript{b}, Jing Dong\textsuperscript{b,2}

\textsuperscript{a}Stanford University, 475 Via Ortega, Stanford, CA 94305. \\
\textsuperscript{b}Columbia University, 3022 Broadway, New York, NY 10027.

Abstract

We develop an exact sampling algorithm for the all time maximum of a Gaussian process with a negative drift and a general correlation structure. Our algorithm requires sampling the Gaussian process for a random amount of time that is adapted to the sample path under construction. The running time of the algorithm has finite moments of all orders.

Keywords: Exact sampling, Gaussian process, long-range dependence

1. Introduction

Let $S = \{S_n : n \in \mathbb{Z}^+\}$ be a Gaussian process. We assume $S_0 \equiv 0$, $\mathbb{E}[S_n] = -n\mu$ for some $\mu > 0$, and

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

In particular, $S_n$ is a negative drifted Gaussian process. Note that $\text{Var}(S_n)$ can grow sublinearly ($H < 1/2$) or superlinearly ($H > 1/2$) in $n$. 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 $\sigma^2$, we have

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

1 Support from NSF grant DMS-132055 is gratefully acknowledged by J. Blanchet. 
2 Support from NSF grant DMS-1720433 is gratefully acknowledged by J. Dong.
where $\rho_i := \text{Corr}(X_n, X_{n+i})$ for $n \in \mathbb{Z}^+$. In this setting, it is commonly understood as long-term or infinite memory the situation in which $\lim_{n \to \infty} \frac{\text{Var}(S_n)}{n} = \infty$ (see [1]). Note that our assumption on $\text{Var}(S_n)$ with $H > 1/2$ accommodates the case of $S_n$ being an infinite memory Gaussian process (corresponding to the case $\sum_{i=1}^{\infty} \rho_i = \infty$).

Most Gaussian processes of practical interests belong to the class of Gaussian processes defined above. For example, let $B^H(t)$ be a factional Brownian motion with Hurst index $H \in (0, 1)$. Then, for $S_n = -n\mu + \sigma B^H(n)$, we have $E[S_n] = -n\mu$ and $\text{Var}(S_n) = \sigma^2 n^{2H}$. Another important class of examples includes fractional ARIMA models.

Let us define the all time maximum of the Gaussian process as

$$M = \max_{n \geq 0} \{S_n\}.$$ 

The main contribution of this paper is the development of an algorithm to draw samples from the distribution of $M$. This is the first work that provides exact simulation algorithm for the all-time maximum of this general class of negative drifted Gaussian processes. The random variable $M$ is often encountered in applications in risk management (e.g. ruining probability) and queueing theory, see [2].

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 to generate an infinite sequence of $S_n$’s, i.e., for all $n \geq 0$, which is clearly infeasible. Our algorithm utilizes a milestone-event construction, which allows us to sample $S_n$’s up to a finite but random time to determine $M$. Of course, such random time cannot be a stopping time with respect to the filtration generated by $\{S_n : n \geq 0\}$ and this is the challenge of the construction.

The milestone-event construction is a variation of the so-called record-breaker strategy, which was first developed in [3] and subsequently applied to several other settings; see [4, 5] for two such examples. The key to successful implementation of this strategy involves two components. The first component is to define a proper sequence of milestone events which will occur only finitely
many times with probability one (making each subsequent milestone less and less likely). The definition and analysis of the milestone events relies on a good understanding of certain analytical property, often corresponding to the large deviations behavior of the underlying stochastic process. The second component is to construct simulation algorithms to efficiently generate these milestone events sequentially until we find the last one of those.

Most of the milestone events have very small probability (less and less likely along the sequence). Thus, developing efficient strategies to sample these events borrows ideas from rare event simulation [6]. The work of [7] is among the first to use rare-event simulation techniques to generate the maximum of a negative-drifted random walk with light-tailed increment. The paper of [8] extends a similar idea to generate the maximum of random walks with heavy-tailed increments and [9] further extends the idea to generate the maximum of a random walk over a nonlinear boundary. Our work extend this line of research by allowing potentially infinite memory in the increments of \( S_n \).

There is also a growing line of literature on generating functions that depend on the whole sequence of Gaussian processes in various applications, especially algorithms for processes that exhibit long-range dependence [10, 11]. The work of [12], for instance, provides an extensive review on simulation algorithms for fractional Brownian motion, which is an important class of Gaussian process with infinite memory. The papers [13] and [14] develop exact simulation algorithms for max-stable fields at a finite collection of locations. These algorithms rely on the ability to generate the maximum of a Gaussian random field. Our work extends this later line of works by allowing more general dependence structures for the Gaussian process.

**Notations.** Throughout the paper, we use \( \Phi(x) \) to denote the cumulative distribution function (cdf) of standard normal distribution, and denote \( \Phi(x) := 1 - \Phi(x) \) as the tail cdf. We define \( \mathbf{S}_n = (S_1, \ldots, S_n)^T \) and \( \Sigma_n \) is covariance function of \( \mathbf{S}_n \). In particular, we denote \( \gamma_{ij} = \text{Cov}(S_i, S_j) \), as the \((i,j)\)-th entry of \( \Sigma_n \). Let \( P_n(\cdot) = P(\cdot|\mathbf{S}_n) \). We also define \( \tilde{\mathbf{S}}_n = (S_1 + \mu, S_2 + 2\mu, \ldots, S_n + n\mu)^T \).
and $U_{nk} = (\gamma_{1k}, \ldots, \gamma_{nk})^T$.

