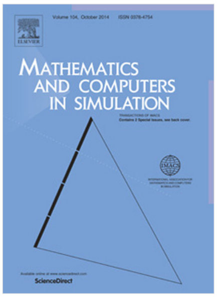


Simulation methods and error analysis for trawl processes and ambit fields

Dan Leonte, Almut E.D. Veraart

PII: S0378-4754(23)00298-7
DOI: https://doi.org/10.1016/j.matcom.2023.07.018
Reference: MATCOM 6150

To appear in: *Mathematics and Computers in Simulation*

Received date : 15 September 2022
Revised date : 6 June 2023
Accepted date : 12 July 2023

Simulation methods and error analysis for trawl processes and ambit fields

Dan Leonte[a,*], Almut E. D. Veraart[a]

[a] *Department of Mathematics, Imperial College London, 180 Queen's Gate, London, SW7 2BX, United Kingdom*

Abstract

Trawl processes are continuous-time, stationary and infinitely divisible processes which can describe a wide range of possible serial correlation patterns in data. In this paper, we introduce new simulation algorithms for trawl processes with monotonic trawl functions and establish their error bounds and convergence properties. We extensively analyse the computational complexity and practical implementation of these algorithms and discuss which one to use depending on the type of Lévy basis. We extend the above methodology to the simulation of kernel-weighted, volatility modulated trawl processes and develop a new simulation algorithm for ambit fields. Finally, we discuss how simulation schemes previously described in the literature can be combined with our methods for decreased computational cost.

Keywords: ambit fields, infinite divisibility, Lévy bases, numerical study of stochastic processes, serial correlation, stochastic simulation, trawl processes
2000 MSC: 6008, 6204, 60G10, 60G57, 60G60

1 Introduction

This paper introduces new simulation algorithms for trawl processes and ambit fields and establishes their error bounds and convergence properties. Deriving efficient simulation schemes, easily adaptable implementations and understanding the corresponding theoretical and numerical errors come naturally as first steps before employing such processes to model real-world data.

Trawl processes and ambit fields have been introduced in the context of Ambit Stochastics, which was first developed to model physical phenomena such as turbulent flow and tumour growth by [1]. Since its introduction, Ambit Stochastics proved to be a powerful modelling tool in other settings, such as spatio-temporal statistics [2], brain imaging [3] and finance [4]. [5] studied trawl

[*] Corresponding author *Email address:* `dl2916@ic.ac.uk` (Dan Leonte)

processes under the name of 'upstairs representations' and used them to model workloads for network communications. Together with the theoretical development of the topic, multiple computer libraries became available, as developed by [6, 7, 8, 9, 10] and [11]. Apart from the extensive analysis of existing and new simulation schemes, we release a unified Python library at [12], which contains efficient implementations of the discussed algorithms.

We start our analysis in the temporal setting with the class of trawl processes, then expand to the spatio-temporal case of ambit fields. Trawl processes are stationary and infinitely divisible stochastic processes which heavily rely on the following two concepts: the trawl set A_t, i.e. the region which influences the value of the trawl process X at time t, and the Lévy basis L, a type of random measure which extends the concept of noise from Gaussian and Poisson random measures to a general infinitely divisible setting. The trawl process X is then defined as the Lévy basis evaluated over the region of interest $X_t = L(A_t)$. This framework enforces the modelling belief that the value of the trawl process at time t is only influenced by a subset of the whole system, represented here by the trawl set A_t, and not by the entire system. In many settings, the choice of the trawl set is inspired by the physical knowledge of the phenomenon to be modelled. A great advantage of trawl processes is the flexibility of the autocorrelation structure and of the marginal distribution, which can be chosen independently. Indeed, the areas of the overlaps $A_t \cap A_s$ determine the correlations $\operatorname{Corr}(X_t, X_s)$ and the Lévy basis determines the marginal law of X, which can be any infinitely divisible distribution. This allows for the modelling of data displaying stylized facts, such as non-Gaussianity or heavy tails and offers a concrete and tractable alternative to modelling via stochastic partial differential equations (SPDEs), whose solutions can even be difficult to approximate numerically. A natural extension of trawl processes to spatio-temporal fields is given by random fields Y defined as $Y(t,x) = Y_t(\mathbf{x}) = L\left(A_t(\mathbf{x})\right)$, where the set $A_t(\mathbf{x}) \subset \mathbb{R}^d$ now depends on both time and $d-1$ spatial coordinates. We call this a simple field. Simulation methods previously described in the literature and which are applicable to trawl processes include simulation via grid discretization in [13], via compound Poisson processes in the case of integer-valued trawls in [4] and by using a slice partition in [14, Chapter 4.4], [15, Chapter 8.6]. We generalize these methods, derive their theoretical errors and computational complexities and discuss practical implementation details. Further, we expand on the slice partition method, develop a novel simulation algorithm for simple ambit fields and explain how the calculations required for higher accuracy can be performed ahead of the simulations, amortising the computational time across simulations. This allows for the implementation of high-accuracy simulation studies and simulation-based inference.

Recent empirical work in areas such as environmental sciences in [16] and energy pricing in [17, 18] suggests the presence of volatility clusters, and hence of a stochastic volatility, which can be easily incorporated into the Ambit Stochastics framework. Indeed, we consider the volatility modulated, kernel-weighted trawl processes $X_t = \int_{A_t} K_t\left(\bar{t}, \bar{\mathbf{x}}\right)\sigma(\bar{t})\mathrm{d}L\left(\bar{t}, \bar{\mathbf{x}}\right)$, and their spatio-temporal analogue, ambit fields $Y_t(\mathbf{x}) = \int_{A_t(\mathbf{x})} K_{t,\mathbf{x}}\left(\bar{t}, \bar{\mathbf{x}}\right)\sigma\left(\bar{t}, \bar{\mathbf{x}}\right)\mathrm{d}L(\bar{t}, \bar{\mathbf{x}})$, where the deterministic kernel K multiplied by the stochastic volatility σ is integrated against the Lévy basis L. The integration is understood in the sense of [19, Theorem 2.7] for deterministic σ and in the sense of [20, 21] and [22] for stochastic σ. This general formulation introduces a kernel and a stochastic volatility with respect to the basis model of trawl processes and simple ambit fields and offers a complex framework which can be used to explicitly construct random fields with certain statistical properties, such as symmetry in space and time [23]. We improve on the grid methods previously used in the literature by [2, 11] and show that despite the added terms, the trawl process simulation algorithms can be directly applied for the efficient simulation of kernel-weighted, volatility modulated trawl processes and ambit fields.

1.1 Contributions of the paper

We consider three simulation algorithms previously described in the literature and which are applicable to trawl processes: the grid discretization, slice partition and compound Poisson methods. We derive for the first time their convergence properties and develop a theoretical analysis for the simulation error. Further, we discuss their computational complexity and provide a Python library containing the simulation algorithms described in this paper, see [12].

Out of the three schemes mentioned above, only the slice partition is an exact algorithm, i.e. incurs no simulation error. The slice partition method accommodates any monotonic trawl shape and any infinitely divisible distribution described via the Lévy basis, requiring only access to samples from the marginal distribution of the Lévy basis. The other two algorithms, the grid discretization and compound Poisson approaches are exact only for particular types of trawl sets and Lévy seeds. In general, they incur a simulation error; we derive their convergence properties, error bounds and computational complexity. We extend the slice partition method from trawl processes and develop a novel simulation algorithm for simple ambit fields. In general, this does not lead to exact simulation of simple ambit fields. However, it has the advantage that the calculations required for higher accuracy only need to be performed once, before the simulation, leading to amortized computational cost across simulations, as opposed to the grid and compound Poisson methods, which in general require an increased cost per simulation; we also discuss in which situations the compound Poisson method might be preferable to the slice partition method. Finally, motivated by the high computational complexity and relative inefficiency of grid methods, as discussed in Subsection 3.4, we show how the compound Poisson and slice partition methods can be generalized to the simulation of volatility modulated, kernel-weighted trawl processes and ambit fields.

Our algorithms allow for the practical implementation of high-accuracy simulation studies and the applications are threefold. The first use-case is parameter inference, where we simulate trawl processes or simple ambit fields and attempt to infer the shape of the trawl set and the parameters of the Lévy seed. The second application is producing Monte Carlo confidence intervals with parametric bootstrap methods for the parameters inferred above. In this case, we generate multiple simulations of the trawl process with the estimated parameters; we then infer the parameters for each of the simulations to get approximate confidence intervals, see [24]. Such simulation studies have already been performed for integer-valued trawls and spatio-temporal Ornstein-Uhlenbeck processes in [4, 2]. Thirdly, trawl processes are non-Markovian stochastic processes with a highly intractable likelihood function, hence indirect inference techniques have to be used. In future work, we aim to use our algorithms within the simulation-based inference framework to fit the parameters of trawl processes and simple ambit fields.

1.2 Structure of the paper

Section 2 defines the notion of Lévy bases, which can be viewed as non-Gaussian extensions of Gaussian white noise, and settles the notation and theoretical framework. In particular, we discuss an extension of the Lévy-Khintchine theorem from Lévy processes and give formulae for the cumulant, autocorrelation structure and marginal distribution of the trawl process. Section 3 presents and compares the three simulation algorithms for trawl processes, employing grid discretizations, compound Poisson processes and slice partitions and analyses the convergence properties and computational complexity, first in the case of bounded trawl sets, and then in the unbounded case. Sections 4 and 5 extend the above methodologies to kernel-weighted, volatility modulated trawl processes and ambit fields. In particular, Subsection 5.1 further derives a new simulation scheme for simple ambit fields via Monte Carlo methods. Proofs that have been omitted from the main body, background material and a discussion of efficient implementations of the discussed algorithms can be found in Sections A, B and C of the supplemental file.

2 Trawl processes and their properties

We first introduce the notation and preliminaries needed in this section. For a set $S \subset \mathbb{R}^d$, let $\mathcal{B}_{\text{Leb}}(S)$ denote the collection of Borel measurable sets of finite Lebesgue measure which are contained in S. We view S as a subset of space-time, where the first coordinate gives the time component and the last $d-1$ coordinates give the spatial component. We say that the measure l is finite if $l(\mathbb{R}) < \infty$ and infinite otherwise. By a Lévy measure l on $\mathbb{R}$ we mean a (possibly infinite) Borel measure with $l(0) = 0$ and $\int_{\mathbb{R}} \min\left(1, y^2\right) l(\mathrm{d}y) < \infty$. Finally, for a random variable X we define the cumulant (log-characteristic) transform $C(\theta, X) = \log\left(\mathbb{E}\left[e^{i\theta X}\right]\right)$ [cf. 25, p. 33] and write $X \overset{d}{=} Y$ if X and Y have the same law.

We formally define Lévy bases and present some of their theoretical properties. We define the trawl process $X_t = L(A_t)$ as the Lévy basis evaluated over a collection of sets of interest A_t and discuss its marginal distribution and autocorrelation structure.

2.1 Lévy bases

Definition 2.1 (Lévy basis)**.** A Lévy basis L on S is a collection $\left\{L(A) : A \in \mathcal{B}_{\mathrm{Leb}}(S)\right\}$ of infinitely-divisible, real-valued random variables such that for any sequence $A_1, A_2, \ldots$ of disjoint sets in $\mathcal{B}_{\mathrm{Leb}}(S)$, the random variables $L(A_1), L(A_2), \ldots$ are independent and further, if $\cup_{j=1}^{\infty} A_j \in \mathcal{B}_{\mathrm{Leb}}(S)$, then $L\left(\cup_{j=1}^{\infty} A_j\right) = \sum_{j=1}^{\infty} L\left(A_j\right)$.

In the following, we assume that the Lévy bases L is homogeneous; a thorough discussion of this property can be found in Chapter 5.1 of [15].

Definition 2.2 (Homogeneous Lévy basis)**.** A Lévy basis L on S is homogeneous if there exist $\xi \in \mathbb{R}$, $a \in \mathbb{R}_{\geq 0}$ and a Lévy measure l on $\mathbb{R}$ such that for any $A \in \mathcal{B}_{\mathrm{Leb}}(S)$, the following holds

$$C(\theta, L(A)) = \left(i\theta\zeta - \frac{1}{2}\theta^2 a + \int_{\mathbb{R}} \left(e^{i\theta y} - 1 - i\theta y \mathbf{1}_{[-1,1]}(y)\right) l(\mathrm{d}y)\right) \mathrm{Leb}(A).$$

Another important concept is that of the Lévy seed, see e.g. in [4].

Definition 2.3 (Lévy seed)**.** A random variable $L^{'}$ is called a Lévy seed of the Lévy basis L if

$$C(\theta, L^{'}) = i\theta\zeta - \frac{1}{2}\theta^2 a + \int_{\mathbb{R}} \left(e^{i\theta y} - 1 - i\theta y \mathbf{1}_{[-1,1]}(y)\right) l(\mathrm{d}y).$$

Then

$$C(\theta, L(A)) = \mathrm{Leb}(A) C(\theta, L^{'}), \tag{2.1}$$

and the distribution of the Lévy seed determines the distribution of the Lévy basis. Note that if $L^{'} \overset{d}{=} L^{''}$, then $L^{''}$ is also a Lévy seed. Further, $L^{'}$ is infinitely divisible and to each Lévy basis L we can associate the Lévy-Khintchine triplet $(\xi,\, a,\, l)$ of $L^{'}$, which fully characterises the distributional properties of L. In the above triplet, ξ denotes the drift term, a the variance of the Gaussian component and l the Lévy measure of the jump part [cf. 25, p. 37]. Differentiating (2.1) once, respectively twice with respect to θ gives

$$\mathbb{E}\left[L(A)\right] = \mathrm{Leb}(A)\, \mathbb{E}\left[L^{'}\right], \tag{2.2}$$

$$\mathrm{Var}\left(L(A)\right) = \mathrm{Leb}(A)\ \mathrm{Var}\left(L^{'}\right), \tag{2.3}$$

and taking higher derivatives gives the relation between the cumulants of $L(A)$ and these of $L^{'}$.

Finally, to construct a trawl process, we need to choose the trawl sets. In the following, we restrict our attention to trawl processes with monotonic trawl functions, i.e. when the trawl sets are of the form

$$A_t = A + (t, 0), \qquad A = \{(s,x) \in \mathbb{R}^2 \colon s < 0, 0 < x < \phi(s)\},$$

where $\phi\colon (-\infty, 0] \to \mathbb{R}_{\geq 0}$ is a continuous increasing function. Define the trawl process $X = (X)_{t\geq 0}$ by the Lévy basis evaluated over the trawl set $X_t = L(A_t)$. We note that, while the trawl process X is defined to take values in $\mathbb{R}$, the trawl set is chosen as a subset of $\mathbb{R}^2$, i.e. it includes an abstract spatial dimension in addition to the temporal dimension. Further, the trawl set is non-anticipative, in the sense that A_t does not contain any points (s,x) with $s > t$. If there is some $T < 0$ such that $\phi(T) = 0$, then A is compactly supported and we say that the trawl is bounded. Otherwise, we say the trawl is unbounded. Generalizations are straightforward for $A \subset \mathbb{R}^d$ with $d > 2$.

2.2 Marginal distribution

As seen in Definition 2.1, the only restriction on the marginal distribution of a trawl process is that it has to be infinitely divisible. This provides a rich class of stochastic processes supported on the integers, on the real line and on the positive or negative real line, with short or long memory and light or heavy tails. Some examples include the following processes.

Integer-valued trawl processes

Example 2.4 (Poisson Lévy basis)**.** Let $L' \sim \text{Poisson}(\nu)$ for some intensity $\nu > 0$. Then $X_t = L(A_t) \sim \text{Poisson}(\nu\text{Leb}\,(A))$.

Example 2.5 (Skellam Lévy basis)**.** Let $L' \sim \text{Skellam}(\mu_1, \mu_2)$, i.e. $L^{'} \sim N_1 - N_2$ with N_1, N_2 independent and Poisson distributed with intensities $\mu_1, \mu_2 > 0$. Then $X_t = L(A_t) \sim \text{Skellam}(\mu_1\text{Leb}\,(A), \mu_2\text{Leb}\,(A))$.

