POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

A general methodology for the simulation of Ornstein–Uhlenbeck-Lévy and Lévy-Ornstein–Uhlenbeck processes applied to option pricing

Tesi di Laurea Magistrale in
Mathematical Engineering - Ingegneria Matematica

Sebastian Castellano, 10582405

Advisor:
Prof. Roberto Baviera

Co-advisors:
Pietro Manzoni

Academic year:
2021-2022

Abstract: The Lévy-Ornstein-Uhlenbeck and Ornstein-Uhlenbeck-Lévy processes are a general class of processes used for a variety of financial applications. In the literature many exact Monte Carlo algorithms exist for their simulation, but although accurate they can be slow, of complex implementation, and are specific to each single process. With this paper we propose a fast Monte Carlo algorithm for their simulation which solves these problems while guaranteeing comparable levels of accuracy. Indeed the fast Monte Carlo algorithm proposed is fast, simple and general. We support this claim by comparing the performance of our algorithm to that of the exact algorithms in terms of both accuracy and computational time.

Key-words: Monte Carlo simulation, non-Gaussian Ornstein-Uhlenbeck, tempered stable, normal tempered stable, option pricing

1. Introduction

In this paper we propose a fast Monte Carlo simulation algorithm for the class of Lévy-Ornstein-Uhlenbeck and Ornstein-Uhlenbeck-Lévy processes. These processes can be used in various applications within mathematical finance: in the context of energy markets (Benth et al. (2007b,a), Benth and Pircalabu (2018) and Sabino and Cufaro Petroni (2021a)), for modelling credit risk (Bianchi and Fabozzi, 2015) or for stochastic volatility modeling (Barndorff-Nielsen and Shephard, 2001). Recently, a fast algorithm has been proposed for additive process, i.e. processes with independent increments (Azzone and Baviera, 2021); we show how to extend the technique to non-Gaussian Ornstein-Uhlenbeck

processes (OU hereinafter) and we compare it with other algorithms present in the literature both in terms of accuracy and of computational time. In particular, we analyse the performance of the algorithm for Lévy processes (characterizing the OU) belonging to two large classes: the tempered stable (TS) and the normal tempered stable processes (NTS). These two classes are among the most used in the financial literature, e.g they include the Variance Gamma studied in Madan and Seneta (1990) or the Normal Inverse Gaussian in Barndorff-Nielsen (1997), and they can accommodate jumps, asymmetry and heavy tails.

In the literature, many papers propose exact Monte Carlo (MC) simulation schemes for the Lévy-OU and OU-Lévy processes, see for example Kawai and Masuda (2011), Zhang (2011), Qu et al. (2021), Sabino and Cufaro Petroni (2022) and Sabino (2022b) for OU-TS and TS-OU, Grabchak (2020) and Grabchak and Sabino (2023) for OU processes driven by a generalized p-tempered TS and Sabino (2022a, 2023) for OU-NTS and NTS-OU. Monte Carlo methods are of particular interest in option pricing, since they can be applied to the pricing of a wide set of discretely-monitoring options with many reset dates. Although the algorithms found in the literature are exact, in the sense that they require no numerical approximation, and therefore they generate unbiased and accurate paths, the main drawback is that they can be relatively slow, of complex implementation and not general, since for each different process a new specific algorithm must be designed and then implemented. The common procedure shared across these exact Monte Carlo algorithms consists in decomposing the Lévy measure of the process into terms which can be recognized as Lévy measures of other processes whose simulation is known, such as tempered stable processes or compound Poisson. Once a possible decomposition is devised, each term must be simulated separately: this part may be non-trivial and computationally intensive. Indeed, to guarantee that the simulations are exact, each term should involve an exact simulation, which can be quite complex and time-consuming when it is not possible to invert the corresponding cumulative distribution function. Even for the terms where there exists a known simulation method, for example for tempered stable random variables, the method may involve acceptance-rejection schemes (see e.g. Flury (1990)), whose acceptance rate is not always close to 1 for every choice of the parameters. This is the main reason why the time needed for simulating with exact methods is non-uniform with respect to the chosen parameters. Moreover, multiple random variables must be sampled for each path, adding to the computational time.

Considering these drawbacks, the need for a fast, simple and general algorithm seems clear. These are indeed the characteristics of the proposed fast MC algorithm: we show that its computational time is significantly lower than the exact methods and is of the same order of magnitude of the time needed to generate a Brownian motion, it is simple to implement since it requires only the characteristic function of the process and it consists of only a few steps, and it is general, since it can be used for all the OU-Lévy and Lévy-OU processes of which we know the characteristic function.

In the literature other "non-exact" Monte Carlo simulation schemes have been developed in the last decades, the Gaussian approximation of Asmussen and Rosiński (2001) being the most efficient (see e.g Cont and Tankov (2003) and references therein). For completeness, we compare the proposed algorithm also to the Gaussian approximation (GA) method.

These are the main contributions of this paper. First, we propose a fast MC algorithm which significantly improves the speed for the simulation of OU-Lévy and Lévy-OU processes with respect to existing methods. Second, we provide a fast simulation technique for infinite variation processes of

OU-tempered stable and tempered stable-OU type, where up to our knowledge no exact MC algorithm exists. Finally, we provide a foundation to extend the algorithm to a large class of OU-Lévy and Lévy-OU processes.

The rest of the paper is organized as follows. In section 2, we introduce the class of OU-Lévy and Lévy-OU processes, explain their different definitions, define the general formulas for their characteristic function and Lévy measure, and show two couples of example applications with a Lévy driver either tempered stable or normal tempered stable. In section 3, we illustrate the fast MC algorithm and give some insights on its implementation. In section 4, we analyze the efficiency of the algorithm when used to simulate the paths of the processes previously introduced in section 2, in terms of both accuracy and computational time. We compare the performance of the fast MC algorithm with the exact MC algorithms and with the GA technique. Finally, section 5 concludes.

2. The Ornstein–Uhlenbeck-Lévy and Lévy-Ornstein–Uhlenbeck processes

In this section, we introduce the class of Ornstein–Uhlenbeck-Lévy (OU-Lévy) and Lévy-Ornstein–Uhlenbeck (Lévy-OU) processes. First, we define them in a general way, then we describe two couples of example applications: the OU-tempered stable (OU-TS) and the tempered stable-OU (TS-OU) processes where the Lévy part is a tempered stable (TS) process, and the OU-normal tempered stable (OU-NTS) and normal tempered stable-OU (NTS-OU) processes, where the Lévy part is a normal tempered stable (NTS) process. For each process we show how to obtain the characteristic function of the increment between 0 and t, which is the only element needed for the fast MC algorithm which we introduce in section 3, and how to obtain its Lévy measure, which can be used for benchmark methods such as Asmussen and Rosiński (2001) Gaussian approximation.

Let us consider the non-Gaussian Ornstein-Uhlenbeck (OU) process $X(t)$ defined via the following stochastic differential equation

$$dX(t) = -bX(t)dt + dL(t), \qquad X(0) = X_0, \qquad b \geq 0, \tag{1}$$

whose solution is

$$X(t) = aX_0 + Z(t), \qquad Z(t) = \int_0^t e^{-b(t-s)} dL(s), \qquad a = e^{-bt} \tag{2}$$

where $L(t)$ is a Lévy process which we call the background driving Lévy process (BDLP) and b is the mean-reversion rate parameter of the process.
When referencing to OU processes two are the main classes, first introduced in Barndorff-Nielsen and Shephard (2001), depending whether they are defined via the BDLP $L(t)$ or via the distribution of the (stationary) process $X(t)$. We call OU-Lévy processes the former and Lévy-OU the latter, as it is used throughout the literature (see e.g. Barndorff-Nielsen and Shephard (2001), Sabino and Cufaro Petroni

(2022), Qu et al. (2021) or Kawai and Masuda (2011)). Hereafter, to make the distinction even clearer, we indicate the process as $\bar{X}(t)$ in the latter case.

By observing the OU solution of equation (2), one can notice that studying the transition law of an OU process reduces to the term $Z(t)$, since $X(t)$ can then be obtained by adding the known (at the initial time) shift aX_0. As a consequence, it is sufficient to define the characteristic function and the Lévy measure of the term $Z(t)$.

OU-Lévy processes For OU-Lévy processes the Lévy measure $\nu_Z(x,t)$ of $Z(t)$ is

$$\nu_Z(x,t) = \frac{1}{b|x|} \begin{cases} \int_{x/a}^{x} \nu_L(y)dy & x < 0 \\ \int_{x}^{x/a} \nu_L(y)dy & x > 0 \end{cases} \tag{3}$$

where $\nu_L(x)$ is the Lévy measure of the BDLP, and b and a are the coefficients introduced in equation (1) and (2).
The formula can be obtained using Cont and Tankov (2003, prop. 15.4) to obtain the Lévy measure of $\bar{X}(t)$, plus the self-decomposability principle of equation (5) to then obtain that of $Z(t)$, and it can be found in Sabino and Cufaro Petroni (2022, eq. 21).
The concept of self-decomposability is central for OU processes, since Barndorff-Nielsen et al. (1998, p. 996) states that the process $X(t)$ in equation (1) is stationary if and only if its distribution is self-decomposable (see also Cont and Tankov (2003, ch. 15.3.2)).
Moreover, Cont and Tankov (2003, prop. 15.4) states that the BDLP must satisfy

$$\int_{|x|\geq 1} log|x|\nu_L(dx) < \infty$$

for $X(t)$ to have a stationary distribution which is self-decomposable.