2. Basic strategy

In this section, we introduce the main idea of our algorithmic development. 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. The idea is to generate these events sequentially until we find the last upward crossing event. As the Gaussian process has negative drift, generating the downward crossing event is straightforward. On the other hand, the upward crossing events in general have very small probability. We thus introduce a rare-event simulation technique called Target Bridge Sampling (TBS), see [15], to efficiently generate the upward crossing events.

2.1. Milestone events

Define
\[
q(n) = \sum_{k=n+1}^{\infty} \mathbb{P}_n \left( s_k > \max_{1 \leq l \leq n} s_l \right).
\]  
(1)

For infinite memory Gaussian processes, we define a sequence of downward-crossing milestone event times and upward-crossing milestone event times as follows. Let $\tau_{0}^+ \equiv 0$. For $k \geq 1$, if $\tau_{k-1}^+ < \infty$, define
\[
\tau_k^- := \inf \{ n > \tau_{k-1}^+ : q(n) < 3/4 \}, \quad \tau_k^+ := \inf \left\{ n > \tau_k^- : s_n > \max_{0 \leq l \leq \tau_k^-} s_l \right\}.
\]

We also define the last upward crossing milestone event time as
\[
T := \sup \{ \tau_k^+ : \tau_k^+ < \infty \}, \quad \text{and} \quad K = \sup \{ k : \tau_k^+ < \infty \}, \quad \text{i.e.,} \quad T = \tau_K^+.
\]

Then, by the definition of the milestone events,
\[
M = \max_{0 \leq n \leq \tau_{K+1}^-} s_n.
\]

In particular, by generating the Gaussian process $S$ up to the random time $\tau_{K+1}^-$, we are able to recover the all time maximum of $S$. 

4
Based on the milestone events, the algorithm can then be decomposed into finding $\tau_k^-$’s and $\tau_k^+$’s sequentially until we find $\tau_{K+1}^-$. We first note that $T$ is well defined. In particular, we have the following theorem.

**Theorem 1** For any $\eta > 0$, $E[T^\eta] < \infty$.

Theorem 1 indicates that even though $\tau_{K+1}^-$ is random, it has finite moments of all orders. The detailed proof of Theorem 1 is deferred to Section 4.

We next explain how to find the milestone events sequentially. Finding the downward crossing events can be done straightforwardly under the nominal measure. The only difficulty is how to check whether $q(n) < \frac{3}{4}$. We defer detailed discussion of this to Section 3. The next lemma ensures that the downward-crossing milestone event is well-defined. In particular, we can find these events in a finite amount of time under the nominal measure. The lemma is also an important intermediate step to establish Theorem 1.

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

**Proof** [Proof of Lemma 1] We start by introduce a few notations. Note that $S_k$ conditional on $S_n$ is still a Gaussian random variable with conditional mean

$$\mu_n(k) = E[S_k | S_n] = -k\mu + U_n^T \Sigma_n^{-1} \tilde{S}_n,$$

and conditional variance

$$\sigma_n(k)^2 = \text{Var}[S_k | S_n] = \sigma^2 k^{2H} - U_{nk}^T \Sigma_n^{-1} U_{nk}.$$

The proof of the lemma is divided into three steps. We first establish bounds for the conditional mean $\mu_n(k)$. Let $\tilde{\mu}_n(k) = U_{nk}^T \Sigma_n^{-1} \tilde{S}_n$. As $\tilde{\mu}_n(k)$ is a linear combination of $S_n$, it follows a Normal distribution with mean 0 and variance $U_{nk}^T \Sigma_n^{-1} U_{nk}$. By the law of total variance, $U_{nk}^T \Sigma_n^{-1} U_{nk} < \sigma^2 k^{2H}$. In this case,
for any fixed \( \delta \in (0, \mu) \), we have
\[
\mathbb{P}(\tilde{\mu}_n(k) > \delta k) \leq \mathbb{P}\left( \frac{\tilde{\mu}_n(k)}{\sqrt{U_n k^T \sum_n^{-1} U_{nk}}} > \frac{\delta k}{\sigma k^H} \right) = \Phi \left( \frac{\delta}{\sigma} k^{1-H} \right) \quad (2)
\]

Then,
\[
\sum_{n=1}^{\infty} \sum_{k=n}^{\infty} \mathbb{P}(\tilde{\mu}_n(k) > \delta k) = \sum_{k=1}^{\infty} \sum_{n=1}^{k} \mathbb{P}(\tilde{\mu}_n(k) > \delta k) \\
\leq \sum_{k=1}^{\infty} k \Phi \left( \frac{\delta}{\sigma} k^{1-H} \right) < \infty.
\]

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

We next establish bounds for \( \sum_{n=1}^{\infty} q(n) \). For \( n + k > L_0 \), we have \( \mu_n(k) \leq - (\mu - \delta) k \) and \( \sigma_n(k) \leq \sigma^2 k^{2H} \). Thus, for any \( b \geq 0 \),
\[
\mathbb{P}_n(S_k > b) \leq \mathbb{P}_n \left( \frac{S_k - \mu_n(k)}{\sigma_n(k)} > \frac{b + (\mu - \delta) k}{\sigma k^H} \right) \leq \Phi \left( \frac{\mu - \delta}{\sigma} k^{1-H} \right). \quad (3)
\]

Based on the analysis above, let \( b = \max_{1 \leq l \leq n} S_l \). We decompose \( \sum_{n=1}^{\infty} q(n) \) into three parts:
\[
\sum_{n=1}^{\infty} \sum_{k=n}^{\infty} \mathbb{P}_n(S_k > b) \leq \sum_{n=1}^{L_0} \sum_{k=n}^{L_0} \mathbb{P}_n(S_k > b) + \sum_{n=1}^{\infty} \sum_{k=L_0}^{L_0} \mathbb{P}_n(S_k > b) + \sum_{n=L_0}^{\infty} \sum_{k=n}^{\infty} \mathbb{P}_n(S_k > b).
\]

