A fast Monte Carlo scheme for additive processes and option pricing

Michele Azzone[‡§] & Roberto Baviera[‡¶]

December 16, 2021

(‡) Politecnico di Milano, Department of Mathematics, 32 p.zza L. da Vinci, Milano
(§) European Central Bank[1], 20 Sonnemannstraße, Frankfurt am Main
(¶) Corresponding author

Abstract

In this paper, we present a fast Monte Carlo scheme for additive processes. We analyze in detail numerical error sources and propose a technique that reduces the two major sources of error. We also compare our results with a benchmark method: the jump simulation with Gaussian approximation.
We show an application to additive normal tempered stable processes, a class of additive processes that calibrates "exactly" the implied volatility surface. Numerical results are relevant. The algorithm is an accurate tool for pricing path-dependent discretely-monitoring options with errors of one bp or below. The scheme is also fast: the computational time is of the same order of magnitude of standard algorithms for Brownian motions.

Keywords: Finance, simulation, Additive process, fast Fourier transform, Lewis formula.

JEL Classification: C51, C63, G12, G13.

Prof. Roberto Baviera
Department of Mathematics
Politecnico di Milano
32 p.zza Leonardo da Vinci
I-20133 Milano, Italy
Tel. +39-02-2399 4575
roberto.baviera@polimi.it

Michele Azzone
Department of Mathematics
Politecnico di Milano
32 p.zza Leonardo da Vinci
I-20133 Milano, Italy
Tel. +39-338-2464 527
michele.azzone@mail.polimi.it
michele.azzone@ecb.europa.eu

[1]The views expressed are those of the author and do not necessarily reflect the views of ECB.

A fast Monte Carlo scheme for additive processes and option pricing

1 Introduction

In this paper, we introduce a fast Monte Carlo simulation technique for additive processes. In option pricing, Monte Carlo methods are attractive because they do not require significant modifications when the payoff structure of the derivative changes. Additive processes are becoming the new frontier in equity derivatives for their ability, on the one hand, to reproduce accurately market data in model calibration, and on the other hand, to keep the process rather elementary (see e.g., Li *et al.* 2016, Azzone and Baviera 2021, Carr and Torricelli 2021). A process $\{X(t)\}_{t\geq 0}$ is said to be an additive process, if it presents independent (but not-stationary) increments and satisfies $X(0) = 0$ a.s.; stationarity is the main difference with Lévy processes.
For most additive processes, the law of increments is not known explicitly, but analytic expressions exist for the characteristic functions thanks to the celebrated Lévy–Khintchine formula (Sato 1999). Given any such characteristic function for increments, this paper aims to describe an efficient and accurate algorithm for Monte Carlo simulations of the increments and to compute the prices of a class of discretely-monitoring path-dependent options.

Up to our knowledge, the unique Monte Carlo (MC) scheme developed for a specific class of additive processes, Sato processes, has been introduced by Eberlein and Madan (2009). They generalize to this class of additive processes, a well-known jump simulation technique developed for Lévy processes, that can be found in many excellent textbooks (see e.g., Cont and Tankov 2003, Asmussen and Glynn 2007). It entails truncating small jumps below a certain threshold and then simulating the finite number of independent jumps; finally, the Asmussen and Rosiński (2001) Gaussian approximation (hereinafter GA) can be applied to substitute small jumps with a diffusive term: this has become a benchmark technique to compare numerical results.

In this paper, we propose a MC technique for additive processes based on a numerical inversion of the cumulative distribution function (CDF). Monte Carlo simulation of additive processes is not straightforward because, in general, the CDF of process increments is not known explicitly. Since the seminal paper of Bohman (1970), general methods have been developed for sampling from Fourier transforms and even some specific methods for some distributions (e.g. stable distributions) that do not require numerical inversion (Samorodnitsky and Taqqu 1994, Sec.1.7, p.41). In the financial literature, these techniques have been considered when the transition probability density of the underlying asset dynamics is not known explicitly; they have been developed specifically in the Lévy case, where it is possible to leverage on the stationary increments (see e.g., Glasserman and Liu 2010, Chen *et al.* 2012, Ballotta and Kyriakou 2014). These techniques are reliable and efficient: they build upon the characteristic function numerical inversion to obtain an estimation of the CDF. Specifically, we use the fast Fourier transform (FFT) method for the numerical inversion as proposed by Lee (2004) and then applied to MC option pricing in the studies of Chen *et al.* (2012) and Ballotta and Kyriakou (2014). Relative to this literature, our contribution lies in analyzing the three sources of error that arise in estimating derivative price expectations and showing how to improve the two largest ones.

Three are the main contributions of this paper. First, we propose a Monte Carlo simulation technique for additive processes based on FFT. Second, we improve the two main sources of numerical error in existing techniques to accelerate convergence, using both an analytic property of Fourier inversion in the complex plane and a spline method for CDF numerical inversion. Finally,

we point out that the proposed technique is accurate and fast: *i*) we compare with traditional GA simulations showing that it is at least one and a half orders of magnitude faster whatever time horizon we consider and *ii*) we observe that, when pricing some discretely-monitoring path-dependent options, the computing time has the same order of magnitude as standard algorithms for Brownian motions.

The rest of the paper is organized as follows. In section 2, we overview the method and recall both Lewis (2001) formula for CDF and the error source in the numerical approximation: we discuss the optimal selection of the integration path. In section 3, we describe the proposed simulation method and present the other main error source in MC option pricing: the interpolation method in numerical inversion. We also discuss how to generalize the GA method for additives in an efficient way. Section 4 presents numerical results for a large class of pure-jump additive processes in the case of both European options (where analytic pricing methods are available), and some discretely-monitoring path-dependent options. Section 5 concludes. We report the proofs in appendix A and a brief description of the algorithm in appendix B.

2 Overview of the MC method for additive process

Pure jump asset pricing models based on additive processes have enjoyed remarkable popularity in recent years. At least for two main reasons. First, they allow a highly tractable closed-form approach with simple analytic expression for European options following Lewis (2001). This formula is computable as fast as the standard Black-Scholes one. Second, additive processes provide an adequate calibration to the implied volatility surface of equity derivatives, as well as they reproduce *stylized facts* as the time scaling of skew in volatility smile (see e.g., Azzone and Baviera 2021).

In this section, we describe a third reason in favor of these models: they allow a simple, accurate, and fast numerical scheme for path-dependent option valuation. We extend to additive processes the preceding literature on Lévy processes' simulation techniques and we discover that, thanks to this Monte Carlo scheme, it is not a challenge to price exotic derivatives as Asian contracts or barrier options with discretely-monitored barriers.

The simulation of a discrete sample path of an additive process reduces to simulating from the distribution of the process increment between time s and time $t > s$. Lévy process simulation is based on time-homogeneity of the jump process: the characteristic function of an increment is the same as the characteristic function of the process itself at time $t = 1$, re-scaled by the time interval $(t - s)$ of interest.

In this paper, we extend the preceding analysis to Lévy processes by *i*) presenting an explicit method for additive processes from their characteristic function and *ii*) analyzing the explicit bound for the total estimation bias. In the Lévy case, thanks to process time homogeneity, the properties of the process characteristic function are immediately extended to its increments. For example, the characteristic function (also of increments) is analytic in a horizontal strip and the purely imaginary points on the boundary of the strip of regularity are singular points (cf. Lukacs 1972, th.3.1, p.12). This identification of process characteristic function and increments' characteristic function is not anymore valid for additive processes. However, the present paper shows that the analyticity strip depends on time and that it is possible to build, in a simple way, a numerical scheme for additive processes requiring an additional condition.

Our method is based on three key observations. First, computing a CDF $P(x)$ corresponds to pricing a digital option: this can be done efficiently in the Fourier space. This step can be crucial, as already highlighted by Ballotta and Kyriakou (2014), the Fourier formula presents some numerical instabilities due to the presence of a pole in the origin. They propose a regularization that leads to an additional numerical error. We propose a different approach that is based on the Lewis (2001)

formula which presents two significant advantages. On the one hand, this technique is exact (thus, no numerical error is associated with it), and, on the other hand, it allows selecting the optimal integration path that reduces the numerical error in the discretization of the CDF.
Second, the Lewis (2001) formula for the CDF can be viewed as an inverse Fourier transform method that can be approximated with a fast Fourier transform (FFT) technique: Lewis-FFT computes multiple values of the CDF simultaneously in a very efficient way.
Finally, knowing CDF approximation $\hat{P}$, we can sample from this distribution by inverting the CDF, i.e. by setting $X = \hat{P}^{-1}(U)$, with U an uniform r.v. in $[0, 1]$. Thus, simulating a r.v. via a numerical CDF (i.e. coupling the Fourier transform with a Monte Carlo simulation), requires a numerical inversion that is realized via an interpolation method. Following Glasserman and Liu (2010), due to its simplicity, a linear interpolation of the CDF is chosen in the existing financial literature (see e.g., Chen *et al.* 2012, Feng and Lin 2013). We propose the spline as interpolation rule because the computational cost is very similar, while the bias associated with the two interpolation rules is significantly different: the upper bound of the bias can be estimated for a given grid spacing γ, and, as we discuss in section 3, it should be at least γ^2 smaller for the spline interpolation. In extensive numerical experiments we observe that, on the one hand, the error decreases even faster as a power of γ than predicted by the upper bound, thanks to the additional properties of the interpolated functions, and on the other hand, it becomes negligible for the grids that are selected in practice.
Due to these three main ingredients (Lewis formula, FFT and Spline interpolation) that play a crucial role in the proposed Monte Carlo simulation technique, we call the method Lewis-FFT-S. The algorithm is reported in appendix B.

The Lewis-FFT-S method extends the Eberlein and Madan (2009) technique to any additive process of financial interest being significantly faster: we show that the proposed Monte Carlo is much faster than any jump-simulation method even considering the Asmussen and Rosiński (2001) Gaussian approximation. Analyzing in detail the numerical errors related to the methodology, we design an algorithm that increases both accuracy and computational efficiency. To the best of our knowledge, the proposed scheme is the first application in financial engineering of the MC simulation based on Lewis formula and FFT, when the underlying is governed by an additive process.

In the next subsection, we also recall explicit and computable expressions for the error estimates.

