Journal of Computational Science 28 (2018) 120–139

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

SWIFT valuation of discretely monitored arithmetic Asian options

Álvaro Leitao[a,b,*], Luis Ortiz-Gracia[c], Emma I. Wagner[d]

[a] Barcelona Graduate School of Mathematics (BGSMath), Spain
[b] Department of Econometrics, Riskcenter-IREA, University of Barcelona, Spain
[c] Department of Econometrics, Statistics and Applied Economics, University of Barcelona, Spain
[d] Delft Institute of Applied Mathematics, Delft University of Technology, The Netherlands

ARTICLE INFO

Article history:
Received 4 March 2018
Received in revised form 15 June 2018
Accepted 25 July 2018
Available online 27 July 2018

MSC:
60E10
60G51
60G52
65T60
65T50

Keywords:
Arithmetic Asian options
Fourier transform
Shannon wavelets
SWIFT method
Exponential Lévy processes
Square-root diffusions
Option pricing

ABSTRACT

In this work, we propose an efficient and robust valuation of discretely monitored arithmetic Asian options based on Shannon wavelets. We employ the so-called *SWIFT* method, a Fourier inversion numerical technique with several important advantages with respect to the existing related methods. Particularly interesting is that SWIFT provides mechanisms to determine all the free-parameters in the method, based on a prescribed precision in the density approximation. The method is applied to two general classes of dynamics: exponential Lévy models and square-root diffusions. Through the numerical experiments, we show that SWIFT outperforms state-of-the-art methods in terms of accuracy and robustness, and shows an impressive speed in execution time.

1. Introduction

In Asian derivatives, the option payoff function relies on some *average* of the underlying values at a prescribed monitoring dates. This fact implies that the final value is less volatile and the option price cheaper. In this work, we will primarily focus on the arithmetic average, i.e. the so-called *arithmetic Asian* options, due to its financial importance and, as we will see, challenging numerical treatment.

In the recent literature of arithmetic Asian valuation, multiple dynamics have been considered, of which two broad classes are addressed here: *exponential Lévy models* and *square-root diffusions*. Exponential Lévy processes have gained popularity in the last decades. Their special definition facilitates the mathematical treatment, usually based on the availability of the *characteristic function*, i.e. the Fourier transform of the density function, and the use of a Fourier inversion technique. The methods relying on Fourier inversion are highly appreciated, particularly for calibration purposes, since they are extremely fast, very accurate and easy to implement. The main drawback attributed to these type of methods is the lack of control in the free-parameter's setting (see [1], for example, and the references therein). The Fourier inversion approaches to price arithmetic Asian products are typically based on the work of Carverhill and Clewlow in [2], where a generic analytical algorithm was introduced. Benhamou in [3] proposed an improved version of Carverhill and Clewlow algorithm applied to arithmetic Asian options via *fast Fourier transform*. In [4] the probability density function of the logarithm of the sum of asset prices is obtained by a series of recursive quadratures. The strategy considered in [5] shows how the characteristic function of the logarithm of the sum of asset prices is obtained numerically by means of a Fourier cosine expansion. The work in [6] follows the same approach, although the cosine expansion is replaced by B-splines. A key aspect about the efficiency of those methods is the numerical computation of

[*] Corresponding author at: Department of Econometrics, Riskcenter-IREA, University of Barcelona, Spain.
E-mail addresses: leitao@ub.edu (Á. Leitao), luis.ortiz-gracia@ub.edu (L. Ortiz-Gracia), emmaingeleiv@gmail.com (E.I. Wagner).

https://doi.org/10.1016/j.jocs.2018.07.004
1877-7503/

certain integrals appearing in the procedure of recovering the characteristic function. While in [5] a Clenshaw–Curtis quadrature is used, a Newton–Cotes quadrature is preferred in [6]. The number of points selected to carry out the quadrature in [5] is tightly related to the number of terms in the cosine expansion (which is unknown a priori) and the valuation of the option may be a matter of trial and error. Moreover, the truncation interval choice appears to be an issue in both cases. Some other approaches have been recently proposed in the literature, based on backward convolution [7], maturity randomization [8] or backward induction [1].

Other important types of asset dynamics are the square-root diffusions. These models are typically found in commodity or interest rate markets, where their properties are very appreciated by the practitioners. In [9], the authors developed a recursive algorithm to obtain the *moment generating function* of the joint distribution of the asset price and its weighted (arithmetic) average for various square-root dynamics, including time-dependent processes. As a numerical inversion technique, *Carr–Madan* method [10] was adopted. In [11], a Malliavin-calculus-based asymptotic expansion for the discretely-monitored Asian option pricing is proposed that can be applied to several diffusion models, but with important practical restrictions in the implementation. Recently, Cai et al. introduce in [12] a general framework for Asian valuation based on that any one-dimensional Markov process can be transformed in a continuous-time Markov chain and then solve the resulting pricing problem under this context. However, this methodology constitutes an approximation in which the errors can be arbitrarily big.

In this work, we employ the wavelets-based *SWIFT* method [13] for pricing arithmetic Asian options under both exponential Lévy models and square-root diffusions. We focus on the derivation of the characteristic function of the associated transitional density appearing in the option pricing problem. Then, the characteristic function is used to recover the series expansion coefficients of the density. SWIFT has already been successfully applied to European options with one and two underlying assets in [13,14] respectively, as well as to early-exercise and discrete barrier options in [15]. It has been shown that among the strengths of SWIFT method is the a priori knowledge of the scale of approximation and equivalently the number of terms in the expansion. Thus, the method provides a great robustness in terms of the free-parameters configuration, given closed-form expressions to determine them based on the level of approximation desired. This constitutes the major advantage of the SWIFT method with respect to other competitor methods, overcoming the parameter selection issue. According to [1], this is precisely the main disadvantage of the techniques in [5,6]. Furthermore, the method is extremely fast and accurate, showing exponential convergence for most of the interesting processes in finance.

In the context of Lévy processes, our approach is conceptually similar to the one in [5,6], but the use of wavelets results in a remarkable improvement. Thanks to the Shannon wavelets properties, we circumvent the numerical computation of the aforementioned integrals (Clenshaw–Curtis in [5] and Newton–Cotes in [6]), avoiding quadrature rules during the recovery of the density function and making the task of pricing Asian options simple in its implementation, very accurate when comparing with benchmark solutions from the literature and extremely fast in its computation. Moreover, the integrals are evaluated in the whole domain, instead of the truncated one like in [5,6], avoiding this way truncation errors.

For the square-root diffusions, we take advantage of the Fusai et al. work [9] and the close relation between the moment generating function and the characteristic function. Again, the use of the SWIFT method in this framework constitutes a significant gain in both precision and efficiency, with respect to the existing methods. The control in the free-parameter choice allows us to avoid numerical errors that can appear in difficult/extreme situations when the "rule-of-thumb" recommendations can fail. In addition, the use of SWIFT allows us to define closed-form formulas for many types of payoff functions, even the ones appearing in interest rate markets. This is a novelty for the techniques based on Fourier inversion.

Furthermore, in this work we develop not only efficient solutions for arithmetic Asian option price but also similar expressions for the option sensitivities, the so-called *Greeks*, the partial derivatives of the option price with respect to a model parameter. The Greeks computation is not a trivial task and of crucial importance in managing the financial risk, which generally implies an extra computational cost or the use of other advanced techniques. SWIFT, however, enables us to straightforwardly obtain the option price and the option sensitivities at once and in a very simple manner.

The paper is organized as follows. Section 1.1 formulates the Asian pricing problem and sets the notation used throughout the paper. We introduce the basis of the SWIFT method in Section 2. The application of SWIFT to the valuation of arithmetic Asian options under exponential Lévy models and square-root diffusions are described in Sections 3 and 4, respectively. Accordingly, we present the conducted numerical experiments in Section 5. Finally, we conclude in Section 6.

1.1. Problem formulation

Lets consider $N+1$ intermediate monitoring dates $t_i \in [0, T]$, $i=0, \ldots, N$ between today, $t_0=0$ and the maturity of the option contract, $t_N=T$, where underlying price, $S(t)$, is observed. We assume here that the initial state of the price process is known, $S(0)=S_0$ and the monitoring dates are equally spaced, i.e. $\Delta t := t_{i+1} - t_i$, $i=0, \ldots, N-1$. The averaged price can be defined according to different metrics. Here we consider a usual arithmetic average, $A_N := \frac{1}{N+1}\sum_{i=0}^{N} S(t_i)$, and a more generic weighted average, $C_N := \sum_{i=0}^{N} \omega_i S(t_i)$, where ω_i are a predefined weights such that $\sum_{i=0}^{N} \omega_i = 1$. A range of payoff functions can be adopted. Let $H_N \in \{A_N, C_N\}$, the payoff of the *European-style* Asian option is defined as follows

$$v(S,T) = \begin{cases} (H_N - K)^+ & \text{for a call,} \\ (K - H_N)^+ & \text{for a put.} \end{cases} \tag{1}$$

When not only the average but also the final state of the underlying price is considered, we have the so-called *floating-strike* Asian option

$$v(S,T) = \begin{cases} \left(\theta\left(S(T) - H_N\right) - K\right)^+ & \text{for a call,} \\ \left(K - \theta\left(S(T) - H_N\right)\right)^+ & \text{for a put,} \end{cases} \tag{2}$$

where $\theta \in \{-1, +1\}$. The price of floating-strike contracts can also be computed by using the equivalence relationship with the *fixed*-strike European-style instruments, presented in [16], valid only in the framework of general Lévy processes.

Finally, we also consider some special Asian-like payoffs typically appearing in interest rate markets. An interesting particularity is that, in this case, the average quantity shows up twice, as the discounted factor also depends on the average and needs to be included within the payoff function. Among them, we can find *regular Asian caps*, *cash Binary Asian caps* and *rate Binary Asian caps* (see [17,18]), whose payoff is

$$\begin{aligned} v(S,T) &= \mathrm{e}^{-H_N T}(H_N - K)^+, \\ v(S,T) &= \mathrm{e}^{-H_N T}1_{\{H_N > K\}}, \quad \text{and} \\ v(S,T) &= H_N T \mathrm{e}^{-H_N T}1_{\{H_N > K\}}, \end{aligned} \tag{3}$$

respectively. The equivalent *floor* contracts can be similarly formulated, but we do not reproduce them here for sake of brevity.

2. The SWIFT method

In this section we give a brief review on the SWIFT method [13], originally developed for pricing European options. In Sections 3 and 4, we will extend the method to Asian options. For sake of completeness we devote a section to the basic theory on Shannon wavelets.

2.1. Multi-resolution analysis and Shannon wavelets

Consider the space of square-integrable functions, denoted by $L^2(\mathbb{R})$. A general structure for wavelets in $L^2(\mathbb{R})$ is called a *multi-resolution analysis*. We start with a family of closed nested subspaces in $L^2(\mathbb{R})$,

$$\ldots \subset \mathcal{V}_{-2} \subset \mathcal{V}_{-1} \subset \mathcal{V}_0 \subset \mathcal{V}_1 \subset \mathcal{V}_2 \subset \ldots, \quad \bigcap_{m\in\mathbb{Z}} \mathcal{V}_m = \{0\}, \quad \overline{\bigcup_{m\in\mathbb{Z}} \mathcal{V}_m} = L^2(\mathbb{R}),$$

where

$$f(x) \in \mathcal{V}_m \Leftrightarrow f(2x) \in \mathcal{V}_{m+1}.$$

If these conditions are met, then there exists a function $\varphi \in \mathcal{V}_0$ that generates an orthonormal basis, denoted by $\{\varphi_{m,k}\}_{k\in\mathbb{Z}}$, for each $\mathcal{V}_m$ subspace, $\varphi_{m,k}(x) = 2^{m/2}\varphi(2^m x - k)$. The function φ is usually referred to as the *scaling function* or *father wavelet*.

For any $f \in L^2(\mathbb{R})$, a projection map of $L^2(\mathbb{R})$ onto $\mathcal{V}_m$, denoted by $\mathcal{P}_m : L^2(\mathbb{R}) \to \mathcal{V}_m$, is defined by means of

$$\mathcal{P}_m f(x) = \sum_{k\in\mathbb{Z}} c_{m,k}\varphi_{m,k}(x), \quad \text{with} \quad c_{m,k} = \langle f, \varphi_{m,k}\rangle, \tag{4}$$

where $\langle f, g\rangle := \int_{\mathbb{R}} f(x)\bar{g}(x)\mathrm{d}x$ denotes the inner product in $L^2(\mathbb{R})$, with $\bar{g}$ being the complex conjugate of g. Considering higher m values (i.e. when more terms are used), the truncated series representation of the function f improves. As opposed to Fourier series, a key fact regarding the use of wavelets is that wavelets can be moved (by means of the k value), stretched or compressed (by means of the m value) to accurately represent the local properties of a function. A basic reference on wavelets is [19].

In this paper, we employ Shannon wavelets [20]. A set of Shannon scaling functions $\varphi_{m,k}$ in the subspace $\mathcal{V}_m$ is defined as,

$$\varphi_{m,k}(x) = 2^{m/2}\frac{\sin(\pi(2^m x - k))}{\pi(2^m x - k)} = 2^{m/2}\varphi(2^m x - k), \quad k \in \mathbb{Z}, \tag{5}$$

where

$$\varphi(z) = \operatorname{sinc}(z) = \begin{cases} \dfrac{\sin(\pi z)}{\pi z} & \text{if } z \neq 0, \\ 1 & \text{if } z = 0, \end{cases}$$

is the scaling function, also called cardinal sine function.