Part (I) only involves a finite number of terms, so it is finite. For part (II), from (3), we have
\[
(II) \leq L_0 \sum_{k=L_0}^{\infty} \Phi \left( \frac{\mu - \delta}{\sigma} k^{1-H} \right) < \infty
\]

Similarly, for part (III), from (3), we have
\[
(III) = \sum_{k=N}^{\infty} \sum_{n=N}^{k} \mathbb{P}_n(S_k > b) \leq \sum_{k=N}^{\infty} (k - N) \Phi \left( \frac{\mu - \delta}{\sigma} k^{1-H} \right) < \infty
\]
Putting part (I), (II) and (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 $E[L^\eta] < \infty$ for any $\eta > 0$. Let $\tilde{L}$ denote a large enough constant, such that

$$\sum_{k=L}^{\infty} \Phi \left( \frac{\mu - \delta}{\sigma} k^{1-H} \right) < a.$$ 

Then, $L \leq \max\{L_0, \tilde{L}\}$. Thus, to prove $E[L^\eta] < \infty$, we only need to show that $E[L_0^\eta] < \infty$. Define $A_n = \bigcup_{k=n}^{\infty} \{ \tilde{\mu}_n(k) > \delta k \}$. Then

$$L_0^\eta \leq \sum_{n=1}^{\infty} 1\{A_n\} n^\eta,$$

and

$$E[L_0^\eta] \leq E \left[ \sum_{n=1}^{\infty} 1\{A_n\} n^\eta \right] = \sum_{n=1}^{\infty} \sum_{k=n}^{\infty} \mathbb{P}(\tilde{\mu}_n(k) > \delta k) n^\eta$$

$$= \sum_{n=1}^{\infty} \sum_{k=1}^{n} n^\eta \mathbb{P}(\tilde{\mu}_n(k) > \delta k)$$

$$\leq \sum_{k=1}^{\infty} k^\eta \Phi \left( \frac{\delta}{\sigma} k^{1-H} \right) < \infty,$$

where the last inequality follows from (2). \(\square\)

Sampling the upward crossing event is more challenging. In particular, since $\mathbb{P}(\tau^+_k < \infty | S_1, \ldots, S_{\tau^-_k}, \tau^-_k) < 1$, given all the information generated up to $\tau^-_k$, if we generate the Gaussian process under its nominal measure until $S_n > \max_{0 \leq i \leq \tau^-_k} S_i$, we may never be able to find $\tau^+_k$, 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 – TBS. The basic idea is to introduce a new measure under which the upward-crossing event happens with probability 1. This provides us with a proposed upward-crossing path. We then apply an acceptance-rejection step. If the path is accepted, it is the path leading to the next upward-crossing event. If the path is rejected, we can claim that there will be no more upward-crossing milestone event.

2.2. Sampling the upward-crossing event

In this section, we provide more details as how to sample the upward-crossing events. To make the discussion concise, we assume that we have found $\tau^-_k$ for
some for some $k > 0$. We write $\tau_k^- = n$ and $\{S_l : 1 \leq l \leq n\} = s_n$. We also define $\mathbb{P}_n(\cdot) = \mathbb{P}(\cdot|s_n)$ and denote $b = \max_{0 \leq l \leq \tau^-} S_l$. Then $\tau_{k+1}^+ = \inf\{l > n : S_l > b\}$. With a little abuse of notation, we define $\tau(b) = \tau_{k+1}^+$, and, naturally, $\mathbb{P}_{\tau(b)}(\cdot) = \mathbb{P}(\cdot|\tau(b), (S_1, \ldots, S_{\tau(b)}))$.

We first introduce a new measure $\mathbb{Q}$ under which the record breaking event happens with probability 1. The construction of the new measure is based on TBS developed in [15]. The new measure is described through the corresponding sampling procedure (Procedure TBS).

**Procedure TBS (Given $\tau_k^- = n$, and $s_n$)**

1. Sample $N(n) \geq n$ with probability density function

   $$f_n(m) = \frac{\mathbb{P}_n(S_m > b)}{\sum_{k=n+1}^{\infty} \mathbb{P}_n(S_k > b)},$$

   for $m = n + 1, n + 2, \ldots$.

2. Given $N(n) = m$, sample $S_m$ according to the conditional measure

   $$\mathbb{P}_n(S_m \in \cdot|S_m \geq b).$$

3. Given $S_m$ and $s_n$, sample the Gaussian bridge $S_{n+1}, \ldots, S_{m-1}$.

4. Set $\tau(b) = \min\{k > n : S_k > b\}$.

5. Output $S_{n+1}, \ldots, S_{\tau(b)}$.

Clearly, $\mathbb{Q}$ is defined in such a way that $\mathbb{Q}(\tau(b) < \infty) = 1$, However, $\mathbb{Q}$ is different from $\mathbb{P}_n(\cdot|\tau(b) < \infty)$. Thus, we need to apply a change of measure, i.e., an acceptance-rejection step. To carry out the acceptance-rejection step, we need to first characterize the corresponding likelihood ratio.

