Malliavin-Based Multilevel Monte Carlo Estimators for Densities of Max-Stable Processes

Jose Blanchet and Zhipeng Liu

Abstract We introduce a class of unbiased Monte Carlo estimators for multivariate densities of max-stable fields generated by Gaussian processes. Our estimators take advantage of recent results on the exact simulation of max-stable fields combined with identities studied in the Malliavin calculus literature and ideas developed in the multilevel Monte Carlo literature. Our approach allows estimating multivariate densities of max-stable fields with precision ε at a computational cost of order $O\left(\varepsilon^{-2}\log\log\log\left(1/\varepsilon\right)\right)$.

Keywords Max-stable process · Density estimation · Malliavin calculus

1 Introduction

Max-stable random fields arise as the asymptotic limit of suitably normalized maxima of many i.i.d. random fields. Intuitively, max-stable fields are utilized to study the extreme behavior of spatial statistics. For instance, if the logarithm of a precipitation field during a relatively short time span follows a Gaussian random field, then extreme precipitations over a long time horizon (which are obtained by taking the maximum at each location of many precipitation fields) can be argued as long as enough temporal

J. Blanchet—Support from NSF grant DMS-132055, NSF grant CMMI-1538217 and NSF grant DMS-1720451 is gratefully acknowledged by J. Blanchet. We also are grateful to the referees for their comments which helped us improve our manuscript.

J. Blanchet (✉) · Z. Liu
Department of Industrial Engineering and Operations Research,
Columbia University, 500 West 120th Street, New York, NY, USA
e-mail: jose.blanchet@columbia.edu

Z. Liu
e-mail: zl2337@columbia.edu

A. B. Owen and P. W. Glynn (eds.), *Monte Carlo and Quasi-Monte Carlo Methods*, Springer Proceedings in Mathematics & Statistics 241,
https://doi.org/10.1007/978-3-319-91436-7_4

independence can be assumed to follow a suitable max-stable process. It is these types of applications in environmental science that motivate the study of max-stable processes (see, for example, [4] for a recent study of this type).

In order to estimate the parameters of a max-stable random field, for instance using maximum-likelihood estimation, it is desirable to evaluate the density over a finite set of locations (i.e. multivariate density of finite-dimensional coordinates of the max-stable field). The recent work in [3] discusses the challenges involved in applying maximum likelihood estimation of max-stable fields. We believe that the algorithms and techniques that we develop in this paper can be used to study maximum likelihood estimators for max-stable fields. For example, our representations can be used to obtain convenient expressions for the derivative of the density. In turn, as explained in [3], differentiability of the density is useful in the asymptotic analysis of maximum likelihood estimators. In addition, our algorithms can be implemented using common random numbers under a wide range of parameters of the max-stable fields. Thus, our algorithms can be used to perform approximate maximum-likelihood estimation by optimizing over a wide range of parameters.

In order to precisely explain our estimators, we now introduce some basic facts about max-stable processes.

We will focus on a class of max-stable random fields which are driven by Gaussian processes. These max-stable fields are popular in practice because their spatial dependence structure is inherited from the underlying Gaussian covariance structure.

To introduce the max-stable field of interest, let us first fix its domain $T \subseteq \mathbb{R}^m$, for $m \geq 1$. We introduce a sequence, $(X_n(\cdot))$, of independent and identically distributed copies of a centered Gaussian random field, $X(\cdot) = (X(t) : t \in T)$. We let (A_n) be the sequence of arrivals in a Poisson process, with unit rate and independent of $(X_n(\cdot))$.

Finally, given a deterministic and bounded function, $\mu : T \longrightarrow \mathbb{R}$, we will focus on developing Monte Carlo methods for the finite dimensional densities of the max-stable field

$$M(t) = \sup_{n \geq 1} \left\{ -\log A_n + X_n(t) + \mu(t) \right\}, \qquad t \in T. \tag{1}$$

(The name max-stable is justified because $M(\cdot)$ turns out to satisfy a distributional equation involving the maximum of i.i.d. centered and normalized copies of $M(\cdot)$, see [9, Theorem 2].)

An elegant argument involving Poisson point processes (see [15] and [2, Lemma 5.1]) allows us to conclude that

$$\begin{aligned} &P(M(t_1) \leq x_1, \ldots, M(t_d) \leq x_d) \\ &= \exp\left(-E\left[\max_{i=1}^{d} \{\exp(X(t_i) + \mu(t_i) - x_i)\} \right] \right). \end{aligned} \tag{2}$$

By redefining x_i as $x_i - \mu(t_i)$, we might assume without loss of generality, for the purpose of computing the density of $M = (M(t_1), \ldots, M(t_d))^T$, that $\mu(t_i) = 0$. We will keep imposing this assumption throughout the rest of the paper.

Throughout the paper we will keep the number of locations, d, over which $M(\cdot)$ is observed, fixed. So, M will remain a d-dimensional vector throughout our discussion. To avoid confusion between M and $M(\cdot)$, note that we use $M(\cdot)$ when discussing the whole max-stable field. We will maintain this convention throughout the rest of the paper for the field $M(\cdot)$ as well as the fields $X_n(\cdot)$, $n \geq 1$.

The joint density of M can be obtained by subsequent differentiation of (2) with respect to $x_1, \ldots, x_d$. However, the final expression obtained for the density contains exponentially many terms. So, computing the density of M using this direct approach becomes quickly intractable, even for moderate values of d. For example, [15] argues that even for $d = 10$, one obtains a sum of more than 10^5 terms.

We will construct an unbiased estimator for the density, $f(x)$, of M evaluated at $x = (x_1, \ldots, x_d)$ for $d \geq 3$. The construction of our estimator, denoted as $V(x)$, is explained in Sect. 2.5. Implementing our estimator avoids the exponential growth issues which arise if one attempts to directly evaluate the density. We concentrate on $d \geq 3$ because the case $d = 2$ leads to only four terms which can be easily computed as explained in [8]. More precisely, our contributions are as follows:

1. The properties of $V(x)$ are summarized in Sect. 3. In particular as shown in (16), $f(x) = E(V(x))$, $Var(V(x)) < \infty$, and given a computational budget of size b, we provide a limit theorem which can be used to estimate $f(x)$ with complexity $O\left((b \cdot \log\log\log(b))^2\right)$ for an error of order $O(1/b)$ – see Theorem 1 and its discussion.
2. As far as we know, this is the first estimator which uses Malliavin calculus in the context of max-stable density estimation. We believe that the techniques that we introduce are of independent interest in other areas in which Malliavin calculus has been used to construct Monte Carlo estimators. For example, we highlight the following techniques in this regard,
   a. We introduce a technique which can be used to estimate the density of the (coordinate-wise) maximum of multivariate variables. We apply this technique to the case of independent Gaussian vectors, but the technique can be used more generally, see the development in Sect. 2.2.
   b. We explain how to extend the technique in item 2(a) to the case of the maxima of infinitely many variables. This extension, which is explained in Sect. 2.3, highlights the role of a recently introduced record-breaking technique for the exact sampling of variables such as M.
   c. We introduce a perturbation technique which controls the variance of so-called Malliavin–Thalmaier estimators (which are explained in Sect. 2.1). These types of estimators have been used to compute densities of multivariate diffusions (see [10]). Our perturbation technique, introduced in Sect. 2.4, can be directly used to improve upon the density estimators in [10], enabling a close-to-optimal Monte Carlo rate of convergence for density estimation of multivariate diffusions.
3. The perturbation technique in Sect. 2.4 is combined with randomized multilevel Monte Carlo techniques (see [13, 14]) in order to achieve the following: Starting

from an infinite variance estimator, we introduce a perturbation which makes the estimator biased, but with finite variance. The randomized multilevel Monte Carlo technique is then used to remove the bias while keeping the variance finite. The price to pay is a small degradation in the rate of convergence in the associated Central Limit Theorem for confidence interval estimation. Instead of an error rate of order $O(1/b^{1/2})$ as a function of the computational budget b, which is the typical rate, we obtain a rate of order $O\left((\log\log\log(b))^{1/2}/b^{1/2}\right)$. The Central Limit Theorem is obtained using recently developed results in [18].