Remark 1. Wavelets are a recent development in the Fourier option pricing context. Alternatively to Shannon wavelets, there is an important research line in methods focusing on local bases, i.e. with compact support (see [21,22], for example). Although these methods generally do not admit the multi-resolution framework, a dual-basis approach (biorthogonality) can be employed to recover the projection coefficients.

2.2. SWIFT method

Given a function $f \in L^2(\mathbb{R})$, we consider its expansion in terms of Shannon scaling functions at the level of resolution m. Our aim is to recover the coefficients $c_{m,k}$ of this approximation from the Fourier transform of the function f, denoted by $\hat{f}$, which is assumed to be known in closed-form. Here,

$$\hat{f}(\xi) = \int_{\mathbb{R}} \mathrm{e}^{-\mathrm{i}\xi x} f(x)\mathrm{d}x, \tag{6}$$

where i is the imaginary unit.

Following wavelets theory, a function $f \in L^2(\mathbb{R})$ can be approximated at the level of resolution m by,

$$f(x) \approx \mathcal{P}_m f(x) = \sum_{k\in\mathbb{Z}} c_{m,k}\varphi_{m,k}(x), \tag{7}$$

where $\mathcal{P}_m f$ converges to f in $L^2(\mathbb{R})$, i.e. $\|f - \mathcal{P}_m f\|_2 \to 0$, when $m \to +\infty$. Here, the coefficients $c_{m,k}$ and the scaling functions $\varphi_{m,k}$ are defined in Eqs. (4) and (5), respectively.

The infinite series in Eq. (7) is well-approximated (see Lemma 1 of [13] for details) by a finite summation without loss of considerable density mass,

$$\mathcal{P}_m f(x) \approx f_m(x) := \sum_{k=k_1}^{k_2} c_{m,k}\varphi_{m,k}(x), \tag{8}$$

for certain accurately chosen values k_1 and k_2.

The next step is the computation of the coefficients in Eq. (8). Recalling Eqs. (4) and (5), we have that,

$$c_{m,k} = \langle f, \varphi_{m,k}\rangle = \int_{\mathbb{R}} f(x)\overline{\varphi}_{m,k}(x)\mathrm{d}x = 2^{m/2}\int_{\mathbb{R}} f(x)\varphi(2^m x - k)\mathrm{d}x.$$

Using the classical Vieta's formula truncated with 2^{J-1} terms, the cosine product-to-sum identity and the definition of the characteristic function in Eq. (6), and after some algebraic manipulations (see [13] for details), the density coefficients, $c_{m,k}$, can be accurately approximated by

$$c_{m,k} \approx \frac{2^{m/2}}{2^{J-1}}\sum_{j=1}^{2^{J-1}} \Re\left[\hat{f}\left(\frac{(2j-1)\pi 2^m}{2^J}\right)\mathrm{e}^{\frac{\mathrm{i}k\pi(2j-1)}{2^J}}\right]. \tag{9}$$

Putting everything together gives the following approximation of f

$$f(x) \approx \sum_{k=k_1}^{k_2} c_{m,k}\varphi_{m,k}(x), \tag{10}$$

where $\varphi_{m,k}$ and $c_{m,k}$ are defined in Eqs. (5) and (9), respectively.

Remark 2. Eq. (9) can be conveniently rearranged to compute the coefficients with the use of the fast Fourier transform (FFT). For a detailed explanation we refer the reader to [13].

2.3. Option valuation formulas with SWIFT

The starting point to solve an option pricing problem is the risk-neutral option valuation formula, where the value of an option at time t and state x, $v(x, t)$, is an expectation under the risk neutral pricing measure, i.e.,

$$v(x,t) = \mathrm{e}^{-r(T-t)}\mathbb{E}[v(y,T)|x] = \mathrm{e}^{-r(T-t)}\int_{\mathbb{R}} v(y,T)f(y|x)\mathrm{d}y, \tag{11}$$

with r the risk-free rate, T the maturity time, $f(y|x)$ the transitional density and $v(y, T)$ represents the option value at maturity time, being the payoff function. The density function f is typically unknown, even for the case of European option pricing where the value of the variable y depends solely on the state at maturity T. In order to compute the value of the option $v(x, t_0)$ in Eq. (11), we can truncate the integration range on $[a, b]$ (we will show later how to select the interval safely) and we can replace the transitional probability function f by its approximation in Eq. (10). Accordingly the SWIFT pricing formula reads

$$v(x,t_0) \approx \mathrm{e}^{-rT}\sum_{k=k_1}^{k_2}\left(c_{m,k}\int_a^b v(y,T)\varphi_{m,k}(y|x)\mathrm{d}y\right) = \mathrm{e}^{-rT}\sum_{k=k_1}^{k_2} c_{m,k}V_{m,k}, \tag{12}$$

with the payoff coefficients $V_{m,k}$ defined as

$$V_{m,k} := \int_a^b v(y,T)\varphi_{m,k}(y|x)\mathrm{d}y.$$

We have reduced the computation of the option price to the dot product between the density coefficients, $c_{m,k}$, and the payoff coefficients, $V_{m,k}$. By employing the Vieta's formula again and interchanging summation and integration operations, we obtain that

$$V_{m,k} \approx \frac{2^{m/2}}{2^{J-1}}\sum_{j=1}^{2^{J-1}}\int_a^b v(y,T)\cos\left(\frac{2j-1}{2^J}\pi(2^m y - k)\right)\mathrm{d}y. \tag{13}$$

2.3.1. Option sensitivities

Under the SWIFT framework, the estimation of the option price sensitivities, the so-called *Greeks*, can be efficiently performed by constructing similar series expansions. The Greeks are defined as the partial derivatives of the option price with respect to some market parameter.

We firstly assume that the option price depends on the parameter of interest only through the density function. Then, the density coefficients (see Eq. (9)) can be rewritten as

$$c_{m,k}(\xi,\varsigma)=\frac{2^{m/2}}{2^{J-1}}\sum_{j=1}^{2^{J-1}}\Re\left[\hat{f}(\xi;\varsigma)e^{\frac{ik\xi}{2^m}}\right],$$

where $\xi=\frac{(2j-1)\pi 2^m}{2^J}$ and ς represents the parameter with respect to we wish to compute the derivative. The corresponding "Greek" density coefficients can be therefore obtained by differentiating (n times) the characteristic function, as follows

$$c_{m,k}^{(n)}(\xi):=\frac{\partial^n c_{m,k}(\xi,\varsigma)}{\partial\varsigma^n}=\frac{2^{m/2}}{2^{J-1}}\sum_{j=1}^{2^{J-1}}\Re\left[\frac{\partial^n\hat{f}\,(\xi;\varsigma)}{\partial\varsigma^n}e^{\frac{ik\xi}{2^m}}\right]. \tag{14}$$

For example, the first and second derivatives with respect to the initial state of the process, S_0, the so-called *Delta*, Δ, and *Gamma*, Γ, respectively, can be computed by plugging the $c_{m,k}^{(n)}$ into Eq. (12),

$$\Delta:=e^{-rT}\sum_{k=k_1}^{k_2}c_{m,k}^{(1)}V_{m,k},\quad \Gamma:=e^{-rT}\sum_{k=k_1}^{k_2}c_{m,k}^{(2)}V_{m,k}. \tag{15}$$

The partial derivative of the characteristic function and, hence, the $c_{m,k}^{(n)}$ coefficients can be analytically computed for many financial models.

A second possible situation appears when the option value depends on the parameter of interest, ς, through the payoff coefficients, i.e., $V_{m,k}(\varsigma)$. Thus, the "Greek" payoff coefficients need to be determined by differentiating $V_{m,k}$ with respect to ς. Particularly, the solution for the Greeks Δ and Γ would be

$$\Delta:=e^{-rT}\sum_{k=k_1}^{k_2}c_{m,k}V_{m,k}^{(1)}(\varsigma),\quad \Gamma:=e^{-rT}\sum_{k=k_1}^{k_2}c_{m,k}V_{m,k}^{(2)}(\varsigma), \tag{16}$$

where now the $c_{m,k}$ are kept invariant and $V_{m,k}^{(n)}$ represents the n-th derivative of $V_{m,k}$. The analytical availability of the $V_{m,k}^{(n)}$ depends on the payoff definition. In the context of Fourier inversion techniques, closed-form solutions for these coefficients can be usually derived. The case of the arithmetic Asian payoff will be addressed in the next section.

Remark 3. The pricing problems studied in this work fall into one of the situations described above. To the best of our knowledge, the option value will never depend on ς through both density and payoff coefficients. Whenever this case occurs, it would need to be treated as a special case.

2.4. Optimal scale m, series truncation bounds k$_1$ and k$_2$, and parameter J

The quality in the approximation provided by the SWIFT method is mainly affected by the scale m, the number of terms in the Vieta's approximation and the series truncation limits, k_1 and k_2 in Eq. (8). In order to optimally determine these parameters we follow the analysis presented in [23,15].

Lets start with the choice of m. It can be proved (see Lemma 3 of [15]) that the error in the projection approximation of function f in Eq. (7), i.e, $\epsilon_p:=|f(x)-\mathcal{P}_m f(x)|$, is bounded by

$$\epsilon_p\leq\frac{1}{2\pi}\int_{|\xi|>2^m\pi}\left|\hat{f}(\xi)\right|d\xi.$$

As the characteristic function, $\hat{f}$, is assumed to be known, we can compute m given a prescribed tolerance ϵ_m. The integral is typically not known in closed-form so it can be approximated by numerical techniques. Applying a simple quadrature rule, the error bound reads

$$\frac{1}{2\pi}\left(\left|\hat{f}(-2^m\pi)\right|+\left|\hat{f}(2^m\pi)\right|\right)\leq\epsilon_m.$$

This approximation is considered sufficient for our purposes. More involved numerical quadratures have been tested, but the observed differences are negligible.

Regarding truncation limits k_1 and k_2, they can be computed based on the integration range $[a, b]$ stated in Eq. (12), as

$$k_1:=\lfloor 2^m a\rfloor\quad\text{and}\quad k_2:=\lceil 2^m b\rceil,$$

where m is the scale of approximation. Therefore we first need to choose the interval limits, a and b, in such a way that the loss of density mass is minimized. We will provide different alternatives to determine the integration range $[a, b]$.

The dependence on m turns out to be very convenient also in the selection of the interval $[a, b]$, facilitating its choice given an approximation scale, m. This constitutes one of the great advantages of the SWIFT method with respect to other Fourier inversion-based techniques, where a and b are arbitrarily selected. Thus, as we know that our approximation at scale m satisfies the tolerance ϵ_m, the error order due to the truncation should not exceed the order of ϵ_m. We can therefore develop an adaptive interval selection algorithm that updates (in some way) the truncated range $[a, b]$ in each iteration, computes the committed error, ϵ (under some measure), in the approximated density using that interval and stops when the same tolerance condition ϵ_m is prescribed. The initial a and/or b can be chosen as suggested in [24,5]. Under these conditions, we can ensure that the overall error in the density is controlled by the scale parameter m. As we know that the density should go to zero in both a and b, the error ϵ is defined here to be the higher value of f evaluated in the interval limits, i.e. $f(a)$ and $f(b)$. In Algorithm 1, we can see a schematic representation of the adaptive procedure, where ϱ is a vector of the parameters involved in the change of $[a, b]$ (for example, it would be L in Eq. (34)).

Algorithm 1. Adaptive interval selection.

```
Data: $\epsilon_m$, $\varrho$
$[a, b] = \mathrm{interval}([\varrho])$ // Set initial interval [as a function of $\varrho$]
while $\epsilon > \epsilon_m$ do
    $f = \mathrm{SWIFT}(a, b)$ // Compute density $f$ in $[a, b]$ by Equation (10)
    $\epsilon = \max(|f(a)|, |f(b)|)$ // Compute the maximum error between the interval
        bounds
    $[a, b] = \mathrm{interval}(\varrho)$ // Update the interval in function of $\varrho$
return $[a, b]$
```

Next, the parameter J is then chosen to be constant[1] based on the previously determined quantities. Doing so, we can benefit from the use of FFT algorithm (further details in [23]). Thus,

$$J := \lceil \log(\pi M_m) \rceil \quad \text{with} \quad M_m := \max_{k_1<k<k_2} M_{m,k},$$

where $M_{m,k} := \max(|2^m \boldsymbol{\mathcal{A}} - k|, |2^m \boldsymbol{\mathcal{A}} + k|)$ and $\boldsymbol{\mathcal{A}} := \max(|a|, |b|)$.