**Lemma 2** For the probability measure $\mathbb{Q}$ induced by the TBS procedure, we have

$$\frac{d\mathbb{P}_n}{d\mathbb{Q}} 1\{\tau(b) < \infty\} = \frac{\sum_{l=n+1}^{\infty} \mathbb{P}_n(S_l > b)}{\sum_{m=\tau(b)}^{\infty} \mathbb{P}_{\tau(b)}(S_m > b)},$$

(5)
Proof | Proof of Lemma 2 We first note that

\[ Q((S_{n+1}, ..., S_k) \in \cdot, \tau(b) = k) \]

\[ = \sum_{m=n+1}^{\infty} f_n(m) P_n((S_{n+1}, ..., S_k) \in \cdot, \tau(b) = k | S_m > b) \]

\[ = \sum_{m=n+1}^{\infty} f_n(m) \frac{P_n((S_{n+1}, ..., S_k) \in \cdot, \tau(b) = k, S_m > b)}{P_n(S_m > b)} \]

\[ = \sum_{m=n+1}^{\infty} \frac{P_n((S_{n+1}, ..., S_k) \in \cdot, \tau(b) = k, S_m > b)}{\sum_{l=n+1}^{\infty} P_n(S_l > b)} \]

\[ \sum_{m=n+1}^{\infty} P_n(S_m > b | (S_{n+1}, ..., S_k) \in \cdot, \tau(b) = k) \]

\[ = \sum_{m=n+1}^{\infty} \frac{\sum_{l=n+1}^{\infty} P_n(S_m > b | (S_{n+1}, ..., S_k) \in \cdot, \tau(b) = k)}{\sum_{l=n+1}^{\infty} P_n(S_l > b)} \]

Thus,

\[ \frac{dP_n}{dQ}(S_{n+1}, ..., S_{\tau(b)}, \tau(b) < \infty) = \frac{\sum_{l=n+1}^{\infty} P_n(S_l > b)}{\sum_{m=\tau(b)}^{\infty} P_{\tau(b)}(S_m > b)}. \]

\[ \square \]

Based on Lemma 2, we propose the following acceptance-rejection procedure.

Procedure AR (Given \( \tau^-_k = n \), and \( \tau(b) = k \) and \( S_k \) from procedure TBS)

1. Sample a Bernoulli random variable \( I \) with probability of success

\[ p(k) = \left( \sum_{l=k}^{\infty} P_k(S_l > b) \right)^{-1}. \tag{6} \]

2. If \( I = 1 \), we output \( \tau(b) = k \) and \( S_k \). If \( I = 0 \), we claim that \( \tau(b) = \infty \).

To see the correctness of Procedure AR, we have the following lemma.

Lemma 3 If we accept the path in Procedure AR, then the generated path follows the measure \( P|\tau(b) < \infty \). In particular,

\[ Q((S_{n+1}, ..., S_k) \in \cdot, \tau(b) = k | I = 1) = P_n((S_{n+1}, ..., S_k) \in \cdot, \tau(b) = k | \tau(b) < \infty). \]
Proof. We denote $E_\theta$ as the expectation with respect to a measure $\theta$. In this proof, the measure can be $Q$ or $P_n$.

From Procedure TBS and Procedure AR, we have

$$Q(I = 1) = E_Q \left( \sum_{l=\tau(b)}^{\infty} P_{\tau(b)}(S_l > b) \right)^{-1}$$

$$= E_{P_n} \left[ \frac{1}{\sum_{l=\tau(b)}^{\infty} P_{\tau(b)}(S_l > b)} \sum_{l=\tau(b)}^{\infty} P_{\tau(b)}(S_l > b) \mathbb{1}\{\tau(b) < \infty\} \right] \ \text{by Lemma 2}$$

$$= E_{P_n} \left[ \sum_{l=n+1}^{\infty} P_n(S_l > b) \right]^{-1} \mathbb{1}\{\tau(b) < \infty\}$$

$$= \frac{P_n(\tau(b) < \infty)}{\sum_{l=n+1}^{\infty} P_n(S_l > b)} \ \text{(7)}$$

Next, we note that

$$Q((S_{n+1}, \ldots, S_{\tau(b)}) \in \cdot, \tau(b) \in \cdot | I = 1) = \frac{Q((S_{n+1}, \ldots, S_{\tau(b)}) \in \cdot, \tau(b) \in \cdot | I = 1)}{Q(I = 1)} \ \text{(8)}$$

As

$$Q(I = 1 | \tau(b), (S_{n+1}, \ldots, S_{\tau(b)})) = \frac{1}{\sum_{l=\tau(b)}^{\infty} P_{\tau(b)}(S_l > b)} P_{\tau(b)}(S_l > b),$$

plugging (7) in (8), we have

$$Q((S_{n+1}, \ldots, S_k) \in \cdot, \tau(b) = k | I = 1)$$

$$= \frac{1}{\sum_{l=k}^{\infty} P_k(S_l > b)} Q((S_{n+1}, \ldots, S_k) \in \cdot, \tau(b) = k) \frac{\sum_{l=n+1}^{\infty} P_n(S_l > b)}{P_n(\tau(b) < \infty)}$$

$$= E_Q \left[ \sum_{l=n+1}^{\infty} P_n(S_l > b) \mathbb{1}\{\tau(b) < \infty\} \right] \ \text{by Lemma 2}$$

$$= E_{P_n} \left[ \mathbb{1}\{(S_{n+1}, \ldots, S_k) \in \cdot, \tau(b) = k\} \right] \ \text{by Lemma 2}$$

$$= P_n(S_{n+1}, \ldots, S_k) \in \cdot, \tau(b) = k | \tau(b) < \infty).$$

We note from the proof of Lemma 3 that

$$Q(I = 0) = 1 - \frac{P_n(\tau(b) < \infty)}{\sum_{l=n+1}^{\infty} P_n(S_l > b)} \neq P_n(\tau(b) = 0).$$

□
Thus, we introduce another Bernoulli random variable $J$ with probability of success $q(n)$ defined in $[1]$. Note that conditional on $S_n$, $J$ is independent of everything else. We also have $q(n) \geq P_n(\tau(b) < \infty)$, and

$$P_n(J = 0) + Q(I = 0, J = 1) = P_n(J = 0) + Q(I = 0)P_n(J = 1) = P_n(\tau(b) < \infty).$$

2.3. The Algorithm

We next summarize the idea discussed above and present the actual simulation algorithm – Algorithm 1.