Real valued trawl processes

Example 2.6 (Gaussian Lévy basis)**.** Let $L^{'} \sim \mathcal{N}(\mu, \sigma^2)$ be Gaussian distributed with mean μ and variance σ^2. Then $X_t = L(A_t) \sim \mathcal{N}\left(\mu\text{Leb}\,(A), \sigma^2\text{Leb}\,(A)\right)$.

Example 2.7 (Cauchy Lévy basis)**.** Let $L^{'} \sim \text{Cauchy}(\gamma)$ with scale parameter $\gamma > 0$. Then $X_t = L(A_t) \sim \text{Cauchy}(\gamma\text{Leb}\,(A))$.

Positive real valued trawl processes

Example 2.8 (Gamma Lévy basis)**.** Let $L^{'} \sim \text{Gamma}(k, \theta)$ with shape and rate parameters $k, \theta > 0$ and pdf $p(x) = \frac{1}{\theta^k\Gamma(k)} x^{k-1} e^{-x/\theta}$, $x > 0$. Then $X_t = L(A_t) \sim \text{Gamma}(k\text{Leb}\,(A), \theta)$.

Example 2.9 (Inverse Gaussian Lévy basis)**.** Let $L' \sim \text{IG}(\delta, \gamma)$, with parameters $\delta, \gamma > 0$ and pdf $p(x) = \frac{\delta}{\sqrt{2\pi x^3}} e^{\delta\gamma - \frac{1}{2}\left(\frac{\delta^2}{x} + \gamma^2 x\right)}$, $x > 0$. Then $X_t = L(A_t) \sim \text{IG}(\delta\text{Leb}\,(A), \gamma)$.

A more general example is given by the class of trawl processes with stable distributions.

Example 2.10 (Stable Lévy basis)**.** Let $L^{'} \sim \text{Stable}(\alpha, \beta, c, \mu)$ have a stable distribution with stability and skewness parameters $\alpha \in (0,2]$, $\beta \in [-1,1]$ and location, scale parameters $\mu \in \mathbb{R}$, $c > 0$, defined through the cumulant function $C(\theta, L^{'}) = i\theta\mu + |c\theta|^{\alpha} \left(1 - i\beta \operatorname{sign}(\theta)\Phi\right)$, where $\Phi = \tan\frac{\pi\alpha}{2}$ if $\alpha \neq 1$ and $-\frac{2}{\pi}\log|\theta|$ if $\alpha = 1$. Then $X_t = L(A_t) \sim \text{Stable}(\alpha, \beta, c\text{Leb(A)}^{1/\alpha}, \mu\text{Leb(A)})$. The support of $L^{'}$ is $[\mu, \infty)$ if $\alpha < 1$, $\beta = 1$, $(-\infty, \mu]$ if $\alpha < 1$, $\beta = -1$ and $\mathbb{R}$ otherwise.

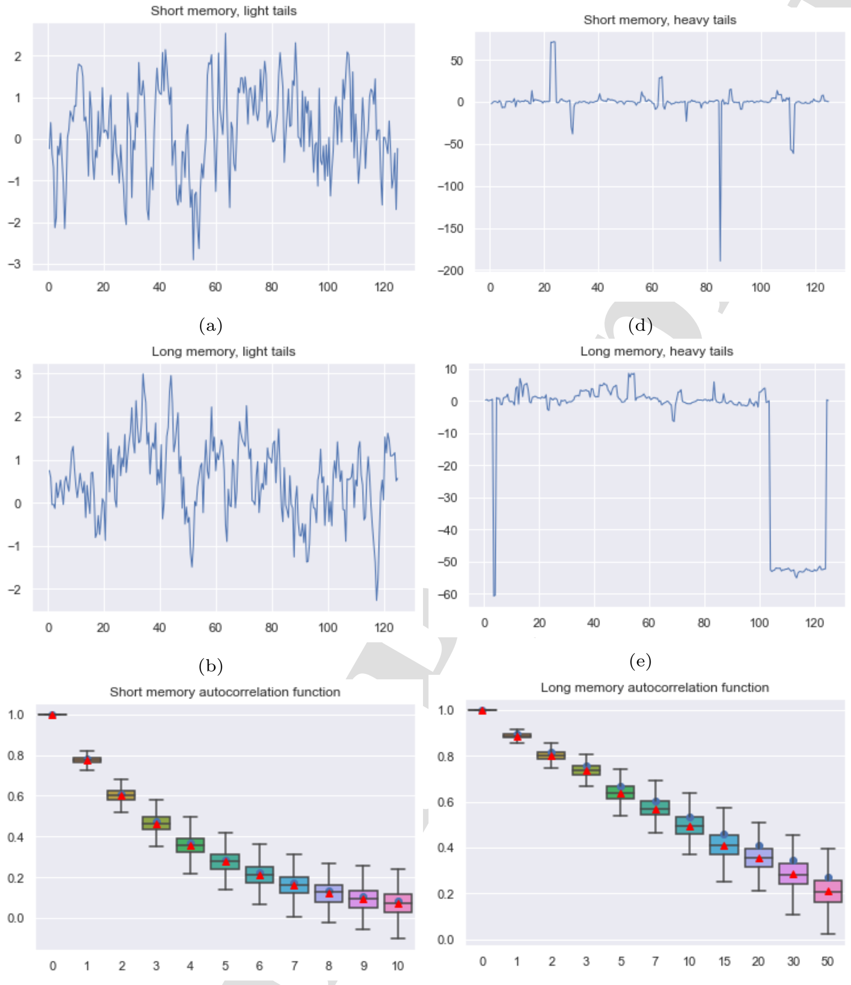
Lévy bases can thus be seen as a generalization of the Gaussian white noise process to a class of random measures with flexible marginal distributions.

2.3 Covariance and Correlation structure

We saw previously that the distribution of $L^{'}$, together with the Lebesgue measure of the trawl set, determines the marginal distribution of the trawl process X_t. Similarly, the shape of the trawl set A, specified here by the trawl function ϕ, determines the autocorrelation structure of X_t. Indeed, note that the sets $A_t \backslash A_{t+h}$, $A_{t+h} \backslash A_t$ and $A_t \cap A_{t+h}$ are disjoint, hence the random variables $L\left(A_t \backslash A_{t+h}\right)$, $L\left(A_{t+h} \backslash A_t\right)$ and $L\left(A_t \cap A_{t+h}\right)$ are independent. By using this decomposition and by (2.2) and (2.3), we obtain that $\text{Cov}(X_t, X_{t+h}) = \text{Leb}\left(A \cap A_h\right) \text{Var}(L^{'}) = \int_{-h}^{0} \phi(s)\text{d}s \, \text{Var}\left(L^{'}\right)$ and further that

$$\rho(h) := \text{Corr}(X_t, X_{t+h}) = \frac{\text{Leb}\left(A \cap A_h\right)}{\text{Leb}\left(A\right)} = \frac{\int_{-h}^{0} \phi(s)\text{d}s}{\int_{-\infty}^{0} \phi(s)\text{d}s}. \tag{2.4}$$

Thus we have a representation of the autocorrelation function ρ solely in terms of the trawl function ϕ. In particular, the trawl process can realize any positive, strictly decreasing autocorrelation function. Figure 1 displays some realizations of the trawl process, with short and long memory, light and heavy tails, simulated by the algorithm from Section 3.3.



(a)

(d)

(b)

(e)

(c)

(f)

Figure 1: a-d) Realizations of the trawl process at times $\tau, \ldots, 250\tau$ with $\tau = 0.5$, obtained by varying the trawl function and Lévy seed. For short and long memory we set ϕ_{sm}, $\phi_{\text{lm}} \colon (-\infty, 0] \to \mathbb{R}_{\geq 0}$ given by $\phi_{\text{sm}}(t) = e^t$ and $\phi(t)_{\text{lm}} = 0.5(1-t)^{-1.5}$, which result in trawl sets of Lebesgue measure 1. The corresponding autocorrelation functions ρ_{sm}, $\rho_{\text{lm}} \colon [0, \infty) \to \mathbb{R}_{\geq 0}$ are given by $\rho_{\text{sm}}(h) = e^{-h}$ for Figures 1a and 1d, and $\rho_{\text{lm}}(h) = (1+h)^{-0.5}$ for Figures 1b and 1e. For light and heavy tails, we set $L^{'}_{\text{lt}} \sim \mathcal{N}(0,1)$, respectively $L^{'}_{\text{ht}} \sim \text{Cauchy}(1)$. Since the Lebesgue measure of both trawl sets is 1, the corresponding marginal distribution is $\mathcal{N}(0,1)$ in Figures 1a and 1b and Cauchy(1) in Figures 1d and 1e. e-f) Theoretical and empirical autocorrelation functions of the trawl process, with trawl function ϕ_{sm} in e) and ϕ_{lm} in f). The boxplots describe the distribution of empirical autocorrelation functions at different lags, based on 500 simulations, in which we simulate the trawl process at times $\tau, \ldots, 1000\tau$ in e) and $\tau, \ldots, 5000\tau$ in f), with $\tau = 0.5$. The blue circle and red triangle show the true, respectively the mean of the empirical autocorrelation functions. Note that the convergence of the empirical autocorrelation function, as a function of the number of simulated trawls, is much faster in the short memory case, where the two markers are superimposed.

3 Simulation algorithms for trawl processes and their convergence properties

In this section, we present and compare three simulation algorithms for trawl processes: the grid discretization, the simulation via compound Poisson processes and the slice partition. Grid methods were previously considered in [13] and [2] for ambit field simulation, whereas simulation via compound Poisson processes was employed in [4] for the simulation of integer-valued trawls. The main disadvantage of these methods is that they are exact only for certain trawl shapes and marginal distributions of $L^{'}$; in general, the computational time increases as a function of the required accuracy. Based on the slice partition ideas from [15, Chapter 8.6] and [14, Chapter 4.4], we describe the slice partition algorithm, which provides an efficient alternative for the exact simulation of monotonic trawls. We establish the convergence properties of these algorithms: in probability, in Skorokhod's topology and uniformly, providing MSE bounds on the theoretical error. In all three algorithms, we require samplers either from the law of $L(A)$ for sets A of various Lebesgue measures or from the Lévy measure l. We postpone discussing this technical but crucial requirement to Subsection 3.4, when the presentation of the simulations schemes has finished and the need for such samplers is clear.

For ease of presentation, assume there is only one spatial component, i.e. $d = 2$. Consider a homogeneous Lévy basis L on $S = \mathbb{R}^2$, with triplet $(\xi,\, a,\, l)$, where $\xi \in \mathbb{R}, a \in \mathbb{R}_{\geq 0}$ are constants and where l is a Lévy measure on $\mathbb{R}$. Let the trawl set A of finite Lebesgue measure be given by

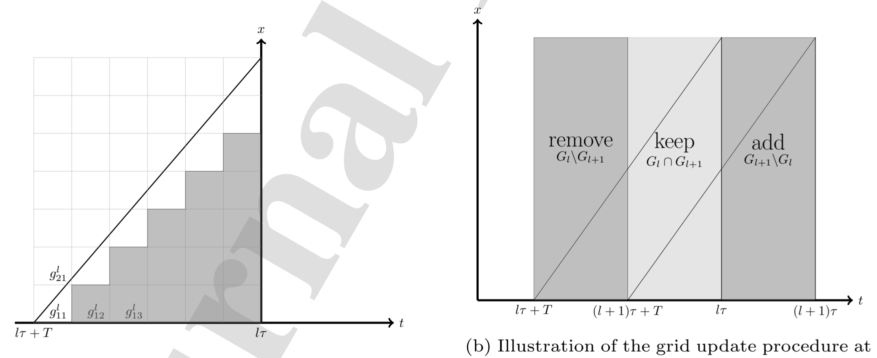
$$A_t = A + (t,0), \qquad A = \{(s,x) \in \mathbb{R}^2 \colon s < 0,\, 0 < x < \phi(s)\},$$

for some smooth, increasing function $\phi\colon (-\infty, 0] \to \mathbb{R}_{\geq 0}$. We aim to simulate the trawl process $X_t = L(A_t) = L(A + (t,0))$ at equidistant times $\tau, \ldots, k\tau$. In the following, we simulate both the Gaussian and jump parts, but note that it is enough to simulate the jump part. Indeed, the covariance matrix Σ of $L_g(A_\tau), \ldots, L_g(A_{k\tau})$ is given by the areas of the overlaps of the translated trawl sets. Thus the Gaussian part can be simulated by Hx, where H comes from the Cholesky decomposition of Σ, i.e. $\Sigma = HH^t$, and x is a vector sampled from the standard k dimensional multivariate Gaussian. In general, the Cholesky factorisation has complexity $\mathcal{O}\left(k^3\right)$ and the matrix-vector multiplication $\mathcal{O}\left(k^2\right)$.

3.1 Algorithm I: grid discretization

Assume that A is bounded, i.e. there exists $T < 0$ such that $\phi(T) = 0$ and $A = \{(t,x) \in \mathbb{R}^2 : T < t \leq 0, 0 < x < \phi(t)\}$; we later relax this assumption. Assume that $T + \tau < 0$; otherwise, the sets $A_\tau, \ldots, A_{k\tau}$ are disjoint and we can simulate the trawl process by drawing k iid samples from the law of $L(A)$. In this algorithm, we discretize the rectangle $[T + \tau, k\tau] \times [0, \phi(0)]$ into a grid of cells, simulate the Lévy basis over these cells and approximate $L(A)$ by $\sum L(c)$, where the sum is over cells c which are fully contained in A.

Let the discretization step-sizes on the time, respectively space axes be $\Delta_t = \tau/N_t$, $\Delta_x = \phi(0)/N_x$ for some positive integers N_t,N_x. If T/Δ_t is not an integer, replace T with $\lfloor T/\Delta_t \rfloor \Delta_t$, where $\lfloor \cdot \rfloor$ is the floor function and let $N = -T/\Delta_t$; this choice excludes boundary effects and ensures that all the cells we consider in this algorithm have equal area. In total, there are $\big((k-1)N_t + N\big) N_x$ cells contained in the grid on $[T+\tau, k\tau] \times [0, \phi(0)]$. Let Y be an $N_x \times \big((k-1)N_t + N\big)$ random matrix with iid entries $Y_{ij} \overset{d}{=} L(c)$, corresponding to the Lévy basis simulated over all the grid cells. Practical experiments show that for small values of Δ_t and Δ_x, it is not feasible to hold a realization of Y in memory. Nevertheless, note that to simulate $L(A_t)$, it is enough to hold the samples $L(c)$ for cells c contained in $[t+T, t] \times [0, \phi(0)]$ in memory. To this end, for each trawl set $A_{l\tau}$, define the corresponding set of $N \cdot N_x$ grid cells on $[l\tau + T, l\tau] \times [0, \phi(0)]$ by $G_l = \{g^l_{ij} : 1 \leq i \leq N_x, 1 \leq j \leq N\}$(see Figure 2a), where each cell is of the form $g^l_{ij} = [l\tau + T + (j-1)\Delta_t, l\tau + T + j\Delta_t] \times [(i-1)\Delta_x, i\Delta_x]$. Define I to be the $N_x \times N$ indicator matrix with entries $I_{ij} = 1$ if $g^l_{ij} \in A_{l\tau}$ and 0 otherwise and note that I does not depend on the chosen trawl set $A_{l\tau}$. Finally, let $Y_l = Y[:, (l-1)N_t + 1 : (l-1)N_t + N]$ be the $N_x \times N$ random matrix obtained by subsetting only the columns of Y corresponding to cells in G_l; the grid approximation of $L(A_{l\tau})$ is then given by $Y_l \odot I$, i.e. the sum of the entries of the component-wise product of matrices Y_l and I. Iteratively, at step $l+1$, we can compute Y_{l+1} from Y_l by discarding $\{L(c)\}_{g \in G_l \setminus G_{l+1}}$ and adding new samples $\{L(c)\}_{g \in G_{l+1} \setminus G_l}$(see Figure 2b). This corresponds to removing the first N_t columns from the left of Y_l, adding N_t new sampled columns to the right of Y_l and approximating $L\left(A_{(l+1)\tau}\right)$ by $Y_{l+1} \odot I$. The full procedure is given in Algorithm 1.