2.1 Lewis CDF via FFT

The proposed MC method simulates from the characteristic function of the additive increments. Due to the Lévy-Khintchine formula, the characteristic function

$$\phi_t(u) := \mathbb{E}\, e^{i\,u\,f_t}$$

of an additive process f_t admits a closed-form expression. Furthermore, as already mentioned in the Introduction, according to Lukacs (1972, th.3.1, p.12), the process characteristic function is analytic in a horizontal strip delimited on the imaginary axis by two values. Similarly to Lee (2004), we define these values $p_t^- > 0$ and $-(p_t^+ + 1) < -1$, s.t. the characteristic function is analytic when $\Im(u) \in (-(p_t^+ + 1), p_t^-)$.
We observe that for Levy processes, the increment $f_t - f_s$ has the same distribution as f_Δ, where $\Delta = t - s$: the same property does not hold for additive processes, due to the time inhomogeneity. For an additive process, the characteristic function of an increment $f_t - f_s$ between times s and $t > s$ is

$$\phi_{s,t}(u) = \mathbb{E}\, e^{i\,u\,(f_t - f_s)} = \frac{\mathbb{E}\, e^{i\,u\,f_t}}{\mathbb{E}\, e^{i\,u\,f_s}},$$

due to the independent increment property of additive processes, then

$$\ln \phi_{s,t} = \ln \phi_t - \ln \phi_s \, .$$

The MC method can be generalized to additive processes if the following assumption holds.

Assumption 1. p_t^+ and p_t^- are non increasing in t ♣

Thanks to Lukacs (1972) theorem and under Assumption 1, we are able to easily identify the strip of regularity in the case of interest: the characteristic function of an increment $f_t - f_s$ is analytic when $\Im(u) \in (-(p_t^+ + 1), p_t^-)$. Lewis (2001) obtains the CDF, shifting the integration path within the characteristic function horizontal analyticity strip. The shift is $-i\ a$ with a a real constant s.t. $a \in (-p_t^-, 1 + p_t^+)$. Lewis deduces this formula using the properties of contour integrals in the complex plane.
If Assumption 1 holds, the CDF $P(x)$ of an additive process increment is (cf. Lee 2001, th.5.1)

$$P(x) = R_a - \frac{e^{-ax}}{\pi} \int_0^\infty du\, Re\left[\frac{e^{-iux}\phi_{s,t}(u - ia)}{i\,u + a}\right] \quad , \tag{1}$$

where

$$R_a = \begin{cases} 1 & 0 < a < p_t^+ + 1 \\ \frac{1}{2} & a = 0 \\ 0 & p_t^- < a < 0 \end{cases} \quad .$$

The case with no shift ($a = 0$) is the Hilbert transform: it has been considered in several studies in the financial literature on MC pricing (see e.g., Chen *et al.* 2012, Ballotta and Kyriakou 2014). In the Hilbert transform case, the singularity in zero in the integration should be taken into account as a Cauchy principal value; as already emphasized by Ballotta and Kyriakou (2014), the method could be not robust enough for applications in the financial industry: they have suggested a regularization technique that introduces an additional error source, while the Lewis method we consider here is exact.

In the following, we focus on $a > 0$: this is a default choice in the equity case (see e.g., Lee 2001, Section 7.4, p.26) because $p_t^+ \geq p_t^-$ is consistent with the negative equity skew. We derive an approximation formula and its error bounds (in sections 2.2 and 3.1). Similar results hold for $a < 0$.
We approximate the Fourier transform with a discrete Fourier transform $\hat{P}(x)$

$$\hat{P}(x) := 1 - \frac{e^{-ax}}{\pi} \sum_{l=0}^{N-1} Re\left[\frac{e^{-i(l+1/2)hx}\phi_{s,t}((l+1/2)h - i\,a)}{i\,(l+1/2)h + a}\right] \quad ,$$

where h is the step size in the Fourier domain and N is the number of points in the grid.

To implement the MC method, we need the CDF function for a large number of values in a regular grid with step size γ. An algorithm that is computationally efficient is the fast Fourier transform (see Lee 2004, for a detailed analysis of the method in derivative pricing): it involves Toeplitz matrix-vector multiplication (see e.g., Press *et al.* 1992, ch.12) and relies on an additional requirement for N, whose simplest choice is $N = 2^M$ with $M \in \mathbb{N}$; hereinafter, we consider an N within this set. The main advantage of the method is that the computational complexity of the FFT is $O(N \log_2 N)$ when computing one time-increment. Moreover, with an FFT, it holds the relationship

$$\gamma\, h = \frac{2\pi}{N} \quad ;$$

i.e., for a given number N of grid points, the step size in the Fourier domain h fixes the step size γ.[2]

2.2 CDF error sources

The numerical Fourier inversion is subject *i*) to a discretization error, because the integrand is evaluated only at the grid points, and *ii*) to a range error, because we approximate with a finite sum.

Assumption 2. $\forall\, t > s \geq 0$ there exists $B > 0$, $b > 0$ and $\omega > 0$ such that, for sufficiently large u, the following bound for the absolute value of the characteristic function holds

$$|\phi_{s,t}(u - i\,a)| < Be^{-b\,u^{\omega}}\ , \qquad \forall a \in (0, p_t^+ + 1) \qquad \clubsuit$$

Leveraging on Assumption 2, we can estimate the explicit bound for the bias in terms of the step size h and the number of grid points N, as shown in the next proposition. The result in the next proposition improves the known bounds for numerical errors when computing the CDF (1), via a discrete Fourier transform, and indicates an optimal integration path that minimizes this error bound.

Proposition 2.1. *If Assumptions 1 and 2 hold, then*

1. *the numerical error* $|P(x) - \hat{P}(x)|$ *for the CDF is bounded by*

$$\mathcal{E}_{h,M}^{CDF}(x) = \frac{e^{-x\,(p_t^+ + 1)/2}}{\omega\; b^{1/\omega}} \frac{1}{Nh} \Gamma\left[\frac{1}{\omega}, b\,(N\,h)^{\omega}\right] + \frac{e^{-\pi(p_t^+ + 1)/h} + e^{-\pi(p_t^+ + 1)/h - (p_t^+ + 1)\,x} \phi_{s,t}(-i\,(p_t^+ + 1))}{1 - e^{-2\pi(p_t^+ + 1)/h}}\ , \tag{2}$$

   where $\Gamma(z, u)$ *is the upper incomplete gamma function and*

$$\frac{1}{N\,h} \Gamma\left[1/\omega, b\,(N\,h)^{\omega}\right] = O\left((N\,h)^{-\omega} e^{-b\,(N\,h)^{\omega}}\right)\ ;$$

2. *the (optimal) bound holds selecting the shift* a *in (1) equal to* $(p_t^+ + 1)/2$.

Proof. See Appendix A □

The first term of $\mathcal{E}_{h,M}^{CDF}(x)$ accounts for the range error in the numerical inversion, while the second one accounts for the discretization error.[3] It is possible to prove, following the same steps of **proposition 2.1**, that in the $a < 0$ case the leading term in $\mathcal{E}_{M}^{CDF}(x)$ is $\exp(-\pi p_t^- / h)$. From this result, we can observe that it is convenient to use $a > 0$ if $p_t^+ + 1 \geq p_t^-$ and $a < 0$ otherwise: as we observe in section 4, $p_t^+ + 1 \geq p_t^-$ is the situation often observed in the relevant equity case. In the financial literature, error estimations have been proposed when approximating a CDF via a discrete Fourier Transform (see e.g., Lee 2004, Chen *et al.* 2012, Ballotta and Kyriakou 2014). The bound in **proposition 2.1** extends these results to the Lewis-FFT case, showing how to

[2] To avoid this constraint, one can consider the fractional fast Fourier transform (Chourdakis 2005) instead of the standard FFT. We have verified that the additional computational cost of the former method is not justified in the CDF simulation described in this paper.

[3] It is possible also to obtain an error bound even when Assumption 2 does not hold. Equation (2) can be extended to the case where the characteristic function has an asymptotical polynomial decay $|\phi_{s,t}(u - i\,a)| \leq B\,|u|^{-b}$, with $b > 0$: in this case, the range error decays only as a power of u due to the polynomial decay of the characteristic function (see e.g., Ballotta and Kyriakou 2014, eq.(14), p.1099). However, in practice, when pricing exotic derivatives, the exponential decay of the characteristic function is a good reason for model selection.

select the optimal integration path in Lewis formula (1) to minimize the exponential decay of the error. Our approach eliminates the source of error originating from the pole in the origin (see e.g., Ballotta and Kyriakou 2014, eq.(4), p.1097), improving the CDF error. Moreover, selecting the optimal path, CDF error is even better than the one proposed by Chen *et al.* (2012, th.2.1, p.14:6) deduced via the sinc expansion technique. The leading term in the discretization error in theorem 2.1 of Chen *et al.* (2012) goes as $\max(e^{-\pi\, p_t^-/h}, e^{-\pi\,(p_t^+ +1)/h})$, while, in our case, the error goes as the minimum of the two terms.
Hence, if $(p_t^+ + 1 \geq p_t^- > 0)$, the discretization error (2) goes as $e^{-\pi\,(p_t^+ +1)/h}$ and it improves the one proposed by Chen *et al.* (2012). If $(0 < p_t^+ + 1 < p_t^-)$, we can choose $a = -p_t^-/2$. In this way, the discretization error goes as $e^{-\pi\, p_t^-/h}$ that also improves the one proposed by Chen *et al.* (2012).

We desire to get a small approximation error increasing N and decreasing h. However, let us observe that, if one takes the limit $h \to 0$ and $N \to \infty$ keeping Nh fixed, then the range error bound does not decrease. Thus, our interest is to select $h = h(N)$ so that the discretization and the range errors have about the same order. Expression (2) allows us to determine the size h and the number N such that the two sources of CDF error are comparable: we can impose that $\exp(-\pi(p_t^+ + 1)/h) = \exp(-b\,(N\,h)^\omega)$, i.e. we select

$$h(N) = \left(\frac{\pi\,(p_t^+ + 1)}{b}\frac{1}{N^\omega}\right)^{1/(\omega+1)} .$$

We define

$$\mathcal{E}_M^{CDF}(x) := \mathcal{E}_{h(2^M),M}^{CDF}(x) \tag{3}$$

the error in this case. $\mathcal{E}_M^{CDF}(x)$ in (3) is the relevant estimation of the CDF error that we use in practice: with this selection of h, the total CDF error is $O(N^{-\omega/(1+\omega)})\exp(-bN^{\omega/(1+\omega)})$ and decays almost exponentially as we increase N; moreover, the step size $\gamma = 2\pi/(hN) = O(N^{-1/(1+\omega)})$.