**Algorithm 1** Structure of Milestone Strategy.

**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 $S_0, \ldots, S_n$.
2. Call Algorithm 2 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 $b = \max\{S_l, l = 1, 2, \ldots, n\}$.
4. Call Algorithm 3 to sample $J \sim \text{Bernoulli}(q(n))$. If $J = 0$, go to **Step 3**. If $J = 1$, Algorithm 3 also outputs $N$ and go to **Step 2**.

**Step 2: ”Upward-crossing event”**.

1. Given $N$, sample $S_N$ according to $P_n(S_N \in \cdot | S_N > b)$.
2. Given $S_N$ and $S_1, \ldots, S_n$, sample the Gaussian bridge $S_{n+1}, \ldots, S_{N-1}$.
3. Set $\tau(b) = \min\{l \geq n : S_l > b\}$.
4. Call Algorithm 4 to sample $I \sim \text{Bernoulli}(p(\tau(b)))$. If $I = 1$, set $k = k + 1$, $\tau^+_k = \tau(b)$, $n = \tau(b)$ and go to **Step 1**. If $I = 0$, go to **Step 3**.

**Step 3: Output.**

1. Output $M = b$.

We comment that to carry out Algorithm 1 we have to develop a few more
intermediate steps: Algorithm 2 - 4. This is because there are still a few pending technical difficulties in implementing the ideas laid out in Section 2.1 and 2.2. The first one is how to check whether \( p(n) < \frac{3}{4} \). The second one is how to sample a Bernoulli random variable with probability of success \( q(n) \) defined in (1). The third one is how to sample \( N \) according to \( f_n(\cdot) \) defined in (4), and the last one is how to sample a Bernoulli random variable with probability of success \( p(N) \) defined in (6). These challenges arise as \( p(n), q(n) \) and \( f_n(\cdot) \) involves evaluating the sum of infinite terms which is analytically intractable. We address these challenges in Section 3.

3. Intermediate Steps in Algorithm 1

In this section, we present the details for the intermediate steps in Algorithm 1. In particular, we introduce Algorithm 2 - 4. The development of the algorithms is based on a sandwiching construction.

The fundamental challenge in these algorithms is that we have to sample Bernoulli random variables whose probability of success \( p \in (0, 1) \) inaccessible. We resolve this by deriving a sequence of upper bounds and lower bounds, \( \{U(l)\}_{l \geq 1} \) and \( \{L(l)\}_{l \leq 1} \) satisfying

\[
L(1) \leq L(2) \leq \cdots \leq p \leq \cdots \leq U(2) \leq U(1)
\]

and

\[
\lim_{l \to \infty} U(l) = \lim_{l \to \infty} L(l) = p.
\]

Based on these bounds, to sample Bernoulli(p), we can first sample \( U \sim \text{Uniform}[0, 1] \), and then sequentially check whether \( U \leq L(l) \) or \( U \geq U(l) \) for some \( l \geq 1 \).

Recall that \( q(n) = \sum_{k=n+1}^{\infty} \mathbb{P}_n(S_k > M_n) \) where \( M_n = \max_{1 \leq l \leq n} S_l \). We define

\[
q(n, l) = \sum_{k=n+1}^{l} \mathbb{P}_n(S_k > M_n)
\]

and

\[
h(l) = \frac{8\sigma^2}{(1 - H)\mu^2} \exp\left(-\frac{\mu^2}{16\sigma^2}l^2 - 2H\right).
\]
We also define
\[ N(n) = \max \left\{ \left( \frac{2\sigma^2 n H \| \Sigma_n^{-1} \|_1 \| \tilde{S}_n \|_1}{\mu} \right)^{\frac{1}{\sqrt{H}}}, \left( \frac{2\sigma^2}{\pi \mu^2} \right)^{\frac{1}{\sqrt{H} - 1}}, \left( \frac{2H - 1 - 16\sigma^2}{1 - H \mu^2} \right)^2 n + 1 \right\}. \]

The following lemma provides a sequence of upper and lower bounds for \( q(n) \).
It lays the theoretical foundation of the algorithmic development in this section.

**Lemma 4** For any \( l \geq N(n) \),
\[ q(n, l) < q(n) \leq q(n, l) + h(l). \]

Moreover, for any \( \eta > 0 \), \( \mathbb{E}[N(n)^\eta] < \infty \).

**Proof** For simplicity of notation, we define
\[ b = \max_{1 \leq l \leq n} S_l. \]
We also write
\[ N_1 = \left( \frac{2\sigma^2 n H \| \Sigma_n^{-1} \|_1 \| \tilde{S}_n \|_1}{\mu} \right)^{\frac{1}{\sqrt{H}}}, N_2 = \left( \frac{2\sigma^2}{\pi \mu^2} \right)^{\frac{1}{\sqrt{H} - 1}}, \]
and
\[ N_3 = \left( \frac{2H - 1 - 16\sigma^2}{1 - H \mu^2} \right)^2. \]

For any \( k > n \), \( S_k \) conditional on \( S_n \) is still a Gaussian random variable with conditional mean
\[ \mu_n(k) = \mathbb{E}[S_k|S_n] = -k\mu + U_{nk}^T \Sigma_n^{-1} \tilde{S}_n, \]
and conditional variance
\[ \sigma_n(k)^2 = \text{Var}[S_k|S_n] = \sigma^2 k^2 H - U_{nk}^T \Sigma_n^{-1} U_{nk}. \]

The lower bound is straightforward.