(a) Illustration of the grid discretization algorithm: the Lévy basis evaluated of the trawl set $A_{l\tau}$ is approximated by $\sum L(c)$, where the sum is taken over cells c which are fully contained in $A_{l\tau}$. These cells are shaded in gray.

(b) Illustration of the grid update procedure at step l: we remove from memory the values $L(c)$ corresponding to cells whose 'upper-left' corners have time coordinates less than $(l+1)\tau + T$ and simulate $L(c)$ over cells whose 'upper-left' corners have time coordinates in the interval $[l\tau, (l+1)\tau]$.

Figure 2

Remark 3.1. Algorithm 1 requires checking if a cell c is fully contained in a monotonic trawl set A_t (step 5) and a sampler for $L(c)$ (steps 12, 17). For

Algorithm 1 Grid discretization

Input: Trawl function ϕ and $T < 0$ such that $\phi(T) = 0$; number of steps N_t,N_x on the time and space axes; sampler $S(\text{area}, m, n)$ which returns an $m \times n$ array of iid samples with the same law as $L(A)$, where $\text{Leb}(A) = \text{area}$; number of trawls to be simulated k and distance τ between them.

Output: Vector X containing the simulated values of the trawl process at times $\tau, \ldots, k\tau$.

```
function ComputeIndicator($\phi$, $T$, $N$, $N_x$, $\Delta_t$, $\Delta_x$)        ▷ Helper function
    $I \leftarrow \text{zeros}(N_x, N)$
    for $i = 1, \ldots, N$ do
        for $j = 1, \ldots, N_x$ do
            if $i\Delta_x \le \phi(T + (j-1)\Delta_t)$ then $I_{ij} \leftarrow 1$   ▷ Check if $g^l_{ij}$ is contained
                                                                  in $A_{l\tau}$ (see Remark 3.1)
    return I
function main($\phi$, $T$, $N_t$, $N_x$, $S$, $k$, $\tau$)
    $T \leftarrow \lfloor T/\Delta_t \rfloor \Delta_t$                    ▷ Ensures all cells have the same area
                                                   and excludes boundary effects
    $\Delta_t \leftarrow \tau/N_t$, $\Delta_x \leftarrow \phi(0)/N_x$, $N \leftarrow -T/\Delta_t$
    $X \leftarrow \text{zeros}(k)$
    $I \leftarrow$ ComputeIndicator($\phi$, $T$, $N$, $N_x$, $\Delta_t$, $\Delta_x$)
    $Y_1 \sim S(\Delta_t\Delta_x, N, N_x)$                          ▷ Sample $L(c)$ for $c \in G_1$
    $X[1] \leftarrow Y_1 \odot I$
    for $2 = 1, \ldots, k$ do
        $Y_l \leftarrow \text{zeros}(N, N_x)$                     ▷ Update step (see Figure 2b)
        $Y_l[:, 1 : N - N_t] \leftarrow Y_{l-1}[:, N_t + 1 : N]$     ▷ Keep $L(c)$ for $c \in G_{l+1} \cap G_l$
        $Y_l[:, N - N_t + 1 : N] \sim S(\Delta_t\Delta_x, N_x, N_t)$        ▷ Sample $L(c)$ for
                                                              $c \in G_{l+1} \backslash G_l$
        $X[l] \leftarrow Y_l \odot I$
    return X
```

the first requirement, note that a cell c is fully contained in a monotonic trawl set A_t iff the 'top-left' corner (s, x) of the cell is in A_t. The cells can then be represented by their 'top-left' corner and the inclusion condition $c \subset A_t$ is equivelent to $x \le \phi(s - t)$, which can be easily checked on a computer. The second requirement is discussed extensively in Subsection 3.4.

Even if A is unbounded, we can still apply Algorithm 1 by truncating and approximating $L(A)$ by $L(A \cap \{t > T\})$. In this case, the algorithm has two sources of error: firstly, taking a grid discretization and considering a cell as part of the trawl if its 'top-left' corner is in the trawl set, and secondly, neglecting $L\left(A \cap \{t < T\}\right)$. We now derive the convergence properties of the grid discretization algorithm in the setting of both infill and increasing domain asymptotics; further, if the Lévy seed $L^{'}$ has finite variance, we provide MSE bounds on the error.

Theorem 3.2. *Let* $A = \{(s, x) \in \mathbb{R}^2 \colon s < 0,\, 0 < x < \phi(s)\}$ *be a monotonic trawl set* *of finite Lebesgue measure, $T < 0$ and $\Delta = (\Delta_t, \Delta_x)$ with $\Delta_t, \Delta_x > 0$. Let $G_{T,\Delta}$ be the set of grid cells on $[T, 0] \times [0, \phi(0)]$ with step-sizes Δ_t, Δ_x on the time, respectively space axes. Let $L_{T,\Delta}(A)$ be the approximation of $L(A)$ with respect to $G_{T,\Delta}$ (see Figure 3)*

$$L_{T,\Delta}(A) = \sum_{g \in G_{T,\Delta} : g \subset A} L(g).$$

Let $T_n \to -\infty, \Delta_n = (\Delta_t^n, \Delta_x^n) \to 0$. Then $L_{T_n,\Delta_n}(A) \to L(A)$ in probability and, if $\operatorname{Var} L^{'}$ is finite, in $\mathcal{L}^2$; further, if $\operatorname{Var} L^{'}$ is finite, then $\mathbb{E}\left[\left(L_{T,\Delta}(A) - L(A)\right)^2\right] \leq C^2_{T,\Delta}\mathbb{E}\left[L^{'}\right] + C_{T,\Delta} \operatorname{Var} L^{'}$, where $C_{T,\Delta} = T\Delta_x + \phi(0)\Delta_x + \int_{-\infty}^{T} \phi(s)ds$. In particular, if A is bounded and $\Delta_t = \Delta_x$, then $\mathbb{E}\left[\left(L_{T,\Delta}(A) - L(A)\right)^2\right] = \mathcal{O}\left(\Delta_t^2\right)$.

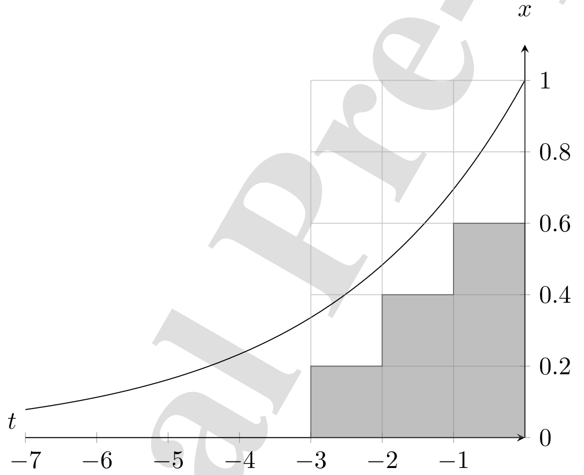


Figure 3: Illustration of the grid discretization for the unbounded trawl set $A = \{(t, x) \in \mathbb{R}^2 : t < 0, 0 < x < e^{2.75t}\}$, with truncation parameter $T = -3$ and step-sizes $\Delta_t = 1$, $\Delta_x = 0.2$. The cells included in the simulation are shaded in gray. For unbounded trawls, we incur both truncation and discretization error.

The MSE bound in Theorem 3.2 is not sharp for unbounded trawls, in the sense that it gives $\mathcal{L}^2$ convergence only when $T_n\Delta_x^n \to 0$ (see proof in Section A). Nevertheless, it provides a way to choose the truncation parameter T and step-sizes Δ_t, Δ_x for the purpose of computer simulations. Despite the convergence properties, Algorithm 1 is computationally expensive, requiring samples for $N_x((k-1)\,N_t + N) = \left((k-1)\,\tau + T\right) \cdot \phi(0)/(\Delta_t\Delta_x)$ cells. In general, if $A \subset \mathbb{R}^d$ with $d > 2$ and with same step-size Δ_t for all dimensions, the number of cells grows like $\mathcal{O}(1/\Delta_t^d)$ and holding the grids G_l in memory, even one at a time, is not feasible. In the following, we discuss two alternatives: the compound Poisson and the slice partition algorithms, which are less computationally intensive.

3.2 Algorithm II: compound Poisson simulation

Given a Lévy basis L with characteristic triplet (ξ, a, l), by the first part of Theorem B.1, there exist a decomposition of L into a Gaussian Lévy basis L_g, with triplet $(\xi, a, 0)$, and a jump Levy basis L_j, with triplet $(0, 0, l)$, such that L_g and L_j are independent and $L = L_g + L_j$. Assume that the Gaussian part has already been simulated, which can be done exactly by Algorithm 3 of the next subsection, or by using a Cholesky decomposition and a standard normal sampler. By the second part of Theorem B.1, there exists a Poisson random measure N on $\mathcal{B}_{\text{Leb}}(\mathbb{R}^2) \times \mathbb{R}$ with intensity measure $\nu := \text{Leb} \otimes l$ such that

$$L_j\,(A) = \int_A \int_{(-1,1)} y \;\mathrm{d}(N - \nu)(\mathbf{z}, y) + \int_A \int_{\mathbb{R}\setminus(-1,1)} y \;\mathrm{d}N(\mathbf{z}, y),$$

where Leb is the Lebesgue measure on $\mathbb{R}^2$, $\mathbf{z} = (t, x) \in \mathbb{R}^2$ and $y \in \mathbb{R}$. In the above, (t, x) are the time and space coordinates at which a jump appears, whereas y is the value of the jump. For ease of presentation, assume that the trawl set is bounded; we relax this assumption at the end of the subsection. We analyse the cases of finite and infinite Lévy measures separately.

If l is a finite measure, i.e. if $c := l(\mathbb{R}) < \infty$, we can simplify the above expression to

$$L_j\,(A) = \int_A \int_{\mathbb{R}\setminus(-1,1)} y \;\mathrm{d}N(\mathbf{z}, y) - \text{Leb}(A) \int_{-1}^{1} y \; l(dy). \tag{3.1}$$

With this representation in mind, to simulate the trawl process, we can simulate a Poisson point process on $\mathbb{R}^2$ with constant intensity c, where to each generated point (t_i, x_i) we associate a corresponding jump sample y_i with law $\tilde{l} := l/c$. Let $\tilde{\xi} = \int_{-1}^{1} y \; l(\mathrm{d}y)$. The value of $L_j(A)$ is then given by the difference between the sum of all jumps y_i corresponding to points $\mathbf{z}_i = (t_i, x_i)$ contained in the trawl set A and the drift term $\tilde{\xi}\text{Leb}(A)$

$$L_j(A) = \sum_{i:\;(t_i,x_i)\in A} y_i - \tilde{\xi}\text{Leb}(A).$$

It follows that $L_j(A)$ is compound Poisson distributed, with jumps distributed according to $\tilde{l}$, minus a constant. This property is also clear from the simplified expression for the cumulant transform from Definition (2.2)

$$C(\theta, L_j(A)) = c\text{Leb}(A) \int_{\mathbb{R}} \left(e^{i\theta y} - 1\right) \tilde{l}(\mathrm{d}y) - i\theta\tilde{\xi}\text{Leb}(A).$$

Finally, as the trawl is bounded, we can choose $T < 0$ be such that $\phi(T) = 0$ and note that we only need to simulate the Poisson point process on $[T + \tau, k\tau] \times [0, \phi(0)]$. The full procedure for the simulation of trawl processes with bounded trawl sets and finite Lévy measures is given in Algorithm 2. Note that the nested for loops in steps 9 and 10 require an expected number of $\mathcal{O}(k^2)$ comparisons of the form $x[i] < \phi(t[i]-l\tau)$. Since A is bounded, at most $\lceil -T/\tau \rceil$ consecutive trawl sets have non empty intersections, hence it is enough to do $\lceil -T/\tau \rceil$ comparisons in step 10 and the complexity reduces to $\mathcal{O}(k)$.

Algorithm 2 Compound Poisson simulation

```
Input: Trawl function $\phi$ and $T < 0$ with $\phi(T) = 0$; sampler $S(n)$ which returns
    an $n$ dimensional vector of iid samples from $\tilde{l}$; $c = l(\mathbb{R})$ and $\tilde{\xi} = \int_{-1}^{1} y\ l(\mathrm{d}y)$;
    number of trawls to be simulated $k$ and distance $\tau$ between trawl sets;
    sampler $U(a, b, n)$ which returns an $n$ dimensional vector of iid samples
    from the uniform distribution on $[a, b]$
Output: Vector $X$ containing the simulated values of the trawl process at times
    $\tau, \dots, k\tau$.
function CppSimulation($S$, $c$, $\tilde{\xi}$, $\phi$, $T$, $k$, $\tau$)
    $\nu \leftarrow c\phi(0)\big((k-1)\tau - T\big)$
    $N \sim Poisson(\nu)$
    $t \sim U\big([T + \tau, k\tau]\big)$              ▷ Generate $N$ jump time samples
    $x \sim U\Big(\big[0, \phi(0)\big]\Big)$              ▷ Generate $N$ jump height samples
    $y \sim S(N)$                              ▷ Generate $N$ jump value samples
    $X \leftarrow -\tilde{\xi}\mathrm{Leb(A)} \cdot \mathrm{ones}(k)$       ▷ Initialization with the compensator
    for $i \in \{1, \dots, N\}$ do
        for $l \in \{1, \dots, k\}$ do
            if $x[i] < \phi(t[i] - l\tau)$ then         ▷ Check if $(t_i, x_i) \in A_{l\tau}$
                $X[l] += y[i]$
    return $X$
```

On the other hand, if l is an infinite measure, we can no longer represent L_j via a compound Poisson process; we need to truncate and discard jumps with magnitude below some threshold, as we would when simulating a Lévy process. Let l^ϵ be the restriction of l to $(-\infty, -\epsilon) \cap (\epsilon, \infty)$ and $\tilde{l^\epsilon} := l^\epsilon / l^\epsilon(\mathbb{R})$. Define the corresponding truncated Lévy basis L_j^ϵ

$$L_j^\epsilon(A) = \int_A \int_{(-1,-\epsilon)\cup(\epsilon,1)} y\ \mathrm{d}(N - \nu)(\mathbf{z}, y) + \int_A \int_{\mathbb{R}\setminus(-1,1)} y\ \mathrm{d}N(\mathbf{z}, y), \tag{3.2}$$

the resulting approximation $X_t^\epsilon = L_j^\epsilon(A_t)$ and $X_t = L_j(A_t)$. Note that X^ϵ can be simulated with Algorithm II and that

$$\begin{aligned}\mathbb{E}\Big[\big(L_j(A) - L_j^\epsilon(A)\big)^2\Big] &= \mathbb{E}\Bigg[\bigg(\int_A \int_{(-1,-\epsilon)\cup(\epsilon,1)} y\ \mathrm{d}(N - \nu)(\mathbf{z}, y)\bigg)^2\Bigg] \\ &= \mathrm{Leb(A)} \int_{-\epsilon}^{\epsilon} y^2 \mathrm{d}l(y) \to 0 \text{ as } \epsilon \to 0,\end{aligned}$$

which gives different convergence rates for different Lévy measures l. Even when the Lévy seed does not have any finite moments, the error $L_j(A) - L_j^\epsilon(A)$ is square integrable and converges to 0 in $\mathcal{L}^2$, which already improves on the convergence of Theorem 3.2. We further establish convergence of X^ϵ as a stochastic process, rather than just at the level of the marginals X_t.

