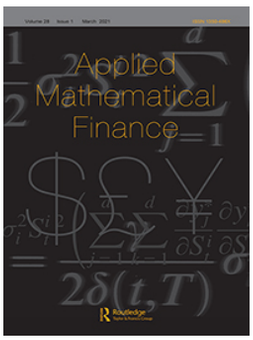


Applied Mathematical Finance

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/ramf20

Fast Pricing of Energy Derivatives with Mean-Reverting Jump-diffusion Processes

Piergiacomo Sabino & Nicola Cufaro Petroni

To cite this article: Piergiacomo Sabino & Nicola Cufaro Petroni (2021) Fast Pricing of Energy Derivatives with Mean-Reverting Jump-diffusion Processes, Applied Mathematical Finance, 28:1, 1-22, DOI: 10.1080/1350486X.2021.1909488

To link to this article: https://doi.org/10.1080/1350486X.2021.1909488

Published online: 21 Apr 2021.

APPLIED MATHEMATICAL FINANCE
2021, VOL. 28, NO. 1, 1–22
https://doi.org/10.1080/1350486X.2021.1909488

Routledge
Taylor & Francis Group

Fast Pricing of Energy Derivatives with Mean-Reverting Jump-diffusion Processes

Piergiacomo Sabino [a] and Nicola Cufaro Petroni [b]

[a]Quantitative Risk Management, E.ON SE, Essen, Germany; [b]Dipartimento di Matematica and TIRES, Università di Bari, INFN Sezione di Bari, Bari, Italy

ABSTRACT

Most energy and commodity markets exhibit mean-reversion and occasional distinctive price spikes, which result in demand for derivative products which protect the holder against high prices. To this end, in this paper we present a few fast and efficient methodologies for the exact simulation of the spot price dynamics modelled as the exponential of the sum of a Gaussian Ornstein-Uhlenbeck process and an independent pure jump process, where the latter one is driven by a compound Poisson process with (bilateral) exponentially distributed jumps. These methodologies are finally applied to the pricing of Asian options, gas and hydro storages and swing options under different combinations of jump-diffusion market models, and the apparent computational advantages of the proposed procedures are emphasized.

ARTICLE HISTORY
Received 12 December 2020
Accepted 23 March 2021

KEYWORDS
Simulation; mean-reverting jump-diffusion processes; compound poisson; energy derivatives

1. Introduction and motivations

The mathematical modelling of the day-ahead price in commodity and energy markets is supposed to capture some peculiarities like mean-reversion, seasonality and jumps. A typical approach consists of resorting to price processes driven either by a standard Ornstein-Uhlenbeck (OU) process, or by a regime switching process. The current literature is very rich with model suggestions: Lucia and Schwartz (2002), for instance, propose a one-factor Gaussian- OU with application to the Nordic Power Exchange, whereas a two factors version can be found in Schwartz and Smith (2000) with an additional Brownian Motion (*BM*).

The extension to models based on OU processes driven by a Normal Inverse Gaussian process or by a Variance Gamma process can be found among others in Benth, Kallsen, and Meyer-Brandis (2007), Cummins, Kiely, and Murphy (2017, 2018) and Sabino (2020a). Finally, a third type of proposed market models are based on mean-reverting jump-diffusion OU processes as originally suggested in Cartea and Figueroa (2005). In this paper, we focus on this last type of models and in particular, we consider mean-reverting jump-diffusion OU processes with exponential-like jumps superposed to standard Gaussian OU processes. A similar combination has been investigated also by other authors: for instance Deng (2000), Kjaer (2008), Hambly, Howison, and Kluge (2009b)

CONTACT Piergiacomo Sabino piergiacomo.sabino@gmail.com Quantitative Risk Management, E.ON SE, Brüsseler Platz 1, Essen 45131, Germany

and Nomikos and Soldatos (2008), or even Benth and Pircalabu (2018) in the context of modelling wind power futures.

Having selected a market model driven by a mean-reverting jump-diffusion dynamics, it is quite common to use Monte Carlo methods to price derivative contracts. To this end, it is very important to design fast and efficient simulation procedures particularly for real-time pricing. Indeed, risk management and trading units have to deal with a large number of contracts whose prices and sensitivities have to be evaluated regularly and, of course, the computational time may become an issue. The simulation of the skeleton of a Gaussian OU process is standard and efficient, whereas the generation of the path of an OU process with exponential jumps deserves particular attention. The simulation of this latter process can be based on the process definition itself, for example using a modified version of Algorithm 6.2 page 174 in Cont and Tankov (2004). Although sometimes referred with different naming convention, a mean-reverting compound Poisson process with exponential jumps is known in the literature as Gamma- OU process (Γ-OU) because it can be proven that its marginal law is a gamma law (see Barndorff-Nielsen and Shephard, (2001)).Recently, two different approaches have been proposed to address the simulation of a Γ-OU process. Based on the decomposition of the OU process into simple components, Qu, Dassios, and Zhao (2019) propose an exact simulation procedure that has the advantage of avoiding the simulation of the jump times. On the other hand, Sabino and Cufaro Petroni (2020) have studied the distributional properties of the Γ-OU and the bilateral- Γ-OU process (biΓ-OU) and they have found the density and characteristic function in closed form. In particular, they have proved that such a law can be seen as a mixture of well-known laws giving, as a side-product, very fast and efficient simulation algorithms.

In the present paper, we apply the new and traditional algorithms and compare their computational performance in the context of the pricing of complex energy derivatives with Monte Carlo simulations. We consider Asian options, gas and hydro storages and swings, that normally require a high computational effort. We assume three types of market models via the superposition of a Gaussian OU process to three different combination of Γ-OU and biΓ-OU processes and provide the risk-neutral conditions. In addition, we give the formal proof that a few commonly used simulation schemes (see for instance Benth, Di Persio, and Lavagnini (2018)) are not always applicable to standard market conditions. The numerical experiments performed below show that the proposed algorithms are unbiased and outperform any other approaches, and therefore they can provide a remarkable advantage in terms of computational time which constitute the main contribution of this paper. In the worst case, they are thirty times faster for the pricing of Asian options and *just* 40% faster for storages and swings using a Monte Carlo based stochastic optimization. These results show that the procedures presented in the following sections are by far the most efficient and are therefore suitable for real-time pricing.

The paper is structured as follows: in Section 2 we introduce the three market models driven by a mean-reverting jump-diffusion dynamics that we will adopt for the pricing of the energy derivatives. Then Section 3 introduces the concept of Lévy -driven OU processes and specifies the algorithms available for the exact simulation of a Γ-OU or a biΓ-OU process, whereas Section 4 illustrates the extensive numerical experiments that have been performed: as mentioned, the pricing of Asian options, gas storages and swings

have been considered. Finally, Section 5 concludes the paper with an overview of future inquiries and possible further applications.

2. Market Models

It is well-known that from a financial standpoint the day-ahead prices exhibit seasonality, mean reversion and jumps, so that a realistic market model has to capture these features. Following Kjaer (2008) and Kluge (2006), we will therefore assume that the dynamics of the day-ahead (spot) price can be decomposed into three independent factors

$$\begin{aligned} S(t) &= S_{\text{season}}(t) \cdot S_{\text{diff}}(t) \cdot S_{\text{jumps}}(t) \\ &= F(0,t)\, e^{h(t)+\sum_{d=1}^{D} X_d(t)+\sum_{j=1}^{J} Y_j(t)} = F(0,t)\, e^{h(t)+H(t)} \end{aligned} \tag{1}$$

where $X_d(t)$ and $Y_j(t)$ respectively represent diffusive and jumping OU processes, $h(t)$ is a deterministic seasonal factor and $F(0,t)$ is the initial forward price with delivery t. Going then to the characteristic functions we find

$$\varphi_H(u,t) = \varphi_{\text{diff}}(u,t) \cdot \varphi_{\text{jumps}}(u,t) = \boldsymbol{E}\left[e^{iuH(t)}\right] = \prod_{d=1}^{D} \varphi_{X_d}(u,t) \prod_{j=1}^{J} \varphi_{Y_j}(u,t) \tag{2}$$

$$\varphi_{X_d}(u,t) = \boldsymbol{E}\left[e^{iuX_d(t)}\right] \qquad \varphi_{Y_j}(u,t) = \boldsymbol{E}\left[e^{iuY_j(t)}\right],$$

whereas according to the risk-neutral arguments of the Lemma 3.1 in Hambly, Howison, and Kluge (2009a), we get the deterministic function $h(t)$ consistent with forward curve

$$h(t) = -\ln \boldsymbol{E}\left[e^{H(t)}\right] = -\log \varphi_H(-i,t). \tag{3}$$

