SIAM J. SCI. COMPUT.
Vol. 38, No. 1, pp. B118–B143

A HIGHLY EFFICIENT SHANNON WAVELET INVERSE FOURIER TECHNIQUE FOR PRICING EUROPEAN OPTIONS*

LUIS ORTIZ-GRACIA† AND CORNELIS W. OOSTERLEE‡

Abstract. In the search for robust, accurate, and highly efficient financial option valuation techniques, we here present the SWIFT method (Shannon wavelets inverse Fourier technique), based on Shannon wavelets. SWIFT comes with control over approximation errors made by means of sharp quantitative error bounds. The nature of the local Shannon wavelets basis enables us to adaptively determine the proper size of the computational interval. Numerical experiments on European-style options show exponential convergence and confirm the bounds, robustness, and efficiency.

Key words. option pricing, European options, Shannon wavelets, sinus cardinal function, Fourier transform inversion

AMS subject classifications. 62P05, 60E10, 65T60

DOI. 10.1137/15M1014164

1. Introduction.

European options are financial derivatives, governed by the solution of an integral, the so-called discounted expectation of a final condition, i.e., the pay-off function. A strain of literature dealing with highly efficient pricing of these contracts already exists, where the computation often takes place in Fourier space. For the computation of the expectation we require knowledge about the probability density function governing the stochastic asset price process, which is typically not available for relevant price processes. Methods based on quadrature and the fast Fourier transform (FFT) [Car99, Lee04, Lin08] and methods based on Fourier cosine expansions [Fan08, Rui12] have therefore been developed because for relevant *log-asset price processes* the characteristic function appears to be available. The characteristic function is defined as the Fourier transform of the density function. In this paper, we will explore the potential of Shannon wavelets [Cat08] for the valuation of European-style options, which is also based on the availability of the characteristic function. We will call the resulting numerical wavelets technique *SWIFT* (Shannon wavelet inverse Fourier technique).

It is well-known that cosines, in corresponding Fourier cosine expansions, form a global basis, and that comes with disadvantages in the vicinity of integration boundaries, where cosines exhibit periodicity behavior. Especially for long-maturity options round-off errors may accumulate near domain boundaries, whereas for short maturity options, governed by a highly peaked density function, many cosine terms may be needed for an accurate representation.

An accurate integration interval is also important to capture the mass of the density when dealing with heavy-tailed asset price distributions, and there is not an a priori uniform prescription for the suitable size of this interval for any asset price

*Submitted to the journal's Computational Methods in Science and Engineering section March 26, 2015; accepted for publication (in revised form) November 23, 2015; published electronically January 26, 2016.

http://www.siam.org/journals/sisc/38-1/M101416.html

†Centre de Recerca Matemàtica, Campus de Bellaterra, Edifici C, 08193 Bellaterra (Barcelona), Spain (lortiz@crm.cat).

‡CWI–Centrum Wiskunde & Informatica, NL-1090 GB Amsterdam, The Netherlands, and Delft University of Technology, Delft Institute of Applied Mathematics, 2628 CD Delft, The Netherlands (C.W.Oosterlee@cwi.nl).

process and option contract. Larger intervals would require a larger number of cosine terms to reach a specific level of accuracy.

For the reasons mentioned above, local wavelet bases have been considered in [Ort13, Ort14], where we focused on Haar and B-spline wavelets. These local bases give flexibility and enhance robustness when treating long maturity options and heavy-tailed asset processes, but at the cost of a more involved computation and certain loss of accuracy for the standard cases, where the COS method [Fan08] exhibits exponential convergence. Employing these local wavelet bases, Kirkby in [Kir14] computes the density coefficients by means of Parseval's identity instead of relying on the Cauchy's integral theorem in [Ort13] and used an FFT algorithm to speed up the method. Although Parseval's identity allows one to get an exact expression of the coefficients, the method involves the truncation of the infinite integration domain onto a finite interval, which restricts the robustness. The convergence of these schemes in the number of wavelet terms is only algebraic.

Shannon wavelets, based on the sinus cardinal (sinc) function, are very interesting alternatives, however. Therefore, in this work we present the SWIFT method. We benefit from the local features of the approximation to the density function, like in [Ort13], but this time the convergence of the presented method is exponential due to the regularity of the employed basis. Long- and short-maturity options are priced robustly and are highly accurate, as are fat-tailed asset price distributions.

A key contribution is that the SWIFT method does not need to rely on an a priori truncation of the integration range as the *local* wavelet basis adaptively indicates whether the density mass is in accordance with a predefined tolerance. Furthermore, the density approximation does not deteriorate with the choice of domain size, and the number of terms needed in the expansion is automatically calculated by the method in terms of the length of the interval. We present two efficient alternatives to compute the density coefficients. First of all, there is Vieta's formula, also called Euler's formula, connecting SWIFT's sinc function to a product of cosines. We will present an accurate error analysis to determine the number of terms needed to meet a predefined error. With this estimate of the error at hand, the density coefficients as well as the pay-off coefficients are efficiently computed. The second alternative for the density coefficients calculation is based on Parseval's identity, like in [Kir14]. However, due to the compact support of the Shannon basis in the transformed domain, the exact coefficients expression avoids the so-called finite domain truncation error. Moreover, we use an FFT algorithm to enhance the speed of the method. It is worth remarking that the SWIFT method is easy to implement and works out well in a wide variety of cases, including pricing of multiple options simultaneously with multiple pay-off functions.

The paper is organized as follows. In section 2 we present the option pricing problem and give a brief summary of Shannon wavelets and multiresolution analysis. In section 3 we present the SWIFT method to recover a density function from its Fourier transform and we compute the pay-off coefficients. We present numerical examples in section 4. Finally, section 5 concludes.

2. Motivation: Option pricing. The pricing of European options in computational finance is governed by the numerical solution of partial differential or partial integro-differential, equations. The corresponding solution, being the option value at time t, can also be found by means of the Feynman–Kac formula as a discounted expectation of the option value at final time $t = T$, the so-called pay-off function.

Here we consider this *risk-neutral option valuation formula*,

$$v(x,t) = e^{-r(T-t)}\mathbb{E}^{\mathbb{Q}}\left(v(y,T)|x\right) = e^{-r(T-t)}\int_{\mathbb{R}} v(y,T)f(y|x)\,dy, \tag{1}$$

where v denotes the option value, T is the maturity, t is the initial date, $\mathbb{E}^{\mathbb{Q}}$ is the expectation operator under the risk-neutral measure $\mathbb{Q}$, x and y are state variables at time t and T, respectively, $f(y|x)$ is the probability density of y given x, and r is the deterministic risk-neutral interest rate.

Whereas f is typically not known, the characteristic function of the log-asset price is often available (sometimes in closed form) as the Fourier transform of f. We represent the option values as functions of the scaled log-asset prices and denote these prices by

$$x = \ln(S_t/K) \quad \text{and} \quad y = \ln(S_T/K)$$

with S_t the underlying price at time t and K the strike price.

The pay-off $v(y,T)$ for European options in log-asset space is then given by

$$v(y,T) = \left[\alpha \cdot K\left(e^y - 1\right)\right]^+ \quad \text{with} \quad \alpha = \begin{cases} 1 & \text{for a call,} \\ -1 & \text{for a put.} \end{cases} \tag{2}$$

The strategy to follow to determine the price of the option consists of approximating the density function f in (1) by means of a *finite combination of Shannon scaling functions* and recovering the coefficients of the approximation from its Fourier transform. The SWIFT method only relies on the availability of the Fourier transform of f and therefore it may be used beyond the risk-neutral measure setting. A brief review of the basic theory on Shannon wavelets is given in the next section.

2.1. Multiresolution analysis and Shannon wavelets. Consider the space

$$L^2(\mathbb{R}) = \left\{ f : \int_{-\infty}^{+\infty} |f(x)|^2\,dx < \infty \right\}$$

of square integrable functions. A general structure for wavelets in $L^2(\mathbb{R})$ is called a *multiresolution analysis*. We start with a family of closed nested subspaces,

$$\cdots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \cdots$$

in $L^2(\mathbb{R})$, where

$$\bigcap_{j\in\mathbb{Z}} V_j = \{0\}, \overline{\bigcup_{j\in\mathbb{Z}} V_j} = L^2(\mathbb{R})$$

and

$$f(x) \in V_j \iff f(2x) \in V_{j+1}.$$

If these conditions are met, then a function $\phi \in V_0$ exists such that $\{\phi_{j,k}\}_{k\in\mathbb{Z}}$ forms an orthonormal basis of V_j, where

$$\phi_{j,k}(x) = 2^{j/2}\phi(2^j x - k).$$

In other words, the function ϕ, called the *scaling function* or *father wavelet*, generates an orthonormal basis for each V_j subspace.

Let us define W_j in such a way that $V_{j+1} = V_j \oplus W_j$. This is, W_j is the space of functions in V_{j+1} but not in V_j, and so $L^2(\mathbb{R}) = \sum_j \oplus W_j$. Then (see [Dau92]) a function $\psi \in W_0$ exists, such that by defining

$$\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k),$$

$\{\psi_{j,k}\}_{k\in\mathbb{Z}}$ gives an orthonormal basis of W_j and $\{\psi_{j,k}\}_{j,k\in\mathbb{Z}}$ is a wavelet basis of $L^2(\mathbb{R})$. The ψ function is also called the *mother wavelet* and the $\psi_{j,k}$ functions are known as *wavelet functions*.