Theorem 3.3. *The process X^ϵ converges uniformly on compacts to X on the space of càdlàg paths as $\epsilon \to 0$.*

We now relax the bounded trawl set assumption. Previous simulation methods for integer-valued trawl processes with unbounded trawls from [15, 26] approximated $L(A)$ by $L(A \cap \{t > T\})$ for some truncation threshold T, as $L(A \cap \{t \leq T\}) \to 0$ in probability as $T \to -\infty$. Apart from losing the uniform convergence properties, T must be chosen with a very large absolute value, significantly increasing the computational time. For example, if the trawl process has long memory, with autocorrelation function given by $\rho(t) = (1+t)^{1/4}$, then even choosing $T = -10^4$ would incur an unacceptable trawl set truncation error. The apparent difficulty comes from simulating a Poisson point process on an unbounded domain of finite area, which we will discuss next.

In the case of unbounded trawl sets, apply Algorithm 2 to simulate L_j^ϵ on $[\tau, k\tau] \times [0, \phi(0)]$ and sample the number of atoms $\mathbf{z}_i = (t_i, x_i)$ of L_j^ϵ on A_τ $N \sim \text{Poisson}(l^\epsilon(\mathbb{R})\text{Leb}(A_\tau))$. Conditionally on N, the t_i's are iid with density $p(t) = \phi(t-\tau) / \int_{-\infty}^0 \phi(s)ds$ for $t < \tau$ and 0 otherwise. Finally, conditionally on N and t_i, $x_i \sim U(0, \phi(t_i))$ and we can continue with Steps $6-11$ of Algorithm 2. By decomposing ϕ into its convex and concave parts, we can draw samples with density p by rejection sampling, as described in [27].

Finally, note that the proof of Theorem 3.3 does not require a bounded trawl set, thus we obtain the same convergence results for unbounded trawls. Nevertheless, for infinite Lévy measures, $l\left(\mathbb{R} \backslash (-\epsilon, \epsilon)\right) \to \infty$ as $\epsilon \to 0$, hence the intensity ν of the Poisson process N to be sampled in Step 3 of Algorithm 2 diverges, leading to an increasing cost per simulation. In the next subsection, we present an algorithm which simulates trawl processes exactly, regardless of the type of Lévy measure.

3.3 Algorithm III: slice partition

In this algorithm we decompose the sets $A_\tau, \ldots, A_{k\tau}$ into a collection $\mathcal{S}$ of disjoint slices S, simulate the values of the Lévy basis L over each slice and then set

$$X_{l\tau} = \sum_{S \subset A_{l\tau}} L(S).$$

Indeed, because of the indepedent-scatteredness of the Lévy basis, i.e. the first property in Definition 2.1, we can sample $\{L(S)\}_{S \in \mathcal{S}}$ independently; then, by the additivity of the Lévy basis, i.e. the second property in Definition 2.1, we can reconstruct the value of the trawl $X_{l\tau}$ by summing the values corresponding to the Lévy basis simulated over the slices contained in $A_{l\tau}$. In general, there could be up to 2^k slices of the form $B_1 \cap \ldots \cap B_k$, where $B_l \in \left\{A_{l\tau}, A_{l\tau}^C\right\}$. For monotonic bounded trawls, the number of slices is $\mathcal{O}(k)$, whereas for monotonic unbounded trawls, it is $\mathcal{O}\left(k^2\right)$, making the simulation scheme feasible. We analyse the slice partition separately in the bounded and unbounded case. For ease of notation and without risk of confusion, we write A_l for $A_{l\tau}$.

If there is some $T < 0$ such that $\phi(T) = 0$, let $I = \lceil -T/\tau \rceil$, where $\lceil \cdot \rceil$ is the ceiling function. For $1 \le i \le I$ define the slice partition (see Figure 4a)

$$
\begin{aligned}
S_{i1} &= \{t \le \tau\} \cap \left(A_i \backslash A_{i+1}\right), \\
S_{ij} &= \left(\{(j-1)\cdot\tau < t \le j\cdot\tau\} \cap A_{i+j-1}\right) \backslash A_{i+j}, \text{ for } j \ge 2.
\end{aligned}
$$

Then exactly I consecutive trawl sets have non-empty intersection and each of the trawl sets $A_{l\tau}$ contains exactly I slices, making up for a total of kI slices. Let $s_{ij} = \operatorname{Leb}\left(S_{ij}\right)$; by the translation invariance of the Lebesgue measure, $s_{ij} = s_{ij'}$ for j, $j' \ge 2$. Hence to determine the areas of the slices S_{ij} it is enough to compute s_{ij} for $i \in \{1,\ldots,k\}$ and $j \in \{1,2\}$. A short calculation shows that

$$
s_{i1} = \int_{-i\cdot\tau}^{(-i+1)\cdot\tau} \phi(t)dt, \tag{3.3}
$$

$$
s_{i2} = s_{i,1} - s_{i+1,1}, \tag{3.4}
$$

where we set $s_{I+1,1} = 0$. The above two equations fully specify the areas s_{ij}. Let Y be the $I \times (I-1+k)$ random matrix of independent random variables $L\left(S_{ij}\right), 1 \le i \le I,\ 1 \le j \le k$, padded with $I-1$ columns of 0's to the left and F be the $I \times I$ lower diagonal matrix filer

$$
Y = \begin{pmatrix}
\underbrace{\begin{matrix} 0 & \ldots & 0 \\ 0 & \ldots & 0 \\ \vdots & & \vdots \\ 0 & \ldots & 0 \end{matrix}}_{I-1} &
\begin{matrix} L(S_{11}) & \ldots & L(S_{1k}) \\ L(S_{21}) & \ldots & L(S_{2k}) \\ \vdots & & \vdots \\ L(S_{I1}) & \ldots & L(S_{Ik}) \end{matrix}
\end{pmatrix},\ F = \begin{pmatrix}
0 & 0 & \ldots & 0 & 1 \\
0 & 0 & \ldots & 1 & 1 \\
\vdots & \vdots & \iddots & \vdots & \vdots \\
0 & 1 & \ldots & 1 & 1 \\
1 & 1 & \ldots & 1 & 1
\end{pmatrix}. \tag{3.5}
$$

The values of the trawl process at times $\tau,\ldots,k\tau$ are given by the convolution $Y \ast F$ (see Section C.1 for the definition of matrix convolution). The full procedure is given in Algorithm 3. Since $s_{ij} = s_{ij'}$ for j, $j' \ge 2$, we can vectorize steps $6-8$ by sampling columns of Y. Further, the convolution step 8 can be computed efficiently by taking advantage of the form of the filter F (see Algorithm 6 in Section C.1), which is also implemented in [12]. In both cases, the number of operations is $\mathcal{O}(k)$.

If the trawl set A is unbounded, define the slice partition

$$
S_{ij} = \left(A_j \cap A_{i+j-1}\right) \backslash A_{i+j},\ 1 \le i, j \le k,\ i+j \le k+1.
$$

In total, we have $k(k+1)/2$ slices (see Figure 4b). Algorithm 3 still applies, with the mention that the areas s_{ij} have different formulae (see Equation C.1 in the supplemental material), Y is $k \times (2k-1)$, F is $k \times k$

$$
Y = \begin{pmatrix}
\underbrace{\begin{matrix} 0 & \ldots & 0 \\ 0 & \ldots & 0 \\ \vdots & & \vdots \\ 0 & \ldots & 0 \end{matrix}}_{k-1} &
\begin{matrix} L(S_{11}) & \ldots & L\left(S_{1,k-1}\right) & L(S_{1k}) \\ L(S_{21}) & \ldots & L\left(S_{2,k-1}\right) & 0 \\ \vdots & \iddots & \vdots & \vdots \\ L(S_{k1}) & \ldots & 0 & 0 \end{matrix}
\end{pmatrix},\ F = \begin{pmatrix}
0 & 0 & \ldots & 0 & 1 \\
0 & 0 & \ldots & 1 & 1 \\
\vdots & \vdots & \iddots & \vdots & \vdots \\
0 & 1 & \ldots & 1 & 1 \\
1 & 1 & \ldots & 1 & 1
\end{pmatrix},
$$

Algorithm 3 Slice partition for bounded, monotonic trawls

```
Input: Sampler $S(\text{area})$ which returns independent samples with the same law
    as $L(A)$, where $\text{Leb}(A) = \text{area}$; number of trawls to be simulated $k$ and
    distance $\tau$ between them; $I = \lceil -T/\tau \rceil$.
Output: Vector containing the simulated values of the trawl process at times
    $\tau, \dots, k\tau$.
function MAIN($S$, $k$, $\tau$, $I$)
    Compute the areas $s_{ij}$ from (3.3),(3.4)  ▷ Requires $I$ integrations and $I$
differences
    $F \leftarrow \text{tril}(I)$        ▷ $I \times I$ lower triangular matrix of ones as in (3.5)
    $Y \leftarrow \text{zeros}(I, I + k - 1)$
    for $i = 1, \dots, I$ do
        for $j = 1, \dots, k$ do
            $Y[i, I - 1 + j] \leftarrow S(s_{ij})$
    return $Y * F$                                ▷ Convolution step
```

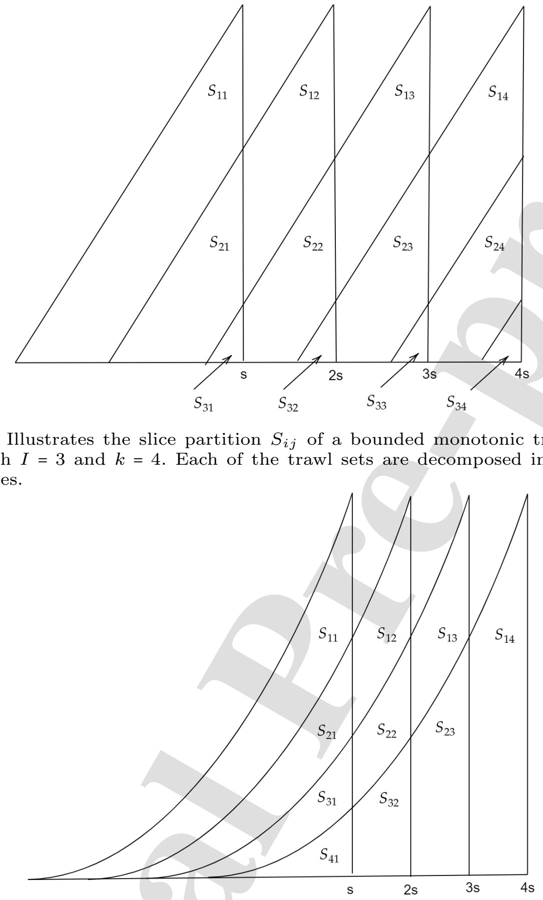
and we now perform $\mathcal{O}\left(k^2\right)$ operations. In practice, some of these $k(k+1)/2$ slices might have areas below machine precision, and for the purpose of computer simulations, we can approximate $Y * F$ by $Y[1:\text{n},:] * F[1:\text{n},:]$, where $Y[1:\text{n},:]$ and $F[1:\text{n},:]$ are obtained from Y and F by discarding the last $k-n$ rows, where $n \in \{1, \dots, k-1\}$. In this case, the first $k-n$ trawls are not simulated exactly and the last n trawls are simulated exactly; further, since the errors $\{\epsilon_l^n\}_{l=1}^{k-n}$ are given by

$$\epsilon_l^n = L\left(\bigcup_{\substack{1\leq j\leq l\\ i>n, i+j\geq l+1}} S_{ij}\right),$$

we have access to the joint distribution of the errors, and in particular, to the mean and variance of the errors. This can be used to calibrate the truncation parameter n.

Remark 3.4. Note that Theorem 3.3 of the previous subsection establishes not just convergence of X^ϵ to X at discretely observed times $\tau, \dots, k\tau$, but convergence of stochastic processes in the supremum norm. A similar result holds for the slice partition algorithm. To this end, simulate X via the slice partition method at discrete times $\mathcal{D}_n = \{0, \frac{1}{2^n}, \dots, 1\}$ and define $X^n(t) = X\left(\frac{\lfloor 2^n t\rfloor}{2^n}\right)$, where $\lfloor\cdot\rfloor$ is the floor function.

Theorem 3.5. *The sequence of stochastic processes X^n converges a.s. to X in Skorokhod's J1 topology.*



(a) Illustrates the slice partition S_{ij} of a bounded monotonic trawl, with $I = 3$ and $k = 4$. Each of the trawl sets are decomposed into I slices.

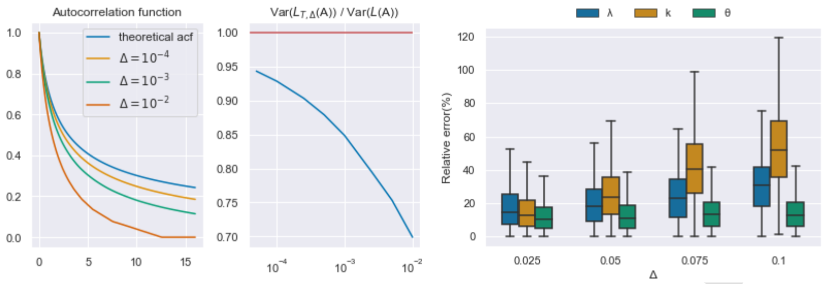
(b) Illustrates the slice partition S_{ij} of monotonic trawl process with infinite decorrelation time and with $k = 4$. For each i, there are $k - i + 1$ slices S_{ij}.

Figure 4: The slice partition for monotonic trawls with (a) finite decorrelation time and (b) infinite decorrelation time

3.4 Summary of convergence properties and discussion of algorithm requirements

In Subsections 3.1-3.3 we presented the grid discretization, compound Poisson and slice partition algorithms for the simulation of trawl processes; we derived their computational complexity and convergence properties and further discussed efficient implementation methods. Based on this analysis, we can compare the schemes and discuss in which cases one should be used over the other.

The grid discretization method is the most general one and can be employed to approximately simulate not only trawl processes, but also volatility-modulated, kernel weighted trawl processes and ambit fields, as defined in Sections 4 and 5. Nevertheless, it does not share the same convergence properties as the other two algorithms and further, it is the only algorithm which affects the autocorrelation function of the simulated trawl process. To quantify the discretization error by statistical measures, let $\rho_{T,\Delta}(h) := \operatorname{Corr}\left(L_{T,\Delta}\left(A_t\right), L_{T,\Delta}\left(A_{t+h}\right)\right)$ be the theoretical autocorrelation function of the trawl process simulated by the grid method with truncation parameter T and step-size $\Delta = \Delta_t = \Delta_x$. Let $\operatorname{Var}\left(L_{T,\Delta}\left(A\right)\right)$ be the corresponding variance. Figure 5a shows slow convergence of $\rho_{T,\Delta}(h)$ to $\rho(h)$ and of $\operatorname{Var}\left(L_{T,\Delta}\left(A\right)\right)$ to $\operatorname{Var}\left(L(A)\right)$ for $A = \{(t,x) \in \mathbb{R}^2 : t < 0, 0 < x < \phi(t)\}$ with $\phi: (-\infty, 0] \to \mathbb{R}_{\geq 0}$ given by $\phi(t) = 0.5(1-t)^{-1.5}$. The simulation error in the finite scale (non-asymptotic) regime depends heavily on the rate of decay of the trawl function ϕ. Consequently, accurately simulating long memory trawl processes, such as the ones with $\phi_H(t) \propto (1-t)^{-H}$, $\rho_H(t) = (1-t)^{-H+1}$ for $H \in (1,2)$ becomes increasingly difficult with the grid method as $H \to 1$. Moreover, the discretization error is reflected not only at the simulation level, through the autcocorrelation function and moments of the marginal dsitribution, but also when inferring the parameters of the simulated trawl process, as shown in Figure 5b. This is particularly important when carrying out simulation-based inference or when comparing two methods to infer the parameters of the trawl process, such as in [28]. Performing high accuracy or exact simulations ensures that the inference error is entirely due to the inference method rather than the simulation error.