Hereafter, however, we will consider just the following particular representation of spot prices with only one diffusing and jumping processes

$$S(t) = F(0,t)\, e^{h(t)+X(t)+Y(t)} \tag{4}$$

where the diffusive part is the Gaussian OU process

$$X(t) = X(0)e^{-\rho t} + \sigma \int_0^t e^{-\rho(t-s)} dW(s) \tag{5}$$

$$\log \varphi_{\text{diff}}(u,t) = iuX(0)e^{-\rho t} - \frac{u^2\sigma^2}{4\rho}\left(1 - e^{-2\rho t}\right). \tag{6}$$

where $W(t)$ is a standard BM . We instead do not consider any additional BM as done in Schwartz and Smith (2000), but we assume that the process $Y(\cdot)$ follows one of the three dynamics below:

1. In the first case the jumping part is the OU process

$$Y(t) = Y(0)e^{-kt} + \sum_{n=1}^{N(t)} e^{-k(t-\tau_n)} J_n = Y(0)e^{-kt} + Z(t) \tag{7}$$

where $N(t)$ is a Poisson process with intensity λ and jump times τ_n; J_n, $n = 1, 2, \ldots$ are independent and identically distributed (*iid*) random variables (*rv*'s) following a double exponential law as defined in Kou (2002), namely a mixture (with mixing parameters p and $q = 1 - p$) of a positive exponential *rv* $U \sim \mathfrak{E}_1(\beta_1)$ and a negative exponential *rv* $-D$ with $D \sim \mathfrak{E}_1(\beta_2)$, so that the probability density (*pdf*) and the characteristic function (*chf*) are

$$f_{\beta_1,\beta_2,p}(x) = p\beta_1 e^{-\beta_1 x} 1_{x\geq 0} + (1-p)\beta_2 e^{\beta_2 x} 1_{x<0} \tag{8}$$

$$\varphi_{\beta_1,\beta_2,p}(v) = p\frac{\beta_1}{\beta_1 - iv} + (1-p)\frac{\beta_2}{\beta_2 + iv} = p\,\varphi_u(v) + (1-p)\varphi_d(v) \tag{9}$$

In other words, in distribution, every jump is $J_n \overset{d}{=} B_n U_n - (1 - B_n)D_n$ where B_n is a binomial *rv* with distribution $\mathfrak{B}(\mathfrak{p})$. In addition, we always assume that J_n, $n = 1, 2, \ldots$ are independent from the process $N(t)$.

The process $Y(\cdot)$ apparently is a mean-reverting non-Gaussian OU process (see next section) with mean-reversion level $k > 0$, and – as shown in Sabino and Cufaro Petroni (2020) – its jumping part turns out to be the difference $Z(t) = Z_1(t) - Z_2(t)$ of two independent processes

$$Z_1(t) = \sum_{n=1}^{N_1(t)} e^{-k(t-\tau_n)} U_n \qquad Z_2(t) = \sum_{n=1}^{N_2(t)} e^{-k(t-\tau_n)} D_n$$

with the same parameter k, and $N_1(t), N_2(t)$ two independent Poisson processes, respectively, of intensities $\lambda_1 = p\lambda$ and $\lambda_2 = (1-p)\lambda$. As a consequence we find

$$\varphi_{\text{jumps}}(u,t) = e^{-iuY(0)e^{-kt}} \varphi_1(u,t)\varphi_2(-u,t) \tag{10}$$

where $\varphi_1(u,t)$ and $\varphi_2(u,t)$ respectively are the *chf*'s of $Z_1(t)$ and $Z_2(t)$.

2. In the second case, our jumping OU process will be the difference $Y(t) = Y_1(t) - Y_2(t)$ with

$$Y_1(t) = Y_1(0)e^{-k_1 t} + \sum_{n=1}^{N_1(t)} e^{-k_1(t-\tau_n^{(1)})} U_n = Y_1(0)e^{-k_1 t} + Z_1(t) \tag{11}$$

$$Y_2(t) = Y_2(0)e^{-k_2 t} + \sum_{m=1}^{N_2(t)} e^{-k_2(t-\tau_m^{(2)})} D_m = Y_2(0)e^{-k_2 t} + Z_2(t) \tag{12}$$

where $N_1(t)$ and $N_2(t)$ are two independent Poisson processes with intensities λ_1 and

λ_2, respectively and U_n, D_m are independent *rv*'s with exponential laws $\mathfrak{E}_1(\beta_1)$ and $\mathfrak{E}_1(\beta_2)$ respectively.

3. Finally in the third case the jumps J_n of the process (7) will be distributed according to a centred Laplace law with parameter β, and therefore the jump process $Z(\cdot)$ can again be seen as the difference of two independent processes $Z(t) = Z_1(t) - Z_2(t)$ as in the case 2, but with the difference that here $Z_1(t)$ and $Z_2(t)$ have the same parameter k and U_n and D_m are independent *rv*'s with the same law $\mathfrak{E}_1(\beta)$.

While the simulation of a Gaussian OU process is standard and very fast, the stepping stone for the simulation of every jump process involved in the previous formulas is the generation of a *rv* distributed according to the law of $Z(t)$. Therefore, the overall computational effort will be deeply affected by that required to simulate these jump processes. To this end, the simulation procedure of the skeleton of the day-ahead price $S(t)$ in (4) over a time grid $t_0, t_1, \ldots, t_M$ ($\Delta t_m = t_m - t_{m-1}$, $m = 1, \ldots, M$) consists of the steps illustrated in Algorithm 1, where we assume for simplicity $X(0) = Y(0) = 0$.

Algorithm 1

1: **for** $m = 1, \ldots, M$ **do**

2: $h(t_m) \leftarrow -\frac{\sigma^2}{4k}(1 - e^{-2\rho\,\Delta t_m}) - \log \varphi_{\text{jumps}}(u, t_m)$

3: Generate a Gaussian $x \sim \mathfrak{N}\left(0, \sigma\sqrt{\frac{1-e^{-2\rho\ \ t_m}}{2\rho}}\right)$

4: Generate $z_1 \stackrel{d}{=} \sum_{n=1}^{N_1(t_m)} e^{-k_1(t_m - \tau_n^{(1)})} U_n$

5: Generate $z_2 \stackrel{d}{=} \sum_{\ell=1}^{N_2(t_m)} e^{-k_2(t_m - \tau_\ell^{(2)})} D_n$

6: $X(t_m) \leftarrow X(t_{m-1}) e^{-\rho \Delta t_m} + x$

7: $Y_i(t_m) \leftarrow Y_i(t_{m-1}) e^{-k_i \Delta t_m} + z_i I = 1, 2.$

8: $S(t_m) \leftarrow e^{h(t_m) + X(t_m) + Y_1(t_m) - Y_2(t_m)}$

9: **end for**

Although sometimes the mean-reverting jump processes with exponential jumps are mentioned under different names in the financial literature (e.g., *MRJD* in Kjaer (2008)), they are generally known as Gamma-OU processes (Γ-OU), because its stationary law is a gamma distribution. On the other hand, the processes $Y(\cdot)$ defined above are in fact bilateral-gamma- OU process – denoted here as biΓ-OU – because their stationary laws are bilateral gamma distributions, namely differences of two independent gamma laws (for details see Sabino and Cufaro Petroni (2020) and Küchler and Tappe (2008)). The exact simulation of the skeleton of $Y(\cdot)$ depends therefore on a fast generation of $Z(t)$, and in the following section we consider three possible alternative simulation algorithms available in the literature.

3. OU Processes with Compound Poisson Noise

Take a one-dimensional Lévy process $L(\cdot)$, and the OU process $Y(\cdot)$ solution of the stochastic differential equation (*SDE*)

$$dY(t) = -kY(t)dt + dL(t) \qquad Y(0) = Y_0 \quad \boldsymbol{P} - a.s. \qquad k>0 \tag{13}$$

that is

$$Y(t) = Y_0 e^{-kt} + Z(t) \qquad Z(t) = \int_0^t e^{-k(t-s)} dL(s). \tag{14}$$

The process $L(\cdot)$ will be called hereafter *background driving Lévy process* (BDLP). Following a Barndorff-Nielsen and Shephard (2001) convention, if $\bar{\mathfrak{D}}$ is the law of the stationary solution of (13) we will say that $Y(\cdot)$ is a $\bar{\mathfrak{D}}$- OU process; when on the other hand the *rv* $L(1)$ of the BDLP is distributed according to the infinitely divisible (*id*) law $\mathfrak{D}$ we will say that $X(\cdot)$ is an OU-$\mathfrak{D}$ process. A well-known result (see for instance Cont and Tankov (2004) or Sato (1999)) states that a distribution $\bar{\mathfrak{D}}$ can be the stationary law of a given OU-$\mathfrak{D}$ process if and only if $\bar{\mathfrak{D}}$ is *self-decomposable* (*sd*, see more below). It is also possible to see (see also Barndorff-Nielsen, Jensen, and Sørensen (1998)) that the solution process (13) is stationary if and only if its *chf* $\varphi_Y(u,t)$ is constant in time and steadily coincides with the *chf* $\overline{\varphi}_Y(u)$ of the (*sd*) invariant initial distribution that in its turn is decomposable according to

$$\overline{\varphi}_Y(u) = \overline{\varphi}_Y(ue^{-kt})\varphi_Z(u,t)$$

where, at every given t, $\varphi_Z(u,t)$ denotes the *chf* of the *rv* $Z(t)$ in (13) .