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

$$\mathcal{P}_m f(x) = \sum_{j=-\infty}^{m-1} \sum_{k=-\infty}^{k=+\infty} d_{j,k}\psi_{j,k}(x) = \sum_{k\in\mathbb{Z}} c_{m,k}\phi_{m,k}(x), \tag{3}$$

where $d_{j,k} = \int_{-\infty}^{+\infty} f(x)\psi_{j,k}(x)\,dx$ are the *wavelet coefficients* and the $c_{m,k} = \int_{-\infty}^{+\infty} f(x)\phi_{m,k}(x)\,dx$ are the *scaling coefficients*. Note that the first part in (3) is a truncated wavelet series. If j were allowed to go to infinity, we would have the full wavelet summation. The second part of (3) gives us an equivalent sum in terms of the scaling functions $\phi_{m,k}$. 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 j value) to accurately represent the local properties of a function.

In our financial mathematics context, we here consider Shannon wavelets (see [Cat08]). The *sinc* function or Shannon scaling function is the starting point for the definition of the Shannon wavelet family. A set of Shannon scaling functions or father wavelets in the subspace V_m is defined as

$$\phi_{m,k}(x) = 2^{m/2}\frac{\sin(\pi(2^m x - k))}{\pi(2^m x - k)}, \quad k \in \mathbb{Z}. \tag{4}$$

It is clear that for $m = k = 0$, we have the basic scaling function or father wavelet,

$$\phi(x) = \mathrm{sinc}(x),$$

where $\mathrm{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$. The wavelet functions in W_m are given by

$$\psi_{m,k}(x) = 2^{m/2}\frac{\sin(\pi(2^m x - k - 1/2)) - \sin(2\pi(2^m x - k - 1/2))}{\pi(2^m x - k - 1/2)}, \quad k \in \mathbb{Z}.$$

Shannon wavelets represent the real part of the so-called harmonic wavelets. They have a slow decay in the time domain but a very sharp compact support in the frequency (Fourier) domain,

$$\hat{\phi}_{m,k}(w) = \frac{e^{-i\frac{k}{2^m}w}}{2^{m/2}}\mathrm{rect}\left(\frac{w}{2^{m+1}\pi}\right) \tag{5}$$

and

$$\hat{\psi}_{m,k}(w) = -\frac{e^{-i\frac{k+1/2}{2^m}w}}{2^{m/2}}\left(\mathrm{rect}\left(\frac{w}{2^m\pi} - \frac{3}{2}\right) + \mathrm{rect}\left(-\frac{w}{2^m\pi} - \frac{3}{2}\right)\right), \tag{6}$$

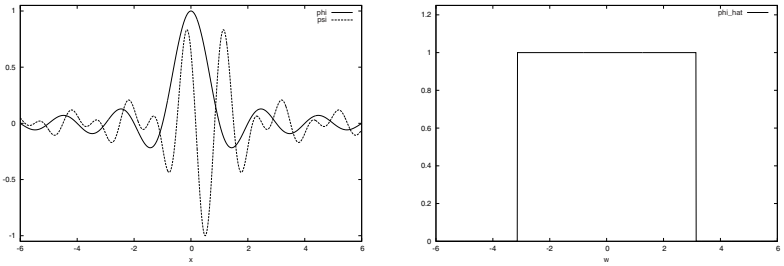

FIG. 1. *Left plot: Shannon scaling function $\phi(x)$ (thick line) and wavelet $\psi(x)$ (dashed line). Right plot: $\hat{\phi}(w)$.*

where rect is the *rectangle* function, defined as

$$\text{rect}(x) = \begin{cases} 1 & \text{if } |x| < 1/2, \\ 1/2 & \text{if } |x| = 1/2, \\ 0 & \text{if } |x| > 1/2. \end{cases}$$

This fact, together with Parseval's identity, is used to easily compute the inner products and the expansion coefficients of the Shannon wavelets, as we will see in the next section. In this work, we consider the Shannon father function in the time domain rather than the mother wavelet, due to its simplicity. Figure 1 shows the Shannon scaling function ϕ, the Shannon mother wavelet ψ, and the Fourier transform of the Shannon scaling function $\hat{\phi}$.

3. SWIFT. In this section we present the SWIFT method, based on the Shannon wavelets, to accurately invert the Fourier transform of density function f in (1). We consider an expansion of function f in terms of the Shannon scaling functions at a fixed approximation level m and recover the coefficients from their Fourier transforms. We present two different alternatives to compute the coefficients. While the first one is based on Vieta's formula [Gea90], which allows us to expand the cardinal sinus function into a combination of cosines, the second one relies on the application of Parseval's identity. In the second method, we benefit from compact support features of the Fourier transform of the Shannon basis functions, avoiding this way the truncation of the infinite domain of integration. Although both methods are very similar in terms of efficiency, we will select the first method throughout the paper, since we can derive a bound to estimate the error when approximating the sinus cardinal function as well as when computing the density coefficients.

Let us consider a probability density function f in $L^2(\mathbb{R})$ associated to a certain continuous random variable X, and its Fourier transform

$$\hat{f}(w) = \int_{\mathbb{R}} e^{-iwx} f(x)\, dx. \tag{7}$$

Following the wavelets theory, function f can be approximated at a level of resolution m, i.e.,

$$f(x) \approx \mathcal{P}_m f(x) = \sum_{k \in \mathbb{Z}} c_{m,k} \phi_{m,k}(x), \tag{8}$$

where $\mathcal{P}_m f$ converges to f in $L^2(\mathbb{R})$, that is, $\|f - \mathcal{P}_m f\|_2 \to 0$, when $m \to +\infty$, $\phi_{m,k}(x) = 2^{m/2}\phi(2^m x - k)$, $\phi(x) = \text{sinc}(x)$, $c_{m,k} = \langle f, \phi_{m,k}\rangle$, and $< f, g >= \int_{\mathbb{R}} f(x)\overline{g(x)}\,dx$ denotes the inner product in $L^2(\mathbb{R})$; the bar denoting complex conjugation. Following the multiresolution analysis theory in section 2.1,

$$\|f - \mathcal{P}_m f\|_2^2 = \sum_{j \geq m} \sum_{k \in \mathbb{Z}} |d_{j,k}|^2, \tag{9}$$

where $d_{j,k} = \langle f, \psi_{j,k}\rangle$. By Parseval's identity and (6),

$$\begin{aligned} \langle f, \psi_{j,k}\rangle =& \frac{1}{2\pi}\left\langle \hat{f}, \hat{\psi}_{j,k}\right\rangle \\ =& -\frac{2^{-j/2}}{2\pi}\left(\int_{2^j\pi}^{2^{j+1}\pi} \hat{f}(w) e^{\frac{iw(k+1/2)}{2^j}}\,dw + \int_{-2^{j+1}\pi}^{-2^j\pi} \hat{f}(w) e^{\frac{iw(k+1/2)}{2^j}}\,dw\right). \end{aligned} \tag{10}$$

Therefore, the approximation error in (9) is highly dependent on the rate of decay of the Fourier transform $\hat{f}$ of f, since by (10),

$$|d_{j,k}| \leq \frac{2^{-j/2}}{2} 2^j \left(\max_{w \in [2^j\pi, 2^{j+1}\pi]} |\hat{f}(w)| + \max_{w \in [-2^{j+1}\pi, -2^j\pi]} |\hat{f}(w)|\right). \tag{11}$$

This is due to the sharp decay of the Shannon wavelets in the Fourier domain and will be illustrated in the numerical experiments section.

Considering the same notation as before, the following lemma holds.

LEMMA 1. *Let us assume that the density function f tends to zero at plus and minus infinite. Then $f\left(\frac{h}{2^m}\right) \approx 2^{\frac{m}{2}} c_{m,h}, h \in \mathbb{Z}$, and the scaling coefficients $c_{m,k}$ satisfy*

$$\lim_{k \to \pm\infty} c_{m,k} = 0.$$

Proof. We observe from (4) that $\phi_{m,k}\left(\frac{h}{2^m}\right) = 2^{\frac{m}{2}}\delta_{kh}$, where δ_{kh} is the Kronecker delta. If we evaluate the approximation to the density function f in (8) at $\frac{h}{2^m}$, this gives us

$$f\left(\frac{h}{2^m}\right) \approx \mathcal{P}_m f\left(\frac{h}{2^m}\right) = 2^{\frac{m}{2}} \sum_{k \in \mathbb{Z}} c_{m,k}\delta_{k,h} = 2^{\frac{m}{2}} c_{m,h}. \tag{12}$$

Since we assume that our density function f tends to zero at plus and minus infinity, this completes the proof. ∎

Lemma 1 guarantees that the infinite series in (8) is well-approximated by a finite summation without loss of considerable density mass,

$$\mathcal{P}_m f(x) \approx f_m(x) := \sum_{k=k_1}^{k_2} c_{m,k}\phi_{m,k}(x), \tag{13}$$

for certain accurately chosen values k_1 and k_2. Taking into account that $\|\phi_{m,k}\| = 1$ (see [Cat08] for details), then

$$\|\mathcal{P}_m f - f_m\|_2^2 = \sum_{k < k_1 \text{ or } k > k_2} |c_{m,k}|^2,$$

and therefore the error of truncating the infinite sum in (8) into a finite sum in (13) depends on the size of scaling coefficients $c_{m,k}$, which in turn depend on the decay rate of f, as seen in Lemma 1.