3 The simulation method

Knowing the CDF approximation $\hat{P}$, we can sample from this distribution by inverting $\hat{P}$, i.e. by setting $X = \hat{P}^{-1}(U)$, with U an uniform r.v. in $[0,1]$.
From the Fourier inversion, we obtain an estimate of $\hat{P}$ on a grid of N points with step γ. As pointed out by Glasserman and Liu (2010, sec.3, pp.1614-1615), an adequate inversion requires to impose that $\hat{P}$ is i) increasing and ii) inside the interval [0,1]. Thus, it is convenient to work with a subset of the grid of N points. We truncate the CDF between $x_0 < 0$ and $x_K > 0$, such that the two conditions hold, and we consider the equally spaced grid (with step γ) $x_0 < x_1 < ... < x_K$ with $K \leq N$.
Simulating a r.v. via a numerical CDF (i.e. coupling the Fourier transform with a MC simulation), requires a numerical inversion that is realized with an interpolation method. As already discussed in section 2, differently from the existing financial literature (see e.g., Glasserman and Liu 2010, Chen *et al.* 2012, Feng and Lin 2013), the proposed method is based on spline interpolation. In the next subsection, we discuss the key idea behind this choice of the interpolation method.

3.1 Simulation error sources: truncation and interpolation

Besides numerical inversion error of the CDF, two are the error sources in the MC, when pricing a contingent claim: truncation and interpolation of the CDF.
Let us consider the expected value $\mathbb{E}V(f_t - f_s)$, with $V(x)$ a derivative contract with a pay-off

differentiable everywhere except in n_V points. It can be proven, similarly to Chen *et al.* (2012, th.4.3, p.14:11), that the pricing error[4] using the Lewis-FFT method with linear interpolation is

$$\mathcal{E} := \int_{-\infty}^{\infty} dx\, V(x)\ [p(x) - \hat{p}(x)] \tag{4}$$

$$< \left(|V(x_0)| + |V(x_K)| + (2K + n_V) \sup_{x\in(x_0,x_K)} |V(x)| + 2 \sup_{x\in(x_0,x_K)} |V'(x)| \right) \mathcal{E}_M^{CDF}(x_0) \tag{5}$$

$$+ \frac{\phi_{s,t}^-}{2\pi} \left(\frac{|V(x_K)| e^{x_K p_t^-}}{|p_t^-|} + \int_{x_K}^{\infty} dx\, V(x) e^{x\, p_t^-} \right) + \frac{\phi_{s,t}^+}{2\pi} \left(\frac{V(x_0) e^{x_0 (p_t^+ + 1)}}{p_t^+ + 1} + \int_{-\infty}^{x_0} dx\, V(x) e^{x\,(p_t^+ + 1)} \right) \tag{6}$$

$$+ \frac{\gamma^2}{2\pi} (x_K - x_0) \sup_{x\in(x_0,x_K)} |V'(x)| \int_{\mathbb{R}} |du\, u\, \phi_{s,t}(u)| \ , \tag{7}$$

where $p(x)$ is the probability density function of $f_t - f_s$, $\hat{p}$ its estimation and

$$\phi_{s,t}^- := \lim_{a\to p_-} \int_{\mathbb{R}} du\, |\phi_{s,t}(u - ia)| \quad \& \quad \phi_{s,t}^+ := \lim_{a\to p_+} \int_{\mathbb{R}} du\, |\phi_{s,t}(u - ia)| \ .$$

Three are the components of the bias error (4) when pricing a derivative: an error related to the numerical approximation of the CDF (5), a truncation error (6) and an interpolation error (7). Let us consider each error source separately.

First, the error related to the numerical approximation of the CDF in (5) is proportional to $\mathcal{E}_M^{CDF}(x_0)$: we have discussed in the previous section how to select the integration path and h in order to minimize it.

Second, we can always choose x_0 and x_K s.t. the truncation error is negligible.
We select a symmetric interval ($x_K = -x_0$), a standard choice in the literature, and x_0 the nearest point to $-D\sqrt{t-s}$ on the grid in which the CDF $\hat{P}$ is computed with $D > 0$.
Why do we select an x_0 that depends on the time-interval $t - s$? It can be easily explained with a graph. In figure 1, as an example, we plot the one-day and one-year normalized probability density functions of the additive process used in the numerical experiments of section 4. As expected, the one-day density is significantly more concentrated around zero than the one-year density when considering a constant x (on the right). Conversely, the ranges of the two densities look similar when considering the rescaled moneyness $x/\sqrt{t-s}$. Thus, the range of probability densities scales approximately with $\sqrt{t-s}$.
Moreover, we choose $D = 5$. In extensive numerical experiments, we observe that for values below x_0 (or above x_K) the probability density function is lower than 10^{-9} whatever time horizon is considered and its contribution to the price is negligible for all practical purposes.

[4]The upper bound on the bias $\mathcal{E}$ can be trivially extended to a payoff with a finite number n of monitoring times. The most relevant case, for $n = 1$, will be discussed in detail in subsection 4.1.

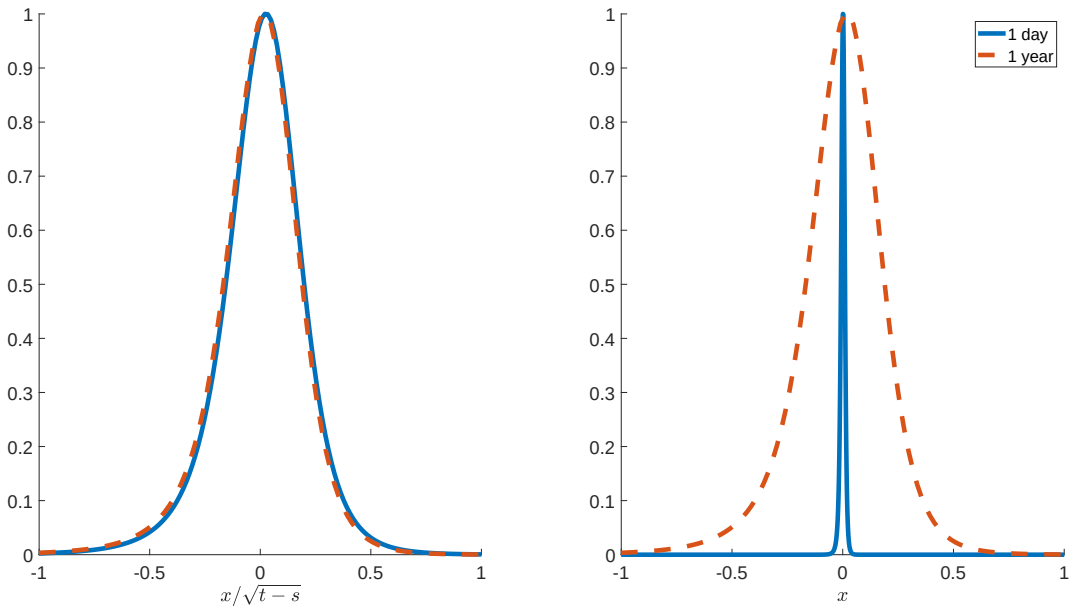

Figure 1: One-day and one-year normalized probability density functions of the additive process that we use in the numerical experiments of section 4 with $s = 0$. On the right, we see that, as expected, the one-day density is significantly more concentrated around zero than the one-year density. Conversely, on the left, we see that the ranges of the two densities wrt to the rescaled $x/\sqrt{t-s}$ are similar. Note that both probability density functions have been divided by their respective maximum for visualization purposes.

Finally, the bias associated with the linear interpolation, when computing the option value, is quadratic in the grid spacing γ; this turns out to be the most significant source of error in most cases, as shown in the next section. It is well known that linear interpolation error goes as γ^2 (see e.g., Quarteroni *et al.* 2007, eq.(8.26), p.339). For this reason, in this paper, we propose a spline interpolation method. In this latter case, it is known that the bias goes, at least, as γ^4 as shown in Hall and Meyer (1976).

As already emphasized by Glasserman and Liu (2010, Sec.3, p.1615), to sample X from $\hat{P}(x)$ with a linear interpolation, after having generated U, a r.v. uniformly distributed in $[0, 1]$, one should

1. select the index j for which $\hat{P}(x_{j-1}) \leq U < \hat{P}(x_j)$;
2. for each j determine the linear interpolation coefficients $c^L_{0,j}$ and $c^L_{1,j}$

$$c^L_{0,j} := \frac{x_j\,\hat{P}(x_j) - x_{j-1}\,\hat{P}(x_{j-1})}{\hat{P}(x_j) - \hat{P}(x_{j-1})} \quad \text{and} \quad c^L_{1,j} := \frac{\gamma}{\hat{P}(x_j) - \hat{P}(x_{j-1})} \;\; ;$$

3. compute

$$X = c^L_{0,j} + c^L_{1,j}\,U \;\; .$$

We discuss the computational cost of each step when sampling $\mathcal{N}_{sim}$ observations. The first step relies on a nearest neighborhood algorithm with an average computational cost proportional to $\mathcal{N}_{sim} \times \log_2 \mathcal{N}_{sim}$ (see e.g., Cormen *et al.* 2001, p.11)[5]. The second step cost is proportional to $6K$. Finally the last step is proportional to $\mathcal{N}_{sim}$.

[5]The computational cost estimation is for the *merge sort* algorithm. Since *merge sort* is a recursive algorithm it could be necessary, for memory efficiency, to recur to *an insertion sort* algorithm which computational cost is roughly proportional to $\mathcal{N}^2_{sim}$ (see e.g., Cormen *et al.* 2001, p.11).

Algorithm	Nearest neighborhood	Linear interpolation	Spline interpolation
time [ms]	1.08	1.13	1.27

Table 1: Computational cost in milliseconds [ms] for the nearest neighborhood (nn), the linear interpolation including nn, and the spline interpolation including nn. We consider a grid size $K = 10^4$ and $\mathcal{N}_{sim} = 10^5$ simulations. Even considering a low number of simulations and a grid size K one order of magnitude above what is used in practice (in the Lewis-FFT-S case K is of order 10^3) the spline simulation cost is just 10% more than the linear simulation one.

Whereas step 1 is shared by both interpolation methods, steps 2 and 3 differ between spline and linear interpolations. In step 2, the additional computational cost of considering spline interpolation boils down to the cost of solving a $K+1$-dimensional linear system with a tridiagonal matrix to determine the spline coefficients $\{c^S_{q,j}\}^3_{q=0}$, cf. Quarteroni *et al.* (2007, ch.8), i.e. the cost is $8K - 7$ (Quarteroni *et al.* 2007, ch.7, p.391). As for step 3, the cost of computing the spline interpolation of U is still proportional to $\mathcal{N}_{sim}$. It is clear that for a sufficiently large number of simulations $\mathcal{N}_{sim}$ and for $\mathcal{N}_{sim} >> K$, for both methods, the most relevant contribution in the computational cost is the one due to step 1, the nearest neighborhood algorithm.