We recall that a law with *pdf* $f(x)$ and *chf* $\varphi(u)$ is said to be *sd* (see Sato (1999) or Cufaro Petroni (2008)) when for every $0<a<1$ we can find another law with *pdf* $g_a(x)$ and *chf* $\chi_a(u)$ such that

$$\varphi(u) = \varphi(au)\chi_a(u). \tag{15}$$

We will accordingly say that a *rv* X is *sd* when its law is *sd*: looking at the definition this means that for every $0<a<1$ we can always find two independent *rv*'s, Y (with the same law of X) and Z_a (here called *a-remainder*, with *pdf* $g_a(x)$ and *chf* $\chi_a(u)$) such that

$$X \stackrel{d}{=} aY + Z_a. \tag{16}$$

As also observed in Sabino (2020a), this apparently means that the law of $Z(t)$ in a stationary solution (14) in nothing else than the a-remainder of its *sd* law provided that $a = e^{-kt}$. Suppose now that the BDLP $L(\cdot)$ in (13) is the compound Poisson process

$$L(t) = \sum_{n=0}^{N(t)} J_n \qquad J_0 = 0 \quad \boldsymbol{P} - a.s.,$$

with intensity λ of the Poisson process $N(t)$, and *iid* jumps $J_n \sim \mathfrak{E}_1(\beta)$: then the solution (14) becomes

$$Y(t) = Y_0 e^{-kt} + Z(t), \qquad Z(t) = \sum_{n=0}^{N(t)} J_n e^{-k(t-\tau_n)}.$$

It is well-known (see for instance Schoutens (2003), 68) that the stationary law of this process is a gamma distribution, and therefore our OU process actually is a Γ-OU (k, λ, β) with all its parameters put in evidence. Extending this naming convention, it is also easy to see that the jump components of the three market models of Section 2 will simply be biΓ-OU processes because it can be shown (see Sabino and Cufaro Petroni (2020)) that their stationary laws are a bilateral gamma distributions, and in particular in the third market model we will have a *symmetric* biΓ-OU process.

All the observations above apparently point to the fact that in every jump process for our market models the law of $Z(t)$ turns out to coincide with that of the a-remainder Z_a of the stationary law of the OU process $Y(\cdot)$, provided that $a = e^{-kt}$. The said law, will therefore be instrumental for the derivation of efficient algorithms for the exact simulation of all the types of $Y(\cdot)$ introduced in the section 2 .

3.1. Positive Jumps: Γ-OU Processes

A straightforward way to sample the *rv* $Z(t)$ to generate the skeleton of a Γ-OU process with parameters k, λ, β (see the step 4 in Algorithm 1) could simply consist in adapting Algorithm 6.2 page 174 in Cont and Tankov (2004) as delineated in the following Algorithm 2.

Algorithm 2

1: Generate $N \sim \mathfrak{P}(\lambda \Delta t_m)$ ▷ Poisson *rv* with intensity $\lambda \Delta t_m$

2: Generate N *iid* uniform *rv*'s $\boldsymbol{u} = (u_1, \ldots, u_N) \sim \mathfrak{U}([0,1]^N)$.

3: Sort $\boldsymbol{u}$, (u(Bardou, Bouthemy, and Pagés 2009) $< \cdots < u_{[N]}$),

4: $\tau n \leftarrow \Delta t_m u_{[n]}$, $n = 1, \ldots N$,

5: Generate N *iid* $J_n \sim \mathfrak{E}_1(\beta), n = 1, \ldots N$, ▷ Exponential *rv* with rate β

6: $z \leftarrow \sum_{n=1}^{N} e^{-k(\Delta t_m - \tau_n)} J_n$

The previous procedure, however, does not rely directly on the statistical properties of the process $Z(\cdot)$, but it is rather based on its definition. Starting instead from a different standpoint, new simulation algorithms that are fully based on the distributional properties of the Γ-OU process have been recently proposed in Sabino and Cufaro Petroni (2020): they rely in fact on the explicit representation of the transition law of a Γ-OU process. In particular, it has been shown that the law of $Z(t)$ for a Γ-OU process with parameters (k, λ, β) coincides with that of the a-remainder Z_a of a gamma law $\Gamma(\alpha, \beta)$ with scale parameter $\alpha = \lambda/k$ and rate parameter β, provided that $a = e^{-kt}$.

We recall that the laws of the gamma family $\Gamma(\alpha, \beta)$ ($\alpha > 0, \beta > 0$) have the following *pdf* and *chf*

$$f_{\alpha,\beta}(x) = \frac{\beta}{\Gamma(\alpha)}(\beta x)^{\alpha-1}e^{-\beta x} \qquad x>0 \tag{17}$$

$$\varphi_{\alpha,\beta}(u) = \left(\frac{\beta}{\beta - iu}\right)^{\alpha}. \tag{18}$$

In particular $\Gamma(n,\beta)$, with $\alpha = n = 1, 2, \ldots$ a natural number, are the Erlang laws $\mathfrak{E}_{\mathfrak{n}}(\beta)$, while $\Gamma(1,\beta)$ is the usual exponential law $\mathfrak{E}_{\mathfrak{1}}(\beta)$. In addition, the *chf* of the a-remainder Z_a of such a gamma law is

$$\chi_a(u,\alpha,\beta) = \left(\frac{\beta - iau}{\beta - iu}\right)^{\alpha}.$$

If now S is a *rv* distributed according to the *negative binomial (a.k.a. Pólya) distribution*, hereafter denoted $\bar{\mathfrak{B}}(\alpha,p)$, $\alpha>0,\ 0<p<1$, namely

$$\boldsymbol{P}\{S=n\} = \binom{\alpha+n-1}{n}(1-p)^{\alpha}p^n \qquad n=0,1,\ldots$$

it has been shown in Sabino and Cufaro Petroni (2020) that the said *chf* of Z_a is representable as

$$\chi_a(u,\alpha,\beta) = \left(\frac{\beta - iau}{\beta - iu}\right)^{\alpha} = \sum_{n=0}^{\infty}\binom{\alpha+n-1}{n}a^{\alpha}(1-a)^n\left(\frac{\beta}{\beta-iau}\right)^n \tag{19}$$

The distribution of Z_a turns out therefore to be an infinite, $\bar{\mathfrak{B}}(\alpha,1-a)$-weighted mixture of Erlang laws $\mathfrak{E}_{\mathfrak{n}}(\beta/a)$, namely the Erlang law $\mathfrak{E}_{\mathfrak{S}}(\beta/a)$ with a Pólya random index S of the following sum of *iid* exponential *rv*'s

$$\sum_{j=0}^{S} X_j \qquad S\sim\bar{\mathfrak{B}}(\alpha,1-a) \qquad X_j \sim \mathfrak{E}_{\mathfrak{1}}(\beta/a) \qquad X_0 = 0,\ \boldsymbol{P}-a.s.$$

The previous remarks entail that the *chf* of $Z(t)$ in a Γ-OU (k,λ,β) process is

$$\varphi_Z(u,t) = \left(\frac{\beta - iue^{-kt}}{\beta - iu}\right)^{\frac{\lambda}{k}} \tag{20}$$

so that the simulation of its innovations z can be carried on according to Algorithm 3.

Algorithm 3

1: $\alpha \leftarrow \lambda/k,\ a \leftarrow e^{-k\Delta tm}$

2: $b \leftarrow B \sim \bar{\mathfrak{B}}(\alpha,1-a)$ ▷ Generate a Pólya $(\alpha,\ 1-a)$ *rv*

3: $z \leftarrow \mathfrak{E}_{\mathfrak{b}}\ (\beta/a)$; ▷ Generate an Erlang *rv* with rate β/a

A different method to simulate a Γ-OU has been recently proposed in Qu, Dassios, and Zhao (2019) and it is based on the following alternative representation of the *chf* of $Z(t)$

$$\varphi_Z(u,t) = e^{\lambda t\left(\varphi_{\tilde{J}}(u)-1\right)} \qquad \varphi_{\tilde{J}}(u) = \int_0^1 \frac{\beta e^{ktv}}{\beta e^{ktv} - iu} dv \tag{21}$$

that coincides with the *chf* of a compound Poisson *rv* with parameter λt and exponentially distributed jumps of random rate $\tilde{\beta} \stackrel{d}{=} \beta e^{ktU}$ (here U is uniform in $[0,1]$). This third procedure is summarized in Algorithm 4. Algorithms 3 and 4 avoid simulating the jump times of the Poisson process, while on the other hand the Algorithms 2 and 4 require similar operations but additional steps w.r.t. to the Algorithm 3.

Algorithm 4

1: Generate $N \sim \mathfrak{P}(\lambda \Delta t_m)$, ◁ Poisson *rv* with intensity $\lambda \Delta t_m$

2: Generate N *iid* uniform *rv*'s $u = (u_1, \ldots, u_N) \sim \mathfrak{U}([0,1]^N)$.