For the upper bound, we first note that for \( k \geq N_1 \),
\[ \mu_n(k) \leq -k\mu + \sigma^2 (nk) H \| \Sigma_n^{-1} \|_1 \tilde{S}_n \|_1 \]
\[ \leq -k\mu + \sigma^2 (nk) H \| \Sigma_n^{-1} \|_1 \| \tilde{S}_n \|_1 \leq -\frac{k\mu}{2}. \]

Next, note that for \( k \geq \max\{N_1, N_2\} \),
\[ \mathbb{P}_n(S_k > b) \leq \frac{1}{\sqrt{2\pi}} \frac{\sigma_n(k)}{b - \mu_n(k)} \exp \left( -\frac{(b - \mu_n(k))^2}{\sigma_n(k)^2} \right) \leq \exp \left( -\frac{\mu^2}{8\sigma^2 k^2 - 2H} \right). \]

(9)
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 $l \geq \max\{N_1, N_2, N_3\}$,
\[
\sum_{k=1+1}^{\infty} \mathbb{P}_n(S_k > b) \leq \sum_{k=1+1}^{\infty} \exp \left( -\frac{\mu^2}{8\sigma^2} k^{2-2H} \right) \quad \text{from } [0] \quad \text{as } l \geq \max\{N_1, N_2\}
\]
\[
\leq \int_l^{\infty} \exp \left( -\frac{\mu^2}{8\sigma^2} k^{2-2H} \right) \, dk
\]
\[
= \frac{1}{2 - 2H} \int_{l^{2-2H}}^{\infty} y^{(2H-1)/(2-2H)} \exp \left( -\frac{\mu^2}{16\sigma^2} y \right) \, dy
\]
\[
\leq \frac{1}{2 - 2H} \int_{l^{2-2H}}^{\infty} \exp \left( -\frac{\mu^2}{16\sigma^2} y \right) \, dy \quad \text{as } l \geq N_3
\]
\[
\leq \frac{8\sigma^2}{(1 - H)\mu^2} \exp \left( -\frac{\mu^2}{16\sigma^2} l^{2-2H} \right) = h(l).
\]
For $\mathbb{E}[\mathbb{N}(n)\gamma]$, we first note that $N_2$ and $N_3$ are finite constants. Thus, we only need to show that $\mathbb{E}[\mathbb{N}(n)\gamma] < \infty$. For any fixed $n$,
\[
\mathbb{E}[\mathbb{N}(n)\gamma] = \mathbb{E} \left[ \left( 2\sigma^2 n^H \| \Sigma_n^{-1} \|_1 \| \tilde{S}_n \|_1 \right)^{\gamma} \right]
\]
\[
= \left( 2\sigma^2 n^H \| \Sigma_n^{-1} \|_1 \right)^{\gamma} \mathbb{E} \left[ \left( \sum_{k=1}^{n} |S_k + k\mu| \right)^{\gamma} \right]
\]
\[
\leq \left( 2\sigma^2 n^H \| \Sigma_n^{-1} \|_1 \right)^{\gamma} n^{1-H} \sum_{k=1}^{n} \mathbb{E} \left[ |S_k + k\mu|^{\gamma} \right]
\]
\[
= \left( 2\sigma^2 \| \Sigma_n^{-1} \|_1 \right)^{\gamma} n^{1-H} \sum_{k=1}^{n} \mathbb{E} \left[ |S_k + k\mu|^{\gamma} \right]
\]
\[
\leq \left( 2^{3/2} \sigma^3 \| \Sigma_n^{-1} \|_1 \right)^{\gamma} \frac{\Gamma(\gamma/(1-H)+1)}{\sqrt{\pi}} n^{2H+\gamma-1}\frac{\Gamma(\eta/(1-H)+1)}{\sqrt{\pi}} n^{2\eta H+\gamma-1}\frac{2\eta H+\gamma}{\sqrt{\pi}} \sum_{k=1}^{n} k^{1-H}
\]
\[
\leq \left( 2^{3/2} \sigma^3 \| \Sigma_n^{-1} \|_1 \right)^{\gamma} \frac{\Gamma(\gamma/(1-H)+1)}{\sqrt{\pi}} n^{2H+\gamma-1}\frac{2\eta H+\gamma}{\sqrt{\pi}} \sum_{k=1}^{n} k^{1-H}
\]
\[
\Box
\]
We note from the proof of Lemma 4 that for \( l \geq N(n) \),

\[
q(n, l) \leq q(n, l + 1) \leq \cdots \leq q(n),
\]

\[
q(n, l) + h(l) \geq q(n, l + 1) + h(l + 1) \geq \cdots \geq q(n),
\]

and

\[
\lim_{l \to \infty} q(n, l) = \lim_{l \to \infty} q(n, l) + h(l) = q(n).
\]

Thus, we have constructed a sequence of lower and upper bounds for \( q(n) \).

The above analysis directly provides us with an Algorithm to check whether we have reached a downward crossing milestone event (Algorithm 2) and to sample \( J \sim \text{Bernoulli}(q(n)) \) (Algorithm 3).