For OU-Lévy processes the log characteristic function (hereinafter log-chf) $\psi_Z(u,t)$ of $Z(t)$ (such that the characteristic function of $Z(t)$ is $\phi(u,t) = e^{\psi_Z(u,t)}$) can be obtained through the characteristic exponent $\varphi_L(u)$ of the BDLP as by Cont and Tankov (2003, lemma 15.1)

$$\psi_Z(u,t) = \int_0^t \varphi_L(ue^{-bs})ds \tag{4}$$

Lévy-OU processes For Lévy-OU processes both the Lévy measure $\nu_Z(x,t)$ and the log-chf $\psi_Z(u,t)$ can be obtained by self-decomposability properties from the Lévy measure $\nu_{\bar{X}}(x)$ and the characteristic exponent $\varphi_{\bar{X}}(u)$ of the stationary process $\bar{X}(t)$ (Sabino and Cufaro Petroni, 2022, prop. 1 and eq. 6)

$$\nu_Z(x,t) = \nu_{\bar{X}}(x) - \frac{\nu_{\bar{X}}(x/a)}{a} \tag{5}$$

$$\psi_Z(u,t) = \varphi_{\bar{X}}(u) - \varphi_{\bar{X}}(au) \tag{6}$$

These equations hold for any self-decomposable distribution of $\bar{X}(t)$. We remind that an infinitely divisible distribution on $\mathbb{R}$ is self-decomposable if and only if its Lévy measure has density $\nu(x) = \frac{k(x)}{|x|}$ with k positive, increasing on $(-\infty, 0)$ and decreasing on $(0, +\infty)$ (see e.g. Cont and Tankov (2003, prop. 15.3)).

Finally, by observing equation (6) one can deduce that, because $a < 1$, the strip of analyticity of the characteristic function of Lévy-OU processes is the same as that of the Lévy part.

In the following subsections we show the applications of these equations to determine the characteristic function and the Lévy measure of these processes when the Lévy part are of tempered stable (TS) and of normal tempered stable (NTS) type.

2.1. The OU-TS and the TS-OU processes

In this section we discuss both the OU-TS and the TS-OU processes, defining their characteristic functions and Lévy measures. These types of processes have been extensively studied by Sabino and Cufaro Petroni (2022) and Qu et al. (2021) in the infinite activity with finite variation setting, by Sabino (2022b) in the finite activity setting and by Zhang (2011); Kawai and Masuda (2011, 2012) has considered TS-OU processes both of finite and infinite variation.

Let's start by recalling the definition of TS processes through their characteristic exponent and their Lévy measure. For their complete definitions we refer the reader to Cont and Tankov (2003, ch. 4.5). A TS process is a pure jump Lévy process (i.e. a Lévy process with no diffusion part), with drift γ_{TS} and the following Lévy measure (Cont and Tankov, 2003, eq. 4.27)

$$\nu_L(x) = c_p \frac{e^{-\beta_p x}}{x^{1+\alpha_p}} \mathbb{1}_{x \geq 0} + c_n \frac{e^{-\beta_n |x|}}{|x|^{1+\alpha_n}} \mathbb{1}_{x<0} \tag{7}$$

with $c_p, c_n \geq 0$, $\beta_p, \beta_n > 0$ and $\alpha_p, \alpha_n < 2$.
This class of Lévy processes has been proposed by Koponen (1995) and introduced in finance by Boyarchenko and Levendorskiĭ (2000) and named KoBoL process; later for a particular choice of parameters the process has been considered in Carr et al. (2002) and called the CGMY model, after the authors of this article.
The stability parameters α_p, α_n determine the behavior of small jumps: for $\alpha_p, \alpha_n < 0$ it is a process of finite activity, for $\alpha_p, \alpha_n < 1$ of finite variation (see e.g. Cont and Tankov (2003, below eq. 4.3)). Notice that in the finite activity case the TS process is not self-decomposable, therefore for TS-OU processes $\alpha_p, \alpha_n < 0$ is not a valid parameter choice.
The characteristic exponent of a TS process (for the general case α_p, α_n different from 0 or 1) is (Cont and Tankov, 2003, prop. 4.2)

$$\begin{aligned} \varphi_L(u) =& iu\gamma_L + c_p \Gamma(-\alpha_p) \beta_p^{\alpha_p} \left[\left(1 - \frac{iu}{\beta_p} \right)^{\alpha_p} - 1 + iu \frac{\alpha_p}{\beta_p} \right] \\ &+ c_n \Gamma(-\alpha_n) \beta_n^{\alpha_n} \left[\left(1 + \frac{iu}{\beta_n} \right)^{\alpha_n} - 1 - iu \frac{\alpha_n}{\beta_n} \right] \end{aligned} \tag{8}$$

where

$$\gamma_L = \gamma_{TS} + \int_{\mathbb{R}\setminus[-1,1]} x\nu_L(x)dx$$

Lemma 1. The strip of analyticity of the characteristic function of a TS process is $\Im(u) \in (-\beta_p, \beta_n)$.

Proof. This property holds because the bounds of the analyticity strip of the characteristic function are on the imaginary axis, due to theorem 3.1 in Lukacs (1972). Thanks to the Lévy-Khintcine formula for pure jump processes, we have that the log-chf is

$$log\mathbb{E}[e^{iuX_t}] = t\left\{i\gamma u + \int_{\mathbb{R}}(e^{iux} - 1 - iux\mathbb{1}_{x\leq 1})\nu_L(x)dx\right\}$$

Due to the form of the Lévy measure ν_L in (7) we can split the integral in two parts. The positive part has the integral

$$c_p\int_{\mathbb{R}}(e^{iux} - 1 - iux\mathbb{1}_{x\leq 1})\frac{e^{-\beta_p x}}{x^{1+\alpha_p}}dx$$

and we are interested in this integral on $u = i\xi$, i.e.

$$c_p\int_{\mathbb{R}}(e^{-\xi x} - 1 + \xi x\mathbb{1}_{x\leq 1})\frac{e^{-\beta_p x}}{x^{1+\alpha_p}}dx$$

The above integral could be not defined for the integrand close to either 0 or $+\infty$. Close to 0 it is always well defined because a Lévy measure is s.t. $\int_0^1 x^2\nu_L(x)dx < +\infty$. Close to $+\infty$ one should check that $e^{-(\xi+\beta_p)x}$ goes to zero, i.e. the integral is well defined $\forall\xi > -\beta_p$. Mutatis mutandis, considering the negative part in equation (7) one obtains that $\xi < \beta_n$. □

OU-TS processes Now, having recalled formula (7) and (8) we can pair them with (3) and (4) to obtain the Lévy measure and the characteristic function of OU-TS processes.
The Lévy measure $\nu_Z(x,t)$ of $Z(t)$ for OU-TS processes is

$$\nu_Z(x,t) = \frac{1}{b|x|}\begin{cases} c_n\beta_n^{\alpha_n}(\Gamma_U(-\alpha_n, -\beta_n x) - \Gamma_U(-\alpha_n, -\frac{\beta_n}{a}x)) & x < 0 \\ c_p\beta_p^{\alpha_p}(\Gamma_U(-\alpha_p, \beta_p x) - \Gamma_U(-\alpha_p, \frac{\beta_p}{a}x)) & x > 0 \end{cases} \tag{9}$$

where $\Gamma_U(a,x)$ is the incomplete upper gamma function (Abramowitz et al., 1988, ch. 6.5). See point (b) in appendix A for the proof.

Lemma 2. The log characteristic function $\psi_Z(u,t)$ of $Z(t)$ for OU-TS processes is

$$\begin{aligned}\psi_Z(u,t) =& iu\frac{1-e^{-bt}}{b}\gamma_L - \frac{c_p\beta_p^{\alpha_p}\Gamma(1-\alpha_p)}{\alpha_p b}\left[I\left(u,\alpha_p,\beta_p,\frac{\beta_p}{a}\right) + \log a + iu\frac{\alpha_p}{\beta_p}(1-a)\right] \\ &- \frac{c_n\beta_n^{\alpha_n}\Gamma(1-\alpha_n)}{\alpha_n b}\left[I\left(-u,\alpha_n,\beta_n,\frac{\beta_n}{a}\right) + \log a - iu\frac{\alpha_n}{\beta_n}(1-a)\right]\end{aligned} \tag{10}$$

with

$$I(u,\alpha,\beta_1,\beta_2) = -\frac{1}{\alpha}\left[\left(\frac{u}{i\beta_2}\right)^{\alpha} {}_2F_1\left(-\alpha,-\alpha;1-\alpha;-\frac{i\beta_1}{u}\right) - \left(\frac{u}{i\beta_1}\right)^{\alpha} {}_2F_1\left(-\alpha,-\alpha;1-\alpha;-\frac{i\beta_1}{u}\right)\right] \tag{11}$$

where ${}_2F_1(a,b;c;x)$ is the Gaussian hypergeometric function (Abramowitz et al., 1988, ch. 15). The Gaussian hypergeometric function is a special case of the generalized hypergeometric function

$${}_pF_q(a_1,...,a_p;b_1,...,b_q;z) = \sum_{n=0}^{\infty} \frac{(a_1)_n...(a_p)_n}{(b_1)_n...(b_q)_n}\frac{z^n}{n!}, \qquad \begin{aligned}(a)_0 &= 1\\ (a)_{n\geq 1} &= a(a+1)...(a+n-1)\end{aligned}$$

with $p=2$ and $q=1$, which in our special case ($a_1=a_2=a$, $b_1=a+1$) reduces to

$${}_2F_1(a,a;1+a;z) = \sum_{n=0}^{+\infty}\frac{(a)_n(a)_n}{(1+a)_n}\frac{z^n}{n!} = a\sum_{n=0}^{+\infty}\frac{(a)_n}{a+n}\frac{z^n}{n!} \tag{12}$$

$(a)_n$ being the Pochhammer symbol (Abramowitz et al., 1988, p. 256, eq. 6.1.22).