3. SWIFT for arithmetic Asian options under exponential Lévy models

In this section we work with exponential Lévy models, i.e. we assume that the log-transformed underlying price, $\log S(t)$, follows a Lévy process. A Lévy process (see [25,26], for further details) is a stochastic process with stationary and independent increments that can be generally written as

$$X(t) = \mu t + W(t) + J(t) + \lim_{\varepsilon \downarrow 0} D^{\varepsilon}(t),$$

where W is a d-dimensional Brownian motion with covariance matrix Σ, drift vector $\mu \in \mathbb{R}^d$, J is a compound Poisson process and D^{ε} is a compensated compound Poisson process. A measure ν on $\mathbb{R}^d$ is adopted, called *Lévy measure*. The Lévy processes are therefore fully determined by the so-called *characteristic triplet* $[\Sigma, \mu, \nu]$. From the well-known *Lévy-Khintchine* formula, the characteristic function is given by

$$\mathbb{E}\left[\mathrm{e}^{\mathrm{i}\xi X(t)}\right] = \mathrm{e}^{t\vartheta(\xi)}, \quad \vartheta(\xi) = \mathrm{i}\mu\cdot\xi + \frac{12}{\Sigma}\xi\cdot\xi + \int_{\mathbb{R}^d}\left(\mathrm{e}^{\mathrm{i}\xi\cdot x} - 1 - \mathrm{i}\xi\cdot x 1_{|x|\le 1}\right)\nu(\mathrm{d}x),$$

where ϑ is often called the *characteristic exponent*. The explicit representation of the characteristic function in the Lévy processes framework supposes a great advantage and allows to recover the density, f, by Fourier inversion numerical techniques and price European options highly efficiently [10,24,21,13]. The characteristic function of exponential Lévy dynamics is often available in a tractable form. Some relevant financial models that fall into this category are Black-Scholes, Merton, Variance Gamma (VG), Normal Inverse Gaussian (NIG) or Carr–Geman–Madan–Yor (CGMY) models.

When dealing with arithmetic Asian options, the derivation of the corresponding characteristic function is somehow more involved. We adopt here an existing approach (see [5] or [3]) based on a recursive procedure that takes advantage of the convenient properties of the exponential Lévy processes. Lets start by defining the *return* or *increment* process R_i,

$$R_i := \log\left(\frac{S(t_i)}{S(t_{i-1})}\right) \quad i = 1, \ldots, N.$$

The increments of a Lévy process are identically and independently distributed, i.e. $R_i \overset{d}{=} R$, and its characteristic function, $\hat{f}_{R_i} = \hat{f}_R$, is therefore the same for all $i = 1, \ldots, N$. Based on R_i, we define a new process

$$Y_i := R_{N+1-i} + Z_{i-1}, \quad i = 2, \ldots, N, \tag{17}$$

where $Y_1 = R_N$ and $Z_i := \log(1 + \mathrm{e}^{Y_i})$, $\forall i$.

Next, we can apply the Carverhill–Clewlow–Hodges [2] factorization scheme to the newly defined stochastic process Y_i. The arithmetic average of $S(t)$ can be then written in terms of the process Y_i as

$$\frac{1}{N+1}\sum_{i=0}^{N} S(t_i) = \frac{\left(1 + \mathrm{e}^{Y_N}\right) S_0}{N+1}.$$

[1] It could be selected as a function of k.

Thus, the option price in Eq. (11) for arithmetic Asian contracts (at time $t = 0$) can be now expressed in terms of the transitional density of the Y_N as

$$v(x, t_0) = \mathrm{e}^{-rT} \int_{\mathbb{R}} v(y, T) f_{Y_N}(y) \mathrm{d}y,$$

where $x = \log S_0$ and the payoff function is given by

$$v(y, T) = \begin{cases} \left(\dfrac{S_0 (1 + \mathrm{e}^y)}{N + 1} - K \right)^+ & \text{for a call,} \\ \left(K - \dfrac{S_0 (1 + \mathrm{e}^y)}{N + 1} \right)^+ & \text{for a put.} \end{cases} \tag{18}$$

Again, the probability density function f_{Y_N} is generally not known, even for Lévy processes. However, as the process Y_N is defined in a recursive manner, the characteristic function of Y_N can be computed iteratively as well. In Section 3.1 this procedure is described.

3.1. Characteristic function of $\mathrm{Y_N}$

In this section an iterative derivation of the characteristic function of the Y_N, defined in the previous section, is presented. Lets assume that the characteristic function of R (as defined above) is available in closed form, as it is for a wide range of exponential Lévy processes. According to the definition of the (backward) sequence Y_i in Eq. (17), the required initial and recursive characteristic functions are given by the following expressions,

$$\begin{aligned} \hat{f}_{Y_1}(\xi) &= \hat{f}_{R_N}(\xi) = \hat{f}_R(\xi), \\ \hat{f}_{Y_i}(\xi) &= \hat{f}_{R_{N+1-i}+Z_{i-1}}(\xi) = \hat{f}_{R_{N+1-i}}(\xi) \cdot \hat{f}_{Z_{i-1}}(\xi) = \hat{f}_R(\xi) \cdot \hat{f}_{Z_{i-1}}(\xi), \end{aligned} \tag{19}$$

where the independence between R_{N+1-i} and Z_{i-1} has been employed.

By definition, the characteristic function of Z_{i-1} reads

$$\hat{f}_{Z_{i-1}}(\xi) := \mathbb{E}\left[\mathrm{e}^{-\mathrm{i}\xi \log(1+\mathrm{e}^{Y_{i-1}})} \right] = \int_{\mathbb{R}} \left(1 + \mathrm{e}^x \right)^{-\mathrm{i}\xi} f_{Y_{i-1}}(x) \mathrm{d}x.$$

We can now use the wavelet approximation from Eq. (8) to the probability density function $f_{Y_{i-1}}$ as follows

$$\begin{aligned} \hat{f}_{Z_{i-1}}(\xi) &= \int_{\mathbb{R}} \left(1 + \mathrm{e}^x \right)^{-\mathrm{i}\xi} f_{Y_{i-1}}(x) \mathrm{d}x \approx \int_{\mathbb{R}} \left(1 + \mathrm{e}^x \right)^{-\mathrm{i}\xi} \sum_{k=k_1}^{k_2} c_{m,k} \varphi_{m,k}(x) \mathrm{d}x \\ &= 2^{\frac{m}{2}} \sum_{k=k_1}^{k_2} c_{m,k} \int_{\mathbb{R}} \left(\mathrm{e}^x + 1 \right)^{-\mathrm{i}\xi} \mathrm{sinc}\left(2^m x - k \right) \mathrm{d}x, \end{aligned} \tag{20}$$

where the $c_{m,k}$ coefficients can be calculated by means of the SWIFT method employing the formula in Eq. (9) and the characteristic function of Y_{i-1}, i.e. $\hat{f}_{Y_{i-1}}$.

The integral on the right hand side of Eq. (20) can be computed at high accuracy by means of Theorem 1 of [27] without the need of using numerical integration. This is a key step within the overall pricing machinery, making the method easily implementable, robust and very fast. Other developments from the literature consist of solving the integral by means of quadratures (see for instance [6,5]).

Theorem 1. Theorem 1.3.2 of [27]

Let f be defined on $\mathbb{R}$ and let its Fourier transform $\hat{f}$ be such that for some positive constant d, $|\hat{f}(\omega)| = \mathcal{O}\left(\mathrm{e}^{-d|\omega|} \right)$ for $\omega \to \pm\infty$, then as $h \to 0$

$$\frac{1}{h} \int_{\mathbb{R}} f(x) S_{j,h}(x) \mathrm{d}x - f(jh) = \mathcal{O}\left(\mathrm{e}^{-\frac{\pi d}{h}} \right),$$

where $S_{j,h}(x) = \mathrm{sinc}\left(\frac{x}{h} - j \right)$ for $j \in \mathbb{Z}$.

Theorem 1 allows us to approximate the integral in Eq. (20) provided that $g(x) := (\mathrm{e}^x + 1)^{-\mathrm{i}\xi}$ satisfies the hypothesis. If we consider $h = \frac{1}{2^m}$, then it follows from Theorem 1 that

$$\int_{\mathbb{R}} g(x) \mathrm{sinc}\left(2^m x - k \right) \mathrm{d}x \approx hg(kh) = \frac{1}{2^m} \left(\mathrm{e}^{\frac{k}{2^m}} + 1 \right)^{-\mathrm{i}\xi}. \tag{21}$$

Thus, Eq. (20) now reads

$$\begin{aligned} \hat{f}_{Z_{i-1}}(\xi) &\approx 2^{\frac{m}{2}} \sum_{k=k_1}^{k_2} c_{m,k} \int_{\mathbb{R}} \left(\mathrm{e}^x + 1 \right)^{-\mathrm{i}\xi} \mathrm{sinc}\left(2^m x - k \right) \mathrm{d}x \\ &\approx 2^{-\frac{m}{2}} \sum_{k=k_1}^{k_2} c_{m,k} \left(\mathrm{e}^{\frac{k}{2^m}} + 1 \right)^{-\mathrm{i}\xi}. \end{aligned} \tag{22}$$

Finally, by using Eq. (19) we find

$$\hat{f}_{Y_i}(\xi) = \hat{f}_R(\xi)\hat{f}_{Z_{i-1}} \approx \hat{f}_R(\xi)2^{-\frac{m}{2}}\sum_{k=k_1}^{k_2} c_{m,k}\left(\mathrm{e}^{\frac{k}{2^m}}+1\right)^{-\mathrm{i}\xi},$$

where the density coefficients $c_{m,k}$ are computed as follows

$$c_{m,k} \approx \frac{2^{m/2}}{2^{J-1}}\sum_{j=0}^{2^{J-1}} \Re\left\{\hat{f}_{Y_{i-1}}\left(\frac{(2j-1)\pi 2^m}{2^J}\right)\mathrm{e}^{\frac{\mathrm{i}k\pi(2j-1)}{2^J}}\right\}.$$

It remains to prove that function $g(x) = (\mathrm{e}^x+1)^{-\mathrm{i}\xi}$ satisfies $|\hat{g}(\omega)| = \mathcal{O}\left(\mathrm{e}^{-d|\omega|}\right)$ for $\omega \to \pm\infty$ for some positive constant d, where the Fourier transform of g is taken with respect to the real variable x and ξ is a real variable that remains fixed. We derive an expression for $\hat{g}(w)$ in Proposition 2 by using Corollary 1, which follows from Proposition 1.

Proposition 1. Theorem 1 of [28]

Let $z \in \mathbb{C}$ and $\binom{z}{n} = \dfrac{z(z-1)(z-2)\cdots(z-n+1)}{n!}$. Then the series $\sum_{n=0}^{\infty}\binom{z}{n}x^n$ converges to $(1+x)^z$ for all complex x with $|x|<1$.

Corollary 1. *Let $z \in \mathbb{C}$. Then the series $\sum_{n=0}^{\infty}\binom{z}{n}x^n y^{z-n}$ converges to $(x+y)^z$ for all complex x, y with $|x|<|y|$.*

Proof. The proof follows from Proposition 1 by taking into account that $(x+y)^z = \left(y\left[\frac{x}{y}+1\right]\right)^z$. □

Proposition 2. *Let $g(x) = (\mathrm{e}^x+1)^z$, where $z =- \mathrm{i}\xi$ and $x, \xi \in \mathbb{R}$. Then,*

$$\hat{g}(\omega) = \sum_{n=0}^{\infty}\binom{z}{n}\frac{2n-z}{(n-\mathrm{i}\omega)(n+\mathrm{i}(\omega+\xi))}, \quad \omega \in \mathbb{R}.$$

Proof. By definition,

$$\hat{g}(\omega) = \int_{\mathbb{R}} \mathrm{e}^{-\mathrm{i}\omega x}g(x)\mathrm{d}x.$$

$$\hat{g}(\omega) = \int_{-\infty}^{0} \mathrm{e}^{-\mathrm{i}\omega x}g(x)dx + \int_{0}^{\infty} \mathrm{e}^{-\mathrm{i}\omega x}g(x)\mathrm{d}x, \tag{23}$$

and observe that, by Corollary 1,

$$\left(\mathrm{e}^x+1\right)^z = \sum_{n=0}^{\infty}\binom{z}{n}\mathrm{e}^{nx}, \quad \text{for}\, x<0, \tag{24}$$

and,

$$\left(\mathrm{e}^x+1\right)^z = \sum_{n=0}^{\infty}\binom{z}{n}\mathrm{e}^{(z-n)x}, \quad \text{for}\, x>0. \tag{25}$$

$$\hat{g}(\omega) = \sum_{n=0}^{\infty}\binom{z}{n}\int_{-\infty}^{0}\mathrm{e}^{-\mathrm{i}\omega x}\mathrm{e}^{nx}\mathrm{d}x + \sum_{n=0}^{\infty}\binom{z}{n}\int_{0}^{\infty}\mathrm{e}^{-\mathrm{i}\omega x}\mathrm{e}^{(z-n)x}\mathrm{d}x. \tag{26}$$

$$\hat{g}(\omega) = \sum_{n=0}^{\infty}\binom{z}{n}\frac{1}{n-\mathrm{i}\omega} + \sum_{n=0}^{\infty}\binom{z}{n}\frac{1}{n+\mathrm{i}(\omega+\xi)} = \sum_{n=0}^{\infty}\binom{z}{n}\frac{2n-z}{(n-\mathrm{i}\omega)(n+\mathrm{i}(\omega+\xi))}. \tag{27}$$

□

Expression in Eq. (27) reveals that it is rather complicated to get a closed-form solution for the modulus of $\hat{g}(\omega)$. We therefore employ Wolfram Mathematica 11.2 to sum the series in Eq. (27), and we obtain,

$$\begin{aligned}\hat{g}(\omega) = \; & \frac{\xi}{2\omega+\xi}\left[\mathrm{e}^{-\pi\omega}(B_{-1}(-\mathrm{i}\omega, 1+z) + 2B_{-1}(1-\mathrm{i}\omega, z))\right. \\ & \left. + \Gamma(\mathrm{i}\omega - z)(2(\mathrm{i}\omega-z)_2\tilde{F}_1(1-z, 1+\mathrm{i}\omega-z; 2+\mathrm{i}\omega-z; -1) - {}_2\tilde{F}_1(-z, \mathrm{i}\omega-z; 1+\mathrm{i}\omega-z; -1))\right],\end{aligned} \tag{28}$$

where ${}_2\tilde{F}_1(a, b; c; \nu)$ is the regularized hypergeometric function, defined as,

$${}_2\tilde{F}_1(a, b; c; \nu) := \frac{{}_2F_1(a, b; c; \nu)}{\Gamma(c)}, \quad {}_2F_1(a, b; c; \nu) := \sum_{n=0}^{\infty}\frac{(a)_n(b)_n}{(c)_n}\frac{\nu^n}{n!},$$