The rest of the paper is organized as follows: In Sect. 2 we explain step-by-step, at a high level, the construction of our estimator. The final form of our estimator is given in Sect. 2.5. The properties of our estimator are summarized in Sect. 3. A numerical experiment is given in Sect. 4. Finally, the details of the implementation of our estimator, in the form of pseudo-codes, are given in the Sect. 5 Appendix.

2 General Strategy and Background

The general strategy is explained in several steps. We first review the Malliavin–Thalmaier identity by providing a brief explanation of its origins and connections to classical potential theory. We finish the first step by noting that there are several disadvantages of the identity having to do with variance properties of the estimator and the implicit assumption that a great degree of information is assumed about the density which we want to estimate. The subsequent steps in our construction are designed to address these disadvantages.

In the second step of our construction, we introduce a series of manipulations which enable the application of the Malliavin–Thalmaier indirectly, by working only with the X_n's. These manipulations are performed assuming that only finitely many Gaussian elements are considered in the description of M.

The third step deals with the fact that the description of M contains infinitely many Gaussian elements. So, first, we need to explain how to sample M exactly. We utilize a recently developed algorithm (see [11]). Based on this algorithm, we explain how to extend the construction from the second step in order to obtain a direct Malliavin–Thalmaier estimator for the density of M.

The fourth step of our construction deals with the fact that a direct Malliavin–Thalmaier estimator will generally have infinite variance. We introduce a small random perturbation to remove the singularity appearing in the Malliavin–Thalmaier estimator, which is the source of the poor variance performance. Unfortunately, such a perturbation also introduces bias in the estimator.

In order to remove the bias, we then apply randomized multilevel Monte Carlo methods (see [13, 14]). Our resulting estimator then is unbiased and has finite variance as we explain in Sect. 3. The price to pay is a small degradation in the rate of convergence of the associated Central Limit Theorem to obtain confidence intervals.

2.1 Step 1: The Malliavin–Thalmaier Identity

The initial idea behind the construction of our estimator comes from the Malliavin–Thalmaier identity, which we shall briefly explain. First, recall the Newtonian potential, given by

$$G(x) = \kappa_d \frac{1}{\|x\|_2^{d-2}},$$

with $\kappa_d = (d(2-d)\omega_d)^{-1}$, where ω_d is the volume of a unit ball in d dimensions, for $d \geq 3$. It is well known, see [5], that $G(\cdot)$ satisfies the equation

$$\Delta G(x-y) = \delta(x-y)$$

in the sense of distributions (where $\delta(x)$ is the delta function). Therefore, if $M \in R^d$ has density $f(\cdot)$ we can write

$$f(x) = \int f(y)\, \Delta G(x-y)\, dy = E(\Delta G(x-M)). \tag{3}$$

But the previous identity cannot be implemented directly because $G(\cdot)$ is harmonic, that is, one can easily verify that $\Delta G(x) = 0$ for $x \neq 0$ (which is not surprising given that one expects ΔG to act as a delta function). The key insight of Malliavin and Thalmaier is to apply integration by parts in the expression (3). So, let us define

$$G_i(x) = \frac{\partial G(x)}{\partial x_i} = (2-d)\kappa_d \frac{x_i}{\|x\|_2^d},$$

and therefore write

$$\Delta G(x-y) = \sum_{i=1}^{d} \frac{\partial^2 G(x-y)}{\partial x_i^2} = \sum_{i=1}^{d} \frac{\partial G_i(x-y)}{\partial x_i}.$$

Consequently, because

$$\frac{\partial G_i(x-y)}{\partial x_i} = -\frac{\partial G_i(x-y)}{\partial y_i},$$

we have that

$$\begin{aligned} E\left(\frac{\partial G_i(x-M)}{\partial x_i}\right) &= \int \cdots \int \frac{\partial G_i(x-y)}{\partial x_i} f(y_1, \ldots, y_d)\, dy_1 dy_2 \ldots dy_d \\ &= -\int \cdots \int \frac{\partial G_i(x-y)}{\partial y_i} f(y_1, \ldots, y_d)\, dy_1 dy_2 \ldots dy_d \end{aligned}$$

$$= \int \cdots \int G_i(x-y) \frac{\partial f(y_1, \ldots, y_d)}{\partial y_i} dy_1 dy_2 \ldots dy_d$$
$$= E\left(G_i(x-M) \frac{\partial}{\partial y_i} \log f(M) \right).$$

Therefore, using (3), we arrive at the following Malliavin–Thalmaier identity,

$$f(x) = E\left(\sum_{i=1}^{d} \frac{\partial G_i(x-M)}{\partial x_i} \right) = \sum_{i=1}^{d} E\left(G_i(x-M) \frac{\partial}{\partial y_i} \log f(M) \right). \quad (4)$$

Refer to [10, 12] for rigorous proof of this identity.

There are two immediate concerns when applying the Malliavin–Thalmaier identity. First, a direct use of the identity requires some basic knowledge of the density of interest, which is precisely the quantity that we wish to estimate. The second issue, which is not evident from (4), is that the singularity which arises when $x = M$ in the definition of $G_i(x-M)$, causes the estimator (4) to typically have infinite variance.

2.2 *Step 2: Applying the Malliavin–Thalmaier Identity to Finite Maxima*

We now shall explain how to address the first issue discussed at the end of the previous subsection. Define

$$M_n(t) = \max_{1 \le k \le n} \{-\log(A_k) + X_k(t)\},$$

where X_k's are i.i.d. centered Gaussian vectors with covariance matrix Σ, and put $M_n = (M_n(t_1), \ldots, M_n(t_d))^T$. Note that

$$\frac{\partial G_i(x-M_n)}{\partial x_i} = -\frac{\partial G_i(x-M_n)}{\partial M_n(t_i)}. \quad (5)$$

In turn, by the chain rule,

$$\frac{\partial G_i(x-M_n)}{\partial X_k(t_i)} = \frac{\partial G_i(x-M_n)}{\partial M_n(t_i)} \frac{\partial M_n(t_i)}{\partial X_k(t_i)}. \quad (6)$$

Further, with probability one (due to the fact that $(A_1, A_2, \ldots, A_k)$ has a density),

$$\sum_{k=1}^{n} \frac{\partial M_n(t_i)}{\partial X_k(t_i)} = \sum_{k=1}^{n} I(M_n(t_i) = X_k(t_i) - \log(A_k)) = 1.$$

Consequently, from Eq. (6) we conclude that

$$\sum_{k=1}^{n} \frac{\partial G_i\left(x - M_n\right)}{\partial X_k\left(t_i\right)} = \frac{\partial G_i\left(x - M_n\right)}{\partial M_n\left(t_i\right)},$$

and therefore, from (5), we obtain

$$\frac{\partial G_i\left(x - M_n\right)}{\partial x_i} = -\sum_{k=1}^{n} \frac{\partial G_i\left(x - M_n\right)}{\partial X_k\left(t_i\right)}.$$

We now can apply integration by parts as we did in our derivation of (4). The difference is that the density of $X_k = \left(X_k\left(t_1\right), \ldots, X_k\left(t_d\right)\right)^T$ is known and therefore we obtain that

$$E\left(\frac{\partial G_i\left(x - M_n\right)}{\partial X_k\left(t_i\right)}\right) = E\left(G_i\left(x - M_n\right) \cdot e_i^T \Sigma^{-1} X_k\right),$$

where e_i is the ith vector in the canonical basis in Euclidean space.