We perform numerical experiments to compare linear and spline interpolation. We observe that, if the number of simulations is significantly above the grid dimension K, the spline interpolation is as expensive as the linear interpolation. Moreover, in table 1, we compare the computational cost of linear interpolation and spline interpolation. We consider a grid of size $K = 10^4$ and $\mathcal{N}_{sim} = 10^5$. In this case, the spline cost is just 10% more than the linear one. The case considered in table 1 is a particularly unfavorable situation, when comparing spline interpolation with linear interpolation: a large grid size $K = 10^4$ and a small number of simulations $\mathcal{N}_{sim} = 10^5$. In this case steps 1, 2 and 3 computational times are comparable while, in practice, most of the computational costs are absorbed by the nearest neighborhood algorithm. For reasonable values of M (e.g. for $M \leq 15$), the dimension of the grid K is always well below 10^4. Thus, for all values of K and $\mathcal{N}_{sim}$ ($\mathcal{N}_{sim} \geq 10^6$) used in practice the incremental cost between Lewis-FFT (with linear interpolation) and Lewis-FFT-S (with spline interpolation) is negligible.

3.2 A simulation benchmark: the Gaussian approximation

In this subsection, we show how to generalize the GA method for additives in an efficient way, when a monotonicity property holds for the Lévy measure and then the Ziggurat method (Marsaglia and Tsang 2000) can be applied.
A generic additive process may have an infinite number of jumps, most of them being small, over an arbitrary finite time horizon, making the simulation of such a process often nontrivial. Defining ν_t the additive process jump measure (see e.g., Sato 1999, def.8.2, p.38), the jump measure of the additive process increment $f_t - f_s$ is $\nu_t - \nu_s$.

Eberlein and Madan (2009), in their study on simulation of additive processes, consider only a class of additive processes (Sato processes): their approach consists in discarding the small jumps that in absolute value are below a given threshold ϵ. It is well known, in the Lévy case, that such an approach is accurate only if there are not too many small jumps (see e.g., Cont and Tankov 2003). Alternatively, the small jump component of an additive process may be approximated by a Brownian motion (Asmussen and Rosiński 2001).

Once the jump measure of the increment (between time s and time $t > s$) is truncated, we have i) to draw a Poisson number of positive and negative jumps and ii) to simulate separately positive

jumps from the probability density $m^+_{s,t}$ and negative jumps from the probability density $m^-_{s,t}$, where

$$m^+_{s,t}(x) := \mathbb{I}_{x>\epsilon}\frac{\nu_t(x)-\nu_s(x)}{\int_\epsilon^\infty dz(\nu_t(z)-\nu_s(z))} \quad \& \quad m^-_{s,t}(x) := \mathbb{I}_{x<-\epsilon}\frac{\nu_t(x)-\nu_s(x)}{\int_{-\infty}^{-\epsilon} dz(\nu_t(z)-\nu_s(z))} \ .$$

To sample positive and negative jumps is extremely costly because often it is not possible to compute explicitly the integrals of $m^+_{s,t}$ and $m^-_{s,t}$.

Assumption 3. $m^+_{s,t}(x)$ is non increasing in x and $m^-_{s,t}(x)$ is non decreasing in x $\forall s,t$ s.t. $0 \le s < t$ ♣

A faster methodology -for sampling from a known distribution without inverting numerically its integral- is based on the Ziggurat method of Marsaglia and Tsang (2000). This method is applicable to probability density functions that are bounded and monotonic. We can apply the algorithm separately to negative and positive jumps. Having truncated small jumps the density functions are bounded, we need to ask the conditions of monotonicity in Assumption 3. The Ziggurat method covers a probability density with N_{ret} rectangles with equal area and a base strip. The base strip contains the tail of the probability density, it is built s.t. it has the same area of the rectangles. The method is composed of two building blocks: first, the rectangles with equal area are identified; second, the random variable is simulated either from a rectangle or from the base strip. Only in the latter case, an inversion of the integral is needed. N_{ret} is a key parameter because it controls the trade-off, in terms of computational time, between this inversion and building the rectangles.

With respect to Eberlein and Madan (2009), to reduce the bias of the method, we also consider the Gaussian approximation of Asmussen and Rosiński (2001).

4 Numerical results

Financial applications provide an important motivation for this study. We show that the proposed Monte Carlo technique for additive processes can price path-dependent options fast and accurately. The computational time is comparable to the case with simple Brownian motion dynamics.
We are interested in simulating a discrete sample path of the process over a finite time horizon: we are only concerned about the values of an additive process on such a discrete-time grid. This arises from situations where only discrete values of the process are concerned as in Chen *et al.* (2012), Ballotta and Kyriakou (2014) (e.g., they consider discrete barrier, lookback, and Asian options).
The case of an additive normal tempered stable (ATS) is discussed in detail. ATS processes present several advantages: they calibrate accurately equity implied volatility surfaces and, in particular, they capture volatility skews (see e.g., Azzone and Baviera 2021).
The Lewis-FFT-S method and the GA benchmark can be used for the ATS because, in the next proposition, we prove that Assumptions 1, 2, and 3 hold for this class of additive processes.

Proposition 4.1. *For ATS processes with $\alpha \in (0,1)$, Assumptions 1, 2 and 3 hold. Furthermore, we have that $p^+_t \ge p^-_t$.*

Proof. See appendix A □

Assumptions 1 and 2 are quite comprehensive and hold also for Sato processes with exponentially decaying characteristic function.[6]

[6] As regard to Assumption 3, Eberlein and Madan (2009, p.30) point out that -for Sato processes- the Lévy measure is decreasing in x for positive x and increasing in x for negative x.

β	δ	$\bar{k}$	$\bar{\eta}$	$\bar{\sigma}$
1	-1/2	1	1	0.2

Table 2: ATS parameters used in all numerical simulations. These selected parameters are consistent with the ones observed in market data.

Proposition 4.2. *For Sato processes with characteristic function $\phi_1(u)$ in $t = 1$ that decays exponentially, Assumptions 1 and 2 hold.*

Proof. See appendix A □

A brief comment on Sato processes can be useful. Thanks to the self-similarity of the processes, if a condition on the characteristic function holds for $t = 1$ then it is satisfied also for all other time intervals. Moreover, a characteristic function that decays exponentially satisfies Assumption 2.[7] Let us emphasize that, in the ATS case, $p_t^+ \geq p_t^-$ and thus it is convenient to use $a > 0$. In particular, for the numerical example, we focus on the power-law scaling ATS (see e.g., Azzone and Baviera 2021, p.3) that is characterized by the parameters

$$k_t = \bar{k}\, t^{\beta}, \qquad \eta_t = \bar{\eta}\, t^{\delta}, \qquad \sigma_t = \bar{\sigma}\ ,$$

where $\bar{\sigma}, \bar{k}, \bar{\eta} \in \mathbb{R}^+$, and $\beta, \delta \in \mathbb{R}$. This model description has been shown to be particularly accurate for equity derivatives.
For all numerical experiments, we use the parameters reported in table 2: these parameters are consistent with the ones observed in market data.

To evaluate the Lewis-FFT-S performances, we consider plain vanilla and exotic derivatives at different moneyness x and at different times to maturity. Deeply out-of-the-money and in-the-money options are less informative on the method performances, as the option value is close to the intrinsic value. In the rest of the section, to ensure that we verify the performance of the method on options in a relevant range of moneyness x, we consider x in the range $\sqrt{t}(-0.2, 0.2)$, where t is the option time to maturity.

In subsection 4.1, we show how the Lewis-FFT-S (with spline interpolation) method significantly outperforms the method with linear interpolation for European options, where - thanks to the closed formula - we can easily verify the accuracy of the numerical method. In subsection 4.2, we provide evidence that Lewis-FFT-S is extremely fast and it is less computationally expensive, by at least 1.5 orders of magnitude than the GA method. In subsection 4.3, we price discretely-monitored Asian options, lookback options, and Down-and-In options with a time to maturity of five years. We also show that the Lewis-FFT-S is particularly efficient. The computational time needed to price path-dependent options with this method is just three times the computational time needed when using standard MC techniques for a geometric Brownian motion.

4.1 European options: accuracy

In the following, the Lewis-FFT-S performances are assessed for the ATS process. First, we show that, when using linear interpolation the leading term in (4) goes as γ^2. Then, we improve the bound by considering spline interpolation (Lewis-FFT-S) and we discuss the excellent performances of the method for the ATS case. Thanks to the FFT approach, Lewis-FFT-S is particularly fast: computational time has the same order of magnitude of standard algorithm that

[7] Eberlein and Madan (2009) consider also some characteristic functions with polynomial decay; in this case, the considerations in note 2 hold.

simulate Brownian motions. Thanks to the spline interpolation, Lewis-FFT-S is also particularly accurate, for 10^7 simulations and for any $M > 9$, the maximum observed error is 0.03 bp.

We do not desire a method that performs well either only OTM or only ITM. We want a MC that prices accurately options with any moneyness: for this reason, we consider the 30 European call options with moneyness in a regular grid with range $\sqrt{t}(-0.2, 0.2)$. The method error is assessed in terms of the maximum error in absolute value (MAX) and the average over the 30 MC standard deviation (SD).
Monte Carlo error is often decomposed into bias and variance (see e.g., Glasserman 2004, Section 1.1.3, pp.9-18). In this paper, we aim to reduce the bias error, but it is relevant to take into account also the variance. For a large number of simulations, confidence intervals estimated via MC are directly linked to this quantity (see e.g., Glasserman 2004, ch.1, eq.(1.10), p.10). In our case, since we are considering the average error over 30 call options, we consider the average standard deviation SD as a rough estimate of the variance error in the estimated prices. When the maximum error is below SD we can infer that the error on bias has been dealt with correctly. In all considered cases, SD is of the order of 0.1 bp.

In figure 2, we plot the three terms that appear in the bias bound of equation (4) for an ATS with $\alpha = 2/3$ over a one-month time interval. The bound is for Lewis-FFT simulation with linear interpolation varying the number of grid points in the FFT via M s.t. $N = 2^M$. We plot the bounds on the error due to i) the truncation error (blue circles) in (6), ii) the linear interpolation of the CDF (red squares) in (7), and iii) the error related to the CDF approximation (green triangles) in (5). As we have already anticipated in subsection 3.1, two are the most relevant error sources: the error originating from the CDF approximation and the one due to the interpolation. The error originating from the truncation is always negligible: at least ten orders of magnitude lower than interpolation error for every M. For the CDF approximation error, as explained in section 2, we have suggested an optimal selection of the shift a in the Lewis-FFT approach. The term that we need to tackle is the interpolation one: for $M > 8$ the unique relevant bound is the one on the interpolation error (e.g. for M=10 it is 10 orders of magnitude above all other errors). Similar results hold $\forall \alpha \in (0, 1)$.