At this stage, it is worth underlining an interesting feature of Shannon scaling functions regarding the area underneath the density function f. This area can already be obtained while the coefficients are being calculated. As shown in expression (12), once coefficient $c_{m,h}$ has been computed, the value of f evaluated at $h/2^m$ is directly obtained, and, therefore, the application of a trapezoidal rule makes sense. If we note by $\mathcal{A}$ the approximation to the area under the curve, then

$$\mathcal{A} = \frac{1}{2^{m/2}} \left(\frac{c_{m,k_1}}{2} + \sum_{k_1<k<k_2} c_{m,k} + \frac{c_{m,k_2}}{2} \right), \tag{14}$$

where $\mathcal{A} \approx 1$.

In the numerical examples section, we will present a procedure to accurately determine values for k_1 and k_2 with the help of the above properties.

3.1. Coefficients computation. We present two alternatives to compute the coefficients in expression (13) and provide an estimate for the error for one of these two methods. We start by considering

$$c_{m,k} = \langle f, \phi_{m,k} \rangle = \int_{\mathbb{R}} f(x) \overline{\phi}_{m,k}(x)\, dx = 2^{m/2} \int_{\mathbb{R}} f(x) \phi(2^m x - k)\, dx. \tag{15}$$

3.1.1. Coefficients via Vieta's formula. Using the classical *Vieta formula* [Gea90], the cardinal sinus can be expressed as an infinite product, i.e.,

$$\operatorname{sinc}(t) = \prod_{j=1}^{+\infty} \cos\left(\frac{\pi t}{2^j}\right). \tag{16}$$

If we truncate the infinite product to a finite product with J factors, then, thanks to the cosine product-to-sum identity [Qui13], we have

$$\prod_{j=1}^{J} \cos\left(\frac{\pi t}{2^j}\right) = \frac{1}{2^{J-1}} \sum_{j=1}^{2^{J-1}} \cos\left(\frac{2j-1}{2^J}\pi t\right). \tag{17}$$

By (16) and (17) the sinc function can thus be approximated as follows:

$$\operatorname{sinc}(t) \approx \operatorname{sinc}^*(t) := \frac{1}{2^{J-1}} \sum_{j=1}^{2^{J-1}} \cos\left(\frac{2j-1}{2^J}\pi t\right). \tag{18}$$

If we replace function ϕ in (15) by its approximation (18), then

$$c_{m,k} \approx c^*_{m,k} := \frac{2^{m/2}}{2^{J-1}} \sum_{j=1}^{2^{J-1}} \int_{\mathbb{R}} f(x) \cos\left(\frac{2j-1}{2^J}\pi(2^m x - k)\right) dx. \tag{19}$$

Taking into account that $\Re(\hat{f}(w)) = \int_{\mathbb{R}} f(x)\cos(wx)\, dx$ in expression (7)[1] and observing that

$$\hat{f}(w) e^{ik\pi\frac{2j-1}{2^J}} = \int_{\mathbb{R}} e^{-i\left(wx - \frac{k\pi(2j-1)}{2^J}\right)} f(x)\, dx,$$

[1] $\Re(z)$ denotes the real part of z.

we end up with the following expression for computing the density coefficients:

$$c_{m,k} \approx c^*_{m,k} = \frac{2^{m/2}}{2^{J-1}} \sum_{j=1}^{2^{J-1}} \Re \left[\hat{f}\left(\frac{(2j-1)\pi 2^m}{2^J} \right) e^{\frac{ik\pi(2j-1)}{2^J}} \right]. \tag{20}$$

The following lemma gives us an estimate of the error when approximating the sinus cardinal function. As we will observe, the convergence in a fixed interval is exponential.

LEMMA 2. *Define the absolute error* $\mathcal{E}_V(t) := sinc(t) - sinc^*(t)$. *Then*

$$|\mathcal{E}_V(t)| \leq \frac{(\pi c)^2}{2^{2(J+1)} - (\pi c)^2}$$

for $t \in [-c, c]$, *where* $c \in \mathbb{R}, c > 0$, *and* $J \geq \log_2(\pi c)$.

Proof. From expression (16) we derive the relation

$$\operatorname{sinc}(t) = \operatorname{sinc}\left(\frac{t}{2^J}\right) \cdot \prod_{j=1}^{J} \cos\left(\frac{\pi t}{2^j}\right),$$

and therefore

$$|\mathcal{E}_V(t)| = \left| \operatorname{sinc}\left(\frac{t}{2^J}\right) - 1 \right| \cdot \left| \prod_{j=1}^{J} \cos\left(\frac{\pi t}{2^j}\right) \right|. \tag{21}$$

If we take into account the Taylor series of $\operatorname{sinc}(t)$ (see, for instance, [Gea90]),

$$\operatorname{sinc}\left(\frac{t}{2^J}\right) = \sum_{n \geq 0} (-1)^n \pi^{2n} \frac{t^{2n}}{(2n+1)! \cdot 2^{2Jn}},$$

and using it in (21) gives us

$$|\mathcal{E}_V(t)| \leq \sum_{n \geq 1} \frac{\pi^{2n}}{(2n+1)! \cdot 2^{2Jn}} |t|^{2n},$$

since $|\prod_{j=1}^{J} \cos\left(\frac{\pi t}{2^j}\right)| \leq 1$. If we consider $|t| \leq c$, for certain $c \in \mathbb{R}, c > 0$, and observe that $\frac{1}{(2n+1)!} < \frac{1}{2^{2n}}$ for all $n \geq 1$, we find

$$|\mathcal{E}_V(t)| \leq \sum_{n \geq 1} \left(\frac{\pi c}{2^{J+1}} \right)^{2n}. \tag{22}$$

The sum at the right-hand side of (22) forms a geometric series which converges when $J \geq \log_2(\pi c)$ and, in this case, we have

$$\sum_{n \geq 1} \left(\frac{\pi c}{2^{J+1}} \right)^{2n} = \frac{(\pi c)^2}{2^{2(J+1)} - (\pi c)^2}. \qquad \square$$

We assess the error estimation in Lemma 2 (that is, the theoretical error) by plotting in Figure 2 the empirical and the theoretical error. For this purpose, we fix

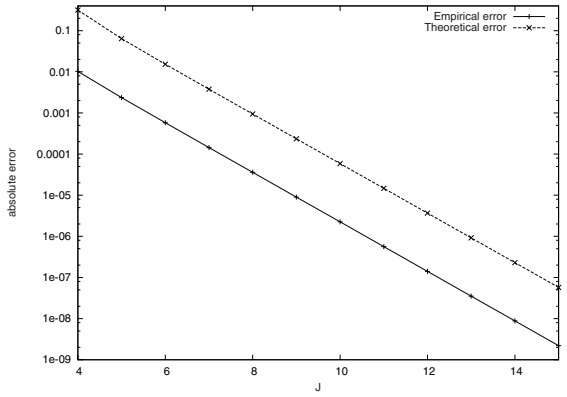

FIG. 2. *Empirical and theoretical error when approximating sinc(t) by sinc***(t) within the interval* $[-5, 5]$.

the interval of approximation $[-5, 5], c = 5$. We consider $J \geq \log_2(\pi c) = 4$ and define the empirical error for each J-value as

$$\varepsilon_{emp} = \max_{\substack{t_j=-5+0.01j \\ j=0,\ldots,1000}} |\text{sinc}(t_j) - \text{sinc}^*(t_j)|.$$

As expected, the thick line corresponding to the empirical error is below the dashed line belonging to the theoretical error.

THEOREM 1. *Let $F(x)$ be the distribution function of a random variable X and define $H(x) := F(-x) + 1 - F(x)$. Let $a > 0$ be a constant such that $H(a) < \epsilon$ for $\epsilon > 0$. Define $M_{m,k} := \max(|2^m a - k|, |2^m a + k|)$ and consider $J \geq \log_2(\pi M_{m,k})$.*

Then,

$$|c_{m,k} - c^*_{m,k}| \leq 2^{m/2} \left(2\epsilon + \sqrt{2a} \|f\|_2 \frac{(\pi M_{m.k})^2}{2^{2(J+1)} - (\pi M_{m.k})^2} \right),$$

and $\lim_{J \to +\infty} c^*_{m,k} = c_{m,k}$.