**Algorithm 2** Given \( n \) and \( S_n \), check whether \( p(n) < 3/4 \).

0. Calculate \( N(n) \) and set \( l = N(n) \).
1. Sample \( U_0 \sim \text{Uniform}[1/2, 3/4] \).
2. Calculate \( L(l) = q(n, l) \) and \( U(l) = q(n, l) + h(l) \).
3. If \( U_0 \leq L(l) \), set \( W = 1 \), and go to Step 4;
   - if \( U_0 \geq U(l) \), set \( W = 0 \) and go to Step 4;
   - otherwise, set \( l = l + 1 \) and go to Step 2.
4. Output \( W \).

Note that in Algorithm 2, we do not compare \( p(n) \) to 3/4 directly. Instead, we compare \( p(n) \) to a uniform random variable on \([1/2, 3/4]\), \( U_0 \). In particular, if \( U_0 > p(n) \), then we know that \( p(n) < 3/4 \), i.e., the criteria for the downward crossing milestone event is met. This extra randomness turns out to be useful for the complexity analysis.

In Algorithm 3 when \( J = 1 \), it also outputs \( N = l \). It is easy to check that

\[
\mathbb{P}(N = l | J = 1) = \frac{\mathbb{P}(S_l > M_n)}{\sum_{k=n+1}^{\infty} \mathbb{P}(S_k > M_n)} = f_n(l).
\]

Thus, as a byproduct of Algorithm 3 we also get a sample of \( N \)
Algorithm 3 Given \( n \) and \( S_n \), sample Bernoulli(\( q(n) \)).

0. Calculate \( \mathcal{N}(n) \) and set \( l = \mathcal{N}(n) \).

1. Sample \( U \sim \text{Uniform}[0,1] \).

2. Calculate \( L(l) = q(n,l) \) and \( U(l) = q(n,l) + h(l) \).

3. If \( U \leq L(l) \), set \( J = 1 \), \( N = l \), and go to Step 4; if \( U \geq U(l) \), set \( J = 0 \) and go to Step 4; otherwise, set \( l = l + 1 \) and go to Step 2.

4. If \( J = 1 \), output \( J \) and \( N = l \); otherwise, output \( J \).

Next, we develop the algorithm to sample Bernoulli(\( p(k) \)) (Algorithm 4).

Recall that \( p(k) = (\sum_{i=k}^{\infty} P_k(S_i > b))^{-1} \) for some fixed \( b \geq 0 \). Define

\[
q_b(k,l) := \sum_{i=k+1}^{l} P_k(S_i > b)
\]

We note from the proof of Lemma 4 that for any \( b \geq 0 \), we have that for \( l \geq \mathcal{N}(k) \),

\[
\sum_{i=k+1}^{l} P_k(S_i > b) \leq \sum_{i=k+1}^{\infty} P_k(S_i > b) \leq \sum_{i=k+1}^{l} P_k(S_i > b) + h(l).
\]

Thus, for \( l \geq \mathcal{N}(k) \),

\[
\frac{1}{1 + q_b(k,l) + h(l)} \leq p(k) \leq \frac{1}{1 + q_b(k,l)}.
\]

Then, Algorithm 4 follows from the same sandwiching construction.

4. Complexity analysis

In this section, we conduct detailed complexity analysis of Algorithm 1. We start by proving Theorem 1. We define \( \tilde{T} \) as the last level \( n \) sampled in Algorithm 1. Due to the extra randomization introduced in Algorithm 2, \( T \leq \tilde{T} \). We next show that for any \( \eta > 0 \), \( \mathbb{E}[\tilde{T}^\eta] < \infty \).

Proof. |Proof of Theorem 1| \( \tilde{T} \) includes the sample generated in two steps (Step 1 and 2 in Algorithm 2). In particular, we write \( \tilde{T} = \tilde{T}_1 + \tilde{T}_1 \), where \( \tilde{T}_i \) counts the number of total samples from Step \( i \), \( i = 1, 2 \).
Algorithm 4 Given $k$, $b$, and $S_k$ with $S_k > b$, sample Bernoulli($p(k)$).

0. Calculate $\mathcal{N}(k)$ and set $l = \mathcal{N}(k)$.

1. Sample $U \sim \text{Uniform}[0, 1]$.

2. Calculate $\mathcal{L}(l) = (q_b(n, l) + h(l) + 1)^{-1}$ and $\mathcal{U}(l) = q_b(n, l)^{-1}$.

3. If $U \leq \mathcal{L}(l)$, set $I = 1$ and go to Step 4;
   if $U \geq \mathcal{U}(l)$, set $I = 0$ and go to Step 4;
   otherwise, set $l = l + 1$ and go to Step 2.

4. Output $I$

For Step 1, we note from Lemma 1 that for $a = 1/2$, there exists a random integer $L$ such that for $n > L$, $q(n) < 1/2$. Thus for $n > L$, we will be able to proceed to Step 1.3 from Step 1.2 directly. When reaching Step 1.4, we need to sample $J \sim \text{Bernoulli}(q(n))$. If $J = 1$, we proceed to Step 2, but there is a chance of revisiting Step 1 after that. We note that at Step 1.4, $q(n) < 3/4$. Thus, the number of times we need to revisit Step 1 from Step 2 can be upper bounded by a Geometric random variable, $G$, with probability of success $3/4$.

The above analysis indicates that

$$\tilde{T}_1 \leq L + G.$$ 

As $L$ and $G$ both have finite moments of all orders, so is $\tilde{T}_1$.