(a)

(b)

Figure 5: (a) Autocorrelation and variance of the trawl process with trawl function $\phi\colon(-\infty,0] \to \mathbb{R}_{\geq 0}$ given by $\phi(t) = 0.5(1-t)^{-1.5}$, simulated with the grid method for different values of Δ. For each Δ, we set $T = \Delta^{-0.5}$, which satisfies Theorem 3.2 and ensures that the discretization error dominates the truncation error. Note that the autocorrelation function and variance are slow to convergence, and that both effects compound on the autocovariance function. By (2.2) and (2.3), $\operatorname{Var}\left(L_{T,\Delta}\left(A\right)\right)/\operatorname{Var}\left(L\left(A\right)\right)$ is equal to $\mathbb{E}\left[L_{T,\Delta}\left(A\right)\right]/\mathbb{E}\left[L\left(A\right)\right]$ and to the ratio of the area of the cells included in the simulation to the area of the trawl set A. (b) Boxplots of relative error for GMM parameter estimates from 1000 simulations for each of $\Delta = 0.025, 0.05, 0.075$ and 0.1. Here $\tau = 0.15$, $k = 1000$, $L^{'} \sim \text{Gamma}(k,\theta)$ with $k = 2$, $\theta = 3$ and $\phi:(-\infty,0] \to \mathbb{R}_{\geq 0}$ given by $\phi(t) = \lambda e^{\lambda t}$ with $\lambda = 1$. We match the first two empirical moments to infer k, θ and the empirical autocorrelation function at lags 1, 3, 5 to infer λ. The results appear to be robust to the choice of lags. Note the improvements across all parameters in the relative error as $\Delta \downarrow 0$.

The above observations agree with Sections 4 and 5 of [2], in which the grid method was applied to simulate a Spatio-temporal Ornstein-Uhlenbeck (STOU) process of the form

$$Y_t(x) = \int_{A_t(x)} e^{-\lambda(t-s)} L(\mathrm{d}\mathbf{z}, \mathrm{d}s).$$

Further, it was shown in [2] that tailoring the type of the grid to match the shape of the set A can reduce the effects of the discretization procedure on spatio-temporal correlations and improve convergence. In particular, the authors approximated $Y_t(\mathbf{x})$ by a finite sum on a rectangular grid and on a diamond grid and noticed faster convergence of the diamond grid scheme in some parameter regimes. The error analysis for STOU processes is complicated by the presence of the kernel $e^{-\lambda(t-s)}$, which makes it difficult to distinguish between truncation error, kernel discretization error and Lévy basis discretization error. All in all, tailoring the grid to each particular trawl shape is time-consuming and the algorithm itself is both memory and computationally expensive. The redundancy can be seen immediately: many of the cells are contained in just one of the sets $A_\tau, \ldots, A_{k\tau}$. Thus, instead of simulating all of these cells, we can either discard grid methods and attempt the compound Poisson method, or we could simulate from the slice partition, which can be seen as the optimal grid.

Accompanied by a Gaussian sampler, the compound Poisson scheme can be used to simulate the jump part of the Lévy basis and is exact when the Lévy measure

l can be normalised to a probability measure. Even if this is not the case, the algorithm has good convergence properties: the approximation X^{ϵ} obtained by truncating the jumps at some threshold ϵ converges in the supremum norm and in $\mathcal{L}^2$ to the trawl process X as $\epsilon \to 0$. Simulating from the potentially truncated Lévy measure can be implemented by methods such concave-convex adaptive rejection sampling [27] or MCMC. In general, this may require custom-made samplers or be computationally expensive.

Alternatively, the slice partition method, which can be seen as a natural generalization of the grid discretization, is the only one to result in exact simulation of the trawl process, regardless of the Lévy measure and trawl set. Further, we can directly trade off the accuracy and speed of the simulation scheme by neglecting slices with small areas, as discussed in Subsection 3.3. For faster, approximate simulation, the choice between the compound Poisson and slice partition schemes depends on the number k of trawls to be simulated, the spacing τ between the trawls and the difference in the cost of obtaining samples with law $L(A)$ for various values of $\text{Leb}(A)$ versus samples from the potentially truncated Lévy measure. Generally speaking, if both samples from $L(A)$ and l are available, the slice partition method is suitable for increasing domain simulations where the trawls are sampled at equidistant times, whereas the compound Poisson method is suitable for infill simulations and non-equidistant times.

Finally, note that Algorithms 1 and 3 require samples from $L(A)$ for some sets A. Let μ be the probability distribution of $L(A)$. In many cases of interest, such as the ones in Subsection 2.2, μ is part of a family of named probability distributions for which efficient samplers are already available. Nevertheless, μ can also be specified through its cumulant transform $C(\theta, \mu)$.We show that samples from μ can be obtained efficiently even in this case, under the mild assumption that the discrete and continuous components of μ can be separated. Indeed, by the Lebesgue decomposition theorem, μ can be decomposed into $\mu_{\text{ac}} + \mu_{\text{sc}} + \mu_{\text{d}}$ for some absolutely continuous measure μ_{ac}, some singular continuous measure μ_{sc} and some discrete measure $\mu_{\text{d}} = \sum_{i=1}^{\infty} p_i \delta_{a_i}$, where δ_x is the Dirac measure at x, p_i are strictly positive and a_i are non-zero real numbers. To exclude pathological cases, assume that $\mu_{\text{sc}} = 0$ and further that given $C(\theta, \mu)$, we can separate $C(\theta, \mu_{\text{ac}})$ from $C(\theta, \mu_d) = \sum_{i=1}^{\infty} e^{i\theta a_i} p_i$. We can sample from μ_d, as it has discrete support. We can also sample from μ_{ac}, by means of efficiently inverting the Fourier transform of μ_{ac}, which we discuss next.

Let f and F be the probability density, respectively cumulative distribution functions of μ_{ac}. The inversion method samples from μ_{ac} by solving $x = F^{-1}(u)$ where u is generated from the uniform distribution on $(0,1)$. When analytic expressions are not available for F^{-1}, a numerical procedure such as Newton-Raphson can be used. This amounts to iterating

$$x_{k+1} = x_k - \frac{F(x_k) - u}{f(x_k)}, \tag{3.6}$$

from a starting point x_0 until a predefined tolerance level is achieved. To account for regions where F is flat, a standard modification of Newton-Raphson can be used, which switches to the bisection method whenever necessary [29]. It remains to approximate the values of $F(x_k)$ and $f(x_k)$ by numerical inversion of the Fourier transform $w \to \int e^{iwx}\mu_{\text{ac}}(\mathrm{d}x) = e^{C(\theta,\mu_{\text{ac}})}$ of μ_{ac}. There is a rich literature on the topic of Fast Fourier Transform algorithms and quadrature methods for the approximation of probability functions. We mention [30], which provides a comprehensive exposition and error analysis, and [31], which employs the Gil-Pelaez inversion formula in conjunction with the fast Fourier transform algorithm to draw samples from distributions with known cumulant transform. In the case of distributions supported on the positive or negative real line, such as Gamma and Inverse Gaussian, we can work with the Laplace transform instead of the Fourier transform. [32] demonstrates that the Post-Widder inversion formula can be used to approximate f and F specifically for infinitely divisible distributions. Alternatively, [29] takes a general approach and approximates the Bromwich inversion integral of the Laplace transform of μ_{ac} by the trapezium rule and by employing the Euler summation to accelerate convergence. The paper provides an extensive error analysis and an R script, which we adapt for Python and make available at [12]. All in all, there are multiple off-the-shelf algorithms which provide arbitrary accuracy and can aid in the simulation of trawl processes and ambit fields.

4 Extensions to volatility modulated, kernel-weighted trawl processes

Trawl processes are stationary, infinitely divisible and ergodic processes which can describe a wide range of possible serial correlation patterns in data. Nevertheless, many systems of interest are inherently non-stationary; examples include precipitation data from [33] and financial time series. To model such behaviour, steps are usually taken to transform the initial process into a stationary one, to which standard methods can be applied. Furthermore, recent empirical work in areas such as environmental sciences in [16] and energy pricing in [17, 18] shows the presence of volatility clusters, and thus of stochastic volatility, which cannot be replicated by traditional models. In the following, we show that both non-stationarity and stochastic volatility can easily be incorporated into the trawl process framework and the same algorithms can be used for efficient simulation, despite the more complicated structure. We present the non-stationary case first and then the volatility modulated one. Finally, we extend the aforementioned methodology to the spatio-temporal case in Section 5.

Since trawl processes are given by the Lévy basis evaluated over the trawl sets, and depending on the kind of desired non-stationary behaviour, we can change either the trawl sets or the Lévy basis. At the level of the trawl sets, we can use different trawl functions $\phi_t \colon (-\infty, 0] \to \mathbb{R}_{\geq 0}$ to define $A_t = \{(s,x) \colon 0 < x < \phi_t(s - t)\}$, which changes both the correlation structure and marginal distributions. At the level of the Lévy basis, we can drop the homogeneity assumption of Definition 2.1, which allows the distribution of the jumps of L_j to depend on the point $\mathbf{z} = (t,x)$ where they appear, the jumps to have a nonuniform intensity measure, and even for the interaction between the jump values and the jump intensity. Similarly, the distribution of $L_g(A)$ is allowed to depend on the points in A, not just on Leb(A). The following result from [19, Proposition 2.1] reflects the extra flexibility of inhomogeneous Lévy basis over homogeneous ones.

Lemma 4.1 (Cumulant of Lévy bases)**.** *Let L be a Lévy basis on $S \subset \mathbb{R}^d, \theta$ a real number and A in $\mathcal{B}_b(S)$. Then*

$$C\left(\theta, L(A)\right) = \int_A \left(i\theta\xi(\mathbf{z}) - \frac{1}{2}\theta^2 a(\mathbf{z}) + \int_{\mathbb{R}} \left(e^{i\theta y} - 1 - i\theta y \mathbf{1}_{[-1,1]}(y) \right) l(\mathrm{d}y, \mathbf{z}) \right) c(\mathrm{d}\mathbf{z}), \tag{4.1}$$

where $\xi\colon S \to \mathbb{R}$, $a\colon S \to \mathbb{R}_{\geq 0}$, $l(\cdot, \mathbf{z})$ is a Lévy measure on $\mathbb{R}$ for each $\mathbf{z}$ in S and c is a measure on S, called the control measure, such that the above integral is well defined. Similarly to Definition 2.3, functions ξ and a give the local drift and variance of the Gaussian component at $\mathbf{z}$.

To each $\mathbf{z} = (t, \mathbf{x})$ in S, we can associate an infinitely divisible random variable $L^{'}(z)$ with

$$C\left(\theta, L^{'}(\mathbf{z})\right) = i\theta\xi(\mathbf{z}) - \frac{1}{2}\theta^2 a(\mathbf{z}) + \int_{\mathbb{R}} \left(e^{i\theta y} - 1 - i\theta y \mathbf{1}_{[-1,1]}(y) \right) l(\mathrm{d}y, \mathbf{z}),$$

which we call the Lévy seed at $\mathbf{z}$. Then

$$C\left(\theta, L(A)\right) = \int_A C\left(\theta, L^{'}(z)\right) c\left(\mathrm{d}\mathbf{z}\right),$$

and the distribution of $L^{'}(\mathbf{z})$ can be seen as the distribution of the infinitesimal $L(\mathrm{d}\mathbf{z})$. As opposed to the homogeneous case in Definition 2.3, this is now a function of $\mathbf{z}$. Further, to exclude pathological cases, assume that c has no discrete or singular continuous part. In this case, without loss of generality, we can incorporate the Radon–Nikodym derivative $\frac{\mathrm{d}c}{\mathrm{d}Leb}$ into $\xi(\mathbf{z})$, $a(\mathbf{z})$, $l(\mathrm{d}y, \mathbf{z})$ and set $c =$ Leb. Then the intensity function of the jumps at $\mathbf{z}$ is given by the total mass $l(\mathbb{R}, \mathbf{z})$.

Despite the great flexibility, we are not aware of settings, apart from theoretical study, where inhomogeneous Lévy bases have been used. We propose the kernel-weighted trawl processes, which encompasses both the changes in the trawl sets and Lévy basis by introducing a kernel with respect to a homogeneous Lévy basis. This results in a more general class of trawl processes with compact notation for which the slice partition and compound Poisson simulation algorithms can be applied with few or no modifications.

Definition 4.2. Let $S \subset \mathbb{R}^d$, $K_t\colon (S, \mathcal{B}_{\mathrm{Leb}}(S)) \to (\mathbb{R}, \mathcal{B}(\mathbb{R}))$ be a family of measurable mappings and $A \in \mathcal{B}_{\mathrm{Leb}}(S)$. The kernel-weighted trawl process over the collection of trawl sets $A_t = A + (t, 0)$ is given by

$$X_t = \int_{A_t} K_t(\bar{t}, \bar{\mathbf{x}}) L(\mathrm{d}\bar{t}, \mathrm{d}\bar{\mathbf{x}}),$$

under mild regularity conditions of the kernel K and where the integration is understood in the $\mathcal{L}^0$ framework of [19, Theorem 2.7]. We mention that it is enough for the $\mathcal{L}^2$-norm of the kernel over the trawl sets $t \to \int_{A_t} K_t^2(\bar{t}, \bar{\mathbf{x}})\, \mathrm{d}\bar{t}\mathrm{d}\bar{\mathbf{x}}$ to be bounded on compacts $[0, T]$ for the existence conditions to be satisfied. Similarly to the case of trawl processes, the cumulant transform of X_t is given by [19, Proposition 2.6]

$$C(\theta, X_t) = C\left(\theta, \int_{A_t} K_t(\bar{t}, \bar{\mathbf{x}}) L(\mathrm{d}\bar{t}, \mathrm{d}\bar{\mathbf{x}})\right) = \int_{A_t} C\left(\theta K_t(\bar{t}, \bar{\mathbf{x}}), L^{'}\right) \mathrm{d}\bar{t}\mathrm{d}\bar{\mathbf{x}}, \quad (4.2)$$

and if $\operatorname{Var}\left(L^{'}\right)$ is finite, the second order structure is given by

$$\mathbb{E}[X_t] = \mathbb{E}\left[L^{'}\right] \int_{A_t} K_t(\bar{t}, \bar{\mathbf{x}}) \mathrm{d}\bar{t}\mathrm{d}\bar{\mathbf{x}},$$
$$\operatorname{Cov}(X_t, X_s) = \operatorname{Var}\left(L^{'}\right) \int_{A_t \cap A_s} K_t(\bar{t}, \bar{\mathbf{x}}) K_s(\bar{t}, \bar{\mathbf{x}}) \mathrm{d}\bar{t}\mathrm{d}\bar{\mathbf{x}}.$$

We discuss the simulation of the kernel-weighted trawl process in increasing order of complexity. In Subsection 4.1, we consider processes of the form $X_t = \int_{A_t} K(\bar{t}, \bar{\mathbf{x}}) L(\mathrm{d}\bar{t}, \mathrm{d}\bar{\mathbf{x}})$. The kernel $K(\bar{t}, \bar{\mathbf{x}})$ can be thought of as the nonstationary component of the trawl process, or equally as a deterministic volatility. In Subsection 4.2, we extend to time dependent kernels $K_t(\bar{t}, \bar{\mathbf{x}})$, which allow for a more complicated joint distribution and, in particular, for the autocorrelation function to take both positive and negative values. Further, we discuss how the joint structure can be controlled solely through the kernel, by using a simple shape for the trawl set, such as a rectangle. In this setting, we recover the Brownian and Levy semistationary processes. Finally, in Subsection 4.3, we discuss modulation of the trawl process by a stochastic volatility. As in Section 3, we only discuss the simulation in the case $S = \mathbb{R}^2$; generalizations to $S \subset \mathbb{R}^d$ for $d > 2$ are straightforward.