Proof. To reduce the amount of calculations we write the proof for a positive TS. The proof for the bilateral case follows the same calculations.
From equations (4) and (8)

$$\begin{aligned}\psi_Z(u,t) &= \int_0^t \varphi_L(ue^{-bs})ds\\ &= \int_0^t iue^{-bs}\gamma_L + c_p\Gamma(-\alpha_p)\beta_p^{\alpha_p}\left[(1-\frac{iue^{-bs}}{\beta_p})^{\alpha_p} - 1 + iue^{-bs}\frac{\alpha_p}{\beta_p}\right]ds\\ &= iu\frac{1-e^{-bt}}{b}\gamma_L + \frac{c_p\Gamma(-\alpha_p)}{b}\int_1^{1/a}\frac{(\beta_p w - iu)^{\alpha_p}}{w^{\alpha_p+1}} - \frac{(\beta_p)^{\alpha_p}}{w} + iu\alpha_p\beta_p^{\alpha_p-1}\frac{1}{w^2}dw\\ &= iu\frac{1-e^{-bt}}{b}\gamma_L + \frac{c_p\beta_p^{\alpha_p}\Gamma(-\alpha_p)}{b}\left[\int_{\beta_p}^{\beta_p/a} z^{-1-\alpha_p}(z-iu)^{\alpha_p}dz + \log a + iu\frac{\alpha_p}{\beta_p}(1-a)\right]\end{aligned}$$

where in the penultimate step we have used the change of variables $e^{bs}=w$, and in the last step $\beta_p w = z$.
To evaluate the remaining integral, observe that in the special case $c=a+1$ the derivative of the Gaussian hypergeometric function is

$$\frac{d}{dx}{}_2F_1(a,b;a+1;x) = \frac{a((1-x)^{-b} - {}_2F_1(a,b;a+1;x))}{x} \tag{13}$$

which can be checked by writing the series representation of the hypergeometric function and of the function $(1-x)^{-b}$. For more insights on the calculations see proof (c) in appendix A.
Then with some algebra we get

$$\frac{d}{dz}\left(z^{-\alpha}{}_2F_1(-\alpha,-\alpha;1-\alpha;-\frac{iz}{u})\right) = -\alpha\left(\frac{i}{u}\right)^{\alpha} z^{-1-\alpha}(z-iu)^{\alpha}$$

therefore we can write

$$\begin{aligned} I(u,\alpha,\beta_1,\beta_2) &= \int_{\beta_1}^{\beta_2} z^{-1-\alpha}(z-iu)^{\alpha}dz \\ &= -\frac{1}{\alpha}\left[\left(\frac{u}{i\beta_2}\right)^{\alpha} {}_2F_1\left(-\alpha,-\alpha;1-\alpha;-\frac{i\beta_1}{u}\right)\right. \\ &\left. -\left(\frac{u}{i\beta_1}\right)^{\alpha} {}_2F_1\left(-\alpha,-\alpha;1-\alpha;-\frac{i\beta_1}{u}\right)\right] \end{aligned}$$

and this concludes the proof. □

The proof reminds the one in Sabino (2022b). Our contribution is to correct some typos and to extend it to infinite variation case (i.e. when either α_p or $\alpha_n \in (1,2)$).

TS-OU processes For TS-OU processes both the characteristic function and the Lévy measure are readily obtained by substituting formulas (7) and (8) in equations (5) and (6), since $\bar{X} \sim TS$. We remind that $\alpha_p, \alpha_n < 0$ is not a valid parameter choice for TS-OU because in this case the TS process is not self-decomposable.

2.2. The OU-NTS and the NTS-OU processes

In this section we discuss both the OU-NTS and the NTS-OU processes, defining their characteristic functions and Lévy measures. These types of processes have been studied by Sabino (2022a, 2023) who focused on symmetric OU-NTS and NTS-OU, by Sabino (2020) and Sabino and Cufaro Petroni (2021b) for the sub-class of Variance Gamma processes, and were already addressed in Barndorff-Nielsen et al. (1998) for the case of Normal Inverse Gaussian-OU.

Let's start by recalling the definition of NTS processes through their characteristic exponent and their Lévy measure. For their complete definitions we refer the reader to Cont and Tankov (2003, ch. 4.4.3).
A NTS process is a Lévy process obtained by time changing an independent Brownian motion by a tempered stable subordinator (TS_p), which is a TS process with $c_n = 0$ and $0 \leq \alpha_p < 1$, see Cont and Tankov (2003, ch. 4.4.2), defined as follows

$$\begin{aligned} L(t) &= \theta S(t) + \sigma\sqrt{S(t)}G \\ S(t) &\sim TS_p(c,\alpha,\beta) \qquad G \sim \mathcal{N}(0,1) \end{aligned} \tag{14}$$

Before proceeding let's introduce a different parametrization for TS_p processes, where we impose that $\mathbb{E}[S(t)] = t$ and $Var(S(1)) = \kappa$. In this way rather than having three parameters (c,α,β) we only have two (α,κ) which are related to the first as (Cont and Tankov, 2003, eq. 4.19)

$$\beta = \frac{(1-\alpha)}{\kappa} \qquad c = \frac{1}{\Gamma(1-\alpha)}\left(\frac{1-\alpha}{\kappa}\right)^{1-\alpha} \tag{15}$$

Within the scope of our study we restrict our analysis to symmetric NTS processes, by setting the parameter $\theta = 0$. With $\theta = 0$ and with the new parametrization of equation (15), the Lévy measure $\nu_L(x)$ of a NTS process is (Cont and Tankov, 2003, eq. 4.24)

$$\nu_L(x) = \frac{C(\alpha, \kappa, \sigma)}{|x|^{\alpha+1/2}} K_{\alpha+1/2}\left(A(\alpha, \kappa, \sigma)|x|\right) \tag{16}$$

with

$$\begin{aligned} C(\alpha, \kappa, \sigma) &= \frac{2^{\alpha/2+5/4}}{\sqrt{2\pi}\Gamma(1-\alpha)} \sigma^{\alpha-1/2} \left(\frac{1-\alpha}{\kappa}\right)^{5/4-\alpha/2} \\ A(\alpha, \kappa, \sigma) &= \sqrt{\frac{2(1-\alpha)}{\kappa\sigma^2}} \end{aligned} \tag{17}$$

and $K_\nu(x)$ is the modified Bessel function of the second kind (Abramowitz et al., 1988, ch. 9.6).
The characteristic exponent $\varphi_L(u)$ of a NTS process in the general case $\alpha > 0$ is (Cont and Tankov, 2003, eq. 4.22)

$$\varphi_L(u) = \frac{1-\alpha}{\kappa\alpha} \left\{ 1 - \left(1 + \frac{\kappa(u^2\sigma^2/2)}{1-\alpha}\right)^{\alpha} \right\} \tag{18}$$

OU-NTS processes Now, like in the previous section, having recalled formulas (16) and (18) we can pair them with (3) and (4) to obtain the characteristic function and the Lévy measure of OU-NTS processes.
The Lévy measure of $Z(t)$ for OU-NTS processes is

$$\nu_Z(x,t) = \frac{C(\alpha, \kappa, \sigma)}{b|x|} \left[J\left(\frac{|x|}{a}, \alpha + \frac{1}{2}, A(\alpha, \kappa, \sigma)\right) - J\left(|x|, \alpha + \frac{1}{2}, A(\alpha, \kappa, \sigma)\right) \right] \tag{19}$$

with

$$\begin{aligned} J(x, \nu, a) = \int y^{-\nu} K_\nu(ay) dy = \frac{\pi x}{2sin(\pi\nu)} \Bigg[& \frac{2^\nu a^{-\nu} x^{-2\nu}}{\Gamma(-\nu+1)(-2\nu+1)} {}_1F_2\left(-\nu + \frac{1}{2}; -\nu+1, -\nu+\frac{3}{2}; \left(\frac{ax}{2}\right)^2\right) \\ & - \frac{a^\nu}{2^\nu\Gamma(\nu+1)} {}_1F_2\left(\frac{1}{2}; \nu+1, \frac{3}{2}; \left(\frac{ax}{2}\right)^2\right) \Bigg] + c \end{aligned} \tag{20}$$

where ${}_1F_2(a; b, c; x)$ is the generalized hypergeometric function.
The correctness of equation (20) can be checked by writing $K_\nu(x)$ through its series representation and then switching the series and integration order. For more insights on the calculations see proof (d) in appendix A.

The log-chf $\psi_Z(u,t)$ of $Z(t)$ for OU-NTS processes is (Sabino, 2023, p. 106)

$$\psi_Z(u,t) = -\frac{c\beta^\alpha\Gamma(1-\alpha)}{2\alpha b} \left[I\left(\frac{i\sigma^2u^2}{2}, \alpha, \beta, \frac{\beta}{\omega}\right) + \log\omega \right], \qquad \omega = a^2 = e^{-2bt} \tag{21}$$

with the term indicated by $I(u, \alpha, \beta_1, \beta_2)$ being the same of equation (11).

NTS-OU processes For NTS-OU processes both the characteristic function and the Lévy measure are readily obtained by substituting the formulas (16) and (18) in equations (5) and (6), because $\bar{X} \sim NTS$.