3: $\tilde{\beta}_n \leftarrow \beta e^{k\Delta t_m u_n}, n = 1, \ldots, N.$

4: Generate N *iid* $\tilde{J}_n \sim \mathfrak{E}_1(\tilde{\beta}_n), n = 1, \ldots, N,$ ◁ Exponential *rv*'s with random rate $\tilde{\beta}_i$

5: $z \leftarrow \sum_{i=n}^{N} \tilde{J}_n$.

Several other alternative schemes based on some numerical approximation are of course conceivable, but they do not appear to be advantageous. Taking for instance – as it is often done – an equally-spaced time grid, one might use an Euler discretization with the assumption that only one jump can occur within each time step with probability $\lambda \Delta t$ so that, if $B_m(1) \sim \mathfrak{B}(1, \lambda \quad t)$ are m independent Bernoulli *rv*'s, it is

$$Y(t_m) = Y(t_{m-1})(1 - k\Delta t) + B_m(1) J_m, \quad m = 1, \ldots M, \tag{22}$$

Setting then for simplicity $b = 1 - \lambda \Delta t$, the *chf* of $B_m(1) Y_m$ is

$$\varphi_m(u,t) = b + \beta \frac{1-b}{\beta - iu} = \frac{\beta - ibu}{\beta - iu} = \frac{\beta - i(1-\lambda\Delta t)u}{\beta - iu}.$$

This *chf* however amounts to a first order approximation of (20) only when $k = \lambda$, while on the other hand, a reduction of the time step would by no means provide an improvement. As a consequence every calibration, or pricing of derivatives relying on the simulation of an Γ-OU with the assumption that only one jump can occur per time step would lead to wrong and biased results.

Following instead Benth, Di Persio, and Lavagnini (2018) in the context of Normal Inverse Gaussian driven OU processes, a further scheme can be developed that approximates in law $Z(\Delta t)$ of (14) by means of $e^{-k\Delta t} L(\Delta t)$ so that the discretized equation of the process becomes

$$Y(t_m) = Y(t_{m-1}) e^{-k\Delta t} + e^{-k\Delta t} \sum_{n=0}^{N(\Delta t)} J_n, \quad m = 1, \ldots M \tag{23}$$

This last approximation however is appropriate under the condition required to approximate in law a Pólya *rv* $S \sim \bar{\mathfrak{B}}(\lambda/k, 1 - e^{-k\Delta t})$ with the Poisson *rv* $N(\Delta t) \sim \mathfrak{P}(\lambda \Delta t)$, namely in the limit $k \to 0$:

$$\boldsymbol{P}\{S=n\} = \frac{\Gamma(\lambda/k + n)}{\Gamma(n+1)\Gamma(\lambda/k)} \left(e^{-k\Delta t}\right)^{\lambda/k} \left(1 - e^{-k\Delta t}\right)^n$$

$$= \frac{e^{-\lambda\Delta t}}{n!} \frac{\lambda}{k} \left(\frac{\lambda}{k} + 1\right) \dots \left(\frac{\lambda}{k} + n - 1\right) (k\Delta t + O(k^2))^n \tag{24}$$

$$= \frac{e^{-\lambda\Delta t}}{n!} (\lambda\Delta t + O(k))^n \left(1 + \frac{k}{\lambda}\right) \dots \left(1 + \frac{(n-1)k}{\lambda}\right) \to \frac{e^{-\lambda\Delta t}}{n!} (\lambda\Delta t)^n$$

and therefore we can conclude that the proposed approximation is good only for $\lambda \gg k$, an assumption that can be rather restrictive in many financial contexts.

3.2. Time-Dependent Poisson Intensity

The jumps are often concentrated in clusters: for instance, energy markets are very seasonal and jumps more often occur either during a period of high demand or in a period of cold spell. A more realistic approach could then be to consider a non-homogeneous Poisson process with a time-dependent intensity $\lambda(t)$ and $\Lambda(t) = \int_0^t \lambda(s)ds$ instead of the usual linear behaviour. In this case, the new Poisson process and its relative compound version have independent, but non-stationary increments. The modelling then becomes more challenging and somehow depends on the choice of the specific intensity function. In many cases, however, one could consider a time grid $t_0, t_1, \dots, t_M$ fine enough that the non-homogeneous Poisson process shows some step-wise intensity, $\lambda(s) = \lambda_m 1_{s\in\Delta t_m}$. Since the non-homogeneous Poisson has independent increments, it behaves at time t as the sum of different independent Poisson processes each with a constant intensity. The main consequence of this simple assumption is that the generation of z at each time step m in Algorithm 1 – in combination to no matter which methodology illustrated in Subsection 3.1 – is performed setting a different intensity λ_m for $m = 1, \dots M$.

3.3. Positive and Negative Jumps: biΓ-OU Processes

The three market models presented in Section 2 exhibit positive and negative jumps that are modelled as biΓ-OU processes. As illustrated in Algorithm 1, the generation of the jump component is simply obtained by running twice one of the algorithms discussed in Subsection 3.1 . On the other hand, as shown in Sabino and Cufaro Petroni (2020), one can also implement simulation procedures customized for the process $Z(\cdot)$ with Laplace jumps. Essentially, the steps 4 and 5 of Algorithm 1 should just be packed together into one because now $i = 1$, and the step 5 in Algorithm 2 should be accordingly reformulated: we omit the details for short. Since on the other hand the *chf* of $Z(t)$ now is

$$\varphi_Z(u,t) = \left(\frac{\beta^2 - u^2a^2}{\beta^2 - u^2}\right)^{\lambda/2k} \qquad a = e^{-kt} \tag{25}$$

the distribution of the process at time t coincides with that of the a-remainder Z_a of a symmetrical $\mathfrak{bГ}(^{\lambda}/_{2\mathfrak{k}}, \beta)$ and Algorithm 3 must be accordingly adapted to the case of a symmetric biГ-OU process (we skip again the unnecessary explicit reformulation).

Recalling finally (see Sabino and Cufaro Petroni (2020)) that the *chf* of $Z(t)$ in the case of a biГ-OU can also be rewritten as

$$\varphi_Z(u,t) = e^{\lambda t\left(\varphi_{\tilde{L}}(u)-1\right)} \tag{26}$$

where

$$\varphi_{\tilde{L}}(u) = \int_0^1 \frac{\lambda_J^2 e^{2ktv}}{\lambda_J^2 e^{2ktv} + u^2}\, dv \tag{27}$$

we also observe that the right-hand side in (27) is the *chf* of compound Poisson *rv* with parameter λt whose jumps are independent copies $\tilde{J}_n$ distributed according to a uniform mixture of centred Laplace laws with random parameter βe^{ktU} with $U \sim \mathfrak{U}([\mathfrak{o}, \mathfrak{1}])$. This result leads to the adaptation of the methodology of Qu, Dassios, and Zhao (2019) to the case of a symmetric biГ-OU as specified in the Algorithm 5

Algorithm 5

1: Generate $N \sim \mathfrak{P}(\lambda \Delta t_m)$, ▷Poisson *rv* with intensity $\lambda \Delta t_m$

2: Generate N *iid* uniform *rv*'s $\boldsymbol{u} = (u_1, \ldots, u_N) \sim \mathfrak{U}([\mathfrak{o}, \mathfrak{1}]^{\mathfrak{N}})$.

3: $\beta_n^{(r)} \leftarrow \beta e^{k\Delta t_m u_n}, n = 1, \ldots, N, r \in \{u, d\}$.

4: Generate N *iid* $U_n \sim \mathfrak{E}_{\mathfrak{1}}(\beta_n^{(u)}), i = 1, \ldots, n$, ▷ Generate N independent exponential *rv*'s with random rate $\beta_i^{(d)}$

5: Generate N *iid* $D_n \sim \mathfrak{E}_{\mathfrak{1}}(\beta_n^{(d)}), n = 1, \ldots, N$, ▷ Generate N independent exponential *rv*'s with random rate $\beta_n^{(d)}$

6: $z \leftarrow \sum_{n=1}^{N}(U_n - D_n)$

4. Numerical Experiments

We compare the computational performance of all the algorithms detailed in Section 3 in combination with Algorithm 1 for the simulation of the path trajectory of each market model introduced in Section 2. We illustrate their differences by pricing energy contracts namely, Asian options, swings and storages with Monte Carlo (MC) methods. The implementation of the pricing of such contracts with MC methods needs to be unbiased and fast especially if it is meant for real-time calculations.

In our numerical experiments, we have decided to assign different mean-reversion rates to the jump and to the diffusive components to better capture the spikes. For example, with respect to the parameter settings used in Deng (2000) and Kjaer (2008), the mean-reversion rates of our jump components are larger than those of their diffusion counterparts. The parameter combination in Kjaer (2008) assumes indeed that the

process $H(\cdot)$ has just one – and small – mean-reversion rate with a high λ, so that $\lambda/k \simeq 7$ and one could implement the simplified version of Algorithm 3 based on the binomial mixture of Erlang laws as explained in Sabino and Cufaro Petroni (2020).