Consequently, we conclude that

$$\begin{aligned} E\left(\frac{\partial G_i\left(x - M_n\right)}{\partial x_i}\right) &= -\sum_{k=1}^{n} E\left(\frac{\partial G_i\left(x - M_n\right)}{\partial X_k\left(t_i\right)}\right) \\ &= -E\left(G_i\left(x - M_n\right) \cdot e_i^T \Sigma^{-1} \sum_{k=1}^{n} X_k\right). \end{aligned}$$

In summary, if f_n is the density of M_n we have that

$$\begin{aligned} f_n\left(x_1, \ldots, x_d\right) &= E\left(\sum_{i=1}^{d} \frac{\partial G_i\left(x - M_n\right)}{\partial x_i}\right) \\ &= -E\left(\sum_{i=1}^{d} \sum_{k=1}^{n} G_i\left(x - M_n\right) \cdot e_i^T \Sigma^{-1} X_k\right). \end{aligned} \tag{7}$$

The verification of this identity follows a very similar argument as that provided for the proof of (4) in [12].

2.3 *Step 3: Extending the Malliavin–Thalmaier Identity to Infinite Maxima*

In order to extend the definition of the estimator (7), we wish to send $n \to \infty$ and obtain a simulatable expression of an estimator. Because we will be using a recently

developed estimator for M in [11], we need to impose the following assumptions on $X_n(\cdot)$.

(B1) In addition to assuming $E[X_n(t)] = 0$, we write $\sigma^2(t) = Var(X(t))$.
(B2) Assume that $\bar{\sigma} = \sup_{t\in T} \sigma(t) < \infty$.
(B3) Suppose that $E\left(\exp\left(\sup_{t\in T} X(t)\right)\right) < \infty$.

A key element of the algorithm in [11] is the idea of record breakers. The general idea is to utilize the properties of those record breakers to construct a Malliavin–Thalmaier estimator for the infinite maxima with finite but random number of Gaussian vectors. In order to describe this idea, let us write $\|X_n\|_\infty = \max_{i=1,\ldots,d} |X_n(t_i)|$.

Following the development in [11], we can identify three random times as follows.

The first is $N_X = N_X(a) < \infty$, defined for any $a \in (0, 1)$, and satisfying that for all $n > N_X$,

$$\|X_n\|_\infty \leq a \log n.$$

The time N_X is finite with probability one because $\|X_n\|_\infty$ is well known to grow at rate $O_p\left((\log n)^{1/2}\right)$ as $n \to \infty$.

The second is $N_A = N_A(\gamma) < \infty$ chosen for any given $\gamma < E(A_1)$, satisfying that for $n > N_A$

$$A_n \geq \gamma n. \tag{8}$$

The time N_A is finite with probability one because of the Strong Law of Large Numbers.

The third is N_a such that, for all $n > N_a$, we have

$$n\gamma \geq A_1 n^a \exp(\|X_1\|_\infty). \tag{9}$$

It is immediate that N_a is finite almost surely because $a \in (0, 1)$.

By successively applying the preceding three equations, we find that for $n > N := \max(N_A, N_X, N_a)$ and any $t = t_1, \ldots, t_d$, we have

$$\begin{aligned} -\log A_n + X_n(t) &\leq -\log A_n + \|X_n\|_\infty \\ &\leq -\log A_n + a \log n \\ &\leq -\log(n\gamma) + a \log n \\ &\leq -\log A_1 - \|X_1\|_\infty \leq -\log A_1 + X_1(t). \end{aligned}$$

Therefore, we conclude that, for $t = t_1, \ldots, t_d$,

$$\sup_{n\geq 1} \{-\log A_n + X_n(t)\} = \max_{1\leq n\leq N} \{-\log A_n + X_n(t)\}.$$

The work in [11] explains how to simulate the random variables N_X, N_A, and N_a, jointly with the sequence $(A_n)_{n\leq N}$ as well as $(X_n)_{n\leq N}$. Moreover, it is also shown in

[11] that the number of random variables required to simulate N_X, N_A and N_a (jointly with $X_1, \ldots, X_N$ and $A_1, \ldots, A_N$) has finite moments of any order. Therefore, N has finite moments of any order. Moreover, $E(N) = O(d^{\varepsilon})$ for any $\varepsilon > 0$. In the appendix, we reproduce the simulation procedure developed in [11].

Now, observe that conditional on $X_1, \ldots, X_{N_X}, N_X$, for $n > N_X$ the random vectors $(X_k)_{k \geq n}$ are independent, but they no longer follow a Gaussian distribution. Nevertheless, the X_n 's still have zero conditional means, given that $n > N$. This is because

$$\begin{aligned} & E\left(X_n \mid \|X_n\|_{\infty} \leq a \log n\right) \\ = & E\left(-X_n \mid \|-X_n\|_{\infty} \leq a \log n\right) = E\left(-X_n \mid \|X_n\|_{\infty} \leq a \log n\right). \end{aligned}$$

Consequently, we have that

$$E\left(e_i^T \Sigma^{-1} X_n | n > N\right) = 0.$$

Therefore, because M is independent of X_n conditional on $n > N$, we obtain that

$$\begin{aligned} & E\left(G_i\left(x - M\right) \cdot e_i^T \Sigma^{-1} X_n | n > N\right) \\ = & E\left(G_i\left(x - M\right) | n > N\right) \cdot E\left(e_i^T \Sigma^{-1} X_n | n > N\right) = 0. \end{aligned} \tag{10}$$

One can let $n \to \infty$ in (7) and formally apply (10) leading to the following result, which is rigorously established in [1].

Proposition 1 *For any* $(x_1, \ldots, x_d) \in R^d$,

$$f\left(x_1, \ldots, x_d\right) = -E\left(\sum_{i=1}^{d} \sum_{k=1}^{N} G_i\left(x - M\right) \cdot e_i^T \Sigma^{-1} X_k\right). \tag{11}$$

2.4 Step 4: Variance Control in Malliavin–Thalmaier Estimators

We now explain how to address the second issue discussed in Sect. 2.1, namely, controlling the variance when using the Malliavin–Thalmaier estimator (11).