3. Simulation of OU-Lévy and Lévy-OU processes through a fast MC scheme

In this section we illustrate a fast way to simulate OU-Lévy and Lévy-OU processes extending the algorithm of Azzone and Baviera (2021), introduced for additive processes. This algorithm allows to sample the trajectories in a simpler, faster and more general way with respect to the exact simulation algorithms found in the literature.

The sequential generation of the process $X(t)$ on a time grid $t_0, t_1, ..., t_M$ consists in implementing recursively the following step with initial condition $X(t_0) = X_0$

$$X(t_i) = a_i X(t_{i-1}) + Z_i, \qquad a_i = e^{-b\Delta t_i}, \qquad \Delta t_i = t_i - t_{i-1}, \qquad i = 1, ..., M \tag{22}$$

It is important to emphasize that $Z_i = Z(\Delta t_i)$

$$Z_i = \int_{t_{i-1}}^{t_i} e^{-b(t_i - s)} dL(s) = \int_0^{\Delta t_i} e^{-b(\Delta t_i - s)} dL(s) = Z(\Delta t_i) \tag{23}$$

due to the independence and stationarity properties of the increments of Lévy processes, i.e. it depends only on the time interval Δt_i and not from t_i and t_{i-1} separately. This implies that if we use a time grid which is uniformly spaced, then the first two steps of the fast MC algorithm which we introduce in the next paragraph can be performed only once, leading to even faster sampling times.

The fast MC method can be summarized in the following steps:

1. Obtain the cumulative distribution function (CDF) of the increment for the time interval Δt by inverting the characteristic function of the increment through the formula (Lee, 2004, th. 5.1)

$$F(x) = R_{\tilde{a}} - \frac{e^{-\tilde{a}x}}{\pi} \int_0^{+\infty} Re\left[\frac{e^{-iux} e^{\psi_Z(u - i\tilde{a}, \Delta t)}}{iu + \tilde{a}} du \right]$$

$$R_{\tilde{a}} = \begin{cases} 1 & 0 < \tilde{a} < p_t^+ + 1 \\ \frac{1}{2} & \tilde{a} = 0 \\ 0 & -p_t^- < \tilde{a} < 0 \end{cases} \tag{24}$$

   Where $F(x)$ is the CDF of the increment between time s and time t, $\tilde{a}$ is the shift in the integration path used to avoid the singularity in the origin, and similarly to Lee (2004) we indicate as $-(p_t^+ + 1)$ and p_t^- the lower and upper bounds of analyticity of the characteristic function $e^{\psi_Z(u, \Delta t)}$, i.e. the characteristic function is analytic when $\Im(u) \in (-(p_t^+ + 1), p_t^-)$.

Azzone and Baviera (2021) suggests to take $\tilde{a} = \frac{p_t^+ + 1}{2}$ or $\tilde{a} = -\frac{p_t^-}{2}$ depending on which has the largest modulus. This means taking the shift on the imaginary axis which is farthest from both the origin and either boundary of the strip of analyticity of the characteristic function.
The integral of equation (24) can be evaluated numerically with the fast Fourier transform (FFT) method (Cooley and Tukey, 1965), in order to get the CDF evaluated on a grid of points $[x_1, x_2, ..., x_N]$.
To implement the FFT one choice is to select M, such that $N = 2^M$ is the number of points in the grid, and h, which is the discretization step used when evaluating the integral for each point x. Then the step γ in the grid of points for which we evaluate the CDF must satisfy

$$\gamma h = \frac{2\pi}{N}$$

Azzone and Baviera (2021, p. 7) suggests a method to select the optimal h such that the truncation error and the discretization error coming from the FFT are of comparable magnitude. The optimal step is

$$h(N, \tilde{b}, \tilde{\omega}) = \left(\frac{\pi p_t^*}{\tilde{b}} \frac{1}{N^{\tilde{\omega}}} \right)^{1/(\tilde{\omega}+1)}, \qquad p_t^* = max(p_t^+ + 1, p_t^-) \tag{25}$$

where $\tilde{b}$ and $\tilde{\omega}$ are the parameters from Assumption 1 in Azzone and Baviera (2021, p. 6). Indeed h is the optimal discretization step only if the following assumption holds (and for h small enough)

Assumption 1. (Azzone and Baviera, 2021, p. 6) $\forall t > s \geq 0$ there exists $B > 0$, $\tilde{b} > 0$ and $\tilde{\omega}$ such that, for sufficiently large $|u|$, the following bound for the absolute value of the characteristic function holds

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

In order to determine the parameters $\tilde{b}$ and $\tilde{\omega}$ we implement the following heuristic estimation

(a) Create a grid of points u such that $|u|$ is sufficiently large.
(b) Evaluate $|\phi_{s,t}(u - i\tilde{a})|$ where $\tilde{a}$ is the shift which we use in equation (24).
(c) Take $-\log(|\phi_{s,t}(u - i\tilde{a})|)$ and perform a linear regression against $|u|^{\tilde{\omega}}$ for different values of $\tilde{\omega}$. Take $\tilde{b}$ as the estimated slope of the regression.
(d) Keep the parameters from the best fitting regression.

We remark that this is just an heuristic way to proceed and it does not prove exactly assumption 1, but nonetheless the numerical results obtained suggest that it is a good approximation to obtain the parameters for equation (25). Moreover we always constrain the value of $h(N, \tilde{b}, \tilde{\omega})$ to be less or equal to 0.1.

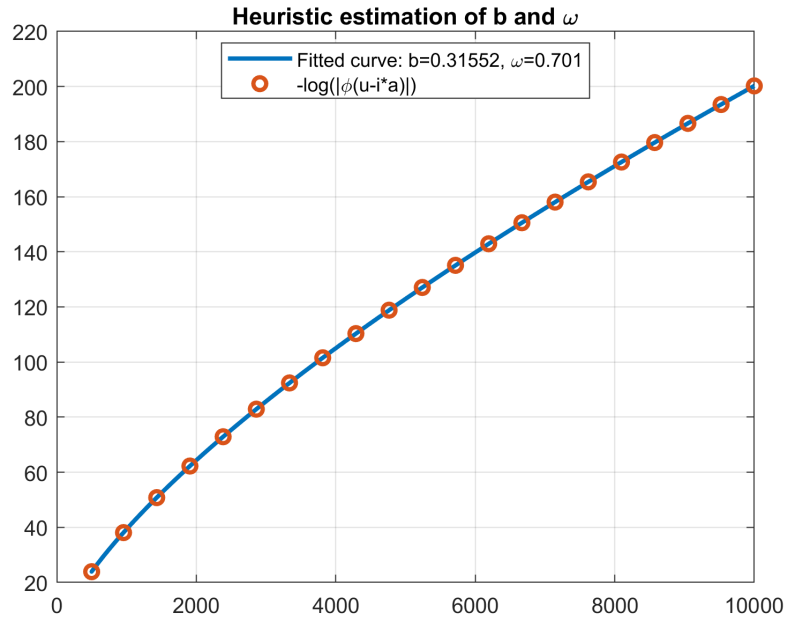

Figure 1: Example illustration of the fitting performed for the heuristic estimation of $\tilde{b}$ and $\tilde{\omega}$. The circles represent the values of $-\log(|\phi_{s,t}(u - i\tilde{a})|)$ while the continuous line represents the function $\tilde{b}|u|^{\tilde{\omega}} + c$. The figure comes from a OU-TS process with $\alpha_p = \alpha_n = 0.7$ and the same parameters which we use in section 4.

2. Once obtained the CDF on a grid of points, remove numerical artifacts by "cleaning" it. Select the largest set of points for which the CDF is monotonous, increasing and within $[0, 1]$.

3. Sample the paths of $Z(\Delta t)$ by inverting the CDF: for each path extract a random standard uniform and invert the CDF to obtain the sample increment, setting $Z_i = F^{-1}(U)$ with $U \sim \mathcal{U}(0, 1)$.
   In practice what we do is interpolate the points where we have evaluated the CDF as a function of F(x) and then project the extracted uniform random variable (r.v.). We use spline interpolation because it has faster error scaling as γ decreases, at the cost of a slight increase in computational time (Quarteroni et al., 2010, ch. 7), but linear interpolation is a valid alternative.
   Before "inverting" the CDF we extend it by adding exponential tails to it, as suggested by Azzone and Baviera (2021, p. 13), in order to account for the uniform samples which fall above the last CDF value or below the first. Taking the last two points for which the CDF has been evaluated and the first two we can interpolate them with exponential functions in order to have a CDF grid which, in the limit, has range equal to $[0, 1]$. Example code for this last step can be found in appendix A.

4. Numerical results

In this section we analyze the performance of the algorithm from the point of view of both accuracy and speed.
First, in section 4.1 we show the method's accuracy with respect to the first four true cumulants of the process (mean, variance, skewness, kurtosis) obtainable by closed formula, and compare them with the ones obtained with exact MC algorithms found in the literature.
Then in section 4.2 we value the performance of the algorithm by reporting the maximum error obtained by pricing a grid of call options with the fast MC with respect to the prices obtained through Lewis formula (Lewis, 2001).
Finally, in subsection 4.3 we compare the computational times needed to generate these results with the fast MC method and the exact MC.
All the numerical results have been obtained using MATLAB R2021b on Intel Core i7-8550U CPU @ 1.80 GHz, 16 GB RAM, Windows 10 64-bit.