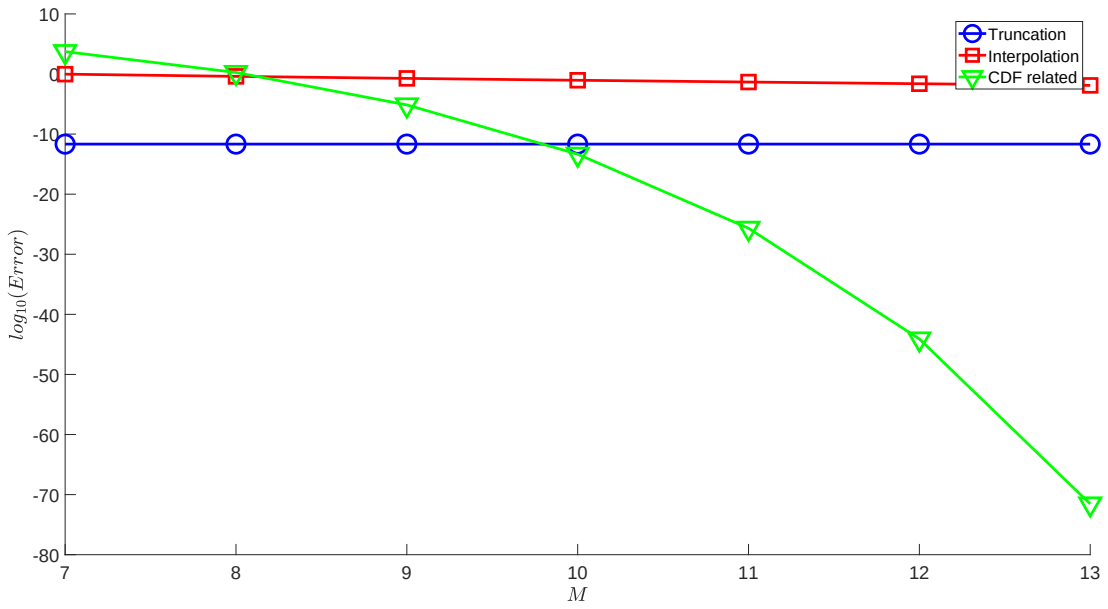

Figure 2: One-month European call options error bounds for an ATS ($\alpha = 2/3$) simulated with Lewis-FFT and linear interpolation. We plot the bounds on the three error sources: *i*) the truncation error (6) (blue circles), *ii*) the error (7) due to the linear interpolation of the CDF (red squares) and *iii*) the error (5) related to numerical CDF (green triangles). Let us emphasize that the truncation error is always negligible wrt the linear interpolation error (at least 10 orders of magnitude smaller for every M). Notice that, for $M > 8$ the unique significant term is the bound on the linear interpolation error (e.g. for $M = 10$, it is at least 10 orders of magnitude above all other errors).

As discussed in subsection 3.1, to reduce the CDF interpolation error, we consider the spline interpolation for the numerical inversion instead of the linear interpolation. With spline interpolation $\mathcal{E}$ should scale as γ^4 instead of γ^2. In figure 3 and 4, we plot the Lewis-FFT maximum error (MAX) for two different times to maturity: the error is for 30 European call options for different values of M using spline (blue circles) and linear (red squares) interpolation. We also plot SD, the average MC standard deviation, with a dashed green line. Notice that, for $M > 7$ the spline interpolation error is significantly below the linear interpolation error. Spline interpolation's error improves significantly faster than the linear interpolation's error: for M in the interval (7,10), the maximum error scales as γ^2 for the linear interpolation and as γ^6 for the spline interpolation. The observed behavior in the latter case -with an error that decreases much faster than γ^4- is probably due to the monotonicity and boundness of the interpolated function (the CDF).

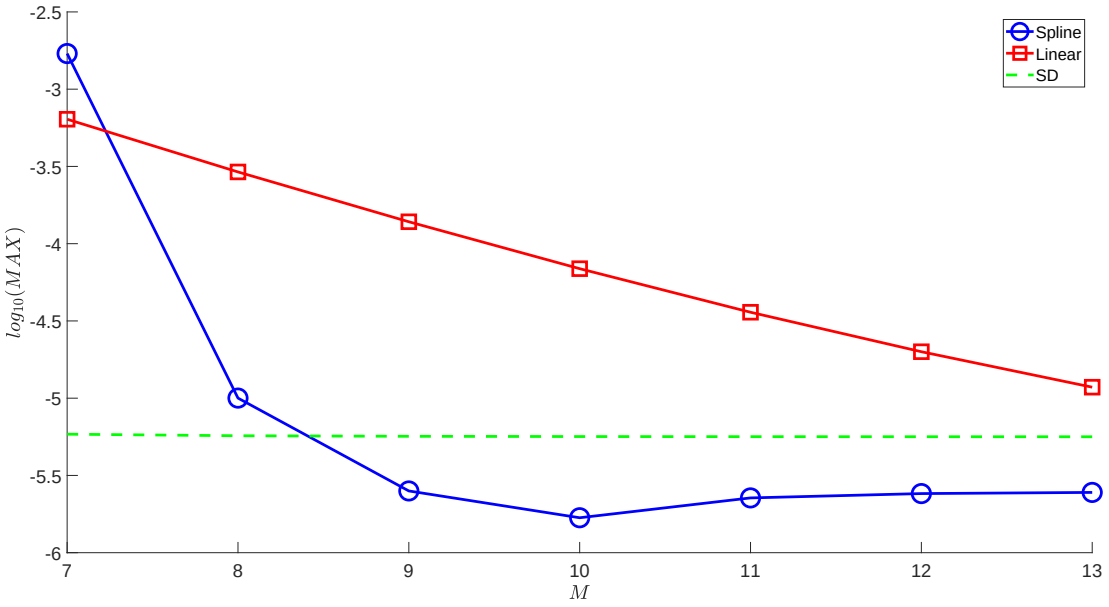

Figure 3: Maximum error for different values of M using Lewis-FFT-S (blue circles) and Lewis-FFT with linear interpolation (red squares). The maximum is computed over 30 call options (one-week maturity), with moneyness in the range $\sqrt{t}$(-0.2,0.2). We consider 10^7 simulations and $\alpha = 2/3$. Notice that, for $M > 7$ the spline interpolation error is significantly below the linear interpolation error. Spline interpolation's error improves significantly faster than the linear interpolation's error: for M in the interval (7,10) the maximum error scales, on average, as γ^6 for the spline interpolation and as γ^2 for the linear interpolation. Moreover, the maximum error becomes significantly lower than the average MC standard deviation in a dashed green line.

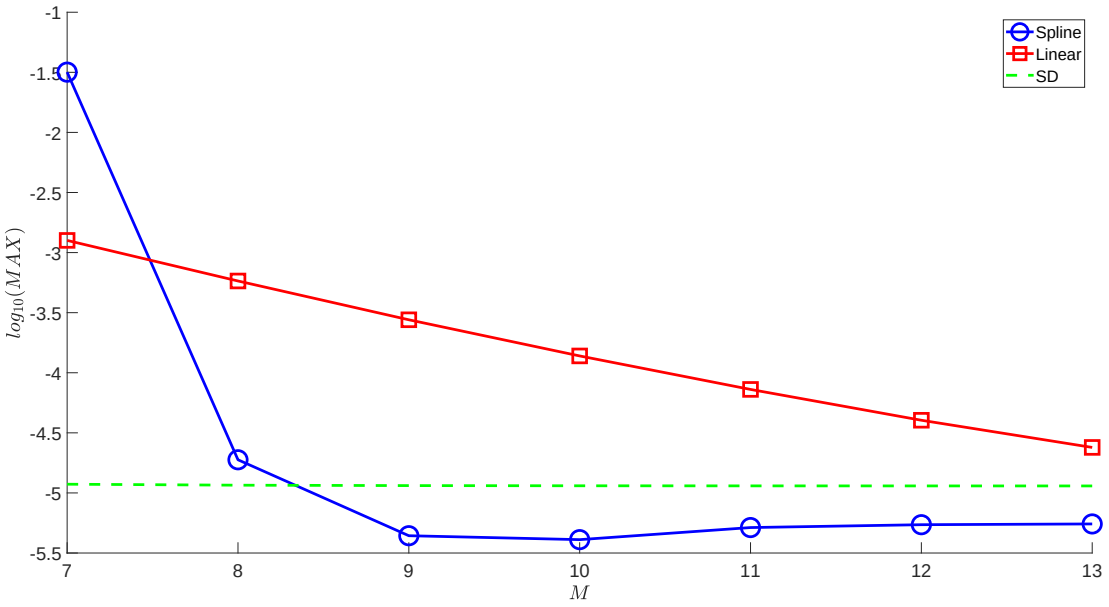

Figure 4: As figure 3 but for one-month maturity. Notice that, for $M > 7$ the spline interpolation error is significantly below the linear interpolation error. Spline interpolation's error improves significantly faster than the linear interpolation's error: for M in the interval (7,10) the maximum error scales, on average, as γ^6 for the spline interpolation and as γ^2 for the linear interpolation.

	M	7	8	9	10	11	12	13
$\alpha = 1/3$	MAX [bp]	7.08	0.32	0.02	0.03	0.03	0.03	0.03
	RMSE [bp]	3.49	0.29	0.01	0.02	0.02	0.02	0.02
	MAPE [%]	2.30	0.21	0.01	0.01	0.01	0.01	0.01
	SD [bp]	0.12	0.12	0.12	0.12	0.12	0.12	0.12
$\alpha = 2/3$	MAX [bp]	317.29	0.19	0.04	0.01	0.02	0.03	0.03
	RMSE [bp]	282.99	0.16	0.03	0.01	0.02	0.02	0.02
	MAPE [%]	185.64	0.11	0.02	0.01	0.01	0.01	0.01
	SD [bp]	0.25	0.11	0.11	0.11	0.11	0.11	0.11

Table 3: Lewis-FFT-S algorithm (with spline) performances wrt different metrics using 10^7 trials for $\alpha = 1/3$ and $\alpha = 2/3$: MAX [bp], RMSE [bp], MAPE [%], SD [bp]. The process is simulated for M that goes from 7 to 13. The metrics are computed for 30 call options (one-month maturity), with moneyness in the range $\sqrt{t}$(-0.2,0.2). We observe that for all $M > 9$ the maximum error is 0.03 bp or below.

	M	7	8	9	10	11	12	13
$\alpha = 1/3$	Time [s]	0.23	0.27	0.28	0.28	0.28	0.28	0.29
$\alpha = 2/3$	Time [s]	0.25	0.27	0.28	0.28	0.28	0.28	0.28

Table 4: Lewis-FFT-S computational time for simulating the ATS (with $\alpha = 1/3$ and $\alpha = 2/3$) over a one-month time-interval using 10^7 trials.