Proof. Considering (15) and (19), we find

$$\begin{aligned} |c_{m,k} - c^*_{m,k}| &= 2^{m/2} \left| \int_{\mathbb{R}} f(x) \left(\text{sinc}(2^m x - k) - \text{sinc}^*(2^m x - k) \right) dx \right| \\ &\leq 2^{m/2} \left[\int_{\mathbb{R} \setminus [-a,a]} f(x) \left| \text{sinc}(2^m x - k) - \text{sinc}^*(2^m x - k) \right| dx \right. \\ &\quad \left. + \int_{-a}^{a} f(x) \left| \text{sinc}(2^m x - k) - \text{sinc}^*(2^m x - k) \right| dx \right]. \end{aligned} \tag{23}$$

Note that for all $\epsilon > 0$, there exists $a > 0$ such that $H(a) < \epsilon$, since the mass in the tails of the density $f(x)$ tends to zero when a tends to infinity. Further, if the characteristic function $\hat{f}$ is analytic in $|z| < R$ for $R > 0$, then $H(a) = \mathcal{O}(e^{-ra})$ for all $0 \leq r \leq R$, when a tends to infinity (see [Ush99] for details). Taking into account that $|\text{sinc}(2^m x - k) - \text{sinc}^*(2^m x - k)| \leq 2$ for all $x \in \mathbb{R}$, the first integral at the right-hand

side of inequality (23) is bounded by 2ϵ. If we apply the Cauchy–Schwarz inequality to the second integral at the right-hand side of inequality (23), we find

$$\int_{-a}^{a} f(x)\,|\mathrm{sinc}(2^m x - k) - \mathrm{sinc}^*(2^m x - k)|\,dx$$
$$\leq \|f\|_2 \left(\int_{-a}^{a} (\mathrm{sinc}(2^m x - k) - \mathrm{sinc}^*(2^m x - k))^2 \right)^{\frac{1}{2}}.$$

We observe that if $-a \leq x \leq a$, then $-2^m a - k \leq 2^m x - k \leq 2^m a - k$. Therefore, $2^m x - k \in [-M_{m,k}, M_{m,k}]$ and the proof concludes by applying Lemma 2. ∎

Remark 1. Although a different J can be selected for each k, we prefer to consider a constant J, defined here by $\jmath := \lceil \log_2(\pi M_m) \rceil$, where $M_m := \max_{k_1<k<k_2} M_{m,k}$ and $\lceil x \rceil$ denotes the smallest integer greater or equal than x. The reason is that, in practice, the computationally most involved part in (20) is the evaluation of $\hat{f}$ at the grid points. Those values can be computed only once and used by the FFT algorithm,[2] as follows. From expression (20),

$$\begin{aligned} c^*_{m,k} &= \frac{2^{m/2}}{2^{\jmath-1}} \sum_{j=1}^{2^{\jmath-1}} \Re \left[\hat{f}\left(\frac{(2j-1)\pi 2^m}{2^\jmath} \right) e^{\frac{ik\pi(2j-1)}{2^\jmath}} \right] \\ &= \frac{2^{m/2}}{2^{\jmath-1}} \Re \left[e^{-\frac{ik\pi}{2^\jmath}} \sum_{j=1}^{2^{\jmath-1}} \hat{f}\left(\frac{(2j-1)\pi 2^m}{2^\jmath} \right) e^{\frac{ik\pi j}{2^{\jmath-1}}} \right] \\ &= \frac{2^{m/2}}{2^{\jmath-1}} \Re \left[e^{-\frac{ik\pi}{2^\jmath}} \sum_{j=0}^{2^{\jmath-1}-1} \hat{f}\left(\frac{(2j+1)\pi 2^m}{2^\jmath} \right) e^{\frac{ik\pi(j+1)}{2^{\jmath-1}}} \right] \\ &= \frac{2^{m/2}}{2^{\jmath-1}} \Re \left[e^{\frac{ik\pi}{2^\jmath}} \sum_{j=0}^{2^{\jmath-1}-1} \hat{f}\left(\frac{(2j+1)\pi 2^m}{2^\jmath} \right) e^{\frac{2\pi ikj}{2^\jmath}} \right]. \end{aligned} \tag{24}$$

Finally, we assume that $\hat{f}(\frac{(2j+1)\pi 2^m}{2^\jmath}) = 0$, from $2^{\jmath-1}$ to $2^\jmath - 1$, so that the last equality in (24) is equivalent to

$$c^*_{m,k} = \frac{2^{m/2}}{2^{\jmath-1}} \Re \left[e^{\frac{ik\pi}{2^\jmath}} \sum_{j=0}^{2^\jmath-1} \hat{f}\left(\frac{(2j+1)\pi 2^m}{2^\jmath} \right) e^{\frac{2\pi ikj}{2^\jmath}} \right], \tag{25}$$

and therefore the FFT algorithm can be applied to compute the density coefficients $c^*_{m,k}$.

The computational complexity associated to a direct computation of the density coefficients by (20) is $\mathcal{O}\left(2^{\jmath-1} \cdot (k_2 - k_1 + 1)\right)$, while the complexity during the application of the FFT algorithm explained above is $\mathcal{O}\left(\log(2) \cdot \jmath \cdot 2^{\jmath+1}\right)$, with k_1 and k_2 fixed indices.

3.1.2. Coefficients via Parseval's identity. As mentioned before, thanks to the compact support of $\hat{\phi}_{m,k}$, the coefficients of the density function can also be accurately calculated by *Parseval's identity*. We further detail this in the next remark.

[2] All the subroutines used throughout the numerical experiments section belong to the C library FFTW.

Remark 2. By Parseval's identity, $\langle f, \phi_{m,k}\rangle = \frac{1}{2\pi}\langle \hat{f}, \hat{\phi}_{m,k}\rangle$ and considering (5),

$$\left\langle \hat{f}, \hat{\phi}_{m,k} \right\rangle = \frac{1}{2^{m/2}} \int_{\mathbb{R}} \hat{f}(w) \cdot e^{i\frac{k}{2^m}w} \cdot \operatorname{rect}\left(\frac{w}{2^{m+1}\pi}\right) dw = \frac{1}{2^{m/2}} \int_{-2^m\pi}^{2^m\pi} \hat{f}(w) \cdot e^{i\frac{k}{2^m}w}\, dw.$$

If we perform the following change of variables, $t = \frac{w}{2^{m+1}\pi}$, then

$$\left\langle \hat{f}, \hat{\phi}_{m,k} \right\rangle = 2\pi \cdot 2^{m/2} \int_{-\frac{1}{2}}^{\frac{1}{2}} \hat{f}(2^{m+1}\pi t) \cdot e^{i2\pi kt}\, dt.$$

Taking into account Parseval's identity and the fact that $c_{m,k} = \langle f, \phi_{m,k}\rangle$ are real-valued coefficients, we have

$$c_{m,k} = 2^{m/2} \int_{-\frac{1}{2}}^{\frac{1}{2}} \Re\left(\hat{f}(2^{m+1}\pi t) \cdot e^{i2\pi kt}\right) dt,$$

where $\Re(z)$ denotes the real part of z, and the coefficients are computed by means of the trapezoidal rule. It is worth remarking that an FFT algorithm can be applied in this case as well to speed up the computations.

3.2. Option pricing problems. In the following subsections, we give details for the computation of the pay-off coefficients for different options. The cash-or-nothing option is described briefly first, as its derivation is easier, and it serves as a first step for the pay-off coefficients of the well-known European options. We consider the approximation to the density function f in terms of the Shannon scaling functions, i.e.,

$$f(y|x) \approx f_m(y|x) = \sum_{k=k_1}^{k_2} c_{m,k}(x)\phi_{m,k}(y), \tag{26}$$

where $c_{m,k}(x)$ are the density coefficients in the derivations.

3.2.1. Cash-or-nothing options pricing. Cash-or-nothing options belong to the class of binary or digital options and they are used as building blocks to set up more complicated financial products. The pay-off function $v(y,T)$ for a cash-or-nothing call option in scaled log-asset space reads

$$v(y,T) = \begin{cases} 1 & \text{if } y > 0, \\ 0 & \text{otherwise.} \end{cases}$$

We replace f by its approximation f_m in the valuation formula (1) to obtain for the cash-or-nothing option

$$\begin{aligned} v(x,t) &= e^{-r(T-t)} \int_0^{+\infty} f(y|x)\, dy \\ &\approx v_1(x,t) = e^{-r(T-t)} \int_0^{+\infty} f_m(y|x)\, dy = e^{-r(T-t)} \sum_{k=k_1}^{k_2} c_{m,k}(x) \cdot V_{m,k}, \end{aligned}$$

where $V_{m,k} = \int_0^{+\infty} \phi_{m,k}(y)\, dy$ are the pay-off coefficients.

For this option we can accurately determine these pay-off coefficients, as stated in the following lemma.

LEMMA 3. *The pay-off coefficients $V_{m,k}$ for a cash-or-nothing call option can be calculated as*

$$V_{m,k} = \frac{1}{2^{m/2}} \left(sgn(k) \cdot Si(|k|) + \frac{1}{2} \right),$$

where

$$sgn(x) := \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{if } x = 0, \\ -1 & \text{if } x < 0 \end{cases}$$

is the sign function and

$$Si(x) := \int_0^x sinc(t)\, dt$$

is the sine integral function.