Let us write

$$W\left(x\right) = -\sum_{i=1}^{d} \sum_{k=1}^{N} G_i\left(x - M\right) \cdot e_i^T \Sigma^{-1} X_k,$$

and observe that

$$W\left(x\right) = \frac{\left\langle M - x, \sum_{i=1}^{N} \Sigma^{-1} X_k \right\rangle}{d w_d ||M - x||^d}.$$

It turns out that the variance of $W(x)$ blows up because of the singularity in the denominator when $M = x$. This is verified in [1], but a similar calculation is also given in the setting of diffusions in [10]. So instead, we consider an approximating sequence defined via $\bar{W}_0(x) = 0$, and

$$\bar{W}_n(x) = \frac{\left\langle M - x, \sum_{i=1}^N \Sigma^{-1} X_k \right\rangle}{dw_d ||M - x||^d + dw_d \delta_n ||M - x||}, \quad n \geq 1,$$

where

$$\delta_n = 1/\log\log\log\left(n + e^e\right). \tag{12}$$

It is apparent that $\lim_{n\to\infty} \bar{W}_n(x) = W(x)$ almost surely. The use of a perturbation in the denominator of the Malliavin–Thalmaier estimator is not new. In [10] also, a small positive perturbation in the denominator is added, but such perturbation is, in their case, deterministic. The difference here is that our perturbation contains the factor $\delta_n \|M - x\|$. We have chosen our perturbation in order to ultimately control both the variance and the bias of our estimator.

In order to quickly motivate the variance implications of our choice, note that

$$\left| \frac{\left\langle M - x, \sum_{i=1}^N \Sigma^{-1} X_k \right\rangle}{dw_d ||M - x||^d + dw_d \delta_n ||M - x||} \right| \leq \left| \frac{\left\langle M - x, \sum_{i=1}^N \Sigma^{-1} X_k \right\rangle}{dw_d \delta_n ||M - x||} \right| \leq \frac{1}{dw_d \delta_n} \left\| \sum_{i=1}^N \Sigma^{-1} X_k \right\|_2,$$

leading to a bound that does not explicitly contain M. Moreover, we mentioned before that N has finite moments of any order and X_k is normally distributed, therefore, one can easily verify that $\left\| \sum_{i=1}^N \Sigma^{-1} X_k \right\|_2$ has finite moments of any order, in particular finite second moment and therefore $\bar{W}_n(x)$ has finite variance.

The reader might wonder why choosing δ_n in the definition of $\bar{W}_n(x)$, because any function of n decreasing to zero will ensure the convergence almost surely of $\bar{W}_n(x)$ toward $W(x)$. The previous upper bound, although not sharp when n is large, might also hint to the fact that it is desirable to choose a slowly varying function of n in the denominator (at least the reader notices a bound which deteriorates slowly as n grows).

The precise reason for the selection of our perturbation in the denominator obeys to a detailed variance calculation which can be seen in [1]. A more in-depth discussion is given in Sect. 2.5 below. For the moment, let us continue with our development in order to give the final form of our estimator.

Even though $\bar{W}_n(x)$ has finite variance and is close to $W(x)$, unfortunately, we have that $\bar{W}_n(x)$ is no longer an unbiased estimator of $f(x)$. In order to remove the bias, we take advantage of a randomization idea from [13, 14], which is related to the multilevel Monte Carlo method in [7], as we shall explain next.

2.5 Final Form of Our Estimator

Let us define $\bar{W}_0(x) = 0$ and for $n \geq 1$ let us write

$$\Delta_n(x) = \bar{W}_n(x) - \bar{W}_{n-1}(x).$$

In order to facilitate the variance analysis of our randomized multilevel Monte Carlo estimator, we further consider a sequence $\left(\bar{\Delta}_n(x)\right)_{n\geq 1}$ of independent random variables so that $\Delta_n(x)$ and $\bar{\Delta}_n(x)$ are equal in distribution.

We let L be a random variable taking values on $n \geq 1$, independent of everything else. Moreover, we let $g(n) = P(L \geq n)$ and assume that

$$g(n) = n^{-1}(\log(n+e-1))^{-1}\left(\log\left(\log\left(n+e^e-1\right)\right)\right)^{-1}. \tag{13}$$

Then, the final form of our estimator is

$$V(x) = \sum_{k=1}^{L} \frac{\bar{\Delta}_k(x)}{g(k)}. \tag{14}$$

The randomized multilevel Monte Carlo idea applied formally yields that

$$\begin{aligned} E(V(x)) &= E\left(\sum_{k=1}^{\infty} \frac{\bar{\Delta}_k(x)\, I(L \geq k)}{g(k)}\right) = \sum_{k=1}^{\infty} E\left(\frac{\bar{\Delta}_k(x)\, I(L \geq k)}{g(k)}\right) \\ &= \sum_{k=1}^{\infty} E\left(\bar{\Delta}_k(x)\right) = E(W(x)) - E\left(\bar{W}_0(x)\right) = E(W(x)) = f(x). \end{aligned} \tag{15}$$

In order to make the previous manipulations rigorous, we must justify exchanging the summation in (15). In addition, we also need to guarantee that $V(x)$ has finite variance. These and other properties will be used to obtain confidence intervals for our estimates, given a computational budget via CLT for renewal processes. In turn, it suffices to make sure the following two conditions hold:

(C1) $\sum_{k\geq 1} E\left(\left|\bar{\Delta}_k(x)\right|\right) < \infty$,

(C2) $\sum_{n\geq 1} \frac{E\left(\left|\bar{\Delta}_n(0)\right|^2\right)}{g(n)} < \infty$.

We shall summarize the properties of $V(x)$ in our main result given in the next section. In particular, we will show that (C1) and (C2) hold with our choice of δ_n and $g(n)$. We also provide a discussion of the running time analysis, which is affected by the choice of $g(n)$.

3 Main Result

Our main contribution is summarized in the following result, which is fully proved in [1]. Our objective now is to give the gist of the technical development in order to have at least an intuitive understanding of the choices behind the design of our estimator (14). We measure computational cost in terms of the elementary random variables simulated.

Theorem 1 *Let ρ be the cost required to generate M so that $V(x)$, defined in (14), has a computational cost equal to $C = \sum_{i=1}^{L} \rho_i + 1$ (where L is independent of $\rho_1, \rho_2, \ldots$, which are i.i.d. copies of ρ). Let $(V_1(x), C_1), (V_2(x), C_2), \ldots$ be i.i.d. copies of $(V(x), C)$ and set $T_n = C_1 + \cdots + C_n$ with $T_0 = 0$. For each $b > 0$ define, $B(b) = \max\{n \geq 0 : T_n \leq b\}$, then we have that*

$$f(x) = E(V(x)) \text{ and } Var(V(x)) < \infty. \tag{16}$$

Moreover,

$$\sqrt{\frac{b}{E(\rho_1) \cdot \log\log\log(b)}} \left(\frac{1}{B(b)} \sum_{i=1}^{B(b)} V_i(x) - f(x) \right) \Rightarrow N(0, Var(V(x))).$$

Before we discuss the analysis of the proof of Theorem 1, it is instructive to note that the previous result can be used to obtain confidence intervals for the value of the density $f(x)$ with precision ε at a computational cost of order $O\left(\varepsilon^{-2} \log\log\log(1/\varepsilon)\right)$, given a fixed confidence level (see Sect. 4 for an example of how to produce such a confidence interval).

The quantity $B(b)$ denotes the number of i.i.d. copies of $V(x)$ which can be simulated with a computational budget b, so the pointwise estimator given in Theorem 1 simply is the empirical average of $B(b)$ i.i.d. copies of $V(x)$.

The rate of convergence implied by Theorem 1 is, for all practical purposes, the same as the highly desirable canonical rate $O\left(\varepsilon^{-2}\right)$, which is rarely achieved in complex density estimation problems, such as the one that we consider in this paper.