All the simulation experiments in the present paper have been conducted using *MATLAB R2019a* with a 64-bit Intel Core i5-6300 U CPU, 8GB. As an additional validation, the comparisons of the simulation computational times have also been performed with *R* and *Python* leading to the same conclusions.

4.1. Numerical Experiments: Asian Options

The first numerical experiment that we have conducted refers to the pricing of an Asian option with European exercise style using MC under the assumption that the jump process $Y(\cdot)$ of the market model (4) is given by (7) (case 1). In virtue of the Lemma 3.1 in Hambly, Howison, and Kluge (2009a), risk-neutral conditions are met if

$$h(t) = -\frac{\sigma^2}{4\rho}\left(1 - e^{-2\rho t}\right) - \frac{p\lambda}{k}\log\left(\frac{\beta_1 - e^{-kt}}{\beta_1 - 1}\right) - \frac{(1-p)\lambda}{k}\log\left(\frac{\beta_2 - e^{-kt}}{\beta_2 - 1}\right).$$

Recalling that the payoff of such an option at maturity T is

$$A(T) = \left(\frac{\sum_{m=1}^{M} S(t_m)}{M} - K\right)^+,$$

we consider an at-the-money Asian option $K = S_0 = 22$ having one year maturity ($T = 1$) with daily settlement ($M = 360$) and with realistic market parameters shown in Table 1 with a flat forward curve.

MC methods are known to be slower than FFT techniques. To this end, one could, for instance, base and benchmark the pricing of discrete Asian options on the Fourier cosine expansion method illustrated in see Zhang and Oosterlee (2013a). In alternative, one could compute optimal bounds relying on the methods of Fusai and Kyriakou (2016) or of Kyriakou et al. (2017). Nevertheless, the MC approach also provides a view on the quantiles of the distribution of the potential cash-flows of derivative contracts giving a precious information to risk managers or to trading units.

Table 2 shows the estimated prices per number of simulations N_S, the standard errors (SE), defined as the sample standard deviations, the percentage errors (%err), defined as

$$\%err = \frac{\text{benchmark value} - \text{estimated value}}{\text{benchmark value}}$$

and the total CPU times in seconds using the different methodologies for the simulation of the process $Y(\cdot)$. We select the MC estimated price with $N_S = 10^7$ simulations obtained using traditional Algorithm 2 as the benchmark.

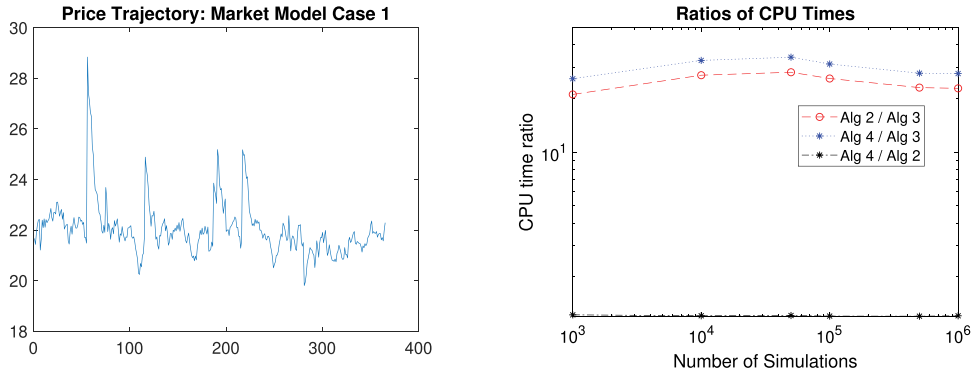
As expected, in terms of convergence and SE's, all the approaches are equally performing while instead, the CPU times are radically different. Algorithm 2 and Algorithm 4

Table 1. Parameters for spot (day-ahead) dynamics (case 1).

S_0	ρ	σ	k	p	λ_2	β_1	β_2
22	67	0.25	50	0.6	20	10	20

Table 2. Asian options: comparison among CPU times in seconds, prices, SE's and %err's with the different algorithms.

	Algorithm 3				Algorithm 2				Algorithm 4			
N_S	Price	%err	SE	CPU	Price	%err	SE	CPU	Price	%err	SE	CPU
10^3	1.0360	− 1.91%	1.6017	0.29	1.0712	− 5.37%	1.6605	6.12	1.0404	− 2.35%	1.5914	7.49
10^4	0.9987	1.76%	1.6012	2.12	0.9887	2.74%	1.5907	57.45	1.0013	1.50%	1.6020	69.56
5×10^4	1.0054	1.10%	1.6129	10.20	1.0198	− 0.32%	1.6186	286.85	1.0219	− 0.52%	1.6232	348.39
10^5	1.0139	0.26%	1.6166	22.14	1.0126	0.39%	1.6158	574.44	1.0140	0.25%	1.6191	692.70
5×10^5	1.0148	0.18%	1.6208	124.90	1.0191	− 0.24%	1.6194	2881	1.0147	0.19%	1.6119	3469
10^6	1.0172	− 0.06%	1.6188	251.45	1.0153	0.13%	1.6173	5745	1.0178	− 0.12%	1.6219	6953



(a) Sample Trajectory. (b) Ratio of CPU times.

Figure 1. Asian options.

have similar computational effort; therefore, their CPU times are comparable as observed in Figure 1(b). From Table 2 it is evident that our methodology provides a remarkable computational advantage.

Indeed, what requires minutes for Algorithms 2 and 4 only requires seconds for Algorithm 3. For example with $N_S = 5 \times 10^5$ simulations, with our computer, the pricing of the Asian option above, is accomplished in almost 2 min whereas, it takes almost 1 h with the other alternatives. Figure 1(b) clearly shows that, in the worst case, our simulation procedure is at least thirty times faster than any other alternative being then suitable for real-time applications.

4.2. Numerical Experiments: Gas and Hydro Storages

Denote by $C(t)$ the volume of a (virtual) gas storage or a hydro pump storage (VPS) or virtual hydro plant (VHP) at time t with $C_{min} \leq C(t) \leq C_{max}$.

The holder of such an energy asset is faced with a timing problem that consists in deciding when to inject, to withdraw or to do-nothing.

Denoting $J(t,x,c)$ the value of such an energy facility at time t given $S(t) = x$, $C(t) = c$, one can write:

$$J(t,x,c) = \sup_{u \in \mathcal{U}} \mathbb{E}\left[\int_t^T \phi_u(S(s))ds + q(S(T), C(T)) \,|S(t) = x, C(t) = c\right], \tag{28}$$

where $\mathcal{U}$ denotes the set of the admissible strategies, $u(t) \in \{-1, 0, 1\}$ is the regime at time t such that

$$\begin{cases} \phi_{-1}(S(t)) &= -S(t) - K_{in}a_{in}, & \text{injection} \\ \phi_0(S(t)) &= -K_N, & \text{donothing} \\ \phi_1(S(t)) &= S(t) - K_{out}a_w & \text{withdrawal} \end{cases}, \tag{29}$$

a_{in} and a_w are the injection (or pump for VPS's) and withdrawal rates, K_{in}, K_{out} and K_N, respectively, represent the costs of injection, do-nothing and withdrawal, and q takes into account the possibility of final penalties.

This problem is similar to the choice of the trading strategy of an American option, although in this case, the decision is more challenging because of the several volumetric constrains and the possibility of multiple actions. For example, VHP's generally have an exogenous inflow that is either described in the contract or is dependent on the nature of the asset and there is no pump flexibility. This particular feature prevents the *do-nothing* action because one is forced to inject. VPS's might also have an external inflow, but that is not always the case.

Based on the Bellman recurrence equation (see Bertsekas (2005)), one can perform the following backward recursion for $m = 1, \ldots, M$:

$$J(t_m, x, c) = \sup_{k \in \{-1,0,1\}} \left\{ \phi_k S(t_m) + \mathbb{E}[J(t_{m+1}, S(t_{m+1}), \tilde{c}_k) | S(t_m) = x, C(t_m) = c] \right\} \quad (30)$$

where

$$\begin{cases} \tilde{c}_{-1} &= \min(c + a_{in}, C_{max}) \\ \tilde{c}_0 &= c \\ \tilde{c}_1 &= \min(c - a_w, C_{min}). \end{cases} \quad (31)$$

Several approaches may be adopted to solve the recursion above: for instance, one may adapt the method proposed by Ben-Ameur et al. (2007) or use the quantization method as explained in Bardou, Bouthemy, and Pagés (2009), or even rely on Fourier techniques described for instance in Jaimungal and Surkov (2011). In our numerical example we consider the modified Least-Squares Monte Carlo (LSMC) method of Boogert and de Jong (2008, 2011) for gas storages. In contrast to the original version introduced in Longstaff and Schwartz (2001) in the context of American options, this modified approach complies with the fact that storages have volumetric restrictions, can have positive and negative payoffs and that the holder must decide among multiple actions at all moment of time (see Boogert and de Jong (2008, 2011) for details). The backward recursion is thus obtained by defining a finite grid of G steps for the admissible capacities c of the plant, $c \in \{C_{min} = c_1, \ldots, C_{max} = c_G\}$, and then by approximating the continuation value per volume step $g = 1, \ldots, G$ and $m = 1, \ldots, M$ with a linear regression

$$\boldsymbol{E}\left[J(t_{m+1}, S(t_{m+1}), \tilde{c}_k) | S(t_m), C(t_m) = c_g\right] \simeq a_0 + a_1 S(t_m) + \ldots, + a_B S^B(t_m).$$

In our experiments, we consider an equally-spaced volume grid with $G = 100$ and use simple power polynomials with $B = 3$, but the regression may be performed on a different set of basis functions as well (see Boogert and de Jong (2011) for a comparison with other basis functions).