We also desire to estimate the method's error with different metrics: besides MAX we consider the root mean squared error (RMSE) and the mean absolute percentage error (MAPE). In table 3, we report the performances of the Lewis-FFT-S algorithm for 10^7 simulations. We consider two values of α for the ATS: $\alpha = 1/3$ and $\alpha = 2/3$. The metrics are computed for 30 call options (one-month maturity) and moneyness in the range $\sqrt{t}$(-0.2,0.2). We observe that for $M > 9$ the error is 0.03 bp or below whatever metric we consider.

The main result of this subsection is that, in the Lewis-FFT-S framework, a Monte Carlo with 10^7 simulations and $M = 13$ provides a very accurate pricing tool whatever time-horizon and $\alpha \in (0, 1)$ we consider.

4.2 European options: computational time

In this subsection, we emphasize that the proposed MC method is fast. We compare the Lewis-FFT-S computational cost both with the simplest possible dynamics for the underlying (geometric Brownian motion) and with the methodology that is often considered a benchmark for simulating jump processes (i.e. the simulation of jumps via Gaussian approximation).

In table 4, we report the performances of the Lewis-FFT-S algorithm for 10^7 simulations. We consider the ATS with $\alpha = 1/3$ and $\alpha = 2/3$. For every choice of M, we register the computational time [s]. The metrics are computed for 30 call options (one-month maturity), with moneyness in the range $\sqrt{t}$(-0.2,0.2). We observe that for $M \geq 10$ the maximum error is 0.03 bp or below.

We point out, that Lewis-FFT-S is considerably efficient. In our machine[8], sampling 10^7 trials of a geometric Brownian motion takes approximately 0.08 seconds: just one-third of the Lewis-FFT-S computational cost (reported in table 4).

[8]We use MATLAB 2021a on an AMD Ryzen 7 5800H, with 3.2 GHz.

In figure 5, we plot the computational time wrt the time to maturity in log-log scale for 10^7 simulations with Gaussian Approximation (blue squares) and Lewis-FFT-S (red circles). Time to maturity goes from one day to two years. To compare the two methods fairly, we need to select M for the Lewis-FFT-S and ϵ for the Gaussian approximation s.t. the two methods provide similar errors. As above, for both methods, we price the 30 call options, with moneyness in the range $\sqrt{t}$(-0.2,0.2). For each time to maturity, we select M and ϵ s.t. the maximum error (MAX) is between 1 bp and 0.1 bp, and s.t. the Lewis-FFT-S error is always below the Gaussian approximation error. Lewis-FFT-S computational time appears constant as the time to maturity increases. While GA computational time improves as the time to maturity reduces. However, Gaussian approximation is always more computationally expensive than Lewis-FFT-S by at least 1.75 orders of magnitude. This difference appears remarkable considering that we have verified that Lewis-FFT-S error is always below Gaussian approximation error.

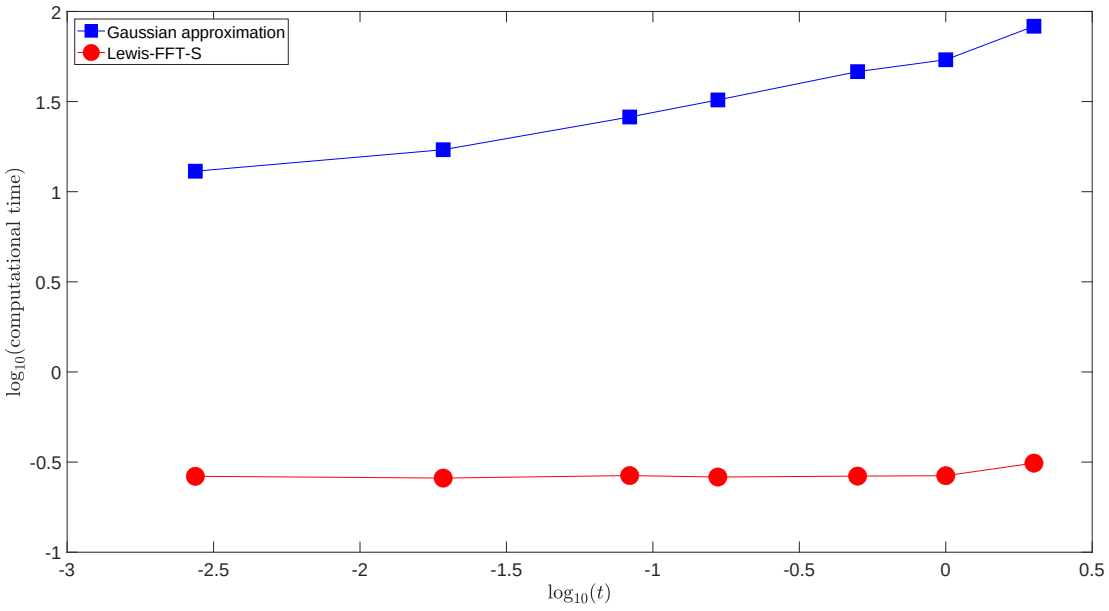

Figure 5: Computational time wrt the time to maturity in log-log scale for 10^7 simulations with GA (blue squares) and Lewis-FFT-S (red circles) techniques. We price 30 European call options, with moneyness in the range $\sqrt{t}(-0.2, 0.2)$ with GA and Lewis-FFT-S. We consider times to maturity, between one day and two years. For each t, we select M and the threshold ϵ s.t. the maximum error is between 1 bp and 0.1 bp and we require that the Lewis-FFT-S error is always below the GA error. The GA computational time improves as the time to maturity reduces because a lower number of jumps is involved, while the Lewis-FFT-S simulation depends weakly on the time horizon considered. We observe that GA is always more computationally expensive than Lewis-FFT-S by at least 1.75 orders of magnitude.

4.3 Discretely monitoring options

In this subsection, to give an idea of an application of the proposed MC, we price discretely-monitored (quarterly) Asian options, lookback options, and Down-and-In options with a time to maturity of five years.

For simplicity, we consider the case with unitary underlying initial value and without interest rates nor dividends: these deterministic quantities can be easily added to simulated prices without any computational effort.

Moneyness	Asian [%]	SD [%]	Lookback [%]	SD [%]	Down-and-In [%]	SD [%]
-0.5	39.79	0.01	3.31	0.00	2.31	0.00
-0.25	24.36	0.01	8.72	0.00	3.98	0.00
0	10.04	0.01	23.07	0.01	6.15	0.01
0.25	2.57	0.00	50.53	0.01	8.95	0.01
0.5	0.55	0.00	86.98	0.01	12.55	0.01

Table 5: Prices and MC standard deviation of Asian calls, lookback puts, and Down-and-In puts for different moneyness. We simulate 10^7 paths of the ATS with $\alpha = 2/3$ and price the discretely-monitored (quarterly) path-dependent options with time to maturity of five years. SD errors are always lower than 1 bp.

Let us call L the Down-and-In barrier. The payoffs of Asian call options, lookback put options and Down-and-In put options we consider are respectively

$$\Big(\sum_{i=0}^{n} e^{f_{t_i}} - e^{-x}\Big)^+ \ ,$$

$$\Big(e^{-x} - \min_i e^{f_{t_i}}\Big)^+ \ \text{and}$$

$$\Big(e^{-x} - e^{f_{t_i}}\Big)^+ \mathbb{I}_{\min_i e^{f_{t_i}} \geq L} \ ,$$

where $0 = t_0 < t_1 < ... < t_i < ... < t_n$ with $n = 20$ the monitoring times, f_{t_i} is the process at time t_i for the logarithm of the underlying price. For example, the process $\{f_t\}_{t\geq 0}$ can be modeled as a Brownian motion in the simplest case (Black-Scholes) or as an ATS process, as discussed in this paper. In both cases, we can simulate the paths of $\{f_t\}_{t\geq 0}$ by simulating the increments $f_{t_i} - f_{t_{i-1}}$.

In table 5, we report prices and MC standard deviation of Asian calls, lookback puts and Down-and-In puts (with a barrier strike $L = 0.6$). We simulate 10^7 paths of the ATS with $\alpha = 2/3$ and price the discretely-monitored (quarterly) path-dependent options with time to maturity five years. We consider options with different moneyness in the range (-0.5,0.5), where $0.5 \approx 0.2\sqrt{t}$ for $t = 5$ years. We use $M = 13$ for the numerical CDF inversion. The method is very precise: the numerical error SD is of the order of one bp (or below) in all considered cases.
As pointed out in the previous subsection, the Lewis-FFT-S is also extremely efficient when pricing discretely-monitored path-dependent exotics: with an ATS, it takes only three times the computational cost that it takes with a standard Brownian motion.

5 Conclusions

In this paper, we propose the Lewis-FFT-S method: a new Monte Carlo scheme for additive processes that leverages on the numerical efficiency of the FFT applied to the Lewis formula for a CDF and on the spline interpolation properties when inverting the CDF. We present an application to the additive normal tempered stable process, which has excellent calibration features on the equity volatility surface (see e.g., Azzone and Baviera 2021). This simulation scheme is accurate and fast.

We discuss in detail the accuracy of the method. In figure 2, we analyze the three-components of the bias error in (4). In this study we have shown how to accelerate convergence improving the two main sources of numerical error (4): the CDF error (5) and the interpolation error (7). First, we sharpen the CDF error considering the Lewis formula (1) for CDF and selecting the optimal

shift that minimizes the error bound in the FFT. Second, we substitute the linear interpolation with the spline interpolation. In this way, the leading term in the interpolation error improves from γ^2 to at least γ^4. This improvement is particularly evident in figures 3-4, where, for $M > 7$, the Lewis-FFT-S maximum error is significantly below the Lewis-FFT version of the method with linear interpolation and it appears to decrease as γ^6 in numerical experiments.

The Lewis-FFT-S is also fast. As discussed in subsection 3.1, for a sufficiently large number of simulations, the increment in computational time due to spline interpolation is negligible. Moreover, as shown in figure 5, the proposed method is at least one and a half orders of magnitude faster than the traditional GA simulations, whatever time horizon we consider. We also observe that, when pricing some discretely-monitoring path-dependent options, the computational time is of the same order of magnitude as standard algorithms for Brownian motions.

References

Abramowitz, M. and Stegun, I.A., 1948. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, vol. 55, US Government printing office.

Asmussen, S. and Glynn, P.W., 2007. *Stochastic simulation: algorithms and analysis*, vol. 57, Springer Science & Business Media.

Asmussen, S. and Rosiński, J., 2001. Approximations of small jumps of Lévy processes with a view towards simulation, *Journal of Applied Probability*, 38 (2), 482–493.