For Step 2, if we have sampled $S_n$ when reaching Step 2, then the number of samples we generate in that iteration is $N(n) - n$. We first note that

$$P_n(N(n) = l) = \frac{P_n(S_l > M_n)}{q(n)} \leq 2P_n(S_l > M_n).$$

Next, following the same line of analysis as the proof of Lemma 1, we have for any $\delta > 0$, there exists $N_0 > 0$ such that for $l > N_0$,

$$P_n(S_l > M_n) \leq \Phi \left( \frac{\mu - \delta}{\sigma l^{1-H}} \right),$$

17
and for any \( \eta > 0 \), \( \mathbb{E}[N_0^\eta] < \infty \). Then for any \( \eta > 0 \),

\[
\mathbb{E}[N(n)^\eta|S_n] = \sum_{l=n+1}^{\infty} l^n \mathbb{P}_n(N(n) = l)
\]

\[
\leq \mathbb{E}[N_0^\eta|S_n] + \mathbb{E} \left[ \sum_{l=N_0+1}^{\infty} l^n \Phi \left( \frac{\mu - \delta}{\sigma} l^{1-H} \right) \bigg| S_n \right].
\]

Thus,

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

Lastly, note that we only visit Step 2 if \( J = 1 \) in Step 1.4. Thus, the number of times we visit Step 2 is upper bounded by \( G \). The above analysis indicates that \( \tilde{T}_2 \leq \sum_{i=1}^{G} N(n_i) \) where \( n_i \) is the number of samples we have generated at the \( i \)-th time we reach Step 2. As for fixed \( n \), \( N(n) \) has finite moments of all orders, \( \tilde{T}_2 \) also has finite moments of all orders.

Putting the two steps together, we have for any \( \eta > 0 \),

\[
\mathbb{E}[\tilde{T}^\eta] = \mathbb{E}[(\tilde{T}_1 + \tilde{T}_2)^\eta] < \infty.
\]

The above analysis does not take into account the sampling complexity of intermediate steps, i.e., the number of iterations in Algorithm 2–4. We next analyze the complexity of these intermediate steps. Based on the sandwiching construction, all three algorithms try to compare a uniform random variable with an unknown probability \( p \) using iteratively updated bounds for \( p \). We present the analysis for one of them (Algorithm 3) next. The others follow exactly the same line of analysis.

Given \( S_n \), let \( \Theta - n \) denote the number of iterations in Algorithm 3. In particular, \( \Theta = \inf\{l > n : U \leq q(n, l) \text{ or } U \geq q(n, l) + h(l)\} \), where \( U \sim \text{Uniform}[0,1] \).

Then for any \( l > N(n) \),

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

We first note that for any \( \eta > 0 \),

\[
\sum_{i=1}^{\infty} l^n h(l) < \infty.
\]
Next, from Lemma 4,
\[ \mathbb{E}[N(n)^n] < \infty. \]
Thus, \( \Theta \) has finite moments of all orders.

Similarly, we can show that the number of iterations in Algorithm 2 and 4 also have finite moments of all orders. Note that for the sandwiching construction in Algorithm 4, we also have
\[ \frac{1}{1 + q_b(n, l)} - \frac{1}{1 + q_b(n, l) + h(l)} \leq h(l). \]

Combining the complexity analysis of the intermediate steps with the proof of Theorem 1, we have that the overall computational complexity of Algorithm 1 has finite moments of all orders.

5. Numerical experiments

In this section, we implement our algorithm and test its performance on some examples. In particular, we consider the case where \( S_n = -n\mu + B^H(n) \) where \( B^H \) is a fractional Brownian motion with Hurst index \( H \in (0, 1) \). We set \( \mu = 1 \) and try three different values of \( H \): \( H = 0.3, 0.5, \) and \( 0.7 \). These correspond to the case where \( X_n \)'s are negatively correlated, uncorrelated, and positively correlated, respectively. Note that when \( H = 0.5 \), \( S_n \) is a discrete-time Brownian motion with drift \( \mu \).

In Figure 1, we plot the histogram of \( M \) based on \( 10^3 \) independent samples of it, i.e. we repeat our sampling procedure \( 10^3 \) times for each model. On the left panel, we plot the histogram of \( M \). We note that there is a very high probability that \( M = 0 \). On the right panel, we plot the histogram of \( M \) conditional on \( M > 0 \). 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.

We next look at the complexity of our algorithm. We analyze two quantities. One is the length of the sample path generated in Algorithm 1, which we denote
as $\tilde{T}$. This is a natural measure of the complexity. However, in actual implementations, we found that the most time-consuming part is the intermediate step – Algorithm 3 i.e., sampling a Bernoulli random variable with probability of success $q(n)$. Recall that the idea of Algorithm 3 is to sequential update $q(n,l)$ and $q(n,l) + h(l)$ for $l \geq N(n)$, until we can tell which side of $q(n)$ the uniform random variable lies. Thus, we also analyze $\Theta_m$, which is defined as the largest level $l$ reached in Algorithm 3.

In Figure 2, we plot the histogram of $\tilde{T}$ (left panel) and $\Theta_m$ (right panel) based on $10^3$ independent replications of Algorithm 1. We observe that as $H$ increases, the length of the sample path generated, $\tilde{T}$, also tend to take larger values. More importantly, as $H$ increases, the tail of distribution of $\Theta_m$ becomes heavier very rapidly. For example, when $H = 0.3$, $\Theta_m$ tend to take very small values, i.e., less than 14. However, when $H = 0.7$, the distribution of $\Theta_m$ has an extremely heavy tail, $m$ can take values as large as $5 \times 10^5$.

References


Figure 1: Histograms of $M$ where $S_n = -n + B^H(n)$ and $H = 0.3, 0.5, \text{ or } 0.7$. 
Figure 2: Histograms of sample path length generated in Algorithm 1 and maximal level reached in Algorithm 3: $S_n = -n + B^H(n)$ and $H = 0.3, 0.5,$ or $0.7.$