4.1. Accuracy with respect to true cumulants

In order to assess the performance of the fast MC method, we first measure the relative error of the estimated cumulants (mean, variance, skewness and kurtosis) with respect to the true cumulants of the process and compare them with the errors of the exact algorithms found in the literature.
The exact algorithms that we consider as benchmark are

- For OU-TS and TS-OU of infinite activity and finite variation ($0 < \alpha_p, \alpha_n < 1$): algorithms 1 and 2 from Sabino and Cufaro Petroni (2022), which are improved versions of the algorithms proposed by Kawai and Masuda (2011) and Qu et al. (2021).
- For OU-TS of finite activity ($\alpha_p, \alpha_n < 0$) algorithm 1 from Sabino (2022b).
- For OU-NTS algorithm 3.1 from Sabino (2023).
- For NTS-OU algorithm 1 from Sabino (2022a).

Up to our knowledge no exact algorithm is found in the literature for OU-TS/TS-OU processes of infinite variation, while we remind that the TS-OU finite activity process does not exist because in the finite activity setting the TS process is not self-decomposable (see e.g. Cont and Tankov (2003, prop. 15.3)).

The performance of the algorithms is evaluated in terms of the relative error of the estimated cumulants with respect to the true cumulants of the process, as of the following formula:

$$\text{relative error} = \frac{\text{true value} - \text{estimated value}}{\text{true value}} \tag{26}$$

The true values for the cumulants are obtained from the standard procedure of differentiating the characteristic function. We report the formulas for the OU-TS and TS-OU cumulants in appendix A at point (e). The formulas for the OU-NTS and NTS-OU cumulants follow from the same reasoning and can be found respectively in Sabino (2022a, p. 81) and Sabino (2023, p. 109)

Table 3 and 4 show the true values of the cumulants and the relative errors for the exact MC and fast MC, for a set of different values of the stability parameter α. The remaining parameters that we use for each process are reported in table 1 and 2.

OU-TS and TS-OU parameters

	γ_L	c_p	c_n	β_p	β_n	b	X_0
OU-TS f.a.	1	0.8	0.8	1.4	1.5	0.5	0
OU-TS i.a. TS-OU i.a.	1	0.8	0.8	1.4	1.5	10	0

Table 1: Parameters for OU-TS and TS-OU processes. In the first two rows, "f.a." stands for finite activity ($\alpha < 0$) and "i.a." for infinite activity ($\alpha > 0$).

OU-NTS and NTS-OU parameters

	σ	κ	b	X_0
OU-NTS	0.201	0.256	52.9	0
NTS-OU	0.201	0.256	1	0

Table 2: Parameters for OU-NTS and NTS-OU processes.

These parameters are taken equal to the ones from the reference articles cited above, with the only exception of γ_L which is not specified in the corresponding exact method article. For all choices of parameters the time step we use is $\Delta t = 1/12$, the number of simulations is 10^7 and we set the parameter $M = 15$ in the fast MC algorithm.
We present the results for TS-OU and OU-NTS. The other two cases (OU-TS and NTS-OU) present similar results in terms of accuracy. Extensive testing for different choices of parameters and time steps has been conducted obtaining similar performances for the fast MC.

TS-OU accuracy with respect to true cumulants

α	Mean			Variance			Skewness			Kurtosis		
	True	Exact MC	Fast MC	True	Exact MC	Fast MC	True	Exact MC	Fast MC	True	Exact MC	Fast MC
0.3	5.65	-0.03	0.02	6.29	0.02	0.08	0.78	-2.36	1.25	16.44	0.63	0.59
0.5	5.65	0.02	0.02	6.60	0.05	0.07	0.67	-2.58	1.33	14.09	-0.25	0.57
0.7	5.65	-0.04	0.03	7.20	0.00	0.07	0.59	1.90	-1.46	12.25	0.21	0.57
1.3	5.65	$\sim$	-0.02	13.00	$\sim$	0.01	0.42	$\sim$	-0.36	8.79	$\sim$	1.54
1.5	5.65	$\sim$	0.02	19.11	$\sim$	0.00	0.39	$\sim$	-1.02	8.14	$\sim$	2.54
1.7	5.65	$\sim$	0.03	43.74	$\sim$	0.00	0.37	$\sim$	-2.83	7.69	$\sim$	7.55

Table 3: Mean, variance, skewness and kurtosis of the TS-OU process and the relative errors obtained with the fast MC, compared to the errors of the exact MC algorithms, for different values of the stability parameter $\alpha = \alpha_p = \alpha_n$. The columns "True" contain the values of the true cumulants (multiplied by 10) while the columns "Exact MC" and "Fast MC" contain the relative error (multiplied by 100) with respect to the true cumulants. The exact values in the last three rows are missing because no exact algorithm exists for the infinite variation setting.

OU-NTS accuracy with respect to true cumulants

α	Variance			Kurtosis		
	True	Exact MC	Fast MC	True	Exact MC	Fast MC
0.1	38.18	-0.35	0.83	0.59	-0.23	7.19
0.3	38.18	-0.34	-0.10	0.59	0.02	-0.44
0.5	38.18	-0.20	0.12	0.59	-1.80	-0.03
0.7	38.18	0.02	-0.31	0.59	3.44	-5.00
0.9	38.18	0.01	0.02	0.59	3.40	4.89

Table 4: Variance and kurtosis errors w.r.t the true cumulants obtained with the fast MC for the OU-NTS process, compared to the errors of the exact MC, for different values of the stability parameter α. The mean and skewness errors are omitted because the true cumulant value is 0. The values of the true cumulants are multiplied by 10^5 while the values under the columns "Exact MC" and "Fast MC" indicate the relative error (multiplied by 100) with respect to the true cumulants.

By observing the values in table 3 and 4 we can say that the relative errors of the fast MC are small and similar to those of the exact methods. In particular the errors on mean and variance are lower than a percentage point for all cases of the fast MC algorithm. We can observe how this is true also for the TS-OU infinite variation cases where our algorithm achieves some of its best results, and where no exact algorithm alternatives exist.

4.2. Accuracy on European option prices

After having observed the relative errors on the true cumulants, another way to assess the performance of the fast MC algorithm is to use the simulated paths for the pricing of European call options and compare it to the pricing obtained with the Lewis formula (Lewis, 2001).

Let's assume that the price process can be modeled as $P(t) = e^{X(t)}$ where $X(t)$ is a Lévy-OU or OU-Lévy process. Consider a grid of 30 European call options with maturity $t = 1/12$ and moneyness in the range $\sqrt{t}(-0.2, 0.2)$. For each of the four process types we consider, Table 5 and 6 report the maximum absolute difference between the prices for the grid of call options obtained by simulating $P(t)$ with the fast MC algorithm w.r.t. the prices obtained with Lewis formula, across different values of the stability parameters $\alpha_p, \alpha_n, \alpha$.

OU-TS and TS-OU accuracy on European options

OU-TS	
α	Max diff [bp]
-1.5	0.06
-0.5	0.41
0.3	0.04
0.7	0.02
1.3	0.25
1.7	0.48

TS-OU	
α	Max diff [bp]
0.3	0.10
0.5	0.13
0.7	0.17
1.3	0.18
1.5	0.25
1.7	0.46

Table 5: Maximum difference between the prices obtained with the fast MC algorithm and the prices obtained with Lewis formula for a grid of 30 European call options with maturity $t = 1/12$ and moneyness in the range $\sqrt{t}(-0.2, 0.2)$, with the price process modeled via OU-TS and TS-OU processes, for different values of the stability parameters $\alpha = \alpha_p = \alpha_n$. The values in the "Max diff" columns are in basis points (bp).

OU-NTS and NTS-OU accuracy on European options

α	OU-NTS Max diff [bp]	NTS-OU Max diff [bp]
0.1	0.06	0.35
0.3	0.02	0.27
0.5	0.07	0.10
0.7	0.09	0.29
0.9	0.03	0.21

Table 6: Maximum difference between the prices obtained with the fast MC algorithm and the prices obtained with Lewis formula for a grid of 30 European call options with maturity $t = 1/12$ and moneyness in the range $\sqrt{t}(-0.2, 0.2)$, with the price process modeled via OU-NTS and NTS-OU processes, for different values of the stability parameter α. The values in the "Max diff" columns are in basis points (bp).

From tables 5 and 6 we can observe how for all processes and for all parameter choices the fast MC algorithm can give prices within a basis point of the Lewis prices, which is an indication that the level of accuracy given by the fast MC algorithm is sufficient to be applied to option pricing.
In the next section we analyze the computational time needed to generate these results and compare it with the time needed for exact MC algorithms.

4.3. Computational time and comparison with the exact methods

The premise of the paper is that the fast MC algorithm is not only as accurate as the exact methods but most importantly much faster. In this section we compare the computational times needed to generate the paths used in the previous sections and compare the times of the fast MC algorithms with those of the exact MC algorithms and also against the times obtained by simulating via Asmussen and Rosiński (2001)'s Gaussian approximation (GA) method as an additional benchmark.
In particular we show that the computational times needed to generate OU-Lévy and Lévy-OU paths via the fast MC algorithm are only a few multiples of those needed to generate a Brownian motion. On our machine it takes approximately 0.09 seconds to generate 10^7 samples from a normal random variable.

To simulate with Gaussian approximation we truncate the Lévy measure of the processes around the origin, calculate the area underneath to obtain the Poisson rate, integrate numerically the associated probability density function (pdf) to obtain its CDF, and finally sample the jumps by generating random uniform r.v. and inverting the CDF, using an interpolation procedure analogous to the last step of the fast MC algorithm. We choose this method rather than the approach suggested in Azzone and Baviera (2021) of sampling from the truncated Lévy measure via the exact Ziggurat method of Marsaglia and Tsang (2000) because the Ziggurat method requires the pdf from which one wants to sample to be invertible and also to know a backup method to sample from the "tail" area, both conditions which are not satisfied in our case. Even by making some modifications to the algorithm, such as

inverting the pdf numerically or using rectangular divisions of the pdf which do not share equal area, it proves to be slower than the numerical integration method we use, and less precise.
We remark that the numerical integration method we use for Gaussian approximation is not the most accurate, and better methods could be devised by studying the Lévy measure of each process, but this goes beyond our scope and for the purpose of obtaining an additional benchmark for computational times we consider the approach adequate.