Azzone, M. and Baviera, R., 2021. Additive normal tempered stable processes for equity derivatives and power law scaling, *Quantitative Finance (to appear), also available at arXiv:1909.07139*.

Ballotta, L. and Kyriakou, I., 2014. Monte Carlo simulation of the CGMY process and option pricing, *Journal of Futures Markets*, 34 (12), 1095–1121.

Bohman, H., 1970. A method to calculate the distribution function when the characteristic function is known, *BIT Numerical Mathematics*, 10 (3), 237–242.

Carr, P. and Torricelli, L., 2021. Additive logistic processes in option pricing, *Finance and Stochastics*, 25, 689–724.

Chen, Z., Feng, L., and Lin, X., 2012. Simulating Lévy processes from their characteristic functions and financial applications, *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 22 (3), 1–26.

Chourdakis, K., 2005. Option pricing using the fractional FFT, *Journal of computational finance*, 8 (2), 1–18.

Cont, R. and Tankov, P., 2003. *Financial Modelling with jump processes*, Chapman & Hall/CRC Financial Mathematics Series.

Cormen, T.H., Leiserson, C.E., Rivest, R.L., and Stein, C., 2001. *Introduction to algorithms*, MIT Press.

Eberlein, E. and Madan, D.B., 2009. Sato processes and the valuation of structured products, *Quantitative Finance*, 9 (1), 27–42.

Feng, L. and Lin, X., 2013. Inverting analytic characteristic functions and financial applications, *SIAM Journal on Financial Mathematics*, 4 (1), 372–398.

Glasserman, P., 2004. *Monte Carlo methods in financial engineering*, vol. 53, Springer Science & Business Media.

Glasserman, P. and Liu, Z., 2010. Sensitivity estimates from characteristic functions, *Operations Research*, 58 (6), 1611–1623.

Hall, C.A. and Meyer, W.W., 1976. Optimal error bounds for cubic spline interpolation, *Journal of Approximation Theory*, 16 (2), 105–122.

Lee, R.W., 2001. Implied and local volatilities under stochastic volatility, *International Journal of Theoretical and Applied Finance*, 4 (1), 45–89.

Lee, R.W., 2004. Option pricing by transform methods: extensions, unification and error control, *Journal of Computational Finance*, 7 (3), 51–86.

Lewis, A.L., 2001. A simple option formula for general jump-diffusion and other exponential Lévy processes, *Available on SSRN, ssrn.com/abstract=282110*.

Li, J., Li, L., and Mendoza-Arriaga, R., 2016. Additive subordination and its applications in finance, *Finance and Stochastics*, 20 (3), 589–634.

Lukacs, E., 1972. A survey of the theory of characteristic functions, *Advances in Applied Probability*, 4 (1), 1–37.

Marsaglia, G. and Tsang, W.W., 2000. The ziggurat method for generating random variables, *Journal of statistical software*, 5 (8), 1–7.

Press, W.H., Teukolsky, S.A., Vetterling, W.T., and Flannery, B.P., 1992. *Numerical recipes in C: The Art of Scientific Computing*, Cambridge university press.

Quarteroni, A., Sacco, R., and Saleri, F., 2007. *Numerical mathematics*, vol. 37, Springer Science & Business Media.

Samorodnitsky, G. and Taqqu, M., 1994. *Stable Non-Gaussian Random Processes: Stochastic Models with Infinite Variance*, Chapman & Hall.

Sato, K.I., 1999. *Lévy processes and infinitely divisible distributions*, Cambridge university press.

Notation and shorthands

Symbol	Description
β	scaling parameter of ATS variance of jumps
δ	scaling parameter of ATS skew parameter
$\mathcal{E}$	total error when pricing the derivative with payoff $V(x)$
η_t	ATS skew parameter
$\bar{\eta}$	ATS constant part of skew parameter
ϵ	small jump threshold for GA
$\mathcal{E}^{CDF}_{h,M}$	CDF error bound as a function of the grid size h and of M
$\mathcal{E}^{CDF}_{M}$	CDF error bound when h is s.t. the two sources of CDF error are comparable
f_t	the additive process at time t
$\phi_{s,t}$	characteristic function of additive increment between s and t time to maturity
γ	grid size in the CDF domain
h	grid size in the Fourier domain
k_t	ATS variance of jumps parameter
$\bar{k}$	ATS constant part of the variance of jumps
K	dimension of the CDF interpolation grid
L	Down-and-In barrier strike
$m^+_{s,t}$	probability density of positive jumps
$m^-_{s,t}$	probability density of negative jumps
M	integer number s.t. N is the number of grid points
n	number of monitoring times in path dependent derivatives
n_v	number of points in which V is not differentiable
N	number of grid points ($N = 2^M$)
$\mathcal{N}_{sim}$	number of simulations
ν_t	jump measure of additive process
$P(x)$	model CDF of the increment between the times s and t
$\hat{P}(x)$	numerical approximation of the CDF of the increment between the times s and t
p^-_t	upper bound of $\phi_{s,t}$ strip of regularity
p^+_t	$-(p^+_t + 1)$ is the lower bound of $\phi_{s,t}$ strip of regularity
$\bar{\sigma}$	ATS diffusion parameter
U	uniform r.v. in (0,1)
$V(x)$	derivative payoff
(x_0, x_K)	interval in which the CDF is interpolated

Shorthands

Symbol	Description
a.s.	almost surely
ATS	Additive normal tempered stable process
bp	basis points
CDF	Cumulative Distribution Function
FFT	Fast Fourier Transform
GA	Gaussian approximation technique
MAPE	MC prices mean absolute percentage error
MAX	Maximum error in MC prices
MC	Monte Carlo
ms	milliseconds
nn	nearest neighborhood algorithm
r.v.	random variable
RMSE	MC prices root mean squared errors
SD	average MC prices standard deviation
wrt	with respect to

Appendices

A Proofs

Proof of proposition 2.1
We bound the range and the discretization error separately.

First, we bound the CDF range error, i.e. the error we introduce considering the integral (1) in the range $(0, Nh)$. Fix h, it exists $N \in \mathbb{N}$ s.t.

$$\begin{aligned}
&\left| P(x) - \left(1 - \frac{e^{-ax}}{\pi} \int_0^{N\,h} du\; Re\left[e^{-iux} \frac{\phi_{s,t}(u - ia)}{iu + a} \right) \right] \right| \\
&\leq \frac{e^{-ax}}{\pi} \int_{N\,h}^{\infty} du\; \frac{B\, e^{-b\, u^{\omega}}}{u} < \frac{e^{-ax}}{N\,h} \frac{1}{\pi} \int_{N\,h}^{\infty} du\; B e^{-b\, u^{\omega}} \qquad (8) \\
&= \frac{e^{-ax}\, B}{N\,h} \frac{1}{\omega\; b^{1/\omega}} \Gamma\left[\frac{1}{\omega}, b\,(N\,h)^{\omega} \right] = O\left((N\,h)^{-\omega} e^{-b\,(N\,h)^{\omega}} \right) \quad . \qquad (9)
\end{aligned}$$

The first inequality is due to $|iu+a| > u$ and to the fact that $|\phi_{s,t}(u-ia)| \leq Be^{-b\, u^{\omega}}$ for sufficiently large values of u, thanks to Assumption 2.

Second, we bound the CDF discretization error.
By theorem 6.2 of Lee (2004), we have that

$$\begin{aligned}
&\left| \frac{e^{-ax}}{2\pi} \int_{\mathbb{R}} du\; e^{-iux} \frac{\phi_{s,t}(u - ia)}{iu + a} - \frac{e^{-ax}}{2\pi} \sum_{j=1}^{M} \frac{e^{-iu_j x} \phi_{s,t}(u_j - ia)}{iu_j + a} \right| \\
&\leq \frac{e^{-2\pi a/h}}{1 - e^{-4\pi a/h}} + \frac{e^{-2\pi(p-a)/h - p\,x}}{1 - e^{-4\pi(p-a)/h}} \phi_{s,t}(-ip) \qquad \forall p < p_t^+ + 1 \quad . \qquad (10)
\end{aligned}$$

Notice that $\phi_{s,t}(-i\,p)$ is well defined because $p < p_t^+ + 1$.

We select a and p to minimize the discretization error. Notice that, for a sufficiently large N, the leading terms in (10) are $e^{-2\pi a/h}$ and $e^{-2\pi(p-a)/h-p\,x}$. Hence, for a given p the best choice of a is

$$\hat{a} = \frac{p}{2}\left(1 - \frac{x}{\pi}\,h\right) \quad .$$

This last quantity, for a sufficiently small h, is close to $p/2$ for any finite x. Thus, to minimize the discretization error, we select $a = p/2$. Then, p can be chosen to its maximum value $p_t^+ + 1$ and we obtain

$$\left| \frac{e^{-ax}}{2\pi} \int_{\mathbb{R}} du\, e^{-iux} \frac{\phi_{s,t}(u - ia)}{iu + a} - \frac{e^{-ax}}{2\pi} \sum_{j=1}^{M} \frac{e^{-iu_j x} \phi_{s,t}(u_j - ia)}{iu_j + a} \right| \tag{11}$$

$$\leq \frac{e^{-\pi(p_t^+ + 1)/h} + e^{-\pi(p_t^+ + 1)/h - (p_t^+ + 1)\,x} \phi_{s,t}(-i(p_t^+ + 1))}{1 - e^{-2\pi(p_t^+ + 1)/h}} \quad . \tag{12}$$

By combining equations (9,12), the thesis follows □

To prove proposition 4.1, we report the ATS characteristic function and the ATS jump measure. At time t, the ATS characteristic function is

$$\phi_t(u) = \mathcal{L}_t\left(iu\left(\frac{1}{2} + \eta_t\right)\sigma_t^2 + \frac{u^2\sigma_t^2}{2};\ k_t,\ \alpha\right) e^{-iu \log \mathcal{L}_t\left(\eta_t \sigma_t^2;\, k_t,\, \alpha\right)} \quad . \tag{13}$$

σ_t, k_t are continuous on $[0, \infty)$ and η_t is continuous on $(0, \infty)$, with $\sigma_t > 0$, $k_t, \eta_t \geq 0$. As in the corresponding Lévy case, we define

$$\ln \mathcal{L}_t\left(u;\ k,\ \alpha\right) := \frac{t}{k}\frac{1-\alpha}{\alpha}\left\{1 - \left(1 + \frac{u\,k}{1-\alpha}\right)^{\alpha}\right\} \quad ,$$

with $\alpha \in (0, 1)$.[9]
The ATS jump measure is

$$\nu_t(x) = \frac{tC\left(\alpha, k_t, \sigma_t, \eta_t\right)}{|x|^{1/2+\alpha}} e^{-(1/2+\eta_t)x} K_{\alpha+1/2}\left(|x|\sqrt{\left(1/2 + \eta_t\right)^2 + 2(1-\alpha)/(k_t\,\sigma_t^2)}\right) \quad , \tag{14}$$

with

$$C\left(\alpha, k_t, \sigma_t, \eta_t\right) := \frac{2}{\Gamma(1-\alpha)\sqrt{2\pi}} \left(\frac{1-\alpha}{k_t}\right)^{1-\alpha} \sigma_t^{2\alpha} \left(\left(1/2 + \eta_t\right)^2 + 2(1-\alpha)/(k_t\,\sigma_t^2)\right)^{\alpha/2+1/4} \quad ,$$

and $K_\nu(x)$ the modified Bessel function of the second kind (see e.g., Abramowitz and Stegun 1948, ch.9, p.376)

$$K_\nu(x) := \frac{e^{-x}}{\Gamma\left(\nu + \frac{1}{2}\right)} \sqrt{\frac{\pi}{2\,x}} \int_0^\infty e^{-z} z^{\nu - 1/2} \left(1 + \frac{z}{2\,x}\right)^{\nu - 1/2} dz \quad .$$

Moreover, we recall a sufficient condition for the existence of ATS is provided in the following theorem (cf. Azzone and Baviera 2021, th.2.1).