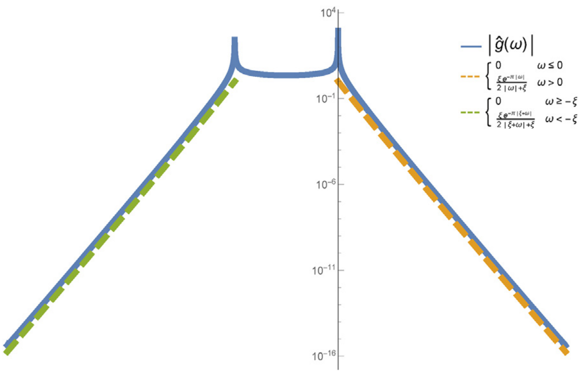

Fig. 1. Modulus of $\hat{g}(\omega)$.

and $(q)_n$ the rising Pochhammer symbol, given by,

$$(q)_n = \begin{cases} 0, & n = 0, \\ q(q+1)\cdots(q+n-1), & n > 0. \end{cases}$$

Finally, the special functions gamma, Γ, and beta, B, are defined as

$$\Gamma(u) := \int_0^\infty t^{u-1}e^{-t}\mathrm{d}t, \quad B_\eta(a,b) := \int_0^\eta t^{a-1}(1-t)^{b-1}\mathrm{d}t.$$

The modulus of $\hat{g}(\omega)$ for a particular value of ξ by using expression in Eq. (28) is presented in logarithmic scale in Fig. 1. We note that the shape of $|\hat{g}(\omega)|$ does not depend on the value given to ξ, since a different ξ just originates a shift of the same function. The two peaks observed in the plot correspond to the poles of $\hat{g}(\omega)$ located at $\omega = 0$ and $\omega = -\xi$. These two poles are clearly identified looking at the first term of the series in Eq. (27). Moreover, function $\hat{g}(\omega)$ presents a symmetry at $\omega = -\xi/2$ since, from Eq. (27), it is straightforward to see that $\hat{g}(\omega - \xi/2) = \hat{g}(-\omega - \xi/2)$.

For the aforementioned reasons, we focus our attention on the right tail of $|\hat{g}(\omega)|$ and observe that its decay appears to be of the same order as $\frac{\xi}{2\omega+\xi}e^{-\pi\omega}$ when $\omega \to +\infty$ (this term is part of the expression in Eq. (28) and is represented in Fig. 1). We therefore conclude that Theorem 1 can be applied to function $g(x)$ with $d = \pi$.

The following theorem generalises the results stated in Theorem 1, showing this way that expression in Eq. (21) can be applied under weaker conditions on the decay of the modulus of the Fourier transform.

Theorem 2. *Let f be defined on $\mathbb{R}$ and let $\hat{f}$ be its Fourier transform. Then,*

$$\left|\frac{1}{h}\int_{\mathbb{R}} f(x)S_{j,h}(x)\mathrm{d}x - f(jh)\right| \le \frac{1}{2\pi}\int_{|w|>\frac{\pi}{h}} \left|\hat{f}(\omega)\right| \mathrm{d}w,$$

where $S_{j,h}(x) = \operatorname{sinc}\left(\frac{x}{h} - j\right)$ for $j \in \mathbb{Z}$.

Proof. As mentioned in Section 2.4, Lemma 3 of [15] shows that the approximation error $|f(x) - \mathcal{P}_m f(x)|$ is uniformly bounded for all $x \in \mathbb{R}$,

$$|f(x) - \mathcal{P}_m f(x)| \le \frac{1}{2\pi}\int_{|\omega|>2^m\pi} |\hat{f}(\omega)|\mathrm{d}\omega, \tag{29}$$

where $\mathcal{P}_m f(x) = \sum_{k\in\mathbb{Z}} c_{m,k}\varphi_{m,k}(x)$. In particular, expression in Eq. (29) is valid for $x = jh$ with $h = 1/2^m$,

$$|f(jh) - \mathcal{P}_m f(jh)| \le \frac{1}{2\pi}\int_{|\omega|>2^m\pi} |\hat{f}(\omega)|\mathrm{d}\omega.$$

$$\mathcal{P}_m f(jh) = 2^{m/2}\cdot 2^{m/2}\int_{\mathbb{R}} f(x)\varphi(2^m x - j)\mathrm{d}x = 2^m\int_{\mathbb{R}} f(x)\operatorname{sinc}(2^m x - j)\mathrm{d}x,$$

and this concludes the proof since $2^m = 1/h$. □

Remark 4. Note that Theorem 1 follows straightforwardly from Theorem 2 when $f(x)$ satisfies $|\hat{f}(\omega)| = \mathcal{O}\left(e^{-d|\omega|}\right)$.

Finally, we establish a bound on the error generated when we approximate the integral in Eq. (22), giving this way an estimation of the accuracy at which $\hat{f}_{Z_{i-1}}(\xi)$ is computed.

Proposition 3. *Let $\mathcal{F}_{Z_{i-1}}(\xi)$, $\mathcal{G}_{Z_{i-1}}(\xi)$ and $\varepsilon(\xi)$ be defined as follows,*

$$\mathcal{F}_{Z_{i-1}}(\xi) = 2^{\frac{m}{2}}\sum_{k=k_1}^{k_2} c_{m,k}\int_{\mathbb{R}}\left(e^{x}+1\right)^{-i\xi}\mathrm{sinc}(2^m x - k)dx,$$

$$\mathcal{G}_{Z_{i-1}}(\xi) = 2^{-\frac{m}{2}}\sum_{k=k_1}^{k_2} c_{m,k}\left(e^{\frac{k}{2^m}}+1\right)^{-i\xi},$$

$\varepsilon(\xi) = \mathcal{F}_{Z_{i-1}}(\xi) - \mathcal{G}_{Z_{i-1}}(\xi)$*. Then,* $|\varepsilon(\xi)|$ *is uniformly bounded by* $|\varepsilon(\xi)| \le \mathcal{C}(k_2 - k_1 + 1)e^{-\pi^2 2^m}$*, where* $\mathcal{C}$ *is a constant.*

Proof. We observe that,

$$\varepsilon(\xi) = 2^{-\frac{m}{2}}\sum_{k=k_1}^{k_2} c_{m,k}[2^m\int_{\mathbb{R}}(e^{x}+1)^{-i\xi}\mathrm{sinc}(2^m x - k)dx - (e^{\frac{k}{2^m}}+1)^{-i\xi}].$$

$$|\varepsilon(\xi)| \le 2^{-\frac{m}{2}}\mathcal{C}\sum_{k=k_1}^{k_2}|c_{m,k}|e^{-\pi^2 2^m},$$

for a certain constant $\mathcal{C}$. The proposition holds by taking into account that,

$$|c_{m,k}| \le \int_{\mathbb{R}} f(x)|\varphi_{m,k}(x)|dx \le 2^{\frac{m}{2}},$$

where the last inequality is satisfied since f is a density function and $|\varphi_{m,k}(x)| \le 2^{\frac{m}{2}}$. □

3.2. Payoff coefficients

In order to complete the SWIFT pricing formula in Eq. (12) for the case of arithmetic Asian options of Lévy processes, the computation of the corresponding payoff coefficients, $V_{m,k}$, needs to be carried out. Lets first plug the arithmetic Asian payoff function in Eq. (18) into Eq. (13). Then we have

$$V_{m,k} = \begin{cases} \dfrac{2^{m/2}}{2^{J-1}}\displaystyle\sum_{j=1}^{2^{J-1}}\int_a^b\left(\frac{S_0\,(1+e^{y})}{N+1}-K\right)^{+}\cos\left(\frac{2j-1}{2^J}\pi\left(2^m y-k\right)\right)dy & \text{for a call,}\\ \dfrac{2^{m/2}}{2^{J-1}}\displaystyle\sum_{j=1}^{2^{J-1}}\int_a^b\left(K-\frac{S_0\,(1+e^{y})}{N+1}\right)^{+}\cos\left(\frac{2j-1}{2^J}\pi(2^m y-k)\right)dy & \text{for a put.}\end{cases}$$

By rearranging the terms, the arithmetic Asian payoff coefficients under Lévy dynamics are given by

$$V_{m,k} = \begin{cases} \dfrac{2^{m/2}}{2^{J-1}}\displaystyle\sum_{j=1}^{2^{J-1}}\left[\frac{S_0}{N+1}\left(I_2^{j,k}(\tilde{x},b)+I_0^{j,k}(\tilde{x},b)\right)-KI_0^{j,k}(\tilde{x},b)\right] & \text{for a call,}\\ \dfrac{2^{m/2}}{2^{J-1}}\displaystyle\sum_{j=1}^{2^{J-1}}\left[KI_0^{j,k}(a,\tilde{x})-\frac{S_0}{N+1}\left(I_2^{j,k}(a,\tilde{x})+I_0^{j,k}(a,\tilde{x})\right)\right] & \text{for a put,}\end{cases} \tag{30}$$

where $\tilde{x} = \log\left(\frac{K(N+1)}{S_0}-1\right)$ and the functions $I_0^{j,k}$ and $I_2^{j,k}$ are defined by the following integrals

$$\begin{aligned} I_0^{j,k}(x_1,x_2) &:= \int_{x_1}^{x_2}\cos\left(C_j\left(2^m y-k\right)\right)dy,\\ I_2^{j,k}(x_1,x_2) &:= \int_{x_1}^{x_2}e^{y}\cos\left(C_j\left(2^m y-k\right)\right)dy,\end{aligned} \tag{31}$$

with $C_j = \frac{2j-1}{2^J}\pi$. The solution of these integrals is known analytically (see Appendix A).

3.3. Greeks

As stated before, series expansions that approximate the option sensitivities or Greeks can also be obtained under the SWIFT method. Here, we have specifically considered Δ and Γ, which are the derivatives with respect to the spot price, S_0. In the context of Asian options under Lévy processes, only the payoff coefficients, $V_{m,k}$ are affected by S_0 (see Eq. (30)). Thus, by differentiating $V_{m,k}$ with respect to S_0, we obtain

$$V_{m,k}^{(1)} = \begin{cases} \dfrac{2^{m/2}}{2^{J-1}} \displaystyle\sum_{j=1}^{2^{J-1}} \left[\dfrac{I_2^{j,k}(\tilde{x}, b) + I_0^{j,k}(\tilde{x}, b)}{N+1} + \dfrac{S_0 \left(\dfrac{\partial I_2^{j,k}(\tilde{x}, b)}{\partial S_0} + \dfrac{\partial I_0^{j,k}(\tilde{x}, b)}{\partial S_0} \right)}{N+1} - K \dfrac{\partial I_0^{j,k}(\tilde{x}, b)}{\partial S_0} \right] & \text{for a call,} \\ \dfrac{2^{m/2}}{2^{J-1}} \displaystyle\sum_{j=1}^{2^{J-1}} \left[K \dfrac{\partial I_0^{j,k}(a, \tilde{x})}{\partial S_0} - \dfrac{I_2^{j,k}(a, \tilde{x}) + I_0^{j,k}(a, \tilde{x})}{N+1} - \dfrac{S_0 \left(\dfrac{\partial I_2^{j,k}(a, \tilde{x})}{\partial S_0} + \dfrac{\partial I_0^{j,k}(a, \tilde{x})}{\partial S_0} \right)}{N+1} \right] & \text{for a put.} \end{cases} \tag{32}$$