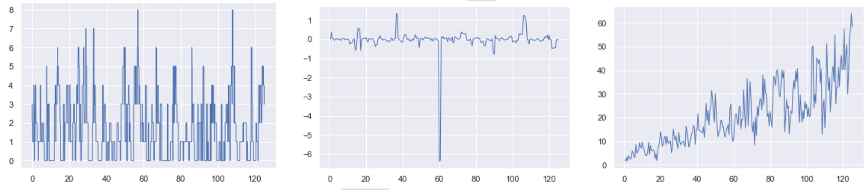
4.1 The non-stationary and deterministic volatility cases

We can directly generalize the slice partition method to simulate the kernel-weighted trawl process at times $\tau, \ldots, k\tau$. We decompose the sets $A_\tau, \ldots, A_{k\tau}$ into a collection $\mathcal{S}$ of disjoint slices S, sample $\int_S K(\bar{t}, \bar{x}) L(\mathrm{d}\bar{t}, \mathrm{d}\bar{x})$ for all $S \in \mathcal{S}$ and set

$$X_{l\tau} = \sum_{S \subset A_{l\tau}} \int_S K(\bar{t}, \bar{x}) L(\mathrm{d}\bar{t}, \mathrm{d}\bar{x}).$$

As discussed in Subsection 3.4, the above sampling may be done analytically or may involve numerically inverting the cumulant transform by a Fast Fourier Algorithm or the Laplace transform by approximating the inversion integral over a contour in the complex plane, such as the Bromwich contour. In turn, this requires multiple evaluations of the integrand, which is itself given as an integral in (4.2). If the integral in (4.2) is difficult to compute, we can separate the kernel-weighted trawl process into $X_t = \int_{A_t} K(\bar{t}, \bar{x}) L_g(\mathrm{d}\bar{t}, \mathrm{d}\bar{x}) + \int_{A_t} K(\bar{t}, \bar{x}) L_j(\mathrm{d}\bar{t}, \mathrm{d}\bar{x})$ and simulate the Gaussian part via the slice partition and the jump part via the compound Poisson method. For the Gaussian part, $\int_S K(\bar{t},\bar{x})L_g(\mathrm{d}\bar{t},\mathrm{d}\bar{x}) \sim \mathcal{N}\left(\mu\int_S K(\bar{t},\bar{x})\mathrm{d}\bar{t}\mathrm{d}\bar{x}, \sigma^2\int_S K^2(\bar{t},\bar{x})\mathrm{d}\bar{t}\mathrm{d}\bar{x}\right)$, where $L_g^{'} \sim \mathcal{N}\left(\mu,\sigma^2\right)$ and which requires only one integral evaluation per slice. For the jump part, note that L_j is a discrete measure supported on at most countably many points (t_i, x_i), with jumps y_i distributed according to the Lévy measure. In particular, we have $L_j = \sum_i y_i \delta_{(t_i,x_i)}$. Then $\int_{A_t} K(\bar{t},\bar{x})L_j(\mathrm{d}\bar{t},\mathrm{d}\bar{x}) = \sum_i y_i K(\bar{t}_i,\bar{x}_i)$. If the Lévy measure is infinite, then we truncate at some small jump threshold ϵ, as in Subsection 3.2, and X_t is approximated by $\sum_{i:y_i>\epsilon} y_i K(\bar{t}_i,\bar{x}_i)$. The uniform convergence of Theorem 3.3 is still valid for kernels K for which $t \to \int_{A_t} K^2\left(\bar{t},\bar{x}\right)\mathrm{d}\bar{t}\mathrm{d}\bar{x}$ is bounded on compacts $[0,T]$, as sketched in Remark A.2.

Figure 6 displays all the possibilities. Figure 6a has $L^{'} \sim \text{Poisson}(5)$, i.e. a finite Lévy measure for which the compound Poisson approach can be used for exact simulation and an integer-valued kernel $K\left(\bar{t},\bar{x}\right) = \left\lfloor 2\left(1+\bar{x}\right)\left(\bar{t}-\lfloor \bar{t}\rfloor\right)\right\rfloor$; note the sharp peaks induced by $\left(1+\bar{x}\right)$, which assigns larger values to points with larger $\bar{x}$ coordinate, i.e. points which 'leave' the trawl sets faster and further note the periodic trend induced by $(t-\lfloor t\rfloor)$. Figure 6b has $L^{'} \sim \text{Cauchy}(1)$ and $K\left(\bar{t},\bar{x}\right) = \sin\bar{t}$; then $\int_S K\left(\bar{t}\right) \sim \text{Cauchy}\left(\int_S K\left(\bar{t},\bar{x}\right)\mathrm{d}\bar{t}\mathrm{d}\bar{x}\right)$ and the slice partition can be applied without modification. Figure 6c has $L^{'} \sim \mathcal{N}(2,4) + \text{Gamma}(2,2)$ and $K\left(\bar{t},\bar{x}\right) = 1+0.1\bar{t}$. The numerical evaluation of the cumulant transform from (4.2) for multiple values of θ is expensive, hence we use the slice partition method for the Gaussian part and the compound Poisson method for the jump part; note the increasing trend.



(a) $L^{'} \sim \text{Poisson}(5)$, $K\left(\bar{t},\bar{x}\right) = \left\lfloor 2\left(1+\bar{x}\right)\left(\bar{t}-\lfloor\bar{t}\rfloor\right)\right\rfloor$.

(b) $L^{'} \sim \text{Cauchy}(1)$, $K\left(\bar{t},\bar{x}\right) = 0.1\sin\bar{t}$.

(c) $L^{'} \sim \mathcal{N}(2,4) + \text{Gamma}(2,2)$, $K\left(\bar{t}\right) = 1+0.1\bar{t}$.

Figure 6: Simulation of three kernel-weighted trawl processes between $t=0$ and $t=120$ for a triangular trawl set $A = \{(s,x) : -2 < s < 0, 0 < x < 1+s/2\}$ in a) and exponential trawl set $A = \{(s,x) : s < 0, 0 < x < e^s\}$ in b) and c); a) is simulated exactly, b) and c) are simulated at equidistant times $\tau,\ldots,250\tau$, where $\tau = 0.5$. Whereas b) just requires one integration per slice to determine the distribution of $\int_S K\left(\bar{t},\bar{x}\right)\mathrm{d}\bar{t}\mathrm{d}\bar{x}$, c) also requires drawing samples from the truncated Lévy measure l^ϵ with density $\frac{\mathrm{d}l^\epsilon}{\mathrm{d}Leb}\left(y\right) = 2y^{-1}e^{-3y}1_{y>\epsilon}$, where we choose $\epsilon = 5\cdot 10^{-3}$. We draw samples from l^ϵ by rejection sampling with a convex envelope of the log density, as inspired by [27].

4.2 Time dependent kernels

The extra dependence of $K_t(\bar{t},\bar{x})$ on t allows for a different kernel for each trawl set A_t and results in a more general joint distribution.The increased flexibility is matched by an increase in computational cost: the slice partition method can only be used for the Gaussian part, as dependent random variable sampling is difficult outside the Gaussian framework. Further, the simulated slices are not reusable, in the sense that we need to sample up to k values $\int_S K_\tau(\bar{t},\bar{x})L_g(\mathrm{d}\bar{t},\mathrm{d}\bar{x}),\ldots,\int_S K_{k\tau}(\bar{t},\bar{x})L_g(\mathrm{d}\bar{t},\mathrm{d}\bar{x})$ for each slice in the partition induced by $X_\tau,\ldots,X_{k\tau}$. Similarly, the approximation of $X_t \approx \sum_{i:y_i>\epsilon} y_i K_t(\bar{t}_i,\bar{x}_i)$ now requires up to k evaluations of the kernels at each jump location (t_i,x_i), the number of evaluations corresponding to the number of trawl sets $A_\tau,\ldots,A_{k\tau}$ which contain (t_i,x_i). Unlike the general algorithms from Section 3, the efficient simulation of kernel-weighted trawl processes depends heavily on the special structure to be exploited in each setting. In particular cases, we are still able to pull back to these off the shelf methods. We study one such example, inspired by [34]. Consider the generalized Ornstein–Uhlenbeck(OU) process $X_t = \int_{A_t} e^{\lambda(\bar{t}-t)}L(\mathrm{d}\bar{t},\mathrm{d}\bar{x})$. By the multiplicative property of the exponential function, we can break $e^{\lambda(\bar{t}-t)}$ into $e^{\lambda\bar{t}}e^{-\lambda t}$, simulate $\int_A e^{\lambda\bar{t}}L(\mathrm{d}\bar{t},\mathrm{d}\bar{x}),\ldots,\int_{A_{k\tau}} e^{\lambda\bar{t}}L(\mathrm{d}\bar{t},\mathrm{d}\bar{x})$ as discussed in Subsection 4.1 and then multiply the above values by $e^{-\lambda\tau},\ldots,e^{-\lambda k\tau}$. The same procedure is applicable if K is given by $\sin(\bar{t}-t)$, $\cos(\bar{t}-t)$, a linear combination of sines and cosines or if K is well approximated by such a linear combination.

More generally, consider $K_t(\bar{t},\bar{x}) = g(\bar{t}-t)$ for some square integrable g and $X_t = \int_{A_t} g(\bar{t}-t)L(\mathrm{d}\bar{t},\mathrm{d}\bar{x})$ for some Lévy basis L with finite $\mathrm{Var}\,(L^{'})$. Although this type of kernel-weighted trawl process is stationary, it is strictly more general than a trawl process, and it can exhibit negative correlations. The Fourier expansion methodology from Section 3 of [34] can be adapted to show that for a slice $S \subset \{(t,x) : (l-1)\tau \leq t \leq l\tau\}$, we have

$$\int_S g(\bar{t}-t)L(\mathrm{d}\bar{t}\mathrm{d}\bar{x}) \approx e^{-\lambda l\tau}\int_S e^{\lambda\bar{t}}\left(a_0 + \sum_{i=1}^{N} a_i \cos\left(n\pi(\bar{t}-l\tau)/\tau\right)\right)L\left(\mathrm{d}\bar{t},\mathrm{d}\bar{x}\right)$$
$$= e^{-\lambda l\tau}\left(a_0\int_S e^{\lambda\bar{t}}L\left(\mathrm{d}\bar{t},\mathrm{d}\bar{x}\right) + \mathfrak{R}\left(\sum_{i=1}^{N} a_i e^{-inl\pi}\int_S e^{in\pi\bar{t}/\tau}L\left(\mathrm{d}\bar{t},\mathrm{d}\bar{x}\right)\right)\right),$$

where the approximation is understood in $\mathcal{L}^2$, N is the number of terms in the approximation, $a_0,\ldots,a_N$ are constants and λ is a parameter to be calibrated. In the above formula, we require N evaluations per slice, as compared to up to k evaluations. This approximation removes the t dependency of the kernel and works well as long as N is smaller than k. The trade off is that the convergence is just in $\mathcal{L}^2$, and not uniformly on compacts. An observation on the type of kernels to be used is in order. Since the autocorrelation function can be modelled through the shape of the trawl set A, the kernel can in principle be chosen from a family of straightforward functions, with which we can work easily, with the aim of inducing drift, seasonal behaviour or deterministic volatility. Consequently, for most practical purposes, it is enough to consider kernels to which the above simplifications apply.

Finally, note that the formulation in Definition 4.2 and the $\mathcal{L}^0$ integration framework of [19] do not require A to be bounded. Indeed, provided the kernel integrability conditions are satisfied, we can choose a trawl set with a simple geometry, such as an unbounded rectangle $A = \{(s,x) : s < 0, 0 < x < 1\}$ and control the joint distribution solely through the kernel. If K is chosen to depend only on t and $\bar{t}$, we recover the Lévy semistationary processes $X_t = \int_{-\infty}^{t} K_t(\bar{t})H(\mathrm{d}\bar{t})$, where H is a two-sided Lévy process with $H_1 \overset{d}{=} L^{'}$ and which are extensively studied in [15, Chapters 1,2 and 10].

4.3 Volatility modulated trawl processes

A stochastic volatility σ can easily be added to the trawl process framework.

Definition 4.3 (Kernel-weighted, volatility modulated trawl processes)**.** Let $S \subset \mathbb{R}^d$, $K_t : (S, \mathcal{B}_{\mathrm{Leb}}(S)) \to (\mathbb{R}, \mathcal{B}(\mathbb{R}))$ be a family of measurable mappings and $A \in \mathcal{B}_{\mathrm{Leb}}(S)$. Let σ be a stochastic process on the same probability space as L. The kernel-weighted, volatility modulated trawl process over the collection of trawl sets $A_t = A + (t,0)$ is given by

$$X_t = \int_{A_t} K_t(\bar{t}, \bar{\mathbf{x}})\sigma(\bar{t})L(\mathrm{d}\bar{t}, \mathrm{d}\bar{\mathbf{x}}).$$

In general, if σ and L are dependent, the integration is understood in the sense of [20] and [21]. We restrict our attention to the case in which σ and L are independent. Then the integration can be defined conditionally on σ, using the same $\mathcal{L}^0$-framework of [19, Theorem 2.7], as for Definition 4.2. The second order structure is given by

$$\mathbb{E}\left[X_t | \mathcal{F}_\sigma^t\right] = \mathbb{E}\left[L^{'}\right] \int_{A_t} K_t(\bar{t}, \bar{\mathbf{x}})\sigma(\bar{t})\mathrm{d}\bar{t}\mathrm{d}\bar{\mathbf{x}}, \tag{4.3}$$

$$\operatorname{Cov}\left(X_t, X_s | \mathcal{F}_\sigma^t \vee \mathcal{F}_\sigma^s\right) = \operatorname{Var}\left(L^{'}\right) \int_{A_t \cap A_s} K_t(\bar{t}, \bar{\mathbf{x}})K_s(\bar{t}, \bar{\mathbf{x}})\sigma^2(\bar{t})\mathrm{d}\bar{t}\mathrm{d}\bar{\mathbf{x}}, \tag{4.4}$$

$$\begin{aligned} C\left(\theta, X_t | \mathcal{F}_\sigma^t\right) &= C\left(\theta, \int_{A_t} K_t(\bar{t}, \bar{\mathbf{x}})\sigma(\bar{t})L\left(\mathrm{d}\bar{t}, \mathrm{d}\bar{\mathbf{x}}\right)\right) \\ &= \int_{A_t} C\left(\theta K_t(\bar{t}, \bar{\mathbf{x}})\sigma(\bar{t}), L^{'}\right)\mathrm{d}\bar{t}\mathrm{d}\bar{\mathbf{x}}, \end{aligned} \tag{4.5}$$

where $\mathcal{F}_\sigma^t$ is the σ algebra generated by $\{\sigma(\bar{t}) : (\bar{t}, \bar{\mathbf{x}}) \in A_t \text{ for some } \bar{\mathbf{x}}\}$. The unconditional structure follows by integrating taking the expectation over σ. An extensive presentation can be found in [15, Chapter 5.3.2.1]. We restrict our attention to the case $S = \mathbb{R}^2$.