Computational times of OU-TS and TS-OU

	OU-TS		
α_p, α_n	Exact MC	GA	Fast MC
-1.5	2.05	1.74	1.44
-0.5	1.73	1.81	1.42
0.3	5.87	2.19	0.47
0.7	7.35	2.90	0.44
1.3	$\sim$	5.01	0.41
1.7	$\sim$	8.62	0.47

	TS-OU		
α_p, α_n	Exact MC	GA	Fast MC
0.3	13.95	4.22	0.35
0.5	10.68	5.31	0.36
0.7	29.17	6.79	0.34
1.3	$\sim$	30.12	0.37
1.5	$\sim$	50.40	0.38
1.7	$\sim$	80.10	0.39

Table 7: Computational times in seconds needed to generate $N = 10^7$ simulations of the OU-TS and TS-OU processes with exact MC algorithms, Gaussian approximation and fast MC algorithm, for different values of the stability parameters α_p, α_n. The missing values in the "exact MC" columns are due to the fact that no exact algorithms exist for the infinite variation setting.

Computational times of OU-NTS and NTS-OU

	OU-NTS			NTS-OU		
α	Exact MC	GA	Fast MC	Exact MC	GA	Fast MC
0.1	38.06	5.94	0.70	5.62	2.03	0.61
0.3	30.46	5.76	0.62	4.79	2.09	0.56
0.5	30.73	5.32	0.63	2.82	2.00	0.56
0.7	90.33	6.16	0.65	3.33	2.03	0.57
0.9	110.53	173.97	0.59	2.77	1.90	0.56

Table 8: Computational times in seconds needed to generate $N = 10^7$ simulations of the OU-NTS and NTS-OU processes with exact MC algorithms, Gaussian approximation and fast MC algorithm, for different values of the stability parameter α.

From tables 7 and 8 we see how for every parameter choice the fast MC algorithm greatly outperforms both the exact MC algorithms and the Gaussian approximation. This can be explained by the different

structure of the fast MC algorithm with respect to the exact MC algorithms and the Gaussian Approximation. For example, for all the exact algorithms and for the GA there is a compound Poisson term, meaning that we have to simulate a Poisson r.v. for each simulation, and then each jump separately. Depending on the parameters of the process, if the Poisson rate is much greater than 1 then we need to simulate an average number of jumps superior to the number of simulations. Moreover we need to know how to simulate these jumps, which may not be trivial especially for exact MC algorithms where we must sample without approximations. When the r.v. do not have an invertible CDF this means needing to devise an ulterior algorithm, which in most cases requires acceptance-rejection schemes and multiple auxiliary r.v. generations.
Take for example algorithm 2 of Sabino and Cufaro Petroni (2022) for the exact simulation of OU-TS processes. For each path we need to generate the following random variables

- One TS r.v., whose sampling is possible by using a combination of the simple acceptance-rejection scheme described in the appendix of Qu et al. (2021) (see originally Chambers et al. (1976)), and the double rejection algorithm of Devroye (2009). Hofert (2011) provides a selection criteria to choose either one of the two algorithms depending on the parameters of the process. Nonetheless the implementation is computationally costly.
- One Poisson r.v, whose rate depends on the parameters of the process and on the time step.
- For each jump of the compound Poisson, one "mixing" r.v. through acceptance-rejection scheme and one gamma r.v.

Moreover, the acceptance rate of the acceptance-rejection schemes is not always close to 1, meaning that for a single sample many trials may be needed. All of this leads to higher computational times which are not uniform across all parameters, indeed for most cases they tend to increase for higher choices of $\alpha_p, \alpha_n, \alpha$.
On the other hand, the fast MC algorithm does not incur in any of these issues since for each path one single uniform r.v generation is needed, and the FFT step is not influenced by the choice of parameters, leading to more uniform sampling times which depend mainly on the choice of the parameter M. To generate all the results of these last sections we use $M = 15$, except for the OU-TS finite activity ($\alpha < 0$) where we set $M = 19$ due to the lower number of jumps. Once again we remark how this choice is sufficient to obtain accurate results with very fast times relatively to the exact methods.
One thing we ought to mention for the reproducibility of our results is that to compute the Gaussian hypergeometric function in the OU-TS characteristic function, see equations (10) and (11), we borrow the Python function `hyp2f1` from the library `scipy.special`, calling the function through MATLAB. We do this because the default MATLAB function `hypergeom` computes the generalized hypergeometric function, while for the special case of the Gaussian hypergeometric function there exist many optimizations which are implemented in Python's `hyp2f1` and that make the computation faster. For further details about the implementation of the `hyp2f1` function we refer the reader to `scipy.special`'s online documentation and source code.

5. Conclusions

In this paper, we propose a new way to generate the paths of OU-Lévy and Lévy-OU processes starting from their characteristic functions, by making use of the MC algorithm recently introduced for additive process Azzone and Baviera (2021), and we show that the algorithm is accurate, fast and general.
We discuss in detail the implementation of the algorithm and analyse its performance both in terms of accuracy and computational time, using as benchmark, when available, the exact MC algorithms present in the literature. In terms of accuracy, we compare the estimated cumulants against the true cumulants of each process, and European option prices against the closed formula of Lewis (2001). In terms of time, we show that computational times are low multiples of the times needed to generate a Brownian motion and are considerably lower with respect to the ones needed by the exact MC algorithm and the Gaussian approximation method of Asmussen and Rosiński (2001).
Moreover, the new MC technique can be used to simulate OU-TS with an infinite variation background driving Lévy process, where, up to our knowledge, no exact algorithm exists.
Therefore, this study shows that the proposed MC algorithm is accurate and significantly faster than the algorithms present in the literature, and it can be extended to a large class of Lévy-OU and OU-Lévy processes for which there are no exact simulation methods.

References

Abramowitz, M., Stegun, I. A., and Romer, R. H. (1988). Handbook of mathematical functions with formulas, graphs, and mathematical tables.

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). A fast Monte Carlo scheme for additive processes and option pricing. *arXiv preprint arXiv:2112.08291*.

Barndorff-Nielsen, O. E. (1997). Processes of normal inverse Gaussian type. *Finance and stochastics*, 2:41–68.

Barndorff-Nielsen, O. E., Jensen, J. L., and Sørensen, M. (1998). Some stationary processes in discrete and continuous time. *Advances in Applied Probability*, 30(4):989–1007.

Barndorff-Nielsen, O. E. and Shephard, N. (2001). Non-Gaussian Ornstein–Uhlenbeck-based models and some of their uses in financial economics. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):167–241.

Benth, F. E., Groth, M., and Kufakunesu, R. (2007a). Valuing Volatility and Variance Swaps for a Non-Gaussian Ornstein–Uhlenbeck Stochastic Volatility Model. *Applied Mathematical Finance*, 14(4):347–363.

Benth, F. E., Kallsen, J., and Meyer-Brandis, T. (2007b). A non-Gaussian Ornstein–Uhlenbeck process for electricity spot price modeling and derivatives pricing. *Applied Mathematical Finance*, 14(2):153–169.

Benth, F. E. and Pircalabu, A. (2018). A non-Gaussian Ornstein–Uhlenbeck model for pricing wind power futures. *Applied Mathematical Finance*, 25(1):36–65.

Bianchi, M. L. and Fabozzi, F. J. (2015). Investigating the performance of non-Gaussian stochastic intensity models in the calibration of credit default swap spreads. *Computational Economics*, 46:243–273.

Boyarchenko, S. I. and Levendorskiĭ, S. Z. (2000). Option pricing for truncated Lévy processes. *International journal of theoretical and applied finance*, 3(03):549–552.

Carr, P., Geman, H., Madan, D. B., and Yor, M. (2002). The fine structure of asset returns: An empirical investigation. *The Journal of Business*, 75(2):305–332.

Chambers, J. M., Mallows, C. L., and Stuck, B. (1976). A method for simulating stable random variables. *Journal of the american statistical association*, 71(354):340–344.

Cont, R. and Tankov, P. (2003). *Financial Modelling with Jump Processes*. Chapman & Hall/CRC.

Cooley, J. W. and Tukey, J. W. (1965). An algorithm for the machine calculation of complex Fourier series. *Mathematics of computation*, 19(90):297–301.

Devroye, L. (2009). Random variate generation for exponentially and polynomially tilted stable distributions. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 19(4):1–20.

Flury, B. D. (1990). Acceptance–rejection sampling made easy. *Siam Review*, 32(3):474–476.

Grabchak, M. (2020). On the simulation of general tempered stable Ornstein–Uhlenbeck processes. *Journal of Statistical Computation and Simulation*, 90(6):1057–1081.

Grabchak, M. and Sabino, P. (2023). Efficient simulation of p-tempered α-stable OU processes. *Statistics and Computing*, 33(1):4.

Hofert, M. (2011). Sampling exponentially tilted stable distributions. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 22(1):1–11.

Kawai, R. and Masuda, H. (2011). Exact discrete sampling of finite variation tempered stable Ornstein–Uhlenbeck processes. *Monte Carlo Methods and Applications*, 17(3):279–300.