We focus then on the LSMC methodology and perform a few numerical experiments selecting the three-factors spot model with the jump component covered by the second case in Section 2 because we want to capture asymmetric jumps (we set $H(0) = 0$): in this case, because of (3) and (20) for $\beta_1, \beta_2 > 1$ it results

$$h(t) = -\frac{\sigma^2}{4\rho}\left(1 - e^{-2\rho t}\right) - \frac{\lambda_1}{k_1}\log\left(\frac{\beta_1 - e^{-k_1 t}}{\beta_1 - 1}\right) - \frac{\lambda_2}{k_2}\log\left(\frac{\beta_2 - e^{-k_2 t}}{\beta_2 - 1}\right).$$

Going back to the initial problem, without loss of generality we consider gas storages only, indeed, apart from a potential hourly stochastic optimization, the pricing technique

Table 3. Parameters for spot (day-ahead) dynamics (case 2).

S_0	ρ	σ	k_1	k_2	λ_1	λ_2	β_1	β_2
22	67	0.25	50	40	20	20	10	20

Table 4. Fast storage parameters.

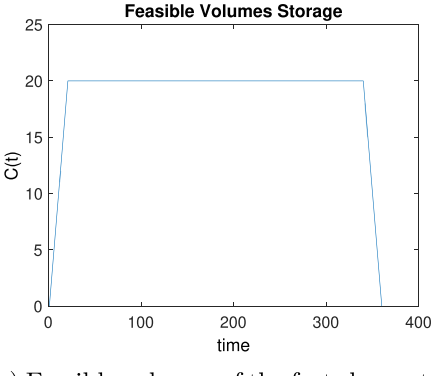
$C(0)$	$C(T)$	a_{in}	a_w	C_{max}
0	0	1	1	20

for VHP or VPS is practically the same. We assume that the units of $C(0), C(T)$ and C_{max} are in MWh, those of the injection and withdrawal rates are in MWh/day, whereas S_0 can be taken in €/MWh; in addition we suppose a flat forward curve. The remaining model parameters are shown in Table 3 and can be considered realistic.

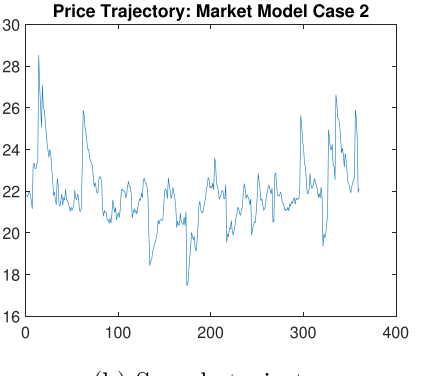
Finally, we consider a one-year ($M = 360$) fast-churn storage with the parameters shown in Table 4 such that 20 days are required to fill or empty the storage as shown in Figure 2(a).

In line with that observed for the pricing of Asian options, Table 5 and Figure 3(a) show that the three types of implementation apparently return comparable gas storage values. Moreover, defining the benchmark as the MC estimated price with Algorithm 2 with $N_S = 10^6$ simulations, one can observe that all approaches are equally convergent with an acceptable error already with $N_S = 2 \times 10^4$.

On the other hand, the ratio of total CPU times (CPUTOT in Figure 3(b)) is not as extreme compared to the Asian option case. Algorithm 3 is 'only' 40% faster, in the worst case, compared to the other two solutions. The reason of this apparent different conclusion compared to the previous section is that the pricing procedure consists of two steps: the simulation of the price trajectory and the stochastic optimization, the latter one being the more computationally intensive. To this end, Table 5 also displays the CPU times required for the path simulation only (CPUPATH) where one can observe that Algorithm 3 is once again tens of times faster. Using Algorithms 2 and 4 the path generation step has a relevant impact on the total time, whereas using our approach it is as if the overall cost coincides with that required by the stochastic optimization. This



(a) Feasible volumes of the fast churn storage.

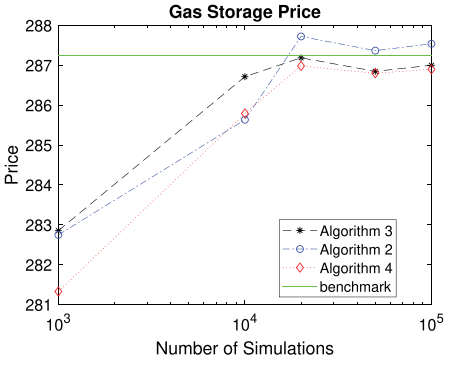


(b) Sample trajectory.

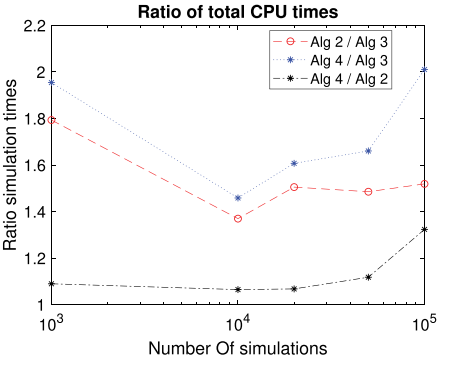
Figure 2. Gas storages.

Table 5. Storage evaluation: prices and computational times in seconds. CPU_{TOT} represents the total time, whereas CPU_{PATH} is time required by the path-generation only.

	Algorithm 3			Algorithm 2			Algorithm 4		
N_S	Price	CPU_{TOT}	CPU_{PATH}	Price	CPU_{TOT}	CPU_{PATH}	Price	CPU_{TOT}	CPU_{PATH}
10^3	282.7	8.5	0.3	282.9	15.3	6.03	281.3	16.7	7.4
10^4	285.6	136.3	1.8	286.7	186.6	57.09	285.8	198.7	69.2
2×10^4	287.7	237.2	3.3	287.2	357.1	113.85	287.0	381.3	138.1
5×10^4	287.4	674.7	8.3	286.8	1002.1	290.34	286.8	1063.7	351.9
10^5	287.5	1197.0	17.9	287.0	1818.3	582.05	286.9	1936.5	700.3

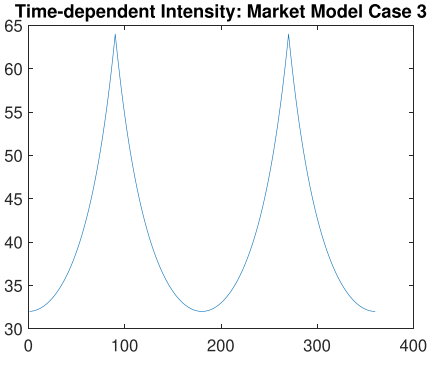


(a) Prices gas storage.

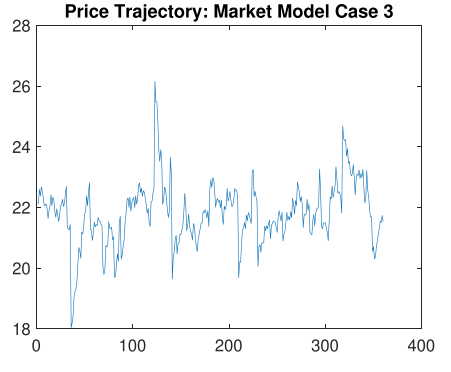


(b) Ratio of total CPU times.

Figure 3. Gas storage results.



(a) Time-dependent intensity.



(b) Sample trajectory.

Figure 4. Market model.

fact provides a computational advantage when one needs to calculate the sensitivities of the storage because a high number of simulations is required.

We finally remark that all the approaches rely on the sequential simulation of the price trajectory forward in time. In combination with LSMC methods, this is not the optimal approach because the entire set of trajectories and simulations are stored in memory with

a risk of memory allocation issues. For instance, Pellegrino and Sabino (2015) and Sabino (2020b) have shown that the backward simulation is preferable with LSMC. Unfortunately, although we know the law of the standard Gaussian-OU bridge, we do not know the law of the Γ-OU bridge which will be one of the topics of our future studies.