The observations from Subsection 4.2 still apply: the slice partition can be used for the Gaussian part and the compound Poisson method for the jump part; in some cases, the t dependence can be removed by means of a Fourier approximation. As explained before, the difficulty in sampling by numerically inverting the cumulant is that every step in the iterative procedure of (3.6) requires evaluations of the cumulant from (4.5) for multiple values of θ. In turn, each of these evaluations requires the values of σ and K for multiple arguments. In this situation, it is usually more expensive to simulate the volatility σ than to evaluate the kernel K, hence inverting the cumulant may not be practical. The difference between the general case and that of a Gaussian Lévy basis L_g is that conditionally on σ, the distribution of $\int_S K(\bar{t},\bar{x})\sigma(\bar{t})L(\mathrm{d}\bar{t},\mathrm{d}\bar{x})$ is fully specified by the two integrals $\int_S K(\bar{t},\bar{x})\sigma(\bar{t})\mathrm{d}\bar{t}\mathrm{d}\bar{x}$ and $\int_S K^2(\bar{t},\bar{x})\sigma^2(\bar{t})\mathrm{d}\bar{t}\mathrm{d}\bar{x}$, and sampling does not require other integral evaluations. The underlying property is that of closure under linear combinations and is satisfied by the family of Lévy stable distributions from Example 2.10. In particular, if $L^{'} \sim \mathrm{Stable}(\alpha,\, \beta,\, c,\, \mu)$ with $\alpha \neq 1$, then $\int_S K(\bar{t},\bar{x})\sigma(\bar{t})L(\mathrm{d}\bar{t},\mathrm{d}\bar{x})|\mathcal{F}_\sigma^t \sim \mathrm{Stable}(\alpha,\tilde{\beta},\tilde{c},\tilde{\mu})$, where $\tilde{c} = c\left(\int_S\left|K(\bar{t},\bar{x})\sigma(\bar{t})\right|^\alpha \mathrm{d}\bar{t}\mathrm{d}\bar{x}\right)^{1/\alpha}$, $\tilde{\beta} = \frac{\beta}{\tilde{c}}\left(\int_S\left|K(\bar{t},\bar{x})\sigma(\bar{t})\right|^\alpha \mathrm{sign}\left(K(\bar{t},\bar{x})\sigma(\bar{t})\right)\mathrm{d}\bar{t}\mathrm{d}\bar{x}\right)^{1/\alpha}$ and $\tilde{\mu} = \int_S K(\bar{t},\bar{x})\sigma(\bar{t})\mathrm{d}\bar{t}\mathrm{d}\bar{x}$, provided the integrals are finite. In the definition of $\tilde{\beta}$, we write x^α for $\mathrm{sign}\,(x)|x|^{1/\alpha}$. A similar, simpler formula holds for $\alpha = 1$, which corresponds to the the Cauchy distribution translated by a location parameter. Consequently, as longs as $L^{'}$ has a stable distribution and the terms $\tilde{c}$, $\tilde{\beta}$, $\tilde{\mu}$ can be approximated well, inverting the cumulant is feasible, by first simulating σ and then sampling X conditionally on σ. Note that the restriction on the distribution of $L^{'}$ is not significant. Although the distribution of X_t conditional on σ is Stable, the unconditional distribution of X_t does not have to be Stable.

The flexible marginal distribution and autocorrelation structure, as well as the computational efficiency and convergence properties of the simulation schemes in Section 3 recommend the trawl process as a candidate for the stochastic volatility component. Thus we model σ^2 with a trawl process. Figure 7 shows such an example, where X_t is conditionally Gaussian (which corresponds to $\alpha = 2$ in the family of Stable distributions) on the volatility. More precisely, we use a Gaussian Lévy basis L and model σ^2 as a stationary trawl process with long memory and Inverse Gaussian marginal distribution.

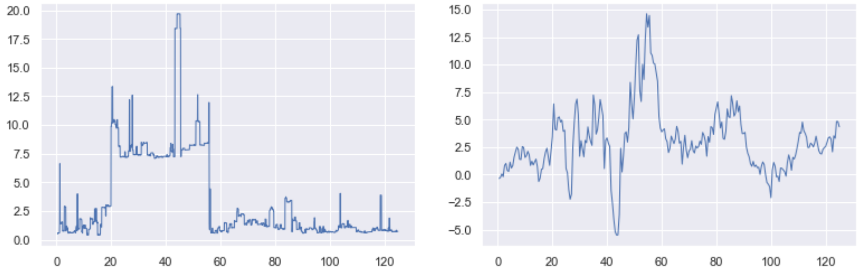
5 Extensions to ambit field simulation

So far we concentrated on the simulation of trawl processes, which amounts to evaluating the Lévy basis, potentially modulated by a kernel and stochastic volatility, over a collection of time-indexed trawl sets A_t. Note again that the trawl sets include an abstract spatial dimension in addition to the temporal dimension, which allows for a flexible joint distribution and autocorrelation function of the trawl process. A natural extension of the trawl process to spatio-temporal fields is the simple ambit field Y given by

$$Y_t(x) = L\left(A_t(\mathbf{x})\right),$$

and more generally, the ambit field Y given by

$$Y_t(\mathbf{x}) = \int_{A_t(\mathbf{x})} K_{t,\mathbf{x}}\left(\bar{t},\bar{\mathbf{x}}\right)\sigma(\bar{t},\bar{\mathbf{x}})L(\mathrm{d}\bar{t},\mathrm{d}\bar{\mathbf{x}}).$$



(a) Realisation of σ^2, modelled as a trawl process with long memory and Inverse Gaussian marginal

(b) Realisation of the volatility modulated trawl process $X_t = \int_{A_t} \sigma(\bar{t}) L(\mathrm{d}\bar{t}, \mathrm{d}\bar{x})$

Figure 7: b) shows the simulation of a volatility modulated trawl process $X_t = \int_{A_t} \sigma(\bar{t}) L(\mathrm{d}\bar{t}, \mathrm{d}\bar{x})$ with $L^{'} \sim \mathcal{N}(1,1)$ and $A = \{(s,x) : 0 < x < 0.25e^{0.25s}\}$ at times $\tau, \ldots, 250\tau$, where $\tau = 0.5$. We use the slice partition method for Lévy stable distributions. Conditionally on the values of σ, $\int_S \sigma_s \mathrm{d}L_s \sim \mathcal{N}(\int_S \sigma(\bar{t})\mathrm{d}\bar{t}\mathrm{d}\bar{x}, \int_S \sigma^2(\bar{t})\mathrm{d}\bar{t}\mathrm{d}\bar{x})$. To approximate these integrals we simulate the volatility on a fine equidistant grid. We model σ^2 as a trawl process with trawl set $B = \{(s,x) : 0 < x < 0.5(1-s)^{0.5}\}$ and Lévy seed $L^{'}_{\sigma} \sim$ Inverse Gaussian$(2,1)$, which we simulate from -10 to 250τ in steps of $\tilde{\tau} = 0.05$. A realisation of σ^2 is shown in a). The trawl functions are normalized such that the areas of A and B are 1.

Ambit fields have already been used in turbulence and tumor growth modelling in [23], and also outside spatio-temporal statistics, for example in electricity futures pricing [35].

We start Subsection 5.1 by showing that the slice partition algorithm can be generalized to the simulation of simple ambit fields. As opposed to simulation via compound Poisson processes, which leads to an increased cost per simulation, the slice partition method can be implemented in a way such that the calculations required for higher accuracy only need to be performed once, before the simulation, leading to lower computational cost across simulations. Finally, we expand on the simulation of ambit fields in Subsection 5.2. To distinguish between the temporal and spatio-temporal cases, we refer to time and space indexed sets $A_t(\mathbf{x})$ as ambit sets.

5.1 The slice partition method for simple ambit fields

As in the trawl case, the autocovariance and autocorrelation structures for the simple ambit field $Y_t(\mathbf{x}) = L(A_t(\mathbf{x}))$

$$\mathrm{Cov}(Y_{t_1}(\mathbf{x_1}), Y_{t_2}(\mathbf{x_2})) = \mathrm{Leb}\left(A_{t_1}(\mathbf{x}) \cap A_{t_2}(\mathbf{x_2})\right) \mathrm{Var}(L^{'})$$
$$\mathrm{Corr}(Y_{t_1}(\mathbf{x_1}), Y_{t_2}(\mathbf{x_2})) = \mathrm{Leb}\left((A_{t_1}(\mathbf{x_1}) \cap A_{t_2}(\mathbf{x_2})\right) / \mathrm{Leb}\left(A\right)$$

and the cumulant transform $C\left(\theta, Y_t(\mathbf{x})\right) = \mathrm{Leb}(A) C(\theta, L^{'})$ present simple ambit fields as a tractable approach to modelling spatio-temporal data. We present the simulation algorithm for $d = 2$ dimensions, as generalizing to more spatial dimensions is straightforward. The goal is then to simulate the simple ambit field $Y_t(x)$ at coordinates $\{(j\tau, ix) : 1 \le j \le k_t, 1 \le i \le k_s\}$.

For ease of notation, let $A_{ij} = A_{j\tau}(ix)$ and define the lexicographic ordering $(m,n) \prec (i,j)$ if $m < i$ or $m = i$ and $n < j$, with equality when $m = i$ and $n = j$. Intuitively, we order trawls from left to right and from bottom to top. For ease of presentation, assume that the ambit set is bounded; we relax this assumption in Section C.2. We say a set $S \subset \mathbb{R}^2$ is a minimal slice if it can be represented as

$$S = \bigcap_{(i,j)\in K} A_{ij}$$

for some indicator set $K \subset \mathbb{Z}^2$ and further $S \cap A_{ij} = \varnothing$ for any $(i,j) \notin K$. The ambit sets can be partitioned into disjoint minimal slices; thus, to simulate the simple ambit field, it is enough to simulate all the minimal slices and keep track of which ambit sets A_{ij} each slice belongs to. In this algorithm, we simulate Lévy basis over trawls sets from left to right along each row, moving over rows from bottom to top. At step (k,l), we simulate slices that belong to A_{kl} and have empty intersection with trawls that are left and on the same row as A_{kl}, or bottom of A_{kl}.

Formally, define $I_t = \left\lceil \frac{-T}{\tau} \right\rceil$ and $I_s = \left\lceil \frac{\phi(0)}{x} \right\rceil$. Note that the sets A_{ij} and $A_{i'j'}$ are disjoint whenever $|i - i^{'}| \geq I_s$ or $|j - j^{'}| \geq I_t$. Thus each minimal slice S can be represented by a minimal pair $(k,l) = \min\{(i,j) \colon S \subset A_{ij}\}$, where the minimum is understood in the sense of $\prec$, and by an $I_s \times I_t$ indicator matrix K^S, where

$$\left(K^S\right)_{ij} = \begin{cases} 1 \text{ if } S \subset A_{i+k-1,j+l-1}, \\ 0 \text{ otherwise.} \end{cases}$$

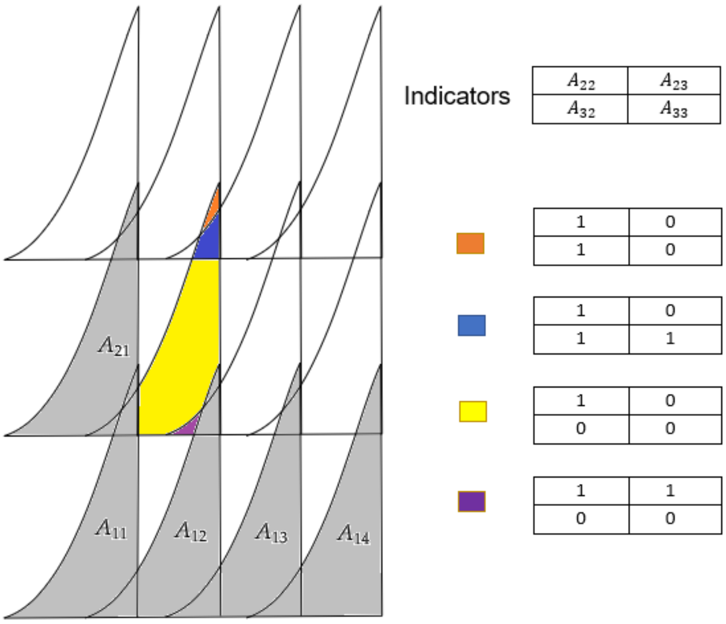
Let $\mathcal{S}_{kl}$ be the set of minimal slices S whose minimal pair is (k,l). Then (see Figure 8)

$$A_{kl} \setminus \bigcup_{(i,j)\prec(k,l)} A_{ij} = \bigcup_{S\in\mathcal{S}_{kl}} S.$$

By the translation invariance property of the grid of ambit sets $\{A_{ij}\}_{-\infty\leq i,j\leq\infty}$, $\mathcal{S}_{kl}$ and $\mathcal{S}_{k'l'}$ contain the same number of minimal slices S, with the same Lebesgue measures and with indicators K^S which are translated by $k - k^{'}$ and $l - l^{'}$. Hence, to simulate the simple ambit field, it is enough to determine the Lebesgue measures and indicators of minimal slices in S_{11}. We can identify the minimal slices S, approximate their Lebesgue measures and corresponding indicators K^S via Monte Carlo methods: sample points uniformly at random, keep track of the indicators K^S, count how many points are in each minimal slice and divide the count by the total number of points to estimate the areas, as described in Algorithm 4. To account for boundary effects and simulate $\{A_{ij}\}_{\substack{1\leq i\leq I_s \\ 1\leq j\leq I_t}}$ exactly, we simulate the minimal slices in

$$\bigcup_{\substack{-I_s+2\leq k\leq k_s \\ -I_t+1\leq l\leq k_t}} \mathcal{S}_{kl},$$

which means that we also simulate subsets of A_{ij} with $-I_s + 2 \leq i \leq 0, k_s < i \leq k_s + I_s - 1$, $-I_t + 2 \leq j \leq 0$, and $k_t < j \leq k_t + I_t - 1$. We discard the extra values. The pseudocode for the slice partition method for simple ambit fields is given in Algorithm 5. Note that the calculations required for a higher accuracy can be performed ahead of the simulation. Regardless of the number of simulations, we only have to perform this procedure once, leading to an amortised computational cost across simulations. Figure 9 displays two simulations of simple ambit fields. The above procedure can be generalized to unbounded trawls sets, as detailed in Section C.2. Just as in the slice partition algorithm for trawl processes, we have more slices to take into account and the computational complexity increases.



Indicators

A_{22}	A_{23}
A_{32}	A_{33}

🟧

1	0
1	0

🟦

1	0
1	1

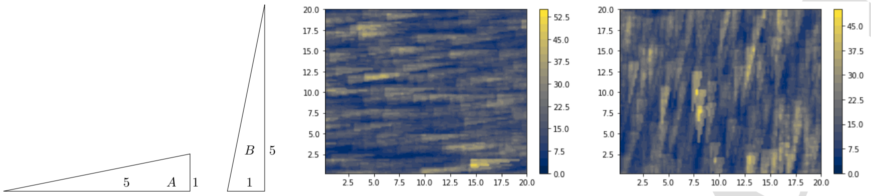
🟨

1	0
0	0

🟪

1	1
0	0

Figure 8: Assume that at step (k,l) with $k = l = 2$, we have already simulated the minimal slices which are situated left and on the same row as A_{22}, or bottom of A_{22}. These slices are shaded in gray and have been simulated at previous steps. The parts coloured in orange, blue, yellow and purple illustrate the minimal slices of $\mathcal{S}_{22}$, together with the corresponding indicator sets K^S. These four minimal slices are the slices to be simulated at step $(2,2)$. Simulating the simple ambit field is then equivalent to simulating all the minimal slices S and keeping track of the indicators K^S.



(a) Ambit sets A and B with side lengths of 1 and 5 units.

(b) Simulation of a simple ambit field with ambit set A.

(c) Simulation of a simple ambit field with ambit set B.

Figure 9: Simulations of two simple ambit fields with triangular ambit sets A and B and with $\tau = x = 0.2$, $k_t = k_s = 100$, $L^{'} \sim \text{Gamma}(2,3)$ and $N = 10^8$ samples for the slice area estimation procedure. The time, space axes are the horizontal, respectively vertical ones. Note that for ambit sets such as A and B, the minimal slices can be worked out by hand and the simulation is then exact. The values of the simulated ambit fields are given by the colorbars of b) and c). The shapes of the ambit sets induce qualitatively different spatio-temporal autocorrelation structures.