Proof. If we consider the definition of $\phi_{m,k}$ and perform the change of variables, $t = 2^m y - k$, we obtain

$$V_{m,k} = 2^{m/2} \int_0^{+\infty} \phi(2^m y - k)\, dy = \frac{1}{2^{m/2}} \int_{-k}^{+\infty} \operatorname{sinc}(t)\, dt.$$

We conclude the proof by taking into account that $\int_0^{+\infty} \operatorname{sinc}(t)\, dt = \frac{1}{2}$. ∎

In practice, the sine integral function $\mathrm{Si}(x)$ is to be computed by means of approximation (18), that is,

$$\mathrm{Si}(x) \approx \mathrm{Si}^*(x) := \int_0^x \operatorname{sinc}^*(x) = \frac{2}{\pi} \sum_{j=1}^{2^{J-1}} \frac{1}{2j-1} \sin\left(\frac{2j-1}{2^J} \pi x \right). \tag{27}$$

PROPOSITION 1. *If we define $V^*_{m,k} := \frac{1}{2^{m/2}} \left(sgn(k) \cdot Si^*(|k|) + \frac{1}{2} \right)$ and assume that $J \geq \log_2(\pi|k|)$, then*

$$|V_{m,k} - V^*_{m,k}| \leq \frac{1}{2^{m/2}} \cdot \frac{sgn(k) \cdot \pi^2 k^3}{2^{2(J+1)} - (\pi k)^2},$$

and $\lim_{J \to +\infty} V_{m,k} = V^*_{m,k}$.

Proof. We have that

$$|V_{m,k} - V^*_{m,k}| = \frac{1}{2^{m/2}} \cdot |\mathrm{sgn(k)}||\mathrm{Si}(|k|) - \mathrm{Si}^*(|k|)|.$$

If $k = 0$, then $V_{m,k} = V^*_{m,k}$; otherwise $|\mathrm{sgn(k)}| = 1$, and therefore

$$|V_{m,k} - V^*_{m,k}| = \frac{1}{2^{m/2}} \cdot |\mathrm{Si}(|k|) - \mathrm{Si}^*(|k|)| \leq \int_0^{|k|} |\operatorname{sinc}(t) - \operatorname{sinc}^*(t)|\, dt.$$

If we take into account that $J \geq \log_2(\pi|k|)$ then, by means of Lemma 2,

$$\int_0^{|k|} |\operatorname{sinc}(t) - \operatorname{sinc}^*(t)|\, dt \leq |k| \frac{(\pi \mathrm{sgn}(k) \cdot k)^2}{2^{2(J+1)} - (\pi \mathrm{sgn}(k) \cdot k)^2}, \tag{28}$$

and this completes the proof. ∎

Remark 3. The most involved part in the computation of the pay-off coefficients $V^*_{m,k}$ is the evaluation of function $\mathrm{Si}^*(x)$ at the integer values $|k|$. Since these values depend on neither the parameters of the option pricing problem nor the scale of approximation m, we can calculate (and store) them by means of expression (27) at a high accuracy level given by (28) and use them later. Another possibility will be computing the pay-off coefficients $V^*_{m,k}$ on the fly by using the error estimate in Proposition 1. We can choose an appropriate value of J for each k in order to meet a predefined tolerance error. Alternatively, as we did with the density coefficients, we can also consider a constant J-value, defined by $\bar{\jmath} := \lceil \log_2(\pi \bar{M}) \rceil$, where $\bar{M} := \max(|k_1|, |k_2|)$, and we can apply an FFT algorithm to evaluate the sine integral function $\mathrm{Si}^*(x)$ at integer values $|k|$. Indeed,

$$\mathrm{Si}^*(|k|) = \frac{2}{\pi} \sum_{j=1}^{2^{\bar{\jmath}-1}} \frac{1}{2j-1} \sin\left(\frac{2j-1}{2^{\bar{\jmath}}} \pi |k|\right) = \frac{2}{\pi} \sum_{j=0}^{2^{\bar{\jmath}-1}-1} \frac{1}{2j+1} \sin\left(\frac{j+1/2}{2^{\bar{\jmath}-1}} \pi |k|\right).$$

The computational complexity associated to a direct computation of the pay-off coefficients by formula (27) is $\mathcal{O}\left(2^{\bar{\jmath}-1} \cdot (k_2 - k_1 + 1)\right)$, while the complexity by the application of the FFT algorithm explained above is $\mathcal{O}\left(\log(2) \cdot (\bar{\jmath} - 1) \cdot 2^{\bar{\jmath}-1}\right)$, with k_1 and k_2 fixed indices.

3.2.2. European options. The pay-off functions for European call or put options have been given in (2). We again consider the approximation to the density function f in terms of Shannon scaling functions, as in (26). We truncate the infinite integration range to a finite domain $\mathcal{I}_m = [\frac{k_1}{2^m}, \frac{k_2}{2^m}]$, with specific k_1 and k_2 values, which gives

$$v(x,t) = e^{-r(T-t)} \int_{\mathbb{R}} v(y,T) f(y|x)\, dy \approx v_1(x,t) = e^{-r(T-t)} \int_{\mathcal{I}_m} v(y,T) f(y|x)\, dy.$$

If we replace f by its approximation f_m, we find

(29)
$$\begin{aligned} v_1(x,t) &= e^{-r(T-t)} \int_{\mathcal{I}_m} v(y,T) f(y|x)\, dy \approx v_2(x,t) = e^{-r(T-t)} \int_{\mathcal{I}_m} v(y,T) f_m(y|x)\, dy \\ &= e^{-r(T-t)} \sum_{k=k_1}^{k_2} c_{m,k}(x) \cdot V^{\alpha}_{m,k}, \end{aligned}$$

with the pay-off coefficients

$$V^{\alpha}_{m,k} := \int_{\mathcal{I}_k} v(y,T) \phi_{m,k}(y)\, dy.$$

Proposition 2. *With the same notation as before, let us define* $\bar{k}_1 := \max(k_1, 0)$, $\bar{k}_2 := \min(k_2, 0)$. *The pay-off coefficients for a European call option are computed as*

$$V^1_{m,k} \approx V^{1,*}_{m,k} := \begin{cases} \frac{K 2^{m/2}}{2^{J-1}} \sum_{j=1}^{2^{J-1}} \left[I_1\left(\frac{\bar{k}_1}{2^m}, \frac{k_2}{2^m}\right) - I_2\left(\frac{\bar{k}_1}{2^m}, \frac{k_2}{2^m}\right) \right] & \text{if } k_2 > 0, \\ 0 & \text{if } k_2 \le 0, \end{cases}$$

and, for the put,

$$V^{-1}_{m,k} \approx V^{-1,*}_{m,k} := \begin{cases} \frac{-K 2^{m/2}}{2^{J-1}} \sum_{j=1}^{2^{J-1}} \left[I_1\left(\frac{k_1}{2^m}, \frac{\bar{k}_2}{2^m}\right) - I_2\left(\frac{k_1}{2^m}, \frac{\bar{k}_2}{2^m}\right) \right] & \text{if } k_1 < 0, \\ 0 & \text{if } k_1 \ge 0, \end{cases}$$

where

$$I_1(a,b) = \frac{C_j 2^m}{1+(C_j 2^m)^2}\left[e^b \sin(C_j(2^m b-k)) - e^a \sin(C_j(2^m a-k)) \right.$$
$$\left. + \frac{1}{C_j 2^m}\left(e^b \cos(C_j(2^m b-k)) - e^a \cos(C_j(2^m a-k))\right)\right],$$
$$I_2(a,b) = \frac{1}{C_j 2^m}\left(\sin(C_j(2^m b-k)) - \sin(C_j(2^m a-k))\right),$$

and $C_j = \frac{2j-1}{2^J}\pi$.

Proof. By definition, we have

$$V_{m,k}^{\alpha} = \int_{\mathcal{I}_m} v(y,T)\phi_{m,k}(y)\,dy.$$

We observe that

$$V_{m,k}^{\alpha} = \begin{cases} K2^{m/2}\int_{\mathcal{I}_m\cap[0,+\infty)}(e^y-1)\cdot \operatorname{sinc}(2^m y-k)\,dy & \text{if } \alpha = 1, \\ -K2^{m/2}\int_{(-\infty,0]\cap\mathcal{I}_m}(e^y-1)\cdot \operatorname{sinc}(2^m y-k)\,dy & \text{if } \alpha = -1. \end{cases} \tag{30}$$

If we replace the function sinc in (30) by approximation formula (18) and we define

$$I_1(a,b) = \int_a^b e^y \cos(C_j(2^m y-k))\,dy$$

and

$$I_2(a,b) = \int_a^b \cos(C_j(2^m y-k))\,dy,$$

then the results in Proposition 2 hold. ∎

PROPOSITION 3. *With the same notation as before, if we consider* $J \geq log_2(\pi N_k)$, *we find for the European option,*

$$|V_{m,k}^{\alpha} - V_{m,k}^{\alpha,*}| \leq \frac{K}{2^{m/2}}\cdot D \cdot \frac{(\pi N_k)^2}{2^{2(J+1)} - (\pi N_k)^2},$$

where $N_k := \max(|\bar{k}_1 - k|, |k_2 - k|)$, $D := |k_2 - \bar{k}_1| \cdot \max_{y\in\left[\frac{\bar{k}_1}{2^m}, \frac{k_2}{2^m}\right]} |e^y - 1|$ *if* $\alpha = 1$, *and* $N_k := \max(|k_1 - k|, |\bar{k}_2 - k|)$, $D := |\bar{k}_2 - k_1| \cdot \max_{y\in\left[\frac{k_1}{2^m}, \frac{\bar{k}_2}{2^m}\right]} |e^y - 1|$ *if* $\alpha = -1$.