4.3. Numerical Experiments: Swings

A swing option is a type of contract used by investors in energy markets that lets the option holder buy a predetermined quantity of energy at a predetermined price (strike), while retaining a certain degree of flexibility in both the amount purchased and the price paid.

We consider a 120-120 swing option having a one year maturity and $M = 360$ with the specifications of Table 7. Such a contract can also be seen as a simplified gas storage where $a_{in} = 0$, $K_N = 0$ and K_w is the strike of the contract, therefore we rely on the LSMC method illustrated before by plugging $C(0) = 120$, $C(T) = 0$, $a_{in} = 0$, $a_w = 1$, $C_{max} = 120$ into ((28)) with an injection cost equal to the strike (see Hambly, Howison, and Kluge (2009a) for an application of the LSMC method to the pricing of swing options). In alternative, one can also use the tree method of Jaillet, Ronn, and Tompaidis (2004) or the Fourier cosine expansion in Zhang and Oosterlee (2013b) taking advantage of the explicit form of the *chf* of the process.

In this last example, we now choose the third market model in Section 2 that consists in a two-factors model with one Gaussian OU diffusion and one symmetric biΓ-OU process – a compound Poisson with Laplace jumps– where once more we set $H(0) = 0$. We also consider a step-wise daily approximation of the following time-dependent intensity

$$\lambda(t) = \frac{2\theta}{1 + |sin(\pi\omega(t - \tau))|} \tag{32}$$

so that for $m = 1, \ldots, M$ and $\beta > 1$ we have

$$h(t_m) = \frac{\sigma^2}{4\rho}\left(1 - e^{-2\rho\, t_m}\right) - \frac{\lambda_m}{2k}\log\left(\frac{\beta^2 - e^{-2kt_m}}{\beta^2 - 1}\right)$$

with the parameters of Table 6; we also consider a flat forward curve. The value of θ is selected such that the average number of jumps per year is about 40 as in the storage example.

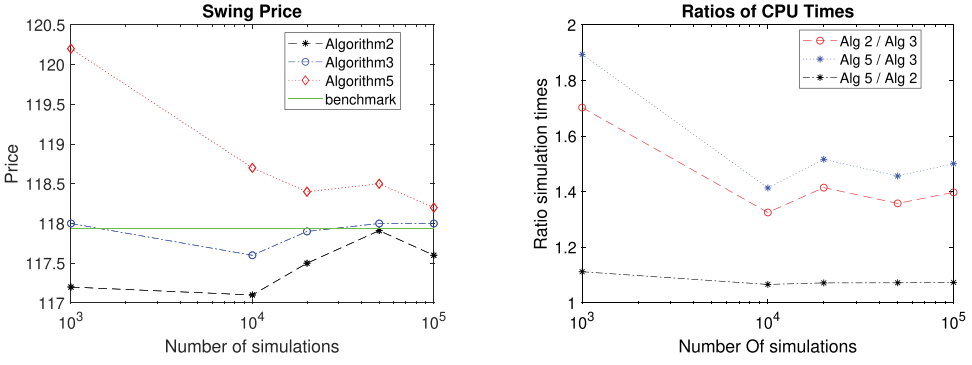
Due to the fact that jump component has now symmetric Laplace jumps, the simulation of the skeleton of $Y(\cdot)$ can be accomplished adapting Algorithms 2, 3 to the case of

Table 6. Parameters for spot (day–ahead) dynamics (case 3).

S_0	ρ	σ	k	θ	ω	τ	β
22	67	0.25	50	32	2	0.25	20

Table 7. Parameters of 120-120 take-or-pay swing.

ACQmin	*ACQmax*	*DCQmin*	*DCQmax*	*M(days)*
120	120	1	1	360



(a) Prices of a 120-120 Swing. (b) Ratio of CPU times.

Figure 5. Swings.

a symmetric biΓ-OU process or using Algorithm 5. The time-dependent intensity function and a sample trajectory of the price are shown in Figure 4.

The conclusions that we can derive from the numerical experiments are very much in line with what is observed for the gas storages. We also define the benchmark as the MC estimated price with Algorithm 2 with $N_S = 10^6$ simulations. As expected, the estimated values of the swing option obtained with the three types of implementation are similar and setting 2×10^4 is already sufficient to achieve a small error as visible in Figure 5(a).

Table 8 and Figure 5(b) show that the total CPU times (CPU_{TOT} in Table 8) using Algorithm 3 are far lower that those required by the other solutions resulting in a competitive advantage of about 40% (in the worst case). This factor becomes even higher if one focuses on the time required by the path-generation (CPU_{PATH} in Table 8).

The contribution of the stochastic optimization step to the total cost is again of about 75% using Algorithm 2 or Algorithm 5, while instead, with our Algorithm 3, the impact of the path generation step becomes almost negligible. We can therefore conclude that Algorithm 3 is the preferable solution for the simulation of the jump component in the market model (4).

5. Conclusions and Future Inquiries

In this paper we have considered the problem of pricing complex energy derivatives with Monte Carlo simulations using mean-reverting jump-diffusion market models. The

Table 8. Swing evaluation: prices and computational times in seconds. CPUTOT represents the total time, whereas CPUPATH is time required by the path-generation only.

	Algorithm 3			Algorithm 2			Algorithm 5		
N_S	Price	CPU_{TOT}	CPU_{PATH}	Price	CPU_{TOT}	CPU_{PATH}	Price	CPU_{TOT}	CPU_{PATH}
10^3	118.0	7.2	0.2	117.2	12.3	5.32	120.2	13.6	5.3
10^4	117.6	137.2	1.2	117.1	181.8	50.45	118.7	193.9	50.4
2×10^4	117.9	237.4	2.4	117.5	335.9	102.04	118.4	360.1	102.0
5×10^4	118.0	625.4	5.7	117.9	849.2	257.82	118.5	910.8	257.8
10^5	118.0	1147.2	11.7	117.6	1603.5	519.85	118.2	1721.7	519.8

jump component that we have chosen is a compound Poisson process with exponentially or bilateral exponentially distributed jumps known in the literature as Γ-OU process or biΓ-OU process, respectively. Although, this is a simple and standard approach, the simulation of the price trajectories may soon become very computational expensive, especially for the pricing of complex derivative contracts. Indeed, the generation of skeleton of the jump process has a relevant impact on the total computational cost.

Based on our results in Sabino and Cufaro Petroni (2020), we have designed new and fast algorithms for the simulation of the spot prices that potentially could be used for real-time pricing. In addition, in contrast to some other simulation schemes based on numerical approximations, our approach is applicable with parameters reflecting any market condition.

We have illustrated the applications of our findings in the context of the pricing of Asian options with standard Monte Carlo and gas storages and swings adopting the Least-Squares Monte Carlo method introduced in Boogert and de Jong (2008). The total computational effort depends on the cost of the path simulation and on that of the stochastic optimization (this last step is not influenced by the particular simulation algorithm).

We have conducted extensive Monte Carlo based simulation experiments and compared the computational performance of our proposal to the traditional approach of Cont and Tankov (2004) and a recent methodology described by Qu, Dassios, and Zhao (2019). Our numerical experiments have shown that our solution outperforms any other alternative because it cuts the simulation time down by a factor larger than forty in the case of Asian options and to a factor of forty percent for the gas storages and swings. In contrast to the other Monte Carlo-based approaches, the numerical tests suggest that our simulation methodology is suitable for real-time pricing.

In a primarily economic and financial perspective, future studies could cover the extension to a multidimensional setting with correlated Poisson processes as those introduced for instance in Lindskog and McNeil (2003) or in Cufaro Petroni and Sabino (2017, 2020), or could investigate the application of other Lévy -driven Ornstein-Uhlebeck processes to the modelling of price dynamics, for instance, using tempered stable and CGMY processes of Ornstein-Uhlenbeck type and relying on the results of Petroni and Sabino (2020).

A last topic deserving further investigation is the enhancement of the computational speed of the Least-Squares Monte Carlo using backward simulations generalizing the results of Pellegrino and Sabino (2015) and Sabino (2020b) to Γ-OU and biΓ-OU processes.

Disclosure statement

No potential conflict of interest was reported by the author(s). The views, opinions, positions or strategies expressed in this article are those of the authors and do not necessarily represent the views, opinions, positions or strategies of, and should not be attributed to E.ON SE.

ORCID

Piergiacomo Sabino http://orcid.org/0000-0003-2072-2353
Nicola Cufaro Petroni http://orcid.org/0000-0001-7857-9081

References

Bardou, O., S. Bouthemy, and G. Pagés. 2009. "Optimal Quantization for the Pricing of Swing Options." *Applied Mathematical Finance* 16 (2): 183–217. doi:10.1080/13504860802453218.

Barndorff-Nielsen, O. E., J. L. Jensen, and M. Sørensen. 1998. "Some Stationary Processes in Discrete and Continuous Time." *Advances in Applied Probability* 30 (4): 989–1007. doi:10.1239/aap/1035228204.