Theorem A.1. Sufficient conditions for existence of ATS
There exists an additive process $\{f_t\}_{t\geq 0}$ with the characteristic function (13) if the following two conditions hold.

[9] We emphasize that we consider $\alpha > 0$. As discussed in subsection 2.2, this is the relevant situation in practice when pricing exotic derivatives: the case with α exactly equal to zero presents a power law decay in the characteristic function.

1. $g_1(t)$, $g_2(t)$, *and* $g_3(t)$ *are non decreasing, where*

$$
\begin{aligned}
g_1(t) &:= (1/2+\eta_t) - \sqrt{(1/2+\eta_t)^2 + 2(1-\alpha)/(\sigma_t^2 k_t)} \\
g_2(t) &:= -(1/2+\eta_t) - \sqrt{(1/2+\eta_t)^2 + 2(1-\alpha)/(\sigma_t^2 k_t)} \\
g_3(t) &:= \frac{t^{1/\alpha}\sigma_t^2}{k_t^{(1-\alpha)/\alpha}} \sqrt{(1/2+\eta_t)^2 + 2(1-\alpha)/(\sigma_t^2 k_t)} \ ;
\end{aligned}
\tag{15}
$$

2. *Both* $t\,\sigma_t^2\,\eta_t$ *and* $t\,\sigma_t^{2\alpha}\,\eta_t^{\alpha}\,/k_t^{1-\alpha}$ *go to zero as* t *goes to zero* □

Proof of proposition 4.1
First, we prove that Assumption 1 holds.
This entails showing that p_t^+ and p_t^- are non increasing for the ATS. At time t, the ATS characteristic function in equation (13) is analytic on the imaginary axis $u = -i\,a$, $a \in \mathbb{R}$ iff

$$
1 + \frac{k_t}{1-\alpha}\left(a\left(\frac{1}{2}+\eta_t\right)\sigma_t^2 - \frac{a^2\sigma_t^2}{2}\right) > 0 \ .
$$

By solving the second order inequality, we get

$$
g_1(t) < a < -g_2(t) \ ,
$$

with $g_1(t)$ and $g_2(t)$ defined in (15). Hence, $p_t^+ := -g_2(t) - 1$ and $p_t^- := -g_1(t)$. Notice that $-g_2(t)$ and $-g_1(t)$ are non increasing wrt t because of condition 1 in theorem A.1 and p_t^+ and p_t^- are positive through direct inspection. Moreover, $p_t^+ \geq p_t^-$ because

$$
p_t^+ - p_t^- = 2\eta_t \geq 0 \ .
$$

Second, we prove that Assumption 2 holds. We observe that, by the condition on $g_1(t)$ and $g_2(t)$ of theorem A.1, we have

$$
g(t) := -(g_1(t) + g_2(t)) = \sqrt{(1/2+\eta_t)^2 + 2(1-\alpha)/(k_t\,\sigma_t^2)}
$$

is non increasing wrt t. Hence, thanks to condition 1 on $g_3(t)$ of theorem A.1

$$
\frac{t}{k_t^{1-\alpha}}\sigma_t^{2\alpha} \ \text{ is increasing in } t \ . \tag{16}
$$

We have to show that, given s and t, there exists $B > 0$, $b > 0$ and $\omega > 0$ such that, for sufficiently large u, Assumption 2 holds for the characteristic function of ATS.
We choose $\log(B) > \frac{1-\alpha}{\alpha}\left(\frac{t}{k_t} - \frac{s}{k_s}\right)$, $0 < b < \frac{(1-\alpha)^{1-\alpha}}{2^\alpha\alpha}\left(\frac{t}{k_t^{1-\alpha}}\sigma_t^{2\alpha} - \frac{s}{k_s^{1-\alpha}}\sigma_s^{2\alpha}\right)$ and $0 < \omega < 2\alpha$.
Notice that it is possible to fix $b > 0$, because (16) holds. Moreover, the imaginary part of the exponent in (13) does not contribute to B, because the absolute value of the exponential of an imaginary quantity is unitary.
For sufficiently large u, $|\phi_{t,s}(u - i\,a)|$ goes to zero faster than $Be^{-b\,u^\omega}$ because $\log\phi_{t,s}(u - i\,a)$ is asymptotic to

$$
-\frac{(1-\alpha)^{1-\alpha}}{2^\alpha\alpha}\left(\frac{t}{k_t^{1-\alpha}}\sigma_t^{2\alpha} - \frac{s}{k_s^{1-\alpha}}\sigma_s^{2\alpha}\right) u^{2\,\alpha} \ ,
$$

that is negative due to (16) for $\alpha \in (0,1)$.

Finally, we prove that Assumption 3 holds.
We have to demonstrate that $m^+_{s,t}(x)$ is non increasing in x and $m^-_{s,t}(x)$ is non decreasing. We prove the thesis by showing that the derivative of $\nu_t(x)$ wrt x is negative and non increasing in t for any $x > 0$. Notice that if this holds then

$$m^+_{s,t}(x) = \mathbb{I}_{x>\epsilon} \frac{\nu_t(x) - \nu_s(x)}{\int_\epsilon^\infty dz(\nu_t(z) - \nu_s(z))}$$

is non increasing in x. Deriving $\nu_t(x)$ in (14), we get

$$\frac{\partial \nu_t(x)}{\partial x} = -C_2 \int_0^\infty dz \frac{e^{-z} z^\alpha g_3(t)\, e^{x g_2(t)}}{x^{2+\alpha}} \left(\alpha + \frac{z}{z/2 + x\, g(t)} + 1 - x\, g_2(t) \right) \ ,$$

where C_2 is a positive constant. The derivative of $\nu_t(x)$ is non increasing in t for any $x > 0$ because

1. $g_3(t)$ is positive and non decreasing in t by condition 1 of theorem A.1;
2. $e^{x\, g_2(t)} \left(\alpha + \frac{s}{s/2 + x\, g(t)} \right)$ is the combination of two non decreasing function in t for any $x > 0$;
3. $g_2(t)$ is negative and non decreasing and $(1 - c\, x)e^{c\, x}$ is non decreasing for $c < 0$.

This proves the thesis for $x > 0$.
The same holds true for $x < 0$. *Mutatis mutandis*, by substituting $g_2(t)$ with $g_1(t)$, the proof is the same □

Proof of proposition 4.2
We recall that, for a Sato process, the characteristic function at time t is $\phi_t(u) = \phi_1(u\, t^\zeta)$, where $\zeta > 0$ is the scaling constant (see e.g., Eberlein and Madan 2009, p.6).

First, we prove that Assumption 1 holds.
This entails showing that p^+_t and p^-_t are non increasing for the process. At time $t = 1$, the characteristic function $\phi_1(u)$ is analytic on the imaginary axis $u = -i\, a$, $a \in \mathbb{R}$ iff

$$-p^-_1 < a < p^+_1 + 1 \ ,$$

where p^-_1 and $p^+_1 + 1$ depend on the process specific characteristic function (cf. Lukacs 1972, th.3.1, p.12). Since $\phi_t(u) = \phi_1(u\, t^\zeta)$, $\phi_t(u)$ is analytic on the imaginary axis iff

$$\frac{-p^-_1}{t^\zeta} < a < \frac{p^+_1 + 1}{t^\zeta} \ .$$

Hence, for a Sato process, $p^+_t = \frac{p^+_1 + 1}{t^\zeta} - 1$ and $p^-_t = \frac{p^-_1}{t^\zeta}$, which are non increasing in t.

Second, we prove that Assumption 2 holds.
If $\phi_1(u)$ decays exponentially as $e^{-\hat{b} u^w}$, with $\hat{b} > 0$, then $\phi_t(u) = \phi_1(u t^\zeta)$ decays as $e^{-\hat{b} u^w\, t^{\zeta\, w}}$. It is possible to select $B > 0$ and $0 < b < \hat{b}(t^{\zeta\, w} - s^{\zeta\, w})$ s.t. $|\phi_{t,s}(u)| = |\phi_t(u)/\phi_s(u)| < B e^{-b u^w}$ for $t > s$ □

B Simulation algorithm

A brief description of the Lewis-FFT algorithm follows

---

procedure LEWIS-FFT($M, \mathcal{N}_{sim}, FlagSpline$)

COMPUTE $h(M)$, N, γ
COMPUTE $\hat{P}$ with FFT ▷ $z_0,\ z_{N-1}$ fixed by FFT

FIX x_K nearest point to $5\sqrt{t-s}$ and $x_0 = -x_K$
$\vec{x} = x_0 : \gamma : x_K$ ▷ Grid dimension: $K+1$

SAMPLE a vector U of $\mathcal{N}_{sim}$ uniform r.v. in [0,1]
J = NearestNeighborhood(U, $\hat{P}(\vec{x})$) ▷ Find next element in the grid

if $FlagSpline = True$ **then**
  COMPUTE spline interpolation coefficients $\{c^S_{q,J}\}_{q=0}^3$ ▷ Solve tridiagonal linear sistem
  X = spline($\hat{P}(\vec{x})$, $\vec{x}$, U,J) ▷ Interpolate on U
else
  COMPUTE linear interpolation coefficients $\{c^L_{q,J}\}_{q=0}^1$
  X $= c^L_{0,J} + U\, c^L_{1,J}$ ▷ Interpolate on U