3.1 Sketching the Proof of Theorem 1

At the heart of the proof of Theorem 1 lies a bound on the size of $|\Delta_n(x)|$. For notational simplicity, let us concentrate on $|\Delta_n(0)|$ and note that for any $\beta \geq 1$

$$|\Delta_n(0)|^\beta \leq \frac{\left\|\Sigma^{-1}\right\|^\beta}{(dw_d)^\beta} \left(\sum_{i=1}^{N} \|X_k\| \right)^\beta \times \left| \frac{\|M\|}{||M||^d + ||M||\delta_{n+1}} - \frac{\|M\|}{||M||^d + ||M||\delta_n} \right|^\beta. \tag{17}$$

We have argued that, because N has finite moments of any order, the random variable $\sum_{i=1}^{N} \|X_k\|_2$ is easily seen to have finite moments of any order. So, after applying Hölder's inequality to the right-hand-side of (17), it suffices to concentrate on estimating, for any $q > 1$,

$$E\left(\left|\frac{1}{||M||^{d-1}+\delta_{n+1}}-\frac{1}{||M||^{d-1}+\delta_n}\right|^{\beta q}\right)^{1/q}$$
$$=E\left(\left|\frac{\delta_n-\delta_{n+1}}{\left(||M||^{d-1}+\delta_{n+1}\right)\left(||M||^{d-1}+\delta_n\right)}\right|^{\beta q}\right)^{1/q}.$$

Let us define

$$a\left(n\right):=\delta_n-\delta_{n+1}\sim\delta_n^2\frac{1}{\log\log\left(n\right)\cdot\log\left(n\right)\cdot n}, \tag{18}$$

and focus on

$$D_{n,\beta}\left(0\right):=E\left(\left|\frac{1}{\left(||M||^{d-1}+\delta_{n+1}\right)\left(||M||^{d-1}+\delta_n\right)}\right|^{\beta q}\right)^{1/q}. \tag{19}$$

Assuming that M has a continuous density in a neighborhood of the origin (a fact which can be shown, for example, from (2), using the Gaussian property of the X_ns), we can directly analyze (19) using a polar coordinates transformation, obtaining that for some $\kappa > 0$

$$D_{n,\beta}^{q}\left(0\right)\leq\kappa\int_0^\infty\int_{\theta\in\mathscr{S}^{d-1}}\frac{f\left(r\cdot\theta\right)r^{d-1}}{\left(r^{d-1}+\delta_{n+1}\right)^{\beta q}\left(r^{d-1}+\delta_n\right)^{\beta q}}drd\theta, \tag{20}$$

where $\mathscr{S}^{d-1}$ represents the surface of the unit ball in d dimensions. Further study of the decay properties of $f\left(r\cdot\theta\right)$ as r grows large, uniformly over $\theta\in\mathscr{S}^{d-1}$, allows us to conclude that

$$D_{n,\beta}^{q}\left(0\right)\leq\kappa'\int_0^\infty\frac{r^{d-1}}{\left(r^{d-1}+\delta_{n+1}\right)^{\beta q}\left(r^{d-1}+\delta_n\right)^{\beta q}}dr, \tag{21}$$

for some $\kappa' > 0$. Applying the change of variables $r = u\delta_n^{1/(d-1)}$ to the right-hand side of (21), allows us to conclude, after elementary algebraic manipulations, that

$$D_{n,\beta}^{q}\left(0\right)=O\left(\delta_n^{d/(d-1)-2\beta q}\right),$$

therefore concluding that

$$E\left(|\Delta_n(0)|^\beta\right) = O\left(\left(\frac{\delta_n - \delta_{n+1}}{\delta_n^2}\right)^\beta \delta_n^{d/(q(d-1))-2\beta}\right). \tag{22}$$

Setting $\beta = 1$ we have (from (18) and the definition of δ_n) that

$$\sum_{n\geq 1} E\left(|\Delta_n(0)|\right) = O\left(\sum_{n\geq 1} \frac{1}{\log\log(n)\cdot\log(n)\cdot n}\delta_n^{d/(q(d-1))}\right) < \infty, \tag{23}$$

because $d/(d-1) > 1$ and $q > 1$ can be chosen arbitrarily close to one. This estimate justifies the formal development in (15) and the fact that $EV(x) = f(x)$.

Now, the analysis in [14] states that $Var(V(x)) < \infty$ if

$$\sum_{n\geq 1} \frac{E\left|\bar{\Delta}_n(0)\right|^2}{g(n)} < \infty. \tag{24}$$

Once again, using (22) and our choice of $g(n)$, we find that (24) holds because of the estimate

$$\sum_{n\geq 1} \frac{n\cdot\log(n)\cdot\log\log(n)}{(\log\log(n)\cdot\log(n)\cdot n)^2}\delta_n^{d/(q(d-1))} < \infty, \tag{25}$$

which is, after immediate cancellations, completely analogous to (23).

Finally, because the cost of sampling M (in terms of the number of elementary random variables, such as multivariate Gaussian random variables) has been shown to have finite moments of any order [11, Theorem 2.2], one can use standard results from the theory of regular variation (see [16, Theorem 3.2]) to conclude that

$$P\left(\sum_{i=1}^{L}\rho_i + 1 > t\right) \sim P\left(L > t/E(\rho_1)\right) \sim E(\rho_1)\, t^{-1}\log(t)^{-1}\log\log(t)^{-1},$$

as $t \to \infty$. Now, the form of the Central Limit Theorem is an immediate application of Theorem 1 in [18].

4 Numerical Examples

In this section, we implement our estimator and compare it against a conventional kernel density estimator. We measure the computational cost in terms of the number of independent samples drawn from **Algorithm M**. This convention translates into assuming that $E(\rho_1) = 1$ in Theorem 1. Given a computational budget b, the estimated density is given by

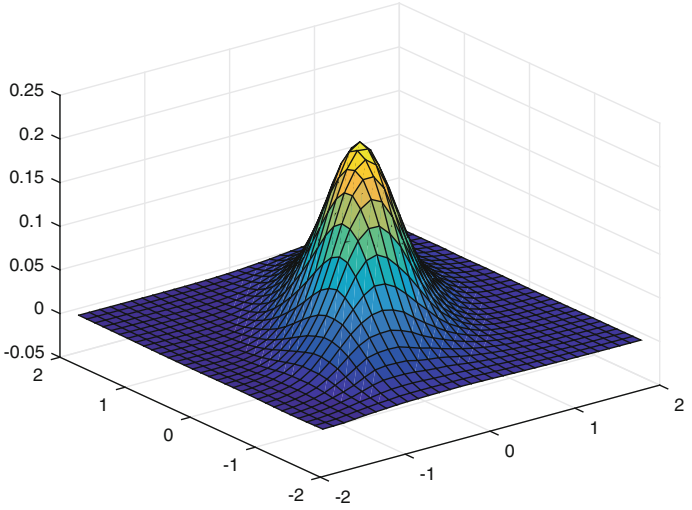

Fig. 1 The estimated three-dimensional joint density of a max-stable process using our algorithm

$$\hat{f}_b(x) = \frac{\sum_{i=1}^{B(b)} V_i(x)}{B(b)}.$$

According to Theorem 1, we can construct the confidence interval for underlying density $f(x)$ with significance level α as

$$\left(\hat{f}_b(x) - z_{\alpha/2}\hat{s}\sqrt{a(b)},\ \hat{f}_b(y) + z_{\alpha/2}\hat{s}\sqrt{a(b)}\right),$$

where $z_{\alpha/2}$ is the quantile corresponding to the $1-\alpha/2$ percentile,

$$\hat{s}^2 = \frac{\sum_{i=1}^{B(b)} \left(V_i(x) - \hat{f}_b(x)\right)^2}{B(b)},$$

and

$$a(b) = \frac{\log\log\log(b)}{b}.$$

We perform our algorithm to estimate the density of the max-stable process. We assume that $T = [0, 1]$ and $X_n(\cdot)$ is a standard Brownian motion. We are interested in estimating the density of $M = (M(1/3), M(2/3), M(1))^T$. That is, the spatial grid is $(1/3, 2/3, 1)$. The graph in Fig. 1 shows a plot of the density on the set $\{x \in \mathbb{R}^3 : x_1 \in (-2, 2), x_2 \in (-2, 2), x_3 = 0\}$. Our estimation of this three-dimensional density has a computation budget of $B = 10^6$ samples from **Algorithm M**.