Algorithm 4 Slice estimation for bounded, monotonic ambit sets

```
Input: Trawl function $\phi$ and $T < 0$ with $\phi(T) = 0$; number of samples $N$ to be
    used in the estimation; sampler $U(a, b, n)$ which returns an $n$ dimensional
    vector of iid samples from the uniform distribution on $[a, b]$
Output: Hash table $H$[key:value] mapping the keys, $I_s \times I_t$ indicator matrices of
    minimal slices, to their correspoding values, given by the estimated Lebesgue
    measures of the minimal slices.
function SliceEstimation($U$, $N$, $\phi$, $T$, $I_t$, $I_s$, $\tau$, $x$)
    $t \sim U(0, \tau, N)$
    $x \sim U\left(x, x + \phi(0), N\right)$
    for $l \in \{1, \dots, N\}$ do
        if $x[l] > \phi(t[l] - \tau)$ and $0 < x[l] - x < \phi(t[l] - \tau)$ then    ▷ Exclude
                                        points which belong to $A_{01}$ and
                                        points which do not belong to $A_{11}$
            $I \leftarrow \text{zeros}(I_s, I_t)$               ▷ Indicator matrix for a minimal slice
            for $i \in \{1, \dots, I_s\}$ do
                for $j \in \{1, \dots, I_t\}$ do
                    if $x[l] - ix < \phi(t[l] - j\tau)$ then  ▷ Check if $(t[l], x[l]) \in A_{ij}$
                        $I[i, j] \leftarrow 1$
            if $I \in \text{keys}(H)$ then $H[I] \leftarrow H[I] + 1$ else $H[I] \leftarrow 1$
    for $I \in \text{keys}(H)$ do
        $H[I] \leftarrow H[I]\tau\phi(0)/N$                ▷ Approximate the area of a min-
                                                  imal slice $S$ with indicator $I$
    return $H$
```

5.2 Ambit field simulation

We turn our attention to the general case of ambit fields Y defined by

$$Y_t(x) = \int_{A_t(x)} K_{t,x}\left(\bar{t}, \bar{x}\right) \sigma(\bar{t}, \bar{x}) L(\mathrm{d}\bar{t}, \mathrm{d}\bar{x}),$$

Algorithm 5 Slice partition for bounded, monotonic ambit sets

Input: Trawl function ϕ and $T < 0$ with $\phi(T) = 0$; number of samples N to be used in the SliceEstimation procedure; sampler $U(a, b, n)$ which returns an n dimensional vector of iid samples from the uniform distribution on $[a, b]$; sampler $T(\text{area})$ which returns one sample from a collection of independent random variables with the same law as $L(A)$, where $\text{Leb}(A) = \text{area}$; number of ambit sets to be simulated k_t, k_s, distance τ, x between ambit sets.

Output: $k_s \times k_t$ matrix Y containing the values of the simulated simple ambit field at coordinates $\{(j\tau, ix) : 1 \leq j \leq k_t, 1 \leq i \leq k_s\}$

```
function SlicePartition($U$, $T$, $N$, $\phi$, $k_t$, $k_s$, $\tau$, $x$)
    $I_t, I_s \leftarrow \left\lceil \frac{-T}{\tau} \right\rceil, \left\lceil \frac{\phi(0)}{x} \right\rceil$
    $H \leftarrow$ SliceEstimation($U$, $N$, $\phi$, $I_t$, $I_s$, $\tau$, $x$)
    $Y \leftarrow \text{zeros}(k_s + 2I_s - 2,\ k_t + 2I_t - 2)$.    ▷ Corresponding to the matrix $L\left(A_{ij}\right)_{\substack{-I_s+2 \leq i \leq k_s+I_s-1 \\ -I_t+2 \leq j \leq k_t+I_t-1}}$
    for $k \in \{1, \dots, k_s + I_s - 1\}$ do
        for $l \in \{1, \dots, k_t + I_t - 1\}$ do
            for $I \in \text{keys}(H)$ do
                $c \leftarrow T(\text{area})$    ▷ Simulate $L(S)$ for each $S \in \mathcal{S}_{kl}$
                $Y[k : k + I_s - 1, l : l + I_t - 1]$ += $cI$.
    return $Y[I_s : I_s + k_s - 1, I_t : I_t + k_t - 1]$.    ▷ Corresponding to the matrix $L\left(A_{ij}\right)_{\substack{1 \leq i \leq k_s \\ 1 \leq j \leq k_t}}$
```

where K is a deterministic kernel and σ is a stochastic volatility field. We aim to simulate $Y_t(x)$ at coordinates $\{(j\tau, ix) : 1 \leq j \leq k_t, 1 \leq i \leq k_s\}$. Assume σ and L are independent. Then the second order structure and cumulant transform follow from Equations (4.3)-(4.5). Similarly to Subsection 4.3, we advocate for the use of a simple ambit fields for the stochastic volatility term σ, which can be simulated efficiently. Conditional on the values of σ, we can simulate Y.

Analogous to Lévy semistationary processes, we can choose ambit sets with simple geometries, such as unbounded rectangles $A_t(x) = \{(\bar{t}, \bar{x}) : \bar{t} < t, x < \bar{x} < x{+}1\}$, and control the joint structure solely through the kernel. In this case, we can identify the minimal slices by hand. In general, if we use a more complicated ambit set and a simple kernel, this is not possible and we identify minimal slices by their indicator matrices, as in Subsection 5.1. In both cases, we separate L into L_g and L_j and simulate their contributions independently, conditionally on σ. The jump part $Y_t(x) = \int_{A_t(x)} K_{t,x}\left(\bar{t}, \bar{x}\right) \sigma(\bar{t}, \bar{x}) L_j(\mathrm{d}\bar{t}, \mathrm{d}\bar{x})$ can be approximated up to small jump truncation by a discrete sum, as in Section 4 and only requires access to the values of the kernel K and volatility σ. The Gaussian part $Y_t(x) = \int_{A_t(x)} K_{t,x}\left(\bar{t}, \bar{x}\right) \sigma(\bar{t}, \bar{x}) L_g(\mathrm{d}\bar{t}, \mathrm{d}\bar{x})$ requires computing the means and covariances

$$\begin{aligned}
&\mathbb{E}\left[L^{'}\right]\int_{A_t(x)} K_{t,x}(\bar{t},\bar{x})\sigma(\bar{t},\bar{x})\mathrm{d}\bar{t}\mathrm{d}\bar{x},\\
&\operatorname{Var}\left(L^{'}\right)\int_{A_{t_1}(x_1)\cap A_{t_2}(x_2)} K_{t_1,x_1}(\bar{t},\bar{x})K_{t_2,x_2}(\bar{t},\bar{x})\sigma^2(\bar{t},\bar{x})\mathrm{d}\bar{t}\mathrm{d}\bar{x},
\end{aligned}$$

which can be approximated similarly to Algorithm 4, by simulating points equidistantly or uniformly at random, evaluating the kernel and volatility at these points and taking the mean.

Further simplifications are possible in particular cases. If the kernel satisfies $K_{t,x}(\bar{t},\bar{x}) = g(\bar{t}-t,\bar{x}-x)$ for some square integrable g and $\operatorname{Var}\left(L^{'}\right)$ is finite, the Fourier approximation methodology from Subsection 4.2 can be used to remove the dependence of the kernel on t and x. Similarly, if $L^{'}$ has a stable distribution, it is enough to first simulate the volatility field σ and then approximate the parameters of the conditional distribution of $\int_S K_{t,x}(\bar{t},\bar{x})\sigma(\bar{t},\bar{x})L\left(\mathrm{d}\bar{t},\mathrm{d}\bar{x}\right)$ for all minimal slices S.

6 Conclusion

After presenting the elementary properties of Lévy bases in Section 2, we introduced three algorithms for the simulation of trawl processes in Section 3. We developed their theoretical error analysis, discussed their computational complexity and provided easily adaptable computer implementations. Further, we investigated the effects of approximating a Lévy basis through grid discretization and small jump truncation. While the three simulation schemes were initially presented in the trawl processes framework, we showed in Sections 4 and 5 that they are directly applicable to the more general settings of kernel-weighted, volatility modulated trawl processes, simple ambit fields and ambit fields. Moreover, we showed in Subsection 4.2 that simulation schemes previously studied in the literature could be combined with our methods for decreased computational cost. All of the above enable the implementation of high-accuracy simulation studies and simulation-based inference and bring Ambit Stochastics closer to widespread use when modelling real-world data.

Acknowledgements

We would like to thank Dan Crisan for constructive discussions and comments on earlier versions of the manuscript. Dan Leonte acknowledges support from the EPSRC Centre for Doctoral Training in Mathematics of Random Systems: Analysis, Modelling and Simulation (EP/S023925/1).

References

[1] O. E. Barndorff-Nielsen, J. Schmiegel, Ambit processes; with applications to turbulence and tumour growth, in: Stochastic analysis and applications, Springer-Verlag, Berlin, 2007, pp. 93–124.
URL https://doi.org/10.1007/978-3-540-70847-6_5

[2] M. Nguyen, A. E. D. Veraart, Spatio-temporal Ornstein–Uhlenbeck processes: Theory, simulation and statistical inference, Scandinavian Journal of Statistics 44 (1) (2017) 46–80.
URL https://doi.org/10.1111/sjos.12241

[3] K. Y. Jónsdóttir, A. Rønn-Nielsen, K. Mouridsen, E. B. Vedel Jensen, Lévy-based modelling in brain imaging, Scandinavian Journal of Statistics 40 (3) (2013) 511–529.
URL https://doi.org/10.1002/sjos.12000

[4] O. E. Barndorff-Nielsen, A. Lunde, N. Shephard, A. E. Veraart, Integer-valued trawl processes: A class of stationary infinitely divisible processes, Scandinavian Journal of Statistics 41 (3) (2014) 693–724.
URL https://doi.org/10.1111/sjos.12056

[5] R. Wolpert, M. Taqqu, Fractional Ornstein-Uhlenbeck Lévy processes and the Telecom process: Upstairs and downstairs, Signal Processing 85 (2005) 1523–1545.
URL https://doi.org/10.1016/j.sigpro.2004.09.016

[6] V. L. A. Courgeau, Extreme-trawl, https://github.com/valcourgeau/extreme-trawl, Github repository (2021).

[7] A. E. D. Veraart, Estimation and simulation of trawl processes, https://CRAN.R-project.org/package=trawl, R package (2021).

[8] M. Bennedsen, Likelihood-based-ivt, https://github.com/mbennedsen/Likelihood-based-IVT, GitHub repository (2021).

[9] M. Nguyen, Mixed spatio-temporal Ornstein-Uhlenbeck processes, https://bitbucket.org/mntd/stou-processes/src/master/, Bitbucket repository (2017).

[10] M. Nguyen, Mixed spatio-temporal Ornstein-Uhlenbeck processes, https://bitbucket.org/mntd/mstou-processes/src/master/, Bitbucket repository (2017).

[11] E. Hedevang, Ambit-stochastics, https://github.com/emilhedevang/ambit-stochastics, GitHub repository (2013).

[12] D. Leonte, Ambit stochastics, https://github.com/danleonte/Ambit_Stochastics, GitHub repository (2022). doi:10.5281/zenodo.7081979.

[13] K. Y. Jónsdóttir, J. Schmiegel, E. B. Vedel Jensen, Lévy-based growth models, Bernoulli 14 (1) (2008) 62–90.
URL https://doi.org/10.3150/07-BEJ6130

[14] R. C. Noven, Statistical models for spatio-temporal extrema and dependencies, Ph.D. thesis, Imperial College London (2016).

[15] O. E. Barndorff-Nielsen, F. E. Benth, A. E. Veraart, Ambit Stochastics, Springer-Verlag, Berlin, 2018.
URL https://doi.org/10.1007/978-3-319-94129-5

[16] W. Huang, K. Wang, F. Jay Breidt, R. A. Davis, A class of stochastic volatility models for environmental applications, Journal of Time Series Analysis 32 (4) (2011) 364–377.
URL https://doi.org/10.1111/j.1467-9892.2011.00735.x

[17] F. E. Benth, The stochastic volatility model of Barndorff-Nielsen and Shephard in commodity markets, Math. Finance 21 (4) (2011) 595–625.
URL https://doi.org/10.1111/j.1467-9965.2010.00445.x

[18] A. E. D. Veraart, L. A. M. Veraart, Modelling electricity day-ahead prices by multivariate Lévy semistationary processes, in: Quantitative energy finance, Springer, New York, 2014, pp. 157–188.
URL https://doi.org/10.1007/978-1-4614-7248-3_6

[19] B. S. Rajput, J. Rosiński, Spectral representations of infinitely divisible processes, Probab. Theory Related Fields 82 (3) (1989) 451–487.
URL https://doi.org/10.1007/BF00339998

[20] J. B. Walsh, An introduction to stochastic partial differential equations, in: École d'été de probabilités de Saint-Flour, XIV—1984, Vol. 1180 of Lecture Notes in Math., Springer, Berlin, 1986, pp. 265–439.
URL https://doi.org/10.1007/BFb0074920

[21] K. Bichteler, J. Jacod, Random measures and stochastic integration, in: G. Kallianpur (Ed.), Theory and Application of Random Fields, Vol. 49 of Lecture Notes in Control and Information Sciences, Springer-Verlag, Berlin, 1983, pp. 1–18.
URL https://doi.org/10.1007/BFb0044678

[22] C. Chong, C. Klüppelberg, Integrability conditions for space–time stochastic integrals: Theory and applications, Bernoulli 21 (2014) 2190–2216.
URL https://doi.org/10.3150/14-BEJ640

[23] O. E. Barndorff-Nielsen, J. Schmiegel, Intermittent stochastic fields and space-time symmetry, Thiele Research Reports, Department of Mathematics, Aarhus University (2015).

[24] A. W. v. d. Vaart, Asymptotic Statistics, Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press, 1998. doi:10.1017/CBO9780511802256.

[25] K. Sato, Lévy processes and infinitely divisible distributions, Vol. 68 of Cambridge Studies in Advanced Mathematics, Cambridge University Press, Cambridge, 1999, translated from the 1990 Japanese original, Revised by the author.

[26] A. E. Veraart, Modeling, simulation and inference for multivariate time series of counts using trawl processes, Journal of Multivariate Analysis 169 (2019) 110–129.
URL https://doi.org/10.1016/j.jmva.2018.08.012

[27] D. Görür, Y. Teh, Concave-convex adaptive rejection sampling, Journal of Computational and Graphical Statistics 20 (2012) 670–691.
URL https://doi.org/10.1198/jcgs.2011.09058

[28] M. Bennedsen, A. Lunde, N. Shephard, A. E. Veraart, Inference and forecasting for continuous-time integer-valued trawl processes and their use in financial economics, Journal of Econometrics, Forthcoming (2023).
URL https://arxiv.org/abs/2107.03674

[29] M. S. Ridout, Generating random numbers from a distribution specified by its Laplace transform, Stat. Comput. 19 (4) (2009) 439–450.
URL https://doi.org/10.1007/s11222-008-9103-x

[30] W. Hürlimann, Improved FFT approximations of probability functions based on modified quadrature rules, Int. Math. Forum 8 (17-20) (2013) 829–840.
URL https://doi.org/10.12988/imf.2013.13087

[31] V. Witkovskỳ, Numerical inversion of a characteristic function: An alternative tool to form the probability distribution of output quantity in linear measurement models, Acta IMEKO 5 (3) (2016) 32–44.
URL http://dx.doi.org/10.21014/acta_imeko.v5i3.382

[32] M. S. Veillette, M. S. Taqqu, A technique for computing the PDFs and CDFs of nonnegative infinitely divisible random variables, J. Appl. Probab. 48 (1) (2011) 217–237.
URL https://doi.org/10.1239/jap/1300198146

[33] L. J. Slater, B. Anderson, M. Buechel, S. Dadson, S. Han, S. Harrigan, T. Kelder, K. Kowal, T. Lees, T. Matthews, et al., Nonstationary weather and water extremes: a review of methods for their detection, attribution, and management, Hydrology and Earth System Sciences Discussions 2020 (2020) 1–54.

[34] F. E. Benth, H. Eyjolfsson, A. E. Veraart, Approximating Lévy semistationary processes via Fourier methods in the context of power markets, SIAM J. Financial Math. 5 (1) (2014) 71–98.
URL https://doi.org/10.1137/130905320

[35] O. E. Barndorff-Nielsen, F. E. Benth, A. E. D. Veraart, Modelling electricity futures by ambit fields, Advances in Applied Probability 46 (3) (2014) 719–745.
URL https://doi.org/10.1239/aap/1409319557

[36] J. Pedersen, The Lévy-Ito decomposition of an independently scattered random measure, MaPhySto, Department of Mathematical Sciences, University of Aarhus, 2003.