Kawai, R. and Masuda, H. (2012). Infinite variation tempered stable Ornstein–Uhlenbeck processes with discrete observations. *Communications in Statistics-Simulation and Computation*, 41(1):125–139.

Koponen, I. (1995). Analytic approach to the problem of convergence of truncated Lévy flights towards the Gaussian stochastic process. *Physical Review E*, 52(1):1197.

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 at SSRN 282110*.

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

Madan, D. B. and Seneta, E. (1990). The variance gamma (VG) model for share market returns. *Journal of business*, pages 511–524.

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

Qu, Y., Dassios, A., and Zhao, H. (2021). Exact simulation of Ornstein–Uhlenbeck tempered stable processes. *Journal of Applied Probability*, 58(2):347–371.

Quarteroni, A., Sacco, R., and Saleri, F. (2010). *Numerical mathematics*, volume 37. Springer Science & Business Media.

Sabino, P. (2020). Exact simulation of variance gamma-related OU processes: application to the pricing of energy derivatives. *Applied Mathematical Finance*, 27(3):207–227.

Sabino, P. (2022a). Exact simulation of normal tempered stable processes of OU type with applications. *Statistics and Computing*, 32(5):81.

Sabino, P. (2022b). Pricing energy derivatives in markets driven by tempered stable and CGMY processes of Ornstein–Uhlenbeck type. *Risks*, 10(8):148.

Sabino, P. (2023). Normal tempered stable processes and the pricing of energy derivatives. *SIAM Journal on Financial Mathematics*, 14(1):99–126.

Sabino, P. and Cufaro Petroni, N. (2021a). Fast pricing of energy derivatives with mean-reverting jump-diffusion processes. *Applied Mathematical Finance*, 28(1):1–22.

Sabino, P. and Cufaro Petroni, N. (2021b). Gamma-related Ornstein–Uhlenbeck processes and their simulation. *Journal of Statistical Computation and Simulation*, 91(6):1108–1133.

Sabino, P. and Cufaro Petroni, N. (2022). Fast simulation of tempered stable Ornstein–Uhlenbeck processes. *Computational Statistics*, 37(5):2517–2551.

Zhang, S. (2011). Exact simulation of tempered stable Ornstein–Uhlenbeck processes. *Journal of Statistical Computation and Simulation*, 81(11):1533–1544.

Notation and abbreviations

Notation

Symbol	Description
a	OU mean-reversion coefficient
$\tilde{a}$	shift in FFT integration path
α	stability parameter of TS_p
α_p	stability parameter for positive jumps of TS
α_n	stability parameter for negative jumps of TS
b	OU mean-reversion rate
$\tilde{b}$	coefficient of Assumption 1
β	decay parameter of TS_p
β_p	decay parameter for positive jumps of TS
β_n	decay parameter for negative jumps of TS
c	scale parameter of TS_p
c_p	scale parameter for positive jumps of TS
c_n	scale parameter for negative jumps of TS
$C(\alpha, \kappa, \sigma, \theta)$	constant in NTS Lévy measure
ϵ	Gaussian approximation truncation
${}_2F_1(a, b; c; x)$	Gauss hypergeometric function
${}_1F_2(a; b, c; x)$	generalized hypergeometric function
$F(x)$	CDF
γ	step between points for which the CDF is evaluated through FFT
γ_L	TS drift term in characteristic exponent
γ_{TS}	TS drift from Lévy triplet
$\Gamma(z)$	gamma function
$\Gamma_U(a, x)$	incomplete upper gamma function
G	standard normal r.v.

h	discretization step of FFT
$I(u, \alpha, \beta_1, \beta_2)$	hypergeometric term in OU-TS and OU-NTS log characteristic functions
$K_\nu(x)$	modified Bessel function of the second kind
κ	variance parameter of NTS
$L(t)$	Lévy process
N	number of grid points in FFT
$\mathcal{N}(0,1)$	standard normal law
$\nu_L(x)$	Lévy measure of $L(1)$
$\nu_{\bar{X}}(x)$	Lévy measure of $\bar{X}(t)$
$\nu_Z(x,t)$	Lévy measure of $Z(t)$
ω	square of OU mean-reversion coefficient
$\tilde{\omega}$	coefficient of Assumption 1
p_t^+	$-(p_t^+ + 1)$ is the lower bound of regularity strip of characteristic function
p_t^-	upper bound of regularity strip of characteristic function
p_t^*	largest bound of regularity in absolute value
$P(t)$	price process
$\phi(u)$	characteristic function
$\varphi_L(u)$	characteristic exponent of $L(t)$
$\varphi_{\bar{X}}(u)$	characteristic exponent of $\bar{X}(t)$
$\psi_Z(u,t)$	log characteristic function of $Z(t)$
$R_{\tilde{a}}$	first term of formula to obtain CDF
$S(t)$	subordinator process of NTS
σ	volatility parameter of NTS
θ	drift parameter of NTS
U	standard normal r.v
$\mathcal{U}(0,1)$	standard normal law
$X(t)$	OU process
$\bar{X}(t)$	OU process (of which we know the self-decomposable distribution)

$Z(t)$	integral term in OU solution

Abbreviations

Shorthand	Description
bp	basis points
BDLP	background driving Lévy process
CDF	cumulative distribution function
f.a.	finite activity
FFT	fast Fourier transform
GA	Gaussian approximation
i.a.	infinite activity
log-chf	log characteristic function
MC	Monte Carlo
NTS	normal tempered stable
NTS-OU	normal tempered stable Ornstein-Uhlenbeck
OU	Ornstein-Uhlenbeck
OU-NTS	Ornstein-Uhlenbeck normal tempered stable
OU-TS	Ornstein-Uhlenbeck tempered stable
pdf	probability density function
r.v.	random variable
TS	tempered stable (bilateral)
TS_p	tempered stable subordinator (unilateral positive)
TS-OU	tempered stable Ornstein-Uhlenbeck
w.r.t.	with respect to

A. Appendix

(a) A simplified version of the MATLAB implementation of the final inversion step of the fast MC algorithm is

```
function [paths] = fastMClaststep(x_grid,cdf_grid,num_simulations)
    % Simplified final step of the fast MC algorithm
    %
    % INPUTS
    % x_grid = grid of points for which the CDF has been evaluated, after cleaning
    % cdf_grid = values of the cdf for the points in x_grid
    % num_simulations = number of paths
    %
    % OUTPUT
    % paths = trajectories sampled from cdf_grid

    % Extract standard normal r.v
    u = rand(num_simulations,1);

    % Separate u
    tail_indexes_right = u > cdf_grid(end);
    tail_indexes_left = u < cdf_grid(1);
    inside_indexes = logical(1 - tail_indexes_left - tail_indexes_right);


    % Interpolation (spline)
    u_inside = u(inside_indexes);
    interp_method = "spline";
    paths_inside = interp1(cdf_grid,x_grid,u_inside,interp_method);
    paths = paths_inside;

    % Left tail y=a*exp(b*x)
    if (any(tail_indexes_left))
        b_left = (log(cdf_grid(2)) - log(cdf_grid(1)))/(x_grid(2) - x_grid(1));
        a_left = cdf_grid(1)*exp(-b_left*x_grid(1));
        paths_left_tail = (log(u(tail_indexes_left)) - log(a_left))/b_left;
        paths = [paths;paths_left_tail];
    end

    % Right tail y=1-a*exp(-b*x);
    if (any(tail_indexes_right))
        b_right = (log(1-cdf_grid(end)) - log(1-cdf_grid(end-1)))/(x_grid(end-1)-
            x_grid(end));
        a_right = (1-cdf_grid(end-1))*exp(b_right*x_grid(end-1));
        paths_right_tail = (log(1-u(tail_indexes_right)) - log(a_right))/(-b_right);
        paths = [paths;paths_right_tail];
    end
end
```

(b) Proof of equation (9) for the Lévy measure of OU-TS processes.

Proof. From equations (3) and (7)

$$\begin{aligned}
\nu_Z(x,t) &= \frac{1}{b|x|}\begin{cases} \int_{x/a}^{x} c_n \frac{e^{-\beta_n|y|}}{|y|^{1+\alpha_n}}dy & x<0 \\ \int_{x}^{x/a} c_p \frac{e^{-\beta_p y}}{y^{1+\alpha_p}}dy & x>0 \end{cases} \\
&= \frac{1}{b|x|}\begin{cases} c_n\beta_n^{\alpha_n}\int_{\beta_n x/a}^{\beta_n x} |s|^{-\alpha_n-1}e^{-|s|}ds & x<0 \\ c_p\beta_p^{\alpha_p}\int_{\beta_p x}^{\beta_p x/a} s^{-\alpha_p-1}e^{-s}ds & x>0 \end{cases} \\
&= \frac{1}{b|x|}\begin{cases} c_n\beta_n^{\alpha_n}(\Gamma_U(-\alpha_n,-\beta_n x)-\Gamma_U(-\alpha_n,-\frac{\beta_n}{a}x)) & x<0 \\ c_p\beta_p^{\alpha_p}(\Gamma_U(-\alpha_p,\beta_p x)-\Gamma_U(-\alpha_p,\frac{\beta_p}{a}x)) & x>0 \end{cases}
\end{aligned}$$

where in the second step we use the change of variables $\beta_n y = s$ and $\beta_p y = s$, and the last step holds from the definition of the incomplete upper gamma function (Abramowitz et al., 1988, ch. 6.5)

$$\Gamma_U(s,x) = \int_x^{+\infty} t^{s-1}e^{-t}dt$$

□

(c) Proof of equation (13) for the property of the derivative of the hypergeometric function in the special case $c = a + 1$.