Table 1 Density estimation with our algorithm

Values (x)	(0, 0, 0)	(0, 0.5, 0)	(0.5, 0, 0)	(0, −0.5, 0)	(−0.5, 0, 0)
est. density $\hat{f}_b(x)$	0.2126	0.106	0.1292	0.1039	0.1439
Lower CI	0.1916	0.0971	0.1180	0.0947	0.1311
Upper CI	0.2336	0.1149	0.14036	0.1131	0.1567
Relative error	5.05%	4.29%	4.41%	4.54%	4.53%

Table 2 Density estimation with KDE

Values (x)	(0,0,0)	(0,0.5,0)	(0.5,0,0)	(0,−0.5,0)	(-0.5,0,0)
est. density $\hat{f}_b^{KDE}(x)$	0.2163	0.0846	0.1143	0.0938	0.1084
Lower CI	0.1953	0.0712	0.0999	0.0800	0.0934
Upper CI	0.2373	0.0980	0.1287	0.1076	0.1234
Relative error	4.94%	8.07%	6.43%	7.51%	7.05%

We calculate the 95% confidence interval of the density on several selected values of the process $M(\cdot)$.

As a comparison, we also calculate the 95% confidence interval of the density using the plug-in kernel density estimation (KDE) method with the same amount ($b = 10^6$) of i.i.d. samples of M. We use the normal density function as the kernel function and select the bandwidth according to [17]. The estimator is obtained as follows. Sample $M^{(1)}, M^{(2)}, \ldots, M^{(d)}$ i.i.d. copies of M, let $h_b = b^{-1/(2d+1)}$ and compute the sample covariance matrix, $\hat{\Sigma}$, based on $(M^{(1)}, M^{(2)}, \ldots, M^{(d)})$. Then, let

$$\hat{f}_b^{KDE}(x) = \frac{1}{bh_b^d} \sum_{i=1}^{b} \phi\left(A^{-\frac{1}{2}} \frac{x - M^{(i)}}{h_b}\right),$$

where $A = \hat{\Sigma}/\det|\hat{\Sigma}|$. We apply the method from [6] to evaluate the corresponding confidence interval, thereby obtaining the estimates shown in Tables 1 and 2.

From the above tables, we can see that our algorithm provides similar estimates to those obtained using the KDE. However, our estimator also has a smaller relative error when the estimated value is relatively small. Also, as discussed in [6], KDE is biased, and one must carefully choose the bandwidth to obtain the optimal convergence rate of mean squared error. In contrast, the construction of confidence intervals with our estimator is straightforward, with a significantly better convergence rate.

5 Appendix: A Detailed Algorithmic Implementation

In order to make this paper as self-contained as possible, we reproduce here the algorithms from [11] which allow us to simulate the random variables N_X, N_A, and N_a, jointly with $(A_n)_{n\leq N}$ and $(X_n)_{n\leq N}$.

5.1 Simulating Last Passage Times of Random Walks

Define the random walk $S_n = \gamma n - A_n$ for $n \geq 0$. Note that $ES_n < 0$, by our choice of $\gamma < E(A_1)$. The authors in [11], argue that the choice of γ is not too consequential so we shall assume that $\gamma = 1/2$.

Here we review an algorithm from [11] for finding a random time N_S such that $S_n < 0$ for all $n > N_S$. Observe that $N_S = N_A$.

The algorithm is based on alternately sampling upcrossings and downcrossings of the level 0. We write $\xi_0^+ = 0$ and, for $i \geq 1$, we recursively define

$$\xi_i^- = \begin{cases} \inf\{n \geq \xi_{i-1}^+ : S_n < 0\} & \text{if } \xi_{i-1}^+ < \infty \\ \infty & \text{otherwise} \end{cases}$$

together with

$$\xi_i^+ = \begin{cases} \inf\{n \geq \xi_i^- : S_n \geq 0\} & \text{if } \xi_i^- < \infty \\ \infty & \text{otherwise.} \end{cases}$$

As usual, in these definitions, the infimum of an empty set should be interpreted as ∞. Writing

$$N_S = \sup\{\xi_n^- : \xi_n^- < \infty\},$$

we have by construction $S_n < 0$ for $n > N_S$. The random variable $N_S - 1$ is an upward last passage time:

$$N_S - 1 = \sup\{n \geq 0 : S_n \geq 0\}.$$

Note that $0 \leq N_S < \infty$ almost surely under P since $(S_n)_{n\geq 0}$ starts at the origin and has negative drift. We will provide pseudo-codes for simulating $(S_1, \ldots, S_{N_S+\ell})$ for any fixed $\ell \geq 0$, but first we need a few definitions.

First, we assume that the *Cramér's root*, $\theta > 0$, satisfying $E(\exp(\theta S_1)) = 1$ has been computed. We shall use $\mathbb{P}_x$ to denote the measure under which $(A_n)_{n\geq 1}$ are arrivals of a Poisson process with unit rate and $S_0 = x$. Then, we define P_x^θ through an exponential change of measure. In particular, on the σ-field generated by $S_1, \ldots, S_n$ we have

$$\frac{dP_x}{dP_x^\theta} = \exp(-\theta(S_n - x)).$$

It turns out that under P_x^θ, $(A_n)_{n\geq 1}$ corresponds to the arrivals of a Poisson process with rate $1-\theta$ and the random walk $(S_n)_{n\geq 1}$ has a positive drift.

To introduce the algorithm to sample $(S_1, \ldots, S_{N_S+\ell})$ we first need the following definitions:

$$\tau^- = \inf\{n \geq 0 : S_n < 0\}, \qquad \tau^+ = \inf\{n \geq 0 : S_n \geq 0\}.$$

For $x \geq 0$, it is immediate that we can sample a downcrossing segment $S_1, \ldots, S_{\tau^-}$ under P_x due to the negative drift, and we record this for later use in a pseudocode function. *Throughout our discussion, 'sample' in pseudocode stands for 'sample independently of anything that has been sampled already'.*

Function SAMPLEDOWNCROSSING(x)**: Samples** $(S_1, \ldots, S_{\tau^-})$ **under** P_x **for** $x \geq 0$
Step 1: Return sample $S_1, \ldots, S_{\tau^-}$ under P_x.
Step 2: EndFunction

Sampling an upcrossing segment is more interesting because it is possible that $\tau^+ = \infty$. So, an algorithm needs to be able to detect this event within a finite amount of computing resources. For this reason, we understand sampling an upcrossing segment under P_x for $x < 0$ to mean that an algorithm outputs $S_1, \ldots, S_{\tau^+}$ if $\tau^+ < \infty$, and otherwise it outputs 'degenerate'. The following pseudo-code samples an upcrossing under P_x for $x < 0$.