Further, $\lim_{J\to+\infty} V_{m,k}^{\alpha,*} = V_{m,k}^{\alpha}$.

Proof. We give the proof for the case $\alpha = 1$ but the case $\alpha = -1$ goes analogously. The error in the computation of the pay-off coefficients comes from replacing the sinc function in (30) by approximation (18). Therefore,

$$|V_{m,k}^{\alpha} - V_{m,k}^{\alpha,*}| \leq K2^{m/2}\int_{\frac{\bar{k}_1}{2^m}}^{\frac{k_2}{2^m}} |e^y - 1||\operatorname{sinc}(2^m y-k) - \operatorname{sinc}^*(2^m y-k)|\,dy.$$

If we make the change of variables, $t = 2^m y - k$, then

$$|V_{m,k}^{\alpha} - V_{m,k}^{\alpha,*}| \leq \frac{K}{2^{m/2}} \max_{y\in\left[\frac{\bar{k}_1}{2^m}, \frac{k_2}{2^m}\right]} |e^y - 1| \int_{\bar{k}_1-k}^{k_2-k} |\operatorname{sinc}(t) - \operatorname{sinc}^*(t)|\,dt.$$

If we assume that $J \geq \log_2(\pi N_k)$, we can apply Lemma 2 to conclude the proof. ∎

Remark 4. Like in the case of cash-or-nothing options, we can choose an appropriate value of J for each k in order to meet a predefined tolerance error, or, alternatively, we can consider a constant J, i.e., $\bar{j} := \lceil \log_2(\pi N) \rceil$, where $N := \max_{k_1<k<k_2} N_k$ and we can apply an FFT algorithm. This is the option chosen in the numerical examples section and we provide details on the application of the algorithm in Appendix B. It is worth mentioning that $V_{m,k}^{\alpha,*}$ can be precomputed for later use with high accuracy by means of the estimates in Proposition 3, saving some CPU time. However, the overall complexity for pricing an option is dominated by the complexity of the computation of the density coefficients.

3.2.3. European option pricing for several strikes. For Lévy jump models and for Heston stochastic volatility models, options for many strike prices can be computed simultaneously in a highly efficient way. We distinguish vectors with boldfaced letters here.

For Lévy processes, whose characteristic functions can be represented by

$$\hat{f}(w;x) = \hat{f}_{\text{Levy}}(w) \cdot e^{-iwx}, \tag{31}$$

the SWIFT option pricing formula simplifies to

$$\begin{aligned} v(\mathbf{x},t) =& e^{-r(T-t)} \frac{2^m \mathbf{K}}{2^{J+\bar{J}-2}} \\ &\times \sum_{k=k_1}^{k_2} \Re \left[e^{\frac{ik\pi}{2^J}} \sum_{j=0}^{2^J-1} \hat{f}_{\text{Levy}} \left(\frac{(2j+1)\pi 2^m}{2^J} \right) e^{-\frac{i(2j+1)\pi 2^m)}{2^J} \cdot \mathbf{x}} e^{\frac{2\pi i k j}{2^J}} \right] \bar{V}_{m,k}^{\alpha,*}, \end{aligned} \tag{32}$$

where

$$\bar{V}_{m,k}^{1,*} := \begin{cases} \sum_{j=1}^{2^{J-1}} \left[I_1\left(\frac{\bar{k}_1}{2^m}, \frac{k_2}{2^m}\right) - I_2\left(\frac{\bar{k}_1}{2^m}, \frac{k_2}{2^m}\right) \right] & \text{if } k_2 > 0, \\ 0 & \text{if } k_2 \leq 0, \end{cases}$$

and

$$\bar{V}_{m,k}^{-1,*} := \begin{cases} -\sum_{j=1}^{2^{J-1}} \left[I_1\left(\frac{k_1}{2^m}, \frac{\bar{k}_2}{2^m}\right) - I_2\left(\frac{k_1}{2^m}, \frac{\bar{k}_2}{2^m}\right) \right] & \text{if } k_1 < 0, \\ 0 & \text{if } k_1 \geq 0. \end{cases}$$

The values $\bar{V}_{m,k}^{\alpha,*}$ are computed only once for $k = k_1, \ldots, k_2$, since they do not depend on the strike values. Details of the relevant characteristic functions are given in Appendix A.

Remark 5. Using pricing formula (32) involves, on the one hand, the application of an FFT algorithm n_K times to compute the pay-off coefficients, where n_K is the number of options in the pricing problem, and, on the other hand, the values $\bar{V}_{m,k}^{\alpha,*}$ that are to be computed only once by the application of the FFT algorithm, as detailed in Remark 4. By doing this, the CPU time employed in the valuation of several options with different strikes is smaller than the direct pricing of each individual option. However, the overall CPU time could be further reduced by reformulating the approximation problem as follows. Observe that the density function f in (31) and

(36) satisfies

$$f(y) = \frac{1}{2\pi}\int_{\mathbb{R}} e^{iyw}\hat{f}(w)\,dw = \frac{1}{2\pi}\int_{\mathbb{R}} e^{iyw}\hat{f}_{\text{Levy}}(w)e^{-iwx}\,dw$$

$$= \frac{1}{2\pi}\int_{\mathbb{R}} e^{i(y-x)w}\hat{f}_{\text{Levy}}(w)\,dw = f_{\text{Levy}}(y-x). \tag{33}$$

This fact allows us to compute the density coefficients associated to f_{Levy} only once (as f_{Levy} does not depend on x). Then, our pricing formula (29) becomes

$$v_1(x,t) = e^{-r(T-t)}\sum_{k=k_1}^{k_2} c_{m,k}\cdot V^{\alpha}_{m,k}(x),$$

where

$$V^{\alpha}_{m,k}(x) := \int_{\mathcal{I}_k} v(y,T)\phi_{m,k}(y-x)\,dy,$$

and $c_{m,k}$ are now the coefficients associated to the density f_{Levy}. Note that we have translated the dependence on x from the density coefficients to the pay-off coefficients. This means that within this new framework, the pay-off coefficients must be computed for each different strike value K. In terms of CPU time, we need only one application of an FFT algorithm for density coefficients and n_K applications of an FFT algorithm for pay-off coefficients (as shown before, the computational complexity for pay-off coefficients is smaller than the computational complexity for density coefficients).

4. Numerical examples. In this section we give several examples of pricing options by the SWIFT method presented. With these examples we can observe high accuracy and robustness of the method while keeping very small CPU times.[3] We present the valuation of a cash-or-nothing option and a European call option under geometric Brownian motion (GBM) dynamics and compare the results with the well-known Black–Scholes formulae. We also consider CGMY and Heston dynamics in our examples.

It is worth mentioning that the SWIFT method does not need to rely on an a priori truncation of the entire real line onto a finite interval. The coefficients can be computed, departing from $k = 0$, by increasing or decreasing the value of this translation parameter. Thanks to the property in expression (12), the value of the density can be computed as a byproduct. Furthermore, the area underneath the curve is to be computed with expression (14) on the fly as well. These two features allow us to control the density computation regardless of features like skewness or heavy tails. This strategy, however, comes at the cost of extra CPU time, since it does not allow the use of an FFT algorithm. To enhance the speed of the method we will depart from an initial interval and we show that the SWIFT method is not sensitive to this choice.

Here we compute the density coefficients following expression (25). We truncate the real line either by choosing an arbitrary interval or using the cumulants[4] (whenever their computation is possible) and adjust (if necessary) the approximation by means

[3]The programs were coded in C language and run on a Dell Vostro 320 with Intel Core 2 Duo E7500 2.93-GHz processor and 4 GB RAM.

[4]The cumulants are the power series coefficients of the cumulant generating function $c(s) = \log \mathbb{E}\left(e^{sX}\right)$.

of the two properties of the density explained before (value of the density and area underneath the curve). This strategy allows us to save a considerable amount of CPU time by computing a unique j-value for all coefficients and by applying an FFT algorithm, as explained in Remark 1. Proceeding this way, the characteristic function is evaluated only once at a fixed grid of points. Following a similar argument, the payoff coefficients are calculated with a unique J-value, called $\bar{j}$, and accompanied with an FFT algorithm, as detailed in Remark 3 for cash-or-nothing options and Remark 4 for European options.

For the asset price processes we are concerned with in this paper, it is relatively easy to compute the cumulants and they give us reliable intervals for the approximation. For this reason, we determine an a priori interval of integration $[a, b]$, based on the approximation

$$[a,b] := \left[x_0 + c_1 - L\sqrt{c_2 + \sqrt{c_4}}, x_0 + c_1 + L\sqrt{c_2 + \sqrt{c_4}}\right] \quad \text{with } L = 10, \tag{34}$$

as in [Fan08], where c_n denotes the nth cumulant of $\ln(S_T/K)$, and $x_t = \ln(S_t/K)$, like in (1). This has been confirmed as being accurate for a variety of asset processes and options with different maturities. Moreover, the area underneath the curve for the resulting interval is very close to one, and therefore it is not necessary to further calculate extra coefficients.