Applying the chain rule, the partial derivatives of $I_u^{j,k}$, $u \in \{0, 2\}$ can be calculated by

$$\begin{aligned} \frac{\partial I_u^{j,k}(\tilde{x}, b)}{\partial S_0} &= \frac{\partial I_u^{j,k}(\tilde{x}, b)}{\partial \tilde{x}} \frac{\partial \tilde{x}}{\partial S_0}, \\ \frac{\partial I_u^{j,k}(a, \tilde{x})}{\partial S_0} &= \frac{\partial I_u^{j,k}(a, \tilde{x})}{\partial \tilde{x}} \frac{\partial \tilde{x}}{\partial S_0}, \end{aligned}$$

where

$$\frac{\partial \tilde{x}}{\partial S_0} = -\frac{K(N+1)}{S_0 K(N+1) - S_0^2},$$

and the partial derivatives $\frac{\partial I_u^{j,k}(\tilde{x},b)}{\partial \tilde{x}}$ and $\frac{\partial I_u^{j,k}(a,\tilde{x})}{\partial \tilde{x}}$ have analytic solution, which can be calculated from the expressions obtained in Appendix A. Following the same procedure, a closed-form solution can be similarly derived for the second derivative of $V_{m,k}$. Thus, $V_{m,k}^{(2)}$ reads

$$V_{m,k}^{(2)} = \begin{cases} \dfrac{2^{m/2}}{2^{J-1}} \displaystyle\sum_{j=1}^{2^{J-1}} \left[\dfrac{2 \left(\dfrac{\partial I_2^{j,k}(\tilde{x}, b)}{\partial S_0} + \dfrac{\partial I_0^{j,k}(\tilde{x}, b)}{\partial S_0} \right)}{N+1} + \dfrac{S_0 \left(\dfrac{\partial^2 I_2^{j,k}(\tilde{x}, b)}{\partial S_0^2} + \dfrac{\partial^2 I_0^{j,k}(\tilde{x}, b)}{\partial S_0^2} \right)}{N+1} - K \dfrac{\partial^2 I_0^{j,k}(\tilde{x}, b)}{\partial S_0^2} \right] & \text{for a call,} \\ \dfrac{2^{m/2}}{2^{J-1}} \displaystyle\sum_{j=1}^{2^{J-1}} \left[K \dfrac{\partial^2 I_0^{j,k}(a, \tilde{x})}{\partial S_0^2} - \dfrac{2 \left(\dfrac{\partial I_2^{j,k}(a, \tilde{x})}{\partial S_0} + \dfrac{\partial I_0^{j,k}(a, \tilde{x})}{\partial S_0} \right)}{N+1} - \dfrac{S_0 \left(\dfrac{\partial^2 I_2^{j,k}(a, \tilde{x})}{\partial S_0^2} + \dfrac{\partial^2 I_0^{j,k}(a, \tilde{x})}{\partial S_0^2} \right)}{N+1} \right] & \text{for a put,} \end{cases} \tag{33}$$

where now the second partial derivatives can be expanded as

$$\begin{aligned} \frac{\partial^2 I_u^{j,k}(\tilde{x}, b)}{\partial S_0^2} &= \frac{\partial^2 I_u^{j,k}(\tilde{x}, b)}{\partial \tilde{x}^2} \left(\frac{\partial \tilde{x}}{\partial S_0} \right)^2 + \frac{\partial I_u^{j,k}(\tilde{x}, b)}{\partial \tilde{x}} \frac{\partial^2 \tilde{x}}{\partial S_0^2}, \\ \frac{\partial I_u^{j,k}(a, \tilde{x})}{\partial S_0} &= \frac{\partial^2 I_u^{j,k}(a, \tilde{x})}{\partial \tilde{x}^2} \left(\frac{\partial \tilde{x}}{\partial S_0} \right)^2 + \frac{\partial I_u^{j,k}(a, \tilde{x})}{\partial \tilde{x}} \frac{\partial^2 \tilde{x}}{\partial S_0^2}, \end{aligned}$$

with

$$\frac{\partial^2 \tilde{x}}{\partial S_0^2} = -\frac{K(N+1)(2S_0 - K(N+1))}{S_0^2 (S_0 - K(N+1))^2},$$

and, as before, $\frac{\partial^2 I_u^{j,k}(\tilde{x},b)}{\partial \tilde{x}^2}$ and $\frac{\partial^2 I_u^{j,k}(a,\tilde{x})}{\partial \tilde{x}^2}$ are analytically available.

We can now employ $V_{m,k}^{(1)}$ and $V_{m,k}^{(2)}$ within Eq. (16) to estimate Δ and Γ, respectively, in the arithmetic Asian option valuation framework under exponential Lévy dynamics.

3.4. Interval selection

Here, we follow an effective *cumulant*-based approach originally described in [24]. Thus, the integration range can be determined relying on the cumulants as follows

$$[a, b] := \left[\kappa_1(Y) - L\sqrt{\kappa_2(Y) + \sqrt{\kappa_4(Y)}}, \kappa_1(Y) + L\sqrt{\kappa_2(Y) + \sqrt{\kappa_4(Y)}} \right], \tag{34}$$

with $\kappa_n(Y)$ representing the n-th cumulant[2] of the random variable Y and L a constant conveniently chosen.

The selection based on cumulants in Eq. (34) has been adapted to the process Y_j in [5]. The approach only requires the cumulants of the logarithmic return process R, $\kappa_n(R)$. These cumulants typically are known since the derivative of the characteristic function of R is usually available. By denoting

$$\begin{aligned} a_i &:= \log(i) + \kappa_1(R) - L\sqrt{i\kappa_2(R) + \sqrt{\kappa_4(R)}}, \\ b_i &:= \log(i) + j\kappa_1(R) + L\sqrt{i\kappa_2(R) + \sqrt{\kappa_4(R)}}, \end{aligned}$$

the interval is selected as

$$[a, b] := \left[\min_{i=1,\ldots,N} a_i, \max_{i=1,\ldots,N} b_i, \right].$$

The interval selection can be now plugged into Algorithm 1, passing an increasing constant L to each iteration of the adaptive procedure presented in Section 2.4 (i.e. $\varrho = L$[3]). The parameter L can be selected from a small range of integers, typically $L \in \{6 - 12\}$.

4. SWIFT for arithmetic Asian options under square-root diffusions

As a second class of processes, we consider the square-root processes, including the usual extensions like mean-reversion or time dependency. The importance of these processes is widely evidenced as they are typically used in fields like commodity and interest rate markets (see [29,30], for example). A well known representative is the so-called Cox–Ingersoll–Ross (CIR) process [31].

We again base our approach on the availability of the characteristic function related to the necessary transitional density. In a recent work of Fusai et al. [9], the authors developed an iterative methodology to compute the *moment generating function*, denoted here by $\mathcal{M}$, of the joint distribution of the final state of the underlying, $S(T)$, and the weighted cumulative price, $C_N = \sum_{i=0}^{N} \omega_i S(t_i)$. It has been shown that this expression can be directly applied to price discretely monitored arithmetic Asian derivatives under square-root dynamics. Here we take advantage of the fact that the characteristic function is related with the moment generating function by

$$\hat{f}(\xi) = \mathcal{M}(\mathrm{i}\xi).$$

In the following, we adapt some of the most important results in [9] and show how to employ the SWIFT method in this context.

4.1. *Characteristic function of* $\mathrm{C_N}$

We start by assuming that the moment generating function of a generic square-root process $S(t)$ at time t_i given the information at time t_{i-1} is available and can be written in the form

$$\mathcal{M}_S(\gamma) = \mathbb{E}\left[\mathrm{e}^{-\gamma S(t_i)} | x = S(t_{i-1})\right] = \mathrm{e}^{-(\psi(\Delta t;\gamma)x + \phi(\Delta t;\gamma))}, \tag{35}$$

where ψ and ϕ are known functions and recall that $\Delta t = t_i - t_{i-1}$. This formulation falls into the broad class of *affine* diffusions [32], whose moment generating function (or characteristic function) has exponential-affine dependence on the state variable. The square-root dynamics treated here fits in the affine diffusions category.

By definition, the moment generating function of the joint distribution of $S(T)$ and C_N, given the information at time $t = 0$ is

$$\mathcal{M}_{S,C_N}(\gamma, \lambda) := \mathbb{E}\left[\mathrm{e}^{-\left(\gamma S(T) + \lambda \sum\limits_{i=0}^{N} \omega_i S(t_i)\right)} | x = S_0\right]. \tag{36}$$

Based on the expression of the moment generating function of $S(t_i)|S(t_{i-1})$ in Eq. (35), the moment generating function of the pair $\left(S(T), \sum_{i=0}^{N} \omega_i S(t_i)\right)$ in Eq. (36) can be computed by

$$\mathcal{M}_{S,C_N}(\gamma, \lambda) = \mathrm{e}^{-\left(\Lambda_0(\gamma,\lambda)x + \sum\limits_{i=0}^{N-1} \phi(\Delta t; \Lambda_{i+1}(\Delta t;\gamma,\lambda))\right)},$$

where the function $\Lambda_i(\Delta t; \gamma, \lambda)$ is constructed recursively as follows

$$\begin{aligned} \Lambda_N(\Delta t; \gamma, \lambda) &= \gamma + \lambda\omega_N, \\ \Lambda_i(\Delta t; \gamma, \lambda) &= \psi(\Delta t, \Lambda_{i+1}(\Delta t; \gamma, \lambda)) + \lambda\omega_i, \quad \text{for} \quad i = N-1, \ldots, 0, \end{aligned} \tag{37}$$

with ψ and ϕ as in Eq. (35). This iterative formulation is obtained by simply applying the definition of moment generating function, the tower law of probabilities and some algebraic manipulations. We refer to Fusai et al. [9] for more details.

[2] Defined from the *cumulant-generating function*, $\mathcal{K}(\tau)$, as $\kappa_n = \mathcal{K}^{(n)}(0)$.

[3] In this work, we employ an initial value $L = 4$ and change rule $L = L + 2$.

Next, it is easy to see that the moment generating function of C_N reads

$$\mathcal{M}_{C_N}(\lambda) := \mathcal{M}_{S,C_N}(0, \lambda). \tag{38}$$

As stated above, the characteristic function of the variable C_N evaluated in ξ is obtained from the moment generating function

$$\hat{f}_{C_N}(\xi) = \mathcal{M}_{C_N}(\mathrm{i}\xi). \tag{39}$$

For floating strike Asian contracts (with payoff function as in Eq. (2)), we would need the joint characteristic function

$$\hat{f}_{S,C_N}(\xi) = \mathcal{M}_{S,C_N}(-\mathrm{i}\xi, \mathrm{i}\xi). \tag{40}$$

Note that $C_N = A_N$ when $\omega_i = \frac{1}{N+1}, \forall i$.

4.2. Payoff coefficients

At this point we recall the SWIFT valuation formula in Eq. (12) where, once the density coefficients are obtained by using the previously developed characteristic function, the remaining part is the computation of the payoff coefficients, $V_{m,k}$. Contrary to the Lévy processes where we employ a logarithmic transformation, for the square-root dynamics we set $y = C_N$ or $y = \theta(S(T) - C_N)$ for European-style (also known as fixed-strike) or floating-strike options, respectively. In equity and commodity markets with payoff functions as in Eqs. (1) or (2), the $V_{m,k}$ for arithmetic Asian options are calculated by

$$V_{m,k} = \begin{cases} \dfrac{2^{m/2}}{2^{J-1}} \displaystyle\sum_{j=1}^{2^{J-1}} \left[I_1^{j,k}(\tilde{x}, b) - K I_0^{j,k}(\tilde{x}, b) \right] & \text{for a call,} \\ \dfrac{2^{m/2}}{2^{J-1}} \displaystyle\sum_{j=1}^{2^{J-1}} \left[K I_0^{j,k}(a, \tilde{x}) - I_1^{j,k}(a, \tilde{x}) \right] & \text{for a put,} \end{cases}$$

with $\tilde{x} = K$, function $I_0^{j,k}$ as in Eq. (31) and function $I_1^{j,k}$ defined as follows

$$I_1^{j,k}(x_1, x_2) := \int_{x_1}^{x_2} y \cos\left(C_j\left(2^m y - k\right)\right) \mathrm{d}y.$$

Again, the integral $I_1^{j,k}$ has analytic solution (see Appendix A).

Furthermore, the SWIFT method can also be efficiently applied when we are dealing with more complex payoff functions like the ones appearing in interest rate derivatives with Asian-like features (see [18], for example, and the references therein). In this context, the option pricing problem needs to be adapted since now the underlying is the interest rate itself. Hence, the valuation formula in interest rate markets holds

$$v(x, t) = \mathbb{E}\left[v(y, T)|x\right] = \int_{\mathbb{R}} v(y, T) f(y|x) \mathrm{d}y,$$

where the payoff function is in the form $v(y, t) = \mathrm{e}^{-y(T-t)} h(y, t)$. In this work, we assume $y = C_N$. Thus, the $V_{m,k}$ payoff coefficients corresponding to the payoff functions in Eq. (3) read

$$\begin{aligned} V_{m,k} &= \frac{2^{m/2}}{2^{J-1}} \sum_{j=1}^{2^{J-1}} \left[I_4^{j,k}(\tilde{x}, b, T) - K I_3^{j,k}(\tilde{x}, b, T) \right], \quad \text{for Asian caps,} \\ V_{m,k} &= \frac{2^{m/2}}{2^{J-1}} \sum_{j=1}^{2^{J-1}} \left[I_3^{j,k}(\tilde{x}, b, T) \right], \quad \text{for cash Binary Asian caps,} \\ V_{m,k} &= \frac{2^{m/2}}{2^{J-1}} \sum_{j=1}^{2^{J-1}} \left[T I_4^{j,k}(\tilde{x}, b, T) \right], \quad \text{for rate Binary Asian caps,} \end{aligned}$$

respectively, with $\tilde{x} = K$ and

$$\begin{aligned} I_3^{j,k}(x_1, x_2, t) &:= \int_{x_1}^{x_2} \mathrm{e}^{-yt} \cos\left(C_j\left(2^m y - k\right)\right) \mathrm{d}y, \\ I_4^{j,k}(x_1, x_2, t) &:= \int_{x_1}^{x_2} y \mathrm{e}^{-yt} \cos(C_j(2^m y - k)) \mathrm{d}y, \end{aligned}$$

solvable again in analytic form (expressions in Appendix A).