Function SAMPLEUPCROSSING(x): **Samples** $(S_1, \ldots, S_{\tau^+})$ **under** P_x **for** $x < 0$
Step 1: $S \leftarrow$ sample $S_1, \ldots, S_{\tau^+}$ under P_x^θ
Step 2: $U \leftarrow$ sample a standard uniform random variable
Step 3: If $U \leq \exp(-\theta(S_{\tau^+} - x))$
Step 4: Return S
Step 5: Else
Step 6: Return 'degenerate'
Step 7: EndIf
Step 8: EndFunction

We next describe how to sample $(S_k)_{k=1,\ldots,n}$ from P_x conditionally on $\tau^+ = \infty$ for $x < 0$. Since $\tau^+ = \infty$ is equivalent to $\sup_{k\leq \ell} S_k < 0$ and $\sup_{k>\ell} S_k < 0$ for any $\ell \geq 1$, after sampling $S_1, \ldots, S_\ell$, by the Markov property we can use SAMPLEUPCROSSING(S_ℓ) to verify whether or not $\sup_{k>\ell} S_k < 0$.

Function SAMPLEWITHOUTRECORDS(x, ℓ)**: Samples** $(S_k)_{k=1,\ldots,\ell}$ **from** P_x **given** $\tau^+ = \infty$ **for** $\ell \geq 1$, $x < 0$
Step 1: Repeat
Step 2: $S \leftarrow$ sample $(S_k)_{k=1,\ldots,\ell}$ under P_x
Step 3: Until $\sup_{1\leq k\leq \ell} S_k < 0$ and SAMPLEUPCROSSING (S_ℓ) is 'degenerate'
Step 4: Return S
Step 5: EndFunction

We summarize our discussion with the full algorithm for sampling $(S_0, \ldots, S_{N_S+\ell})$ under P given some $\ell \geq 0$.

Algorithm S: Samples $S = (S_0, \ldots, S_{N_S+\ell})$ ***under*** P ***for*** $\ell \geq 0$
We use S_{end} to denote the last element of S.
Step 1: $S \leftarrow [0]$
Step 2: Repeat
Step 3: DowncrossingSegment $\leftarrow$ SAMPLEDOWNCROSSING(S_{end})
Step 4: $S \leftarrow [S, \text{DowncrossingSegment}]$
Step 5: UpcrossingSegment $\leftarrow$SAMPLEUPCROSSING(S_{end})
Step 6: If UpcrossingSegment is not 'degenerate'
Step 7: $S \leftarrow [S, \text{upcrossingSegment}]$
Step 8: EndIf
Step 9: Until UpcrossingSegment is 'degenerate'
Step 10: If $\ell > 0$
Step 11: $S \leftarrow [S, \text{SAMPLEWITHOUTRECORDS}(S_{\text{end}}, \ell)]$
Step 12: EndIf

5.2 Simulating Last Passage Times for Maxima of Gaussian Vectors

The technique is similar to the random walk case using a sequence of record-breaking times. The parameter $a \in (0, 1)$ can be chosen arbitrarily, but [11] suggests selecting a such that

$$\exp\left(\frac{\overline{\sigma}}{a}\overline{\Phi}^{-1}\left(\delta\sqrt{2\pi}\frac{\phi(\overline{\sigma}/a)}{d\overline{\sigma}/a}\right)+\frac{\overline{\sigma}^2}{a^2}\right)=E\left[\left(\frac{A_1\,\exp(\|X\|_\infty)}{\gamma}\right)^{\frac{1}{1-a}}\right],$$

where $\Phi(\cdot)$ is the cumulative distribution function of a standard Gaussian random variable and $\overline{\Phi} = 1 - \Phi$.

Now, assume that $\eta_0 \geq 0$ is given (we will choose it specifically in the sequel). Let $(X_n)_{n\geq 1}$ be i.i.d. copies of X and define, for $i \geq 1$, a sequence of *record breaking times* (η_i) through

$$\eta_i = \begin{cases} \inf\{n > \eta_{i-1} : \|X_n\|_\infty > a \log n\} & \text{if } \eta_{i-1} < \infty \\ \infty & \text{otherwise.} \end{cases}.$$

We provide pseudo-codes which ultimately will allow us to sample $(X_1, \ldots, X_{N_X+\ell})$ for any fixed $\ell \geq 0$, where

$$N_X = \max\{\eta_i : \eta_i < \infty\}.$$

First, we shall discuss how to sample (X_n) up to a η_1. In order to sample η_1, $\eta_0 = n_0$ needs to be chosen so that $P(\|X\|_\infty > a \log n)$ is controlled for every $n > n_0$. Given the choice of $a \in (0, 1)$, select n_0 such that

$$d\overline{\Phi}\left(\frac{a \log n_0}{\overline{\sigma}} - \frac{\overline{\sigma}}{a}\right) \leq \frac{1}{2}\sqrt{\frac{\pi}{2}}\frac{\phi(\overline{\sigma}/a)}{\overline{\sigma}/a}.$$

Define

$$T_{n_0} = \inf\{k \geq 1 : \|X_k\|_\infty > a \log(n_0 + k)\}. \tag{26}$$

We describe an algorithm that outputs 'degenerate' if $T_{n_0} = \infty$ and $(X_1, \ldots, X_{T_{n_0}})$ if $T_{n_0} < \infty$.

First, we describe a simple algorithm to simulate from X conditioned on $\|X\|_\infty > a \log n$. Our algorithm makes use of a probability measure $P^{(n)}$ defined through

$$\frac{dP^{(n)}}{dP}(x) = \frac{\sum_{i=1}^d \mathbf{1}(|x(t_i)| > a \log n)}{\sum_{i=1}^d P(|X(t_i)| > a \log n)}.$$

It turns out that the measure $P^{(n)}$ approximates the conditional distribution of X given that $\|X\|_\infty > a \log n$ for n large.

Now, define $w^j(t) = \text{Cov}(X(t), X(t_j))/\text{Var}\left(X\left(t_j\right)\right)$ and note that $X(\cdot) - w^v(\cdot) X(t_v)$ is independent of $X(t_v)$ given v. This property is used in [11] to show that the following algorithm outputs from $P^{(n)}$. We will let U be a uniform random variable in $(0, 1)$ and J is independent of U and such that $P(J = 1) = 1/2 = P(J = -1)$.)

Function CONDITIONEDSAMPLEX (a, n)**: Samples** X **from** $P^{(n)}$

Step 1: $v \leftarrow$ sample with probability mass function

$$P(v = j) = \frac{P(\left|X(t_j)\right| > a \log n)}{\sum_{i=1}^d P(|X(t_i)| > a \log n)}$$

Step 2: $U \leftarrow$ sample a standard uniform random variable

Step 3: $X(t_v) \leftarrow \sigma(t_v) \cdot J \cdot \Phi^{-1}(U + (1 - U)\Phi(a(\log n)/\sigma(t_v)))$ #Conditions on $|X(t_v)| > a \log n$

Step 4: $Y \leftarrow$ sample of X under P

Step 5: Return $Y(t) - w^v(t)Y(t_v) + X(t_v)$

Step 6: EndFunction

We now explain how CONDITIONEDSAMPLEX is used to sample T_{n_0}. Define, for $k \geq 1$,

$$g_{n_0}(k) = \frac{\int_{k-1}^k \phi((a \log(n_0 + s))/\overline{\sigma})ds}{\int_0^\infty \phi((a \log(n_0 + s))/\overline{\sigma})ds},$$

where $\phi(x) = d\Phi(x)/dx$. Note that $g_{n_0}(\cdot) \geq 0$ defines the probability mass function of some random variable K. It turns out that if $U \sim U(0, 1)$ then we can sample

$$K = \left\lceil \exp\left\{ \frac{\overline{\sigma}^2}{a^2} + \frac{\overline{\sigma}}{a} \overline{\Phi}^{-1}\left(U\,\overline{\Phi}\left(\frac{a \log n_0}{\overline{\sigma}} - \frac{\overline{\sigma}}{a} \right) \right) \right\} - n_0 \right\rceil .$$

The next function samples $(X_1, \ldots, X_{T_{n_1}})$ for $n_1 \geq n_0$.