4.1. Scale of approximation. The remaining important parameter to fix is the scale of approximation m. Although we do not provide an exact prescription of how to select this parameter, we know (due to the estimation of the size of the detail coefficients in (11)) that the error of the approximation to the density function at scale m decreases when the characteristic function decays rapidly. In these cases, we obtain very accurate values at the starting scales (like $m = 0, m = 1$) and therefore we do not need to decide anything about the scale. Besides, in most of the cases of interest, the characteristic function is available in analytic form, and we can study its decay properties. To illustrate this fact we consider the pricing of a cash-or-nothing option under GBM dynamics with parameters $S_0 = K = 100, r = 0.1, T = 1, \sigma = 0.25$ (with reference value 0.5504504967481910 computed by means of the Black–Scholes formula), and CGMY dynamics with parameters $S_0 = K = 100, r = 0.1, T = 1, C = 1, G = 5, M = 5, Y = 1.5$ (with reference value 0.262562626927812 computed by means of the COS method with a very large number of cosine terms). The absolute errors at scales $m = 0$ and $m = 1$ corresponding to the underlying process CGMY are respectively $1.2 \cdot 10^{-5}$ and $4.7 \cdot 10^{-15}$, whereas in the case of GBM similar errors are obtained at scale $m = 2$ (absolute error $2.5 \cdot 10^{-4}$) or $m = 4$ (absolute error $2.2 \cdot 10^{-16}$).

Now we consider $Y = 0.1$ and we keep the rest of the CGMY parameters the same. Then, the absolute error at scale $m = 4$ is $3.6 \cdot 10^{-5}$ (the reference value is in this case 0.543271332426876). The densities are plotted in Figure 3 both in the time domain (left) as well as in the frequency domain (right), where they have been recovered at the lower reported scales. As we observe, we reach engineering accuracy at scale 0 for rapidly decaying densities in the Fourier domain.

4.2. Exponential convergence of SWIFT method. As a second example, we compute the price of a cash-or-nothing call option under GBM dynamics, with parameters

$$S_0 = 100, r = 0.1, T = 0.1, \text{ and volatility } \sigma = 0.25. \tag{35}$$

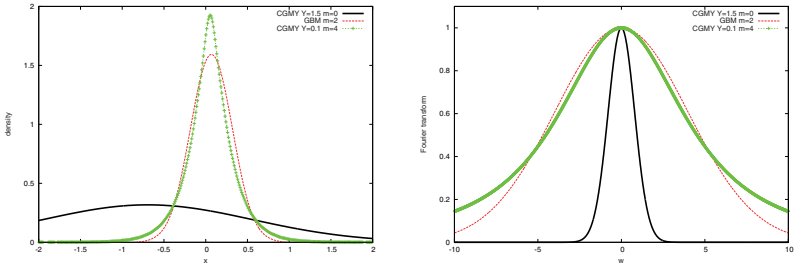

FIG. 3. *Density function (left) and modulus of the characteristic function (right) corresponding to CGMY and GBM dynamics.*

The results are presented in Table 1. We observe exponential convergence in terms of the scale of approximation used (see Figure 4). We get engineering accuracy with 25 terms and machine accuracy with 50 terms. Regarding the CPU time, the speed of the SWIFT method is impressive. We reach machine accuracy within one-tenth of a millisecond. The CPU time can be further reduced if we compute the values for the integral of the cardinal sinus in advance and store them for later use (as in Remark 3).

TABLE 1

Maximum absolute errors corresponding to the valuation of a cash-or-nothing call under the GBM dynamics by means of the SWIFT method with parameters (35) *and different strikes* $K = 80, 100, 120$. *The reference values have been computed using the Black-Scholes formulae:* 0.9882579795645030 ($K = 80$), 0.5293295436540910 ($K = 100$), *and* 0.0131034102155745 ($K = 120$).

m	K	k_1	k_2	$\jmath$	$\bar{\jmath}$	Error	CPU time (milliseconds)
1	80	-1	2	4	3	$1.93e-01$	0.02
2	120	-3	2	5	4	$4.42e-02$	0.02
3	120	-7	4	6	5	$1.06e-02$	0.04
4	80	-8	16	7	6	$6.36e-06$	0.07
5	80	-17	32	8	7	$3.33e-16$	0.14

4.3. Size of integration interval. Figure 5 shows the errors corresponding to the pricing of a cash-or-nothing call and a European call under GBM dynamics by means of the SWIFT and COS methods. We select $N = 40$ for the COS method in order to consider the same number of terms as the SWIFT method at scale $m = 3$. The horizontal axis represents parameter L in expression (34). In the case of the COS method, the accuracy deteriorates if we consider large values of L (and therefore wider intervals) and keep the number of cosine terms, N, fixed. On the contrary, a high accuracy by the SWIFT method remains when widening the interval. The difference between the two methods comes from the fact that SWIFT automatically calculates the number of coefficients that should be used to recover the density, while there is no prescription for how to increase N to preserve accuracy for the COS method. Table 2 presents the number of coefficients used by the SWIFT method with varying L-values.

The ratio between the CPU time employed by the SWIFT ($m = 3$) and COS ($N = 40$) methods to value a cash-or-nothing option with parameters $S_0 = K = 100, r = 0.1, T = 1, \sigma = 0.25$ under the GBM dynamics is $0.07/0.02 = 3.5$. The number of cosine terms for the COS method has been selected to meet the same

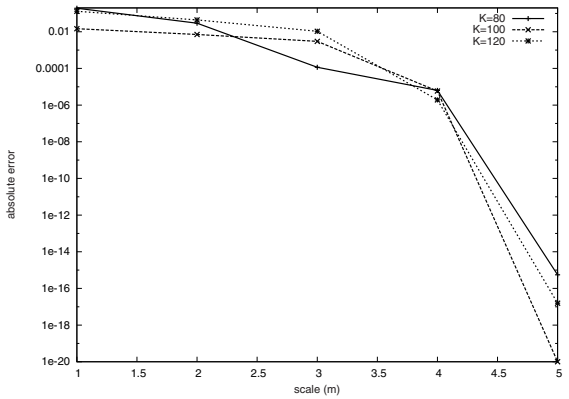

FIG. 4. *Absolute errors corresponding to the valuation of a cash-or-nothing call under the GBM dynamics by means of the SWIFT method at different scales of approximation m with parameters* (35) *and strikes* $K = 80, 100, 120$. *The reference values have been computed using the Black–Scholes formulae:* 0.9882579795645030 ($K = 80$), 0.5293295436540910 ($K = 100$), *and* 0.0131034102155745 ($K = 120$).

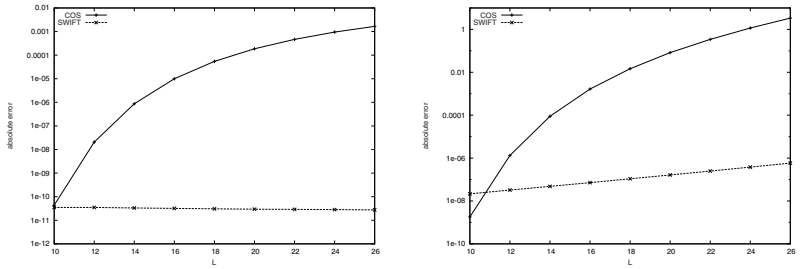

FIG. 5. *Absolute errors corresponding to the valuation of a cash-or-nothing call (left) and a European call (right) under the GBM dynamics by means of the SWIFT* ($m = 3$) *and COS* ($N = 40$) *methods with parameters* $S_0 = K = 100, r = 0.1, T = 1, \sigma = 0.25$. *The reference values have been computed using the Black–Scholes formulae:* 0.5504504967481910 *(cash-or-nothing) and* 14.9757907783113000 *(call).*

accuracy as for the SWIFT method at scale $m = 3$, which is order 10^{-11} as in Figure 5. The value of parameter L used in this case is the default value $L = 10$. If we measure the CPU time required in this case to compute only the density coefficients (and the pay-off coefficients have been precomputed as explained in Remark 3), the ratio decreases to $0.06/0.02 = 3$. So the most involved part is the computation of the density coefficients. To confirm this, we consider the pricing of a call option under CGMY with parameters $S_0 = K = 100, r = 0.1, T = 1, C = 1, G = 5, M = 5, Y = 0.1$ (with reference value 15.86966263787780 computed by means of the COS method). In this case, the rate of decay of the characteristic function is low and we perform the approximation at scale $m = 6$, getting an absolute error of $1.6 \cdot 10^{-4}$ (with $\jmath = 11, k_1 = -304, k_2 = 311$). The CPU time needed to calculate the density plus the pay-off coefficients is 1.32 milliseconds, while the CPU time to calculate the density coefficients is 0.94 milliseconds.