4.3. Greeks

Regarding the Greeks, the SWIFT method also allows a straightforward computation of Δ and Γ under square-root dynamics. Eq. (15) is suitable in this framework since the option value depends on $x = S_0$ through the density coefficients. Combining Eq. (40) (or (39) for the special case of fixed-strike contracts) with Eq. (36) and differentiating the resulting characteristic function, it turns out that, for the first derivative, we have

$$\frac{\partial \hat{f}_{S,C_N}(\xi)}{\partial x} = \frac{\partial \mathcal{M}_{S,C_N}(-\mathrm{i}\xi, \mathrm{i}\xi)}{\partial x} = -\Lambda_0\left(-\mathrm{i}\xi, \mathrm{i}\xi\right) \mathrm{e}^{-\left(\Lambda_0(-\mathrm{i}\xi,\mathrm{i}\xi)x+\sum_{i=0}^{N-1}\phi(\Delta t;\Lambda_{i+1}(\Delta t;-\mathrm{i}\xi,\mathrm{i}\xi))\right)}, \tag{41}$$

and, for the second derivative

$$\frac{\partial^2 \hat{f}_{S,C_N}(\xi)}{\partial x^2} = \frac{\partial^2 \mathcal{M}_{S,C_N}(-\mathrm{i}\xi, \mathrm{i}\xi)}{\partial x^2} = \left(\Lambda_0\left(-\mathrm{i}\xi, \mathrm{i}\xi\right)\right)^2 \mathrm{e}^{-\left(\Lambda_0(-\mathrm{i}\xi,\mathrm{i}\xi)x+\sum_{i=0}^{N-1}\phi(\Delta t;\Lambda_{i+1}(\Delta t;-\mathrm{i}\xi,\mathrm{i}\xi))\right)}, \tag{42}$$

where Λ_i is as in Eq. (37). As mentioned, the same expression can be employed for the European-style case, by setting $\gamma = 0$, i.e. $\mathcal{M}_{S,C_N}(0, \mathrm{i}\xi)$.

By using the expressions above, we can now compute the coefficients $c^{(1)}_{m,k}$ and $c^{(2)}_{m,k}$ via Eq. (14) and, therefore obtain the Greeks Δ and Γ by Eq. (15).

4.4. *Interval selection*

Again, a key aspect in the computational performance of the SWIFT method is the choice of the truncated interval. Its smart selection is crucial to reduce the execution time keeping a prescribed accuracy.

As mentioned before, we can follow the cumulant-based approach as in Eq. (34). By definition the *cumulant generating function* can be expressed in terms of the moment generating function as

$$\kappa(\tau) = \log \mathcal{M}(\tau),$$

and the cumulants can be obtained from the cumulant generating function by simply differentiating it with respect to the function variable, τ. Next, the n-th cumulant, κ_n, is calculated by evaluating the n-th derivative of $\kappa(\tau)$ at zero, i.e.

$$\kappa_n = (-1)^n \kappa^{(n)}(0).$$

In our context, we are particularly interested in the cumulants of C_N. In this case

$$\begin{aligned}\kappa_n(C_N) &= (-1)^n \kappa^{(n)}_{C_N}(0)\\ &= -(-1)^n (\Lambda^{(n)}_0(\Delta t; 0, \lambda)x + \sum_{i=0}^{N-1} \phi^{(n)}(\Delta t; \Lambda_{i+1}(\Delta t; 0, \lambda))\Lambda^{(n)}_{i+1}(\Delta t; 0, \lambda))\Big|_{\lambda=0}.\end{aligned}$$

The derivatives $\Lambda^{(n)}_i(\Delta t; 0, \lambda)$ with respect to the parameter λ can be computed again in recursive way applying the chain rule. Thus, for example, the first derivative would be

$$\begin{aligned}&\Lambda'_N(\Delta t; 0, \lambda) = \omega_N,\\ &\Lambda'_i(\Delta t; 0, \lambda) = \psi'(\Delta t, \Lambda_{i+1}(\Delta t; 0, \lambda))\,\Lambda'_{i+1}(\Delta t; 0, \lambda) + \omega_i, \quad \text{for} \quad i = N-1, \ldots, 0.\end{aligned}$$

Similarly, an iterative sequence can be defined to obtain higher derivatives. Once the cumulants are available, they can be used together with Eq. (34) in the interval selection procedure described in Algorithm 1.

4.4.1. *Cumulants approximation*

In order to avoid the differentiation of a function formulated recursively, we propose an approximated expression for the cumulant generating function of C_N, κ_{C_N}. For that, we first assume that $\mathbb{E}[\log(\cdot)]$ is a good proxy of $\log\mathbb{E}[\cdot]$, i.e. $\mathbb{E}[\log(\cdot)] \approx \log\mathbb{E}[\cdot]$. Applying this log-interchange in combination with Equations (38) and(36), the cumulant generating function of C_N is approximated as follows

$$\kappa_{C_N}(\lambda) \approx \hat{\kappa}_{C_N}(\lambda) := \mathbb{E}[\log(\mathrm{e}^{-\lambda\sum_{i=0}^{N}\omega_i S(t_i)})] = \mathbb{E}[-\lambda\sum_{i=0}^{N}\omega_i S(t_i)] = -\lambda\sum_{i=0}^{N}\omega_i \mathbb{E}[S(t_i)],$$

where ω_i the arithmetic weights. By Jensen's inequality, this approximation implies an upper bound (since the function log(·) is concave). We have constructed an approximated cumulant generating function, $\hat{\kappa}_{C_N}$ that depends only on expectations of the underlying process, often available. However, that restricts the use of $\hat{\kappa}_{C_N}$ to compute only the first cumulant, $\hat{\kappa}_1(C_N) \approx \kappa_1(C_N)$, since the second and higher derivatives are always zero. The use of only one cumulant may be justified in some special situations. For example, when we deal with European-style interest rate Asian derivatives, the lower bound a can be safely set to zero, $a = 0$, since the arithmetic average of the underlying cannot be negative but it is typically close to zero. The upper bound of the interval, b, can be determined by simply setting it equal to the approximated first cumulant, i.e. $b = \hat{\kappa}_1(C_N)$, and increasing it in each iteration of the adaptive interval selection explained in Algorithm 1 of Section 2.4.

Table 1
GBM. The reference values are 11.9049157487 (N = 12), 11.9329382045 (N = 50) and 11.9405631571 (N = 250).

# Decimals	Method	N = 12	N = 50	N = 250
6	ASCOS	$N_c = 144$, $n_q = 225$	$N_c = 384$, $n_q = 600$	$N_c = 384$, $n_q = 600$
	SWIFT	$m = 5$	$m = 6$	$m = 7$
8	ASCOS	$N_c = 192$, $n_q = 300$	$N_c = 384$, $n_q = 600$	$N_c = 768$, $n_q = 1200$
	SWIFT	$m = 5$	$m = 6$	$m = 8$
10	ASCOS	$N_c = 256$, $n_q = 400$	$N_c = 512$, $n_q = 800$	$N_c = 5120$, $n_q = 8000$
	SWIFT	$m = 6$	$m = 7$	$m = 8$

Table 2
NIG. The reference values are 1.0135 (N = 12), 1.0377 (N = 50) and 1.0444 (N = 250).

# Decimals	Method	N = 12	N = 50	N = 250
2	ASCOS	$N_c = 128$, $n_q = 200$	$N_c = 128$, $n_q = 200$	$N_c = 192$, $n_q = 300$
	SWIFT	$m = 6$	$m = 5$	$m = 5$
3	ASCOS	$N_c = 128$, $n_q = 200$	$N_c = 192$, $n_q = 300$	$N_c = 192$, $n_q = 300$
	SWIFT	$m = 6$	$m = 5$	$m = 7$
4	ASCOS	$N_c = 256$, $n_q = 400$	$N_c = 256$, $n_q = 400$	$N_c = 512$, $n_q = 800$
	SWIFT	$m = 7$	$m = 8$	$m = 9$

A great advantage is that, in some cases, this approximation results in a closed-form expression which facilitates the interval selection based on cumulants. For example, when the basic square-root process (Eq. (43)) is considered and $\omega_i = \omega$, $\forall\, i$, the previous expression becomes

$$\hat{\kappa}_{C_N}(\lambda) = -\lambda\sum_{i=0}^{N}\omega\mathbb{E}[S(t_i)] = -\lambda\sum_{i=0}^{N}\omega S_0 e^{rt_i} = -S_0\lambda\omega\sum_{i=0}^{N}e^{rt_i} = -S_0\lambda\omega\frac{1-e^{r(N+1)\Delta t}}{1-e^{r\Delta t}}.$$

5. Numerical results

In this section, we present the performance of the SWIFT method against the state-of-the-art methods existing in the recent literature. In the framework of exponential Lévy models, we will consider the well-known COS method, particularly the COS variant for arithmetic Asian option, called *ASCOS* method [5]. To the best of our knowledge, the ASCOS method provides the best balance between accuracy and efficiency in arithmetic Asian pricing under Lévy processes. For the case of square-root dynamics, we employ the well-known *Carr–Madan* method, as it was the author's choice in [9], where the algorithm followed in this work for square-root processes was originally presented. This allows a fair comparison in the numerical tests. All the experiments have been conducted in a computer system with the following characteristics: CPU Intel Core i7-4720HQ 2.6GHz and memory of 16GB RAM. The employed software package is Matlab R2017b.

5.1. Results under Lévy process

In the context of arithmetic Asian options under exponential Lévy models, the first issue arises from the unavailability of reliable reference prices, even in the simplest case of Black-Scholes model. In order to overcome this inconvenience, our first experiment consists of determining reference values that are sufficiently consistent. In [5], the authors considered an extreme configuration of the ASCOS method as a reference. We follow here a similar approach, but employing two different methods. We use increasingly "extreme" settings for the ASCOS method and our SWIFT method aiming to find a prescribed number of coincident decimals in the solution. The procedure is stopped when each method's converging value starts to diverge. The ASCOS method includes a numerical quadrature (Clenshaw–Curtis) where, the number of points in the quadrature, n_q, has an unknown relation with the number of COS expansion terms N_c. We follow here the "rule-of-thumb" suggested by the authors in [5], i.e. $n_q = \lceil 100N_c/64\rceil$. Note however that, as this is an empirical rule, ASCOS can suffer from instabilities. In contrast, SWIFT employs only one parameter, the scale of approximation m, whose value ranges in a very small set of integers, typically $m \in \{4-8\}$, and can be determined given a required tolerance. This provides a great advantage with respect to ASCOS, where the number of coefficients, N_c (and therefore the quadrature points, n_q), are chosen by a "trial-and-error" procedure.

The number of decimals in the references highly depends on the option payoff function and underlying model. Here, an arithmetic Asian valuation is considered, with varying number of monitoring dates, N = 12 (monthly), N = 50 (weekly) and N = 250 (daily), and assuming conceptually different underlying Lévy dynamics: GBM (Black-Scholes) and NIG [33]. In Table 1, we can see that, under GBM, we find up to ten coincident digits between ASCOS and SWIFT methods. However, when employing NIG model (see Table 2), we are not able to ensure more than four concordant decimals in the reference solution. In the subsequent experiments, this fact needs to be taken into consideration to fairly evaluate the numerical results.

Next, we employ the previously determined reference values (see Tables 1 and 2) to conduct a full pricing experiment. We then assess not only the accuracy in the solution but also the computational performance of two methods: the ASCOS method and the SWIFT method. The measure of the error is the absolute difference. When all the digits in the reference are coincident, (error below 10^{-10} or 10^{-4} for GBM or NIG models, respectively), the solution is considered exact (=). We can observe in Tables 3 and 4 that both methods provide satisfactory results for the evaluated models. It is worth highlighting that, given a certain precision, SWIFT is, in general, competitive with ASCOS, outperforming it specially when few monitoring dates and high accuracy is prescribed. In this situation, the impact on the computational cost of the numerical quadrature[4] employed in ASCOS is relatively more important. By using the SWIFT method we avoid any quadrature by using Theorem 1, which significantly reduces the cost of evaluating the integral in Eq. (20).

[4] Here, an improved version of the quadrature implementation has been used to be fair in the comparison. This implies that the computational cost of ASCOS is much lower than that reported in [5].

Table 3
SWIFT vs. ASCOS. Absolute error and execution time for the arithmetic Asian call option. Setting [5]: **GBM**, $S_0 = 100$, $r = 0.0367$, $\sigma = 0.17801$, $T = 1$ and $K = 90$. The "=" sign means that the result matches all the decimal digits of the reference value.

GBM		$N = 12$	$N = 50$	$N = 250$
ASCOS				
$N_c = 64$, $n_q = 100$	Error	3.75×10^{-4}	8.34×10^{-4}	7.17×10^{-3}
	Time (s)	0.03	0.02	0.01
$N_c = 128$, $n_q = 200$	Error	8.37×10^{-7}	7.43×10^{-6}	3.82×10^{-5}
	Time (s)	0.03	0.02	0.02
$N_c = 256$, $n_q = 400$	Error	=	5.33×10^{-7}	1.58×10^{-7}
	Time (s)	0.16	0.12	0.11
$N_c = 512$, $n_q = 800$	Error	=	=	3.04×10^{-8}
	Time (s)	1.96	1.80	1.85
$N_c = 1024$, $n_q = 1600$	Error	=	=	=
	Time (s)	13.99	13.99	14.25
SWIFT				
$m = 4$	Error	2.70×10^{-4}	1.27×10^{-2}	3.82×10^{-2}
	Time (s)	0.01	0.01	0.03
$m = 5$	Error	7.47×10^{-9}	9.78×10^{-5}	4.01×10^{-3}
	Time (s)	0.01	0.02	0.06
$m = 6$	Error	=	3.55×10^{-10}	6.96×10^{-4}
	Time (s)	0.02	0.10	0.40
$m = 7$	Error	=	=	1.21×10^{-8}
	Time (s)	0.08	0.34	1.37
$m = 8$	Error	=	=	=
	Time (s)	0.33	1.31	5.11

Table 4
SWIFT vs. ASCOS. Absolute error and execution time for the arithmetic Asian call option. Setting [5]: **NIG**, $S_0 = 100$, $r = 0.0367$, $\sigma = 0.0$, $\alpha = 6.1882$, $\beta = -3.8941$, $\delta = 0.1622$, $T = 1$ and $K = 110$. The "=" sign means that the result matches all the decimal digits of the reference value.