Function SAMPLESINGLERECORD (a, n_0, n_1)**: Samples** $(X_1, \ldots, X_{T_{n_1}})$ **for** $a \in (0, 1), n_1 \geq n_0 \geq 0$
Step 1: Sample K
Step 2: $[X_1, \ldots, X_{K-1}] \leftarrow$ i.i.d. sample from P
Step 3: $X_K \leftarrow$ CONDITIONEDSAMPLEX $(a, n_1 + K)$
Step 4: $U \leftarrow$ sample a standard uniform random variable
Step 5: If $\|X_k\|_\infty \leq a \log(n_1 + k)$ for $k = 1, \ldots, K-1$ and $U\, g_{n_0}(K) \leq dP/dP^{(n_1+K)}(X_K)$
Step 6: Return $(X_1, \ldots, X_K)$
Step 7: Else
Step 8: Return 'degenerate'
Step 9: EndIf
Step 10: EndFunction

We next describe how to sample $(X_k)_{k=1,\ldots,n}$ conditionally on $T_{n_0} = \infty$. This is a simple task because the X_ns are independent.

Function SAMPLEWITHOUTRECORDX (n_1, ℓ)**: Samples** $(X_k)_{k=1,\ldots,\ell}$ **conditionally on** $T_{n_1} = \infty$ for $\ell \geq 1$
Step 1: Repeat
Step 2: $X \leftarrow$ sample $(X_k)_{k=1,\ldots,\ell}$ under P
Step 3: Until $\sup_{1 \leq k \leq \ell}[X_k - a \log(n_1 + k)] < 0$
Step 4: Return X
Step 5: EndFunction

We now can explain how to sample $(X_1, \ldots, X_{N_X+\ell})$ under P given some $\ell \geq 0$. The idea is to successively apply SAMPLESINGLERECORD to generate the sequence $(\eta_i : i \geq 1)$ defined at the beginning of this section. Starting from $\eta_0 = n_0$, then n_1 is replaced by each of the subsequent η_is.

Algorithm X: Samples $(X_1, \ldots, X_{N_X+\ell})$ ***given*** $a \in (0, 1), \overline{\sigma} > 0, \ell \geq 0$
Step 1: $X \leftarrow [\,], \eta \leftarrow n_0$
Step 2: $X \leftarrow$ sample $(X_k)_{k=1,\ldots,\eta}$ under P
Step 3: Repeat
Step 4: segment $\leftarrow$ SAMPLESINGLERECORD (a, n_0, η)
Step 5: If segment is not 'degenerate'
Step 6: $X \leftarrow [X, \text{segment}]$
Step 7: $\eta \leftarrow \text{length}(X)$
Step 8: EndIf

Step 9: Until segment is 'degenerate'
Step 10: If $\ell > 0$
Step 11: $X \leftarrow [X, \text{SAMPLEWITHOUTRECORDX}(\eta, \ell)]$
Step 12: EndIf

5.3 *Algorithm to Sample $X_1, \ldots, X_N, N$*

The final algorithm for sampling $M, X_1, \ldots, X_N, N$ is given next.

Algorithm M: Samples $M, X_1, \ldots, X_N, N$ ***given*** $a \in (0, 1), \gamma < E(A_1)$, and $\overline{\sigma}$
Step 1: Sample $A_1, \ldots, A_{N_A}$ using Steps 1–9 from **Algorithm S** with $S_n = \gamma n - A_n$.
Step 2: Sample $X_1, \ldots, X_{N_X}$ using Steps 1–9 from **Algorithm X**.
Step 3: Calculate N_a with (9) and set $N = \max(N_A, N_X, N_a)$.
Step 4: If $N > N_A$
Step 5: Sample $A_{N_A+1}, \ldots, A_N$ as in Step 10–12 from **Algorithm S** with $S_n = \gamma n - A_n$.
Step 6: EndIf
Step 7: If $N > N_X$
Step 8: Sample $X_{N_X+1}, \ldots, X_N$ as in Step 10–12 from **Algorithm X**.
Step 9: EndIf
Step 10: Return $M(t_i) = \max_{1 \le n \le N} \{-\log A_n + X_n(t_i) + \mu(t_i)\}$ for $i = 1, \ldots, d$, and $X_1, \ldots, X_N, N$.

References

1. Blanchet, J.H., Liu, Z.: Efficient conditional density estimation for max-stable fields. Submitted
2. Dieker, A.B., Mikosch, T.: Exact simulation of Brown–Resnick random fields at a finite number of locations. Extremes **18**(2), 301–314 (2015)
3. Dombry, C., Engelke, S., Oesting, M.: Asymptotic properties of the maximum likelihood estimator for multivariate extreme value distributions. arXiv:1612.05178
4. Embrechts, P., Klüppelberg, C., Mikosch, T.: Modelling Extremal Events for Insurance and Finance. Springer, New York (1997)
5. Evans, L.C.: Partial Differential Equations: Second Edition. American Mathematical Society, Providence (2010)
6. Fiorio, C.V.: Confidence intervals for kernel density estimation. Stata J. **4**, 168–179 (2004)
7. Giles, M.B.: Multilevel Monte Carlo path simulation. Oper. Res. **56**(3), 607–617 (2008)
8. Huser, R., Davison, A.C.: Composite likelihood estimation for the Brown–Resnick process. Biometrika **100**(2), 511–518 (2013)
9. Kabluchko, Z., Schlather, M., de Haan, L.: Stationary max-stable fields associated to negative definite functions. Ann. Probab. **37**(5), 2042–2065
10. Kohatsu-Higa, A., Yasuda, K.: Estimating multidimensional density functions using the Malliavin–Thalmaier formula. SIAM J. Numer. Anal. **47**, 1546–1575 (2009)
11. Liu, Z., Blanchet, J.H., Dieker, A.B., Mikosch, T.: On optimal exact simulation of max-stable and related random fields. Submitted, arXiv:1609.06001

12. Malliavin, P., Thalmaier, A.: Stochastic Calculus of Variations in Mathematical Finance. Springer, New York (2006)
13. McLeish, D.: A general method for debiasing a Monte Carlo estimator. Monte Carlo Methods Appl. **17**(4) (2011)
14. Rhee, C.H., Glynn, P.W.: Unbiased estimation with square root convergence for SDE models. Oper. Res. **63**(5), 1026–1043 (2015)
15. Ribatet, M.: Spatial extremes: max-stable processes at work. Journal de La Société Française de Statistique **154**(2), 156–177 (2013)
16. Robert, C.Y., Segers, J.: Tails of random sums of a heavy-tailed number of light-tailed terms. Insur. Math. Econ. **43**(1), 85–92 (2008)
17. Scott, D.W., Sain, S.R.: Multidimensional density estimation. Handb. Stat. **24**, 229–261 (2005)
18. Zheng, Z., Blanchet, J., Glynn, P.W.: Rates of convergence and CLTs for subcanonical debiased MLMC. Submitted