We repeated the same experiment with parameter L for the CGMY process, which exhibits heavier tails than GBM. Again, we choose N to be the same number of terms

TABLE 2
Number of terms used with the SWIFT method with respect to the size of the interval (34) *determined by* L.

L	10	12	14	16	18	20	22	24	26
k_1	-19	-23	-27	-31	-35	-39	-43	-47	-51
k_2	20	24	28	32	36	40	44	48	52

for the SWIFT and COS methods for $L = 10$. A behavior very similar to the former example was observed. It is worth underlining that the SWIFT method is capable of achieving engineering accuracy at scale $m = 0$, which basically means that the wavelet basis has not been compressed to carry out the approximation, highlighting the robustness of the method.

4.4. Long maturity options and fat-tailed densities. Our next example is devoted to exploring the performance of SWIFT for long maturity options. We picked maturities $T = 50$ and $T = 100$, as we sometimes encounter in the insurance and pension industry. We compare our results with the COS method and show our findings in Table 3. Again the SWIFT method shows robustness, being capable of highly accurately pricing the European call option at very low scales of approximation. The COS method shows a large absolute error for $T = 100$ and $N = 35$, although the accuracy increases with $N = 70$. The SWIFT method at scale $m = 0$ shows improved accuracy for long maturities. The pay-off function grows exponentially fast and round-off errors may appear and hamper the approximation. When we use Shannon wavelets, this growth is compensated by very small density coefficients, whereas this is not the case with the cosines expansion. Shannon wavelets perform a local approximation, while cosines present a global approximation.

TABLE 3
Absolute errors corresponding to the valuation of a call option under the GBM dynamics by means of the SWIFT and COS methods with parameters $S_0 = 100, K = 120, r = 0.1, \sigma = 0.25$. *The reference values have been computed using the Black–Scholes formulae:* 99.2025928525532000 *(*$T = 50$*) and* 99.9945609694213000 *(*$T = 100$*).*

Method	Error ($T = 50$)	Error ($T = 100$)
SWIFT ($m = 0$)	$1.91e-01$	$2.50e-05$
COS ($N = 35$)	$4.98e-01$	$2.05e+02$
SWIFT ($m = 1$)	$7.78e-09$	$3.20e-06$
COS ($N = 70$)	$2.79e-08$	$2.02e-05$

Now we focus on a fat-tailed density arising from the CGMY dynamics with parameters as in [Ort13]. The recovered density by means of the SWIFT method at scale $m = 0$ is plotted in Figure 6. We consider several intervals for the approximation and compute the value of the densities at the boundaries of the intervals and the absolute error when calculating the area underneath the density by the SWIFT method at scale $m = 0$. The results are summarized in Table 4. The last row of the table presents the interval given by the cumulants. This is an example of how the interval can be calculated in absence of cumulants. As explained before, the way to proceed is by a combination of the value of the density at the boundaries and the area underneath the curve. While the density evaluated at the right boundary of the intervals forms a decreasing sequence, the evaluation at the left boundary is not

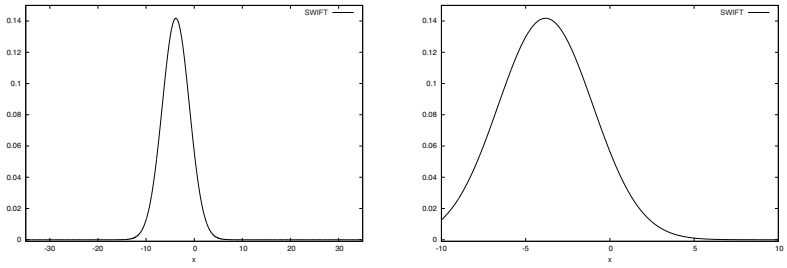

FIG. 6. *Recovered density (left) and zoom of the density (right) by the SWIFT method at scale* $m = 0$ *corresponding to the CGMY dynamics with parameters* $S_0 = 100, K = 110, r = 0.1, q = 0.05, T = 5, C = 1, G = 5, M = 5, Y = 1.5$ *as in [Ort13].*

monotonic. For example, $f(-5) > f(-2) > f(-1)$ but $f(-10) < f(-5)$. Thanks to the computed area we see that substantial mass of the density is still to be recovered.

Another interesting application consists in determining the length of the interval to meet a certain criteria regarding the area underneath the density. We may want to keep a shorter interval (than the one suggested by the cumulants) to use fewer terms in the approximation.

TABLE 4

Absolute error of the computed area under the recovered density by the SWIFT method at scale $m = 0$ *corresponding to the CGMY dynamics with parameters* $S_0 = 100, K = 110, r = 0.1, q = 0.05, T = 5, C = 1, G = 5, M = 5, Y = 1.5$. *The interval in the last row of the table has been computed with the help of cumulants:* $[-32.83, 25.19]$.

$[a, b]$	$\lvert f\left(\frac{k_1}{2^m}\right)\rvert$	$\lvert f\left(\frac{k_2}{2^m}\right)\rvert$	Error (area)
$[-1, 1]$	$5.97e-02$	$4.20e-02$	$9.82e-01$
$[-2, 2]$	$1.15e-01$	$1.65e-02$	$7.59e-01$
$[-5, 5]$	$1.30e-01$	$1.06e-03$	$3.40e-01$
$[-10, 10]$	$1.27e-02$	$8.92e-07$	$1.49e-02$
$[-20, 20]$	$1.10e-08$	$1.82e-15$	$7.05e-09$
$[-32.83, 25.19]$	$1.02e-15$	$1.32e-15$	$6.00e-15$

4.5. Multiple strikes valuation. In our last example we consider the simultaneous pricing of several call options with different strikes under the Heston stochastic volatility dynamics. The results are presented in Table 5. The CPU time for pricing 21 strikes at scale 6 is only 6.6 times the CPU time for pricing one option. For practical implementation, we consider as the left boundary of the approximation interval the minimum between the left boundaries of the intervals corresponding to each separate option as in expression (34). Similarly, we set up the right-side boundary of the approximation interval. By doing this, we ensure that the resulting interval is appropriate for pricing all options in a portfolio simultaneously, but the integration interval is thus wider than the intervals selected for pricing one option. Although the interval has been widened, the accuracy remains without changing the scale of approximation, as we observed in the previous examples. In the case of the COS method, the accuracy may be hampered when the number of terms in the cosines expansion remains constant.

TABLE 5

Simultaneous valuation of 21 *European call options (top) with strikes ranging from* 50 *to* 150 *and valuation of only one European call option (bottom) under the Heston dynamics with parameters* $S_0 = 100, \mu = 0, \lambda = 1.5768, \eta = 0.5751, \bar{u} = 0.0398, u_0 = 0.0175, \rho = -0.5711$. *The reference values have been computed using the COS method with* 50,000 *terms and* $L = 10$.

21 strikes	Scale m	4	5	6
	CPU time (milliseconds)	3.04	3.84	6.62
	Max. absolute error	$2.04e-02$	$5.63e-05$	$3.63e-06$
1 strike ($K = 100$)	Scale m	4	5	6
	CPU time (milliseconds)	0.32	0.52	1.00
	Absolute error	$4.78e-03$	$1.61e-05$	$6.56e-07$

5. Conclusions. In this paper we advocated the use of Shannon wavelets within the European option pricing framework. Specifically, we have focused on the discounted expected pay-off pricing formula, where the price of the option is computed by integrating the pay-off function multiplied by the density function. We have presented a novel method based on Shannon wavelets to recover the density function from its characteristic function and we have called it SWIFT. In detail, we use the Shannon father wavelet, which is the sinus cardinal function, and carry out the approximation in a multiresolution analysis environment where the desired scale of approximation is fixed. The density coefficients as well as the pay-off coefficients are computed using Vieta's formula. An estimate of the error for Vieta's formula is also presented and used subsequently throughout the paper. The presented numerical method shows a high accuracy and robustness, as well as a fast convergence to the solution at low scales with only a few terms in the expansion. The SWIFT method is efficient for the valuation of long- and short-maturity contracts and it does not rely on an a priori truncation of the entire real line, since it can do it adaptively in order to meet a predefined tolerance error regarding the mass underneath the density function. The accuracy of the method is not affected by the width of the interval for the approximation and it automatically computes the number of coefficients needed for the new interval.

As part of our future research, the SWIFT method will be applied to early-exercise options as well as in the context of risk management to compute the risk measures.

Appendix A. Characteristic functions. Here, we provide the definition of the characteristic function which is consistent with the definition of the Fourier transform in (7). We focus on the details of the characteristic functions and refer the reader to the literature [Car02, Con04, Gat06] for further information on these processes. Two of the models treated in this paper are the well-known GBM and CGMY processes [Car02], which are particular cases of Lévy models. The GBM process has the following characteristic function:

$$\hat{f}_{\mathrm{GBM}}(w) = \exp\left(-iw\left(r - q - \frac{1}{2}\sigma^2\right)(T-t) - \frac{1}{2}\sigma^2 w^2 (T-t)\right),$$

where r is the interest rate, σ volatility, and T maturity time. Parameter q represents a dividend yield, which is often set to zero in our experiments (the parameter is not given in the experiments unless it is chosen differently from zero). One problem with the GBM model is that it is not able to reproduce the volatility skew or smile present in most financial