NIG		$N = 12$	$N = 50$	$N = 250$
ASCOS				
$N_c = 64$, $n_q = 100$	Abs error	7.78×10^{-3}	1.71×10^{-1}	8.75×10^{-2}
	CPU time	0.03	0.03	0.02
$N_c = 128$, $n_q = 200$	Abs error	2.60×10^{-4}	5.89×10^{-3}	1.49×10^{-2}
	CPU time	0.03	0.03	0.03
$N_c = 256$, $n_q = 400$	Abs error	=	=	1.42×10^{-4}
	CPU time	0.19	0.17	0.15
$N_c = 512$, $n_q = 800$	Abs error	=	=	=
	CPU time	1.98	1.96	2.02
$N_c = 1024$, $n_q = 1600$	Abs error	=	=	=
	CPU time	14.38	14.22	14.71
SWIFT				
$m = 4$	Abs error	9.72×10^{-2}	9.27×10^{-2}	4.01×10^{-2}
	CPU time	0.02	0.02	0.04
$m = 5$	Abs error	5.69×10^{-3}	6.92×10^{-4}	4.50×10^{-3}
	CPU time	0.02	0.03	0.08
$m = 6$	Abs error	2.13×10^{-4}	9.12×10^{-4}	9.11×10^{-4}
	CPU time	0.02	0.12	0.48
$m = 7$	Abs error	=	=	=
	CPU time	0.13	0.47	1.52
$m = 8$	Abs error	=	=	=
	CPU time	0.39	1.46	5.85

5.2. Results under square-root diffusions

We now wish to test the SWIFT method applied to square-root processes, applying the methodology explained in Section 4. Let us first consider the most basic square-root price dynamics

$$\mathrm{d}S(t) = \mu S(t)\mathrm{d}t + \sigma\sqrt{S(t)}\mathrm{d}W(t), \tag{43}$$

where μ and σ are constants and W is a Brownian motion. This formulation falls into the category presented in Section 4. Furthermore, the functions ψ and ϕ in Eq. (35) are analytically available,

$$\begin{aligned}\psi(\Delta t;\gamma) &= \frac{\gamma \mathrm{e}^{\mu\Delta t}}{1 + \dfrac{1}{2}\sigma^2\gamma\dfrac{\mathrm{e}^{\mu\Delta t} - 1}{\mu}},\\ \phi(\Delta t;\gamma) &= 0.\end{aligned}$$

We compare the results reported by Fusai et al. in [9], based on the Carr–Madan method, and those obtained by the SWIFT method presented here. We are able to reproduce exactly the same option prices with different configurations of strikes and monitoring dates for both European-style and floating-strike options. However, what we wish to highlight in this test is the computational efficiency of the SWIFT method with respect to Carr–Madan. Note that the number of terms in the expansion (N_{cm}) and the dampening factor used in this

Table 5
SWIFT vs. Carr–Madan. Execution time and speedup for the arithmetic Asian call option. Setting [9]: **Basic square-root**, $S_0 = 1$, $r = 0.04$, $\sigma = 0.7$ and $T = 1$.

# Decimals	Method	$N = 12$	$N = 50$	$N = 250$
4	Carr–Madan SWIFT Speedup	0.01 ($N_{cm} = 2^{14}$) 0.05 ($m = 3$) ×0.33	0.03 ($N_{cm} = 2^{14}$) 0.05 ($m = 3$) ×0.50	0.09 ($N_{cm} = 2^{14}$) 0.05 ($m = 2$) ×1.68
6	Carr–Madan SWIFT Speedup	0.03 ($N_{cm} = 2^{16}$) 0.05 ($m = 4$) ×0.58	0.09 ($N_{cm} = 2^{16}$) 0.06 ($m = 4$) ×1.47	0.32 ($N_{cm} = 2^{16}$) 0.07 ($m = 4$) ×4.39
8	Carr–Madan SWIFT Speedup	3.68 ($N_{cm} = 2^{23}$) 0.07 ($m = 5$) ×47.98	1.21 ($N_{cm} = 2^{20}$) 0.08 ($m = 5$) ×15.00	5.04 ($N_{cm} = 2^{20}$) 0.11 ($m = 5$) ×42.73

Table 6
SWIFT vs. Fusai et al. Option value for the arithmetic floating-strike Asian call option. Setting [9]: **Time-dependent square-root**, $S_0 = 7.1409$, $r = 0.0525$, $\sigma = 0.7$, $T = 1$, $\theta = -1$ and $m = 6$. Curve F is obtained by cubic spline interpolation, using the market quotes of Natural Gas at NYMEX on March 1, 2007 (see Table 6 in [9]).

Strike	Method	$N = 12$	$N = 50$	$N = 250$
$K = -0.02$	Fusai et al. SWIFT MC interval	0.151519 0.149538 [0.149074, 0.150394]	0.158608 0.156629 [0.155629, 0.156989]	0.160253 0.158235 [0.157383, 0.158754]
$K = 0.0$	Fusai et al. SWIFT MC interval	0.146705 0.144774 [0.144366, 0.145662]	0.153680 0.151750 [0.150683, 0.152020]	0.155301 0.153333 [0.152327, 0.153675]
$K = 0.02$	Fusai et al. SWIFT MC interval	0.141999 0.140118 [(0.139050, 0.140319]	0.148860 0.146978 [0.146085, 0.147399]	0.150458 0.148539 [0.147667, 0.148991]

experiment were not specified in the paper. Thus, after setting the dampening factor[5] to 0.05, an increasing N_{cm} is employed. In Table 5 we show the execution time required to achieve a certain precision (number of coincident digits) for both techniques. As shown, when mid-high precision is demanded, SWIFT method performs significantly faster than Carr–Madan, reaching up to 40 times reduction of the computational cost for eight decimal digits precision.

Secondly, we introduce a time-dependency in the model parameters, which are no longer constants but functions of the time, i.e. $\mu(t)$ and $\sigma(t)$. Lets assume that we have a number of market quotes of the forward prices, $F(t)$, observed at time $t = 0$. This is commonly the case in commodity and interest rate markets. Under these premises, closed-form expression for ψ (ϕ is again zero, $\phi(\Delta t;\gamma) = 0$) can be derived as well (see [9] for further details),

$$\begin{aligned}\psi(\Delta t;\gamma) &= \frac{\gamma \mathrm{e}^{\int_t^{t+\Delta t}\mu(s)\mathrm{d}s}}{1+\frac{1}{2}\gamma\int_t^{t+\Delta t}\sigma^2(s)\mathrm{e}^{\int_s^{t+\Delta t}\mu(u)\mathrm{d}u}\mathrm{d}s}\\ &= \frac{\gamma\frac{F(t+\Delta t)}{F(t)}}{1+\frac{1}{2}\gamma F(t+\Delta t)\int_t^{t+\Delta t}\frac{\sigma^2(s)}{F(s)}\mathrm{d}s},\end{aligned}$$

where we use the matching condition $F(t) = \mathbb{E}[S(t)] := S_0\mathrm{e}^{\int_0^t \mu(s)\mathrm{d}s}$.

We consider now a floating-strike arithmetic Asian option pricing problem. In Table 6 the prices reported by Fusai et al. in [9], and the ones computed by SWIFT are shown. As we can observe, there is a relatively large divergence from the second decimal digit. A Monte Carlo confidence (95%) interval is therefore included as a reference. We can observe that, while Fusai et al. results tend to slightly overestimate the option value, SWIFT method performs more accurately with all the produced prices within the Monte Carlo (MC) confidence interval. This can occur because the time dependency in the dynamics results in a transitional density with a heavy left tail and high negative skewness. In Fig. 2, we depict the recovered densities with and without time-dependency where the mentioned effect is clearly observed. Heavy-tailed distributions are typically harder to recover for the classical Fourier inversion-based techniques like Carr–Madan, where several free-parameters need to be arbitrarily chosen without any prior knowledge or optimal selection procedure. In the case of the Carr–Madan method, the number of terms in the expansion and the dampening constant need to be set. A wrong choice of these parameters can cause, like in this experiment, inaccurate results. In contrast, due to the control of the density approximation error by the scale of approximation m and its locality properties (by construction), SWIFT is able to efficiently recover the density function in this situation, capturing its singularities and providing therefore an accurate and fast option valuation.

As a third square-root diffusion, the underlying price is assumed to be driven by the classical CIR model [31], whose definition reads

$$\mathrm{d}S(t) = (a_c - b_c S(t))\mathrm{d}t + \sigma\sqrt{S(t)}\mathrm{d}W(t),$$

[5] We have not observed a significant impact of this parameter in the results when it is chosen among the usual values.

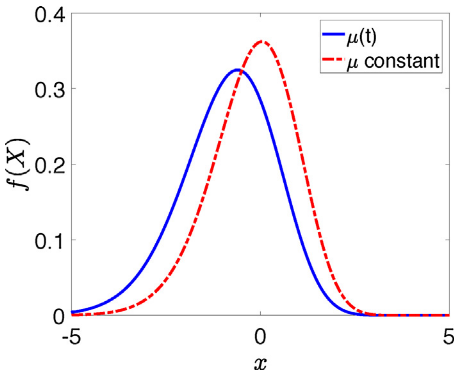

Fig. 2. Influence of the time-dependency on the transitional density.

Table 7
Option value for several types of arithmetic Asian caps. Setting [18]: **CIR**, $S_0 = 0.1$, $a_c = 0.15$, $b_c = 1.5$, $\sigma = 0.2$, $T = 1$, $K = S_0$ and $m = 6$.

	$N = 12$	$N = 50$	$N = 250$
Asian cap			
SWIFT	0.00750	0.00770	0.00776
MC interval	[0.00747, 0.00752]	[0.00766, 0.00771]	[0.00774, 0.00778]
Cash Binary Asian cap			
SWIFT	0.41225	0.41239	0.41241
MC interval	[0.41070, 0.41240]	[0.41182, 0.41352]	[0.41181, 0.41350]
Rate Binary Asian cap			
SWIFT	0.04873	0.04894	0.04900
MC interval	[0.04864, 0.04885]	[0.04890, 0.04911]	[0.04895, 0.04915]

Table 8
Option sensitivities, Greeks Δ and Γ. Setting: $K = S_0$, $N = 12$, $m = 6$.

Greek	Method	GBM	NIG	Square-root	Square-root, $\mu(t)$	CIR
Δ	SWIFT	0.57645	0.67561	0.53930	0.87165	0.27083
	Ref.	0.57036	0.67220	0.53705	0.87106	0.27111
Γ	SWIFT	0.03788	0.03639	0.97123	0.27936	5.14647
	Ref.	0.03777	0.03617	0.97356	0.27556	5.14679

where model parameter a_c introduces a mean-reverting term, while b_c controls the mean-reversion speed. Under the CIR dynamics, analytical solutions for both ψ and ϕ functions in Eq. (35) can be obtained (see [34], for example),

$$\begin{aligned}\psi(\Delta t;\gamma) &= \frac{2\gamma b_c}{\sigma^2\gamma\left(\mathrm{e}^{b_c\Delta t}-1\right)+2b_c\mathrm{e}^{b_c\Delta t}},\\ \phi(\Delta t;\gamma) &= \frac{-2a_c}{\sigma^2}\log\left(\frac{2b_c\mathrm{e}^{b_c\Delta t}}{\sigma^2\gamma\left(\mathrm{e}^{b_c\Delta t}-1\right)+2b_c\mathrm{e}^{b_c\Delta t}}\right).\end{aligned}$$

The CIR model is of particular importance in interest rate markets, where it appeared as an alternative of the well-known Vasicek model to model the short rate. For that, we here employ these dynamics to price Asian-like options appearing in the interest rate context. Specifically, three different Asian caps are chosen, arithmetic Asian caps, cash Binary Asian caps and rate Binary Asian caps, according to the payoff functions described in Eq. (3). In Table 7 we present the option prices computed by means of the SWIFT method. As a reference, a Monte Carlo (MC) confidence interval is again employed. Our technique turns to be highly precise, always falling into the confidence interval, even when it is extremely narrow. Regarding the computational cost, SWIFT execution time ranges around [0.08, 0.15] seconds with scale of approximation $m = 6$.

5.3. Greeks

By employing the SWIFT method, the option sensitivities with respect to the initial price, the Greeks Δ and Γ, can be readily computed from Eqs. (32) and (33) or from Eqs. (41) and (42), for exponential Lévy models or square-root diffusions, respectively. The resulting values are shown in Table 8, where a regular arithmetic Asian call is priced and we have employed the same parameter configuration as in the previous experiments corresponding to each dynamics. The reference values are computed by the *data-driven COS* method [35] and *Rolling Adjoints* method [36], for Lévy and square-root dynamics, respectively. These Monte Carlo-based techniques provide stable and accurate sensitivities. Note that the computation of the "Greek" coefficients in SWIFT can be done at once together with the payoff (or density) coefficients. Thus, it does not imply an increment in the computational cost. With $m = 6$, the execution time remains around 0.1 s.