Proof. First, an useful property for the derivative of any Gaussian hypergeometric function is that

$$\frac{d}{dx}{}_2F_1(a,b;c;x) = \frac{ab}{c}{}_2F_1(a+1,b+1;c+1;x)$$

Knowing this, let's rewrite equation (13) by bringing to the left side both the hypergeometric terms

$$\begin{aligned}
\frac{x}{a}\frac{d}{dx}{}_2F_1(a,b;a+1;x) + {}_2F_1(a,b;a+1;x)) &= \\
\frac{x}{a}\frac{ab}{a+1}{}_2F_1(a+1,b+1;a+2;x) + {}_2F_1(a,b;a+1;x)) &= \\
xb\sum_{n=0}^{+\infty}\frac{(b+1)_n}{\alpha+1+n}\frac{x^n}{n!} + a\sum_{n=0}^{+\infty}\frac{(b)_n}{a+n}\frac{x^n}{n!} &= \\
\sum_{n=0}^{+\infty}(b)_n\frac{x^n}{n!} &= \\
(1-x)^{-b}&
\end{aligned}$$

$(b)_n$ being the Pochhammer symbol (Abramowitz et al., 1988, p. 256, eq. 6.1.22). This concludes the proof since equation (13) can be obtained by rearranging the terms above.

□

(d) Proof of equation (20) for the antiderivative of the combination of a power function and the modified Bessel function of the second kind

Proof. First, let's introduce the series representation of the modified Bessel function of the second

kind (Abramowitz et al., 1988, p. 375, eq. 9.6.2)

$$K_\nu(z) = \frac{\pi}{2}\frac{I_{-\nu}(z) - I_\nu(z)}{sin(\pi\nu)}$$

with $I_\nu(z)$ being the modified Bessel function of the first kind, which has the following series representation (Abramowitz et al., 1988, p. 375, eq. 9.6.10)

$$I_\nu(z) = \sum_{k=0}^{+\infty} \frac{(z/2)^{\nu+2k}}{k!\Gamma(k+\nu+1)}$$

Thus,

$$\int x^{-\nu} K_\nu(ax)dx = \frac{\pi}{2sin(\pi\nu)}\left(\int x^{-\nu} I_{-\nu}(ax)dx - \int x^{-\nu} I_\nu(ax)dx\right)$$

Let's develop the calculations for the first integral

$$\begin{aligned}
\int x^{-\nu} I_{-\nu}(ax)dx &= \int x^{-\nu} \sum_{k=0}^{+\infty} \frac{(x/2)^{-\nu+2k}}{k!\Gamma(k-\nu+1)}dx \\
&= \int \sum_{k=0}^{+\infty} \frac{a^{-\nu+2k}x^{-2\nu+2k}}{2^{-\nu+2k}k!\Gamma(-\nu+1)(-\nu+1)_k}dx \\
&= \left(\frac{a}{2}\right)^{-\nu} \frac{1}{\Gamma(-\nu+1)} \sum_{k=0}^{+\infty} \int \frac{a^{2k}x^{-2\nu+2k}}{2^{2k}k!(-\nu+1)_k}dx \\
&= \left(\frac{a}{2}\right)^{-\nu} \frac{1}{\Gamma(-\nu+1)} \sum_{k=0}^{+\infty} \frac{a^{2k}x^{-2\nu+2k+1}}{2^{2k}k!(-\nu+1)_k(-2\nu+2k+1)}dx \\
&= \left(\frac{a}{2}\right)^{-\nu} \frac{x^{-2\nu+1}}{\Gamma(-\nu+1)} \sum_{k=0}^{+\infty} \frac{1}{(-\nu+1)_k(-2\nu+2k+1)} \frac{(ax/2)^{2k}}{k!} \\
&= \left(\frac{a}{2}\right)^{-\nu} \frac{x^{-2\nu+1}}{\Gamma(-\nu+1)} \frac{1}{-2\nu+1} {}_1F_2\left(-\nu+1/2; -\nu+1, -\nu+3/2; \left(\frac{ax}{2}\right)^2\right)
\end{aligned}$$

Following the same steps for the second integral we get

$$\int x^{-\nu} I_\nu(ax)dx = \left(\frac{a}{2}\right)^{\nu} \frac{x}{\Gamma(\nu+1)} {}_1F_2\left(1/2; \nu+1, 3/2; \left(\frac{ax}{2}\right)^2\right)$$

By substituting in the equation above we obtain equation (20) and this concludes the proof.

□

(e) Formulas for the cumulants of OU-TS and TS-OU processes.

For a random variable, $\mathbb{E}[X^k]$ can be obtained by calculating the k-th derivative of the characteristic function by u, dividing by i^k and evaluating for $u = 0$.

Therefore for the OU-TS, whose characteristic function can be obtained through equation (4), the k-th cumulant

$$c_{X(t),k} = \begin{cases} \mathbb{E}[(X(t) - \mathbb{E}[X(t)])^k] & k = 1, 2, 3 \\ \mathbb{E}[(X(t) - \mathbb{E}[X(t)])^k] - 3(\mathbb{E}[(X(t) - \mathbb{E}[X(t)])^{k/2}])^{k/2} & k = 4 \end{cases}$$

is

$$c_{X(t),k} = \begin{cases} X_0 e^{-bt} + \gamma_L \dfrac{1 - e^{-bt}}{b} & k = 1 \\ c_{L,k} \dfrac{(1 - e^{-kbt})}{kb} & k = 2, 3, 4 \end{cases}$$

where

$$c_{L,k} = \int_{-\infty}^{+\infty} x^k \nu_L(x) dx$$

For TS-OU the k-th cumulant, which can be derived from equation (6), is

$$c_{X(t),k} = \begin{cases} X_0 e^{-bt} + \gamma_L (1 - e^{-bt}) & k = 1 \\ c_{\bar{X},k} (1 - e^{-kbt}) & k = 2, 3, 4 \end{cases}$$

where

$$c_{\bar{X},k} = \int_{-\infty}^{+\infty} x^k \nu_{\bar{X}}(x) dx$$

For the TS Lévy measure we have

$$c_{TS,k} = \int_{-\infty}^{+\infty} x^k \nu_{TS}(x) dx = c_p \beta_p^{\alpha_p - k} \Gamma(k - \alpha_p) + (-1)^k c_n \beta_n^{\alpha_n - k} \Gamma(k - \alpha_n)$$

Abstract in lingua italiana

I processi Lévy-Ornstein-Uhlenbeck e Ornstein-Uhlenbeck-Lévy sono una classe generale di processi utilizzata per una varietà di applicazioni finanziarie. In letteratura esistono molti algoritmi Monte Carlo esatti per la loro simulazione, ma sebbene accurati possono essere lenti, di implementazione complessa, e sono specifici per ogni singolo processo. Con questo articolo proponiamo un algoritmo Monte Carlo più veloce per la simulazione di questi processi, risolvendo i problemi degli algoritmi esatti e al contempo garantendo livelli di accuratezza comparabili. Infatti l'algoritmo proposto è veloce, semplice e generale. Supportiamo questa affermazione confrontando le prestazioni del nostro algoritmo con quelle degli algoritmi esatti in termini sia di accuratezza che di tempo computazionale.

Parole chiave: simulazioni di Monte Carlo, Ornstein-Uhlenbeck non Gaussiano, tempered stable, normal tempered stable, prezzatura di opzioni

Acknowledgements

This thesis marks the end of a journey which I started five years ago. During this time I have grown and changed significantly, and this was only possible thanks to my friends, my family and the professors which have taught me at Politecnico di Milano.

Foremost, I would like to thank my supervisor Professor Roberto Baviera for allowing me to complete this master thesis and for helping me along the way with his valuable knowledge, both within the context of the thesis and outside of it.

A special thanks goes to my friends: to those who have always been by my side since high school and even before, to those which I met the first day at Politecnico, to those which I encountered later on during my Master's degree and just as importantly to those which I lost and found again along the way.

Finally, thank you to my family for always believing in me and supporting me throughout my experience. All of this would not have been possible without you.

The end of this path marks only the beginning of a new one, but I am certain that I will always be accompanied by these treasured moments and bonds which I gathered during these years.

Thank you all for your support!

Ringraziamenti

Questa tesi segna la fine di un percorso iniziato cinque anni fa. Durante questo tragitto sono cresciuto e sono cambiato significativamente, e questo è stato possibile solo grazie ai miei amici, alla mia famiglia e ai professori che mi hanno insegnato al Politecnico di Milano.

Innanzitutto, vorrei ringraziare il mio supervisore, il Professor Roberto Baviera, per avermi permesso di completare questa tesi e per avermi aiutato durante il percorso con le sue preziose conoscenze, sia nel contesto della tesi che al di fuori di esso.

Un ringraziamento speciale va ai miei amici: a quelli che sono sempre stati al mio fianco dai tempi del liceo e perfino prima, a quelli che ho conosciuto il mio primo giorno al Politecnico, a quelli che ho incontrato più avanti durante gli ultimi anni di laurea Magistrale e non di meno a quelli che ho smarrito e che ho ritrovato lungo il percorso.

Infine, un grazie alla mia famiglia per aver sempre creduto in me e per avermi supportato durante l'intero percorso. Tutto ciò non sarebbe stato possibile senza di voi.

La fine di questo cammino segna solo l'inizio di uno nuovo, ma sono sicuro che sarò sempre accompagnato dai preziosi momenti e legami che ho collezionato durante questi anni.

Grazie a tutti per il supporto!