Barndorff-Nielsen, O. E., and N. Shephard. 2001. "Non-Gaussian Ornstein-Uhlenbeck- Based Models and Some of Their Uses in Financial Economics." *Journal of the Royal Statistical Society: Series B* 63 (2): 167–241. doi:10.1111/1467-9868.00282.

Ben-Ameur, H., M. Breton, L. Karoui, and P. L'Ecuyer. 2007. "A Dynamic ProgrammingApproach for Pricing Options Embedded in Bonds." *Journal of Economic Dynamics and Control* 31 (July): 2212–2233. doi:10.1016/j.jedc.2006.06.007.

Benth, F. E., and A. Pircalabu. 2018. "A Non-gaussian Ornstein-Uhlenbeck Model for Pricing Wind Power Futures." *Applied Mathematical Finance* 25 (1): 36–65. doi:10.1080/1350486X.2018.1438904.

Benth, F. E., J. Kallsen, and T. Meyer-Brandis. 2007. "A non-Gaussian Ornstein- Uhlenbeck Process for Electricity Spot Price Modeling and Derivatives Pricing." *Applied Mathematical Finance* 14 (2): 153–169. doi:10.1080/13504860600725031.

Benth, F. E., L. Di Persio, and S. Lavagnini. 2018. "Stochastic Modeling of Wind Derivatives in Energy Markets." *Risks, MDPI, Open Access Journal* 6 (2): 1–21.

Bertsekas, D. P. 2005. *Dynamic Programming and Optimal Control, Volume I*. 3rd ed. Belmont, MA: Athena Scientific.

Boogert, A., and C. de Jong. 2008. "Gas Storage Valuation Using a Monte Carlo Method." *Journal of Derivatives* 15 (3): 81–91. doi:10.3905/jod.2008.702507.

Boogert, A., and C. de Jong. 2011. "Gas Storage Valuation Using a Multifactor Price Model." *The Journal of Energy Markets* 4: 29–52. doi:10.21314/JEM.2011.067.

Cartea, A., and M. Figueroa. 2005. "Pricing in Electricity Markets: A Mean Reverting Jump Diffusion Model with Seasonality." *Applied Mathematical Finance, No. 4, December 2005* 12 (4): 313–335.

Cont, R., and P. Tankov. 2004. *Financial Modelling with Jump Processes*. New York, NY: Chapman and Hall.

Cufaro Petroni, N. 2008. "Self-decomposability and Self-similarity: A Concise Primer." *Physica A, Statistical Mechanics and Its Applications* 387 (7–9): 1875–1894. doi:10.1016/j.physa.2007.11.036.

Cufaro Petroni, N., and P. Sabino. 2017. "Coupling Poisson Processes by Selfdecomposability." *Mediterranean Journal of Mathematics* 14 (2): 69. doi:10.1007/s00009-017-0847-4.

Cufaro Petroni, N., and P. Sabino. 2020. "Tempered Stable Distribution and Finite Variation Ornstein-Uhlenbeck Processes." https://arxiv.org/abs/2011.09147

Cufaro Petroni, N., and P. Sabino. 2020. "Pricing Exchange Options with Correlated Jump Diffusion Processes." *Quantitative Finance* 20 (11): 1811–1823.

Cummins, M., G. Kiely, and B. Murphy. 2017. "Gas Storage Valuation under Lévy Processes Using Fast Fourier Transform." *Journal of Energy Markets* 4: 43–86. doi:10.21314/JEM.2017.169.

Cummins, M., G. Kiely, and B. Murphy. 2018. "Gas Storage Valuation under Multifactor Lévy Processes." *Journal of Banking and Finance* 95: 167–184. doi:10.1016/j.jbankfin.2018.02.012.

Deng, S. 2000. *Stochastic Models of Energy Commodity Prices and Their Applications: Mean-reversion with Jumps and Spikes*. Citeseer.

Fusai, G., and I. Kyriakou. 2016. "General Optimized Lower and Upper Bounds for Discrete and Continuous Arithmetic Asian Options." *Mathematics of Operations Research* 41 (2): 531–559. doi:10.1287/moor.2015.0739.

Hambly, B., S. Howison, and T. Kluge. 2009a. "Information-Based Models for Finance and Insurance." *Quantitative Finance* 9 (8): 937–949.

Hambly, B., S. Howison, and T. Kluge. 2009b. "Modelling Spikes and Pricing Swing Options in Electricity Markets." *Quantitative Finance* 9 (8): 937–949.

Jaillet, P., E. I. Ronn, and S. Tompaidis. 2004. "Valuation of Commodity-Based Swing Options." *Management Science* 50 (7): 909–921. doi:10.1287/mnsc.1040.0240.

Jaimungal, S., and V. Surkov. 2011. "Lévy Based Cross-commodity Models and Derivative Valuation." *SIAM Journal of Financial Mathematics* 2: 464–487. doi:10.1137/100791609.

Kjaer, M. 2008. "Pricing of Swing Options in a Mean Reverting Model with Jumps." *Applied Mathematical Finance* 15 (5–6): 479–502. doi:10.1080/13504860802170556.

Kluge, T. 2006. "Pricing Swing Options and Other Electricity Derivatives." Technical report. PhD thesis, University of Oxford. http://perso-math.univ-mlv.fr/users/bally.vlad/publications.html

Kou, S. G. 2002. "A Jump-Diffusion Model for Option Pricing." . *Management Science* 48 (August): 1086–1101. doi:10.1287/mnsc.48.8.1086.166.

Küchler, U., and S. Tappe. 2008. "Bilateral Gamma Distributions and Processes in Financial Mathematics." *Stochastic Processes and Their Applications* 118 (2): 261–283. doi:10.1016/j.spa.2007.04.006.

Kyriakou, I., P. K. Pouliasis, N. C. Papapostolou, and K. Andriosopoulos. 2017. "Freight Derivatives Pricing for Decoupled Mean-reverting Diffusion and Jumps." *Transportation Research Part E: Logistics and Transportation Review* 108: 80–96. doi:10.1016/j.tre.2017.09.002.

Lindskog, F., and J. McNeil. 2003. "Common Poisson Shock Models: Applications to Insurance and Credit Risk Modelling." *ASTIN Bulletin* 33 (2): 209–238. doi:10.1017/S0515036100013441.

Longstaff, F. A., and E. S. Schwartz. 2001. "Valuing American Options by Simulation: A Simple Least-Squares Approach." *Review of Financial Studies* 14 (1): 113–147. doi:10.1093/rfs/14.1.113.

Lucia, J. J., and E. S. Schwartz. 2002. "Electricity Prices and Power Derivatives: Evidence from the Nordic Power Exchange." *Review of Derivatives Research* 5 (January): 5–50. doi:10.1023/A:1013846631785.

Nomikos, N. K., and O. Soldatos. 2008. "Using Affine Jump Diffusion Models for Modelling and Pricing Electricity Derivatives." *Applied Mathematical Finance* 15 (1): 41–71. doi:10.1080/13504860701427362.

Pellegrino, T., and P. Sabino. 2015. "Enhancing Least Squares Monte Carlo with Diffusion Bridges: An Application to Energy Facilities." *Quantitative Finance* 15 (5): 761–772. doi:10.1080/14697688.2014.941913.

Qu, Y., A. Dassios, and H. Zhao. 2019. "Exact Simulation of Gamma-driven Ornstein–Uhlenbeck Processes with Finite and Infinite Activity Jumps." *Journal of the Operational Research Society* 1–14.

Sabino, P. 2020a. "Exact Simulation of Variance Gamma Related OU Proceses: Application to the Pricing of Energy Derivatives." *Applied Mathematical Finance* 27 (3): 207–227. doi:10.1080/1350486X.2020.1813040.

Sabino, P. 2020b. "Forward or Backward Simulations? A Comparative Study." *Quantitative Finance* 20 (7): 1213–1226. doi:10.1080/14697688.2020.1741668.

Sabino, P., and N. Cufaro Petroni. 2020. "Gamma-related Ornstein–Uhlenbeck Processes and Their Simulation*." *Journal of Statistical Computation and Simulation* 1–26.

Sato, K. 1999. *Lévy Processes and Infinitely Divisible Distributions*. Cambridge: Cambridge U.P.

Schoutens, W. 2003. *Lévy Processes in Finance: Pricing Financial Derivatives*. John Wiley and Sons.

Schwartz, P., and J. E. Smith. 2000. "Short-term Variations and Long-term Dynamics in Commodity Prices." *Management Science* 46 (7): 893–911. doi:10.1287/mnsc.46.7.893.12034.

Zhang, B., and C. W. Oosterlee. 2013a. "An Efficient Pricing Algorithm for Swing Options Based on Fourier Cosine Expansions." *Journal of Computational Finance* 16 (4): 1–32. doi:10.21314/JCF.2013.268.

Zhang, B., and C. W. Oosterlee. 2013b. "Efficient Pricing of European-style Asian Options under Exponential Lévy Processes Based on Fourier Cosine Expansions." *SIAM Journal on Financial Mathematics* 4 (1): 399–426. doi:10.1137/110853339.