6. Conclusions

In this paper, a new Fourier inversion-based technique has been proposed in the framework of discretely monitored Asian options. We have studied the performance of the SWIFT method on two broad classes of financial models, the exponential Lévy processes and square-root diffusions. The application of SWIFT to the Asian pricing problem allows to overcome the main drawbacks attributed to this type of methods, like the free-parameters control and the numerical integration. Furthermore, SWIFT results in a highly accurate and fast technique, outperforming the competitors in most of the analysed experiments. We have also shown that the method robustness avoids possible numerical errors appearing in involved situations, like heavy tailed distributions. Under the SWIFT context, not only the option price but also the option sensitivities can also be efficiently computed at the same time.

Acknowledgements

Álvaro Leitao acknowledges financial support from the Spanish Ministry of Economy and Competitiveness, through the María de Maeztu Programme for Units of Excellence in R&D (MDM-2014-0445).

Luis Ortiz-Gracia acknowledges the Spanish Ministry of Economy and Competitiveness for funding under Grants ECO2016-76203-C2-2 and MTM2016-76420-P (MINECO/FEDER, UE).

Emma I. Wagner acknowledges the support from the Delft University of Technology for partially funding her six-months research stay at University of Barcelona where part of the research presented in this manuscript was carried out.

Finally, the authors would like to thank Bowen Zhang from Rabobank and Cornelis W. Oosterlee from Centrum Wiskunde & Informatica (CWI) and Delft University of Technology (TU Delft) for providing the original codes of the ASCOS method for comparisons.

Appendix A. Solution of integrals $I_0^{j,k}, I_1^{j,k}, I_2^{j,k}, I_3^{j,k}, I_4^{j,k}$

The integrals appearing in the computation of the payoff coefficients $V_{m,k}$ have analytic solution,

$$\begin{aligned}
I_0^{j,k}(x_1,x_2) &= \int_{x_1}^{x_2} \cos\left(C_j\left(2^m y - k\right)\right)\mathrm{d}y \\
&= \frac{1}{2^m C_j}\left(\sin\left(C_j(2^m x_2 - k)\right) - \sin\left(C_j(2^m x_1 - k)\right)\right), \\
I_1^{j,k}(x_1,x_2) &= \int_{x_1}^{x_2} y\cos\left(C_j\left(2^m y - k\right)\right)\mathrm{d}y \\
&= \frac{1}{\left(2^m C_j\right)^2}\left(\cos\left(C_j(k - x_2 2^m)\right) - \cos\left(C_j(k - x_1 2^m)\right)\right) \\
&\quad + \frac{C_j 2^m}{\left(2^m C_j\right)^2}\left(x_1\sin\left(C_j(k - x_1 2^m)\right) - x_2\sin\left(C_j(k - x_2 2^m)\right)\right), \\
I_2^{j,k}(x_1,x_2) &= \int_{x_1}^{x_2} \mathrm{e}^{y}\cos\left(C_j\left(2^m y - k\right)\right)\mathrm{d}y \\
&= \frac{1}{1+(2^m C_j)^2}\left(\mathrm{e}^{x_2}\cos\left(C_j(2^m x_2 - k)\right) - \mathrm{e}^{x_1}\cos\left(C_j(2^m x_1 - k)\right)\right) \\
&\quad + \frac{2^m C_j}{1+(2^m C_j)^2}\left(\mathrm{e}^{x_2}\sin\left(C_j(2^m x_2 - k)\right) - \mathrm{e}^{x_1}\sin\left(C_j(2^m x_1 - k)\right)\right), \\
I_3^{j,k}(x_1,x_2,t) &= \int_{x_1}^{x_2} \mathrm{e}^{-yt}\cos\left(C_j\left(2^m y - k\right)\right)\mathrm{d}y \\
&= \frac{t}{(2^m C_j)^2 + t^2}\left(\mathrm{e}^{-x_1 t}\cos\left(C_j(k - 2^m x_1)\right) - \mathrm{e}^{-x_2 t}\cos\left(C_j(k - 2^m x_2)\right)\right) \\
&\quad + \frac{2^m C_j}{(2^m C_j)^2 + t^2}\left(\mathrm{e}^{-x_1 t}\sin\left(C_j(k - 2^m x_1)\right) - \mathrm{e}^{-x_2 t}\sin\left(C_j(k - 2^m x_2)\right)\right), \\
I_4^{j,k}(x_1,x_2,t) &= \int_{x_1}^{x_2} y\mathrm{e}^{-yt}\cos\left(C_j\left(2^m y - k\right)\right)\mathrm{d}y \\
&= \frac{(2^m C_j)^2 - t^2}{\left((2^m C_j)^2 + t^2\right)^2}\left(\mathrm{e}^{-x_2 t}\cos\left(C_j(k - 2^m x_2)\right) - \mathrm{e}^{-x_1 t}\cos\left(C_j(k - 2^m x_1)\right)\right) \\
&\quad + \frac{t}{(2^m C_j)^2 + t^2}\left(x_1\mathrm{e}^{-x_1 t}\cos\left(C_j(k - 2^m x_1)\right) - x_2\mathrm{e}^{-x_2 t}\cos\left(C_j(k - 2^m x_2)\right)\right) \\
&\quad + \frac{t 2^{m+1} C_j}{\left((2^m C_j)^2 + t^2\right)^2}\left(\mathrm{e}^{-x_1 t}\sin\left(C_j(k - 2^m x_1)\right) - \mathrm{e}^{-x_2 t}\sin\left(C_j(k - 2^m x_2)\right)\right) \\
&\quad + \frac{2^m C_j}{(2^m C_j)^2 + t^2}\left(x_1\mathrm{e}^{-x_1 t}\sin\left(C_j(k - 2^m x_1)\right) - x_2\mathrm{e}^{-x_2 t}\sin\left(C_j(k - 2^m x_2)\right)\right).
\end{aligned}$$

where we recall that $C_j = \frac{2j-1}{2J}\pi$.

References

[1] S. Levendorskii, Pricing arithmetic Asian options under Lévy models by backward induction in the dual space, SIAM J. Financ. Math. 9 (1) (2018) 1–27.
[2] A.P. Carverhill, L.J. Clewlow, Flexible convolution: valuing average rate (Asian) options, Risk Mag. 3 (4) (1990) 25–29.
[3] E. Benhamou, Fast Fourier transform for discrete Asian options, J. Comput. Finance 6 (1) (2002) 49–68.
[4] G. Fusai, A. Meucci, Pricing discretely monitored Asian options under Lévy processes, J. Bank. Finance 32 (10) (2008) 2076–2088.
[5] B. Zhang, C.W. Oosterlee, Efficient pricing of European-style Asian options under exponential Lévy processes based on Fourier cosine expansions, SIAM J. Financ. Math. 4 (1) (2013) 399–426.
[6] J.L. Kirkby, An efficient transform method for Asian option pricing, SIAM J. Financ. Math. 7 (1) (2016) 845–892.
[7] A. Cerný, I. Kyriakou, An improved convolution algorithm for discretely sampled Asian options, Quant. Finance 11 (3) (2011) 381–389.
[8] G. Fusai, D. Marazzina, M. Marena, Pricing discretely monitored Asian options by maturity randomization, SIAM J. Financ. Math. 2 (1) (2011) 383–403.
[9] G. Fusai, M. Marena, A. Roncoroni, Analytical pricing of discretely monitored Asian-style options: theory and application to commodity markets, J. Bank. Finance 32 (10) (2008) 2033–2045.
[10] P. Carr, D.B. Madan, Option valuation using the Fast Fourier transform, J. Comput. Finance 2 (1999) 61–73.
[11] N. Cai, C. Li, C. Shi, Closed-form expansions of discretely monitored Asian options in diffusion models, Math. Oper. Res. 39 (3) (2014) 789–822.
[12] N. Cai, Y. Song, S. Kou, A general framework for pricing Asian options under Markov processes, Oper. Res. 63 (3) (2015) 540–554.
[13] L. Ortiz-Gracia, C.W. Oosterlee, A highly efficient Shannon wavelet inverse Fourier technique for pricing European options, SIAM J. Sci. Comput. 38 (1) (2016) 118–143.
[14] G. Colldeforns-Papiol, L. Ortiz-Gracia, C.W. Oosterlee, Two-dimensional Shannon wavelet inverse Fourier technique for pricing European options, Appl. Numer. Math. 117 (Suppl. C) (2017) 115–138.
[15] S.C. Maree, L. Ortiz-Gracia, C.W. Oosterlee, Pricing early-exercise and discrete barrier options by Shannon wavelet expansions, Numer. Math. 136 (4) (2017) 1035–1070.
[16] E. Eberlein, A. Papapantoleon, Equivalence of floating and fixed strike Asian and lookback options, Stoch. Processes Appl. 115 (1) (2005) 31–40.
[17] A. Dassios, J. Nagaradjasarma, Pricing of Asian Options on Interest Rates in the CIR Model, Department of Statistics, London School of Economics and Political Science, London, UK, 2011, Available at: http://eprints.lse.ac.uk/32084.
[18] A. Prayoga, N. Privault, Pricing CIR yield options by conditional moment matching, Asia-Pac. Financ. Mark. 24 (1) (2017) 19–38.
[19] I. Daubechies, Ten Lectures on Wavelets, Society for Industrial and Applied Mathematics, 1992.
[20] C. Cattani, Shannon wavelets theory, Math. Probl. Eng. 2008 (2008) 164808.
[21] L. Ortiz-Gracia, C.W. Oosterlee, Robust pricing of European options with wavelets and the characteristic function, SIAM J. Sci. Comput. 35 (5) (2013) B1055–B1084.
[22] J.L. Kirkby, Efficient option pricing by frame duality with the fast Fourier transform, SIAM J. Financ. Math. 6 (1) (2015) 713–747.
[23] G. Colldefons-Papiol, L. Ortiz-Gracia, Computation of market risk measures with stochastic liquidity horizon, J. Comput. Appl. Math. 342 (2018) 431–450.
[24] F. Fang, C.W. Oosterlee, A novel pricing method for European options based on Fourier-cosine series expansions, SIAM J. Sci. Comput. 31 (2008) 826–848.
[25] W. Schoutens, Lévy Processes in Finance: Pricing Financial Derivatives, John Wiley & Sons, Ltd, 2003.
[26] P. Tankov, Pricing and Hedging in Exponential Lévy Models: Review of Recent Results, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 319–359.
[27] F. Stenger, Handbook of Sinc Numerical Methods, CRC Press, Inc., Boca Raton, FL, USA, 2010.
[28] D. Goss, The Ongoing Binomial Revolution, Birkhauser, Basel, 2014, pp. 183–193.
[29] A. Roncoroni, G. Fusai, M. Cummins, Handbook of Multi-Commodity Markets and Products: Structuring, Trading and Risk Management, John Wiley & Sons, Ltd, 2015.
[30] D. Brigo, F. Mercurio, Interest Rate Models – Theory and Practice, Springer Finance, Springer-Verlag Berlin Heidelberg, 2006.
[31] J.C. Cox, J.E. Ingersoll, S.A. Ross, A theory of the term structure of interest rates, Econometrica 53 (2) (1985) 385–407.
[32] D. Duffie, D. Filipović, W. Schachermayer, Affine processes and applications in finance, Ann. Appl. Probab. 13 (3) (2003) 984–1053.
[33] O.E. Barndorff-Nielsen, Normal inverse Gaussian distributions and stochastic volatility modelling, Scand. J. Stat. 24 (1) (1997) 1–13.
[34] D. Lamberton, B. Lapeyre, Introduction to Stochastic Calculus Applied to Finance, Chapman & Hall, London, 1996.
[35] A. Leitao, C.W. Oosterlee, L. Ortiz-Gracia, S.M. Bohte, On the data-driven COS method, Appl. Math. Comput. 317 (Suppl. C) (2018) 68–84.
[36] S. Jain, A. Leitao, C.W. Oosterlee, Rolling adjoints: fast Greeks along Monte Carlo scenarios for early-exercise options, Appl. Math. Comput. (2017), Available at SSRN: https://ssrn.com/abstract=3093846 (in press).

Currently, I am a BGSMath-María de Maeztu JUNIOR postdoctoral researcher in the Riskcenter-IREA research group at the University of Barcelona, funded by the Barcelona Graduate School of Mathematics after competitive call. I have obtained my BSc. degree in Computer Science at the University of A Coruña. After that, I got an MSc. in Mathematical Engineering at the University of Vigo with specialization in numerical techniques for mathematical finance. Then, I have worked at the Department of Mathematics in University of A Coruña on the high-performance implementation of mathematical models with financial applications. In 2013, I moved to The Netherlands to start my doctoral education awarded by a Marie Curie fellowship, in the framework of an International Training Network (ITN) called STRIKE – Novel Methods in Computational Finance. I carried out my doctoral studies at the Delft Institute of Applied Mathematics (DIAM) in the Delft University of Technology (TU Delft) and the Scientific Computing (SC) research group of the Centrum Wiskunde & Informatica (CWI), the National research center in mathematics and computer science in Amsterdam. My research topic is focused on the development of efficient numerical solution techniques in financial mathematics, a discipline that lies at the intersection of numerical analysis and stochastic calculus. My expertise includes stochastic processes and stochastic differential equations (SDEs), Monte Carlo methods and Fourier inversion techniques, applied to problems appearing in the financial sector. Particularly, I am interested in hybrid solutions for advanced quantitative problems, combining several methodologies (including computation) aiming a satisfactory balance between precision, robustness and efficiency. My expertise also includes GPU parallel computing. Regarding my programming skills, I regularly use C, Python and Matlab. I have taught several courses on Python programming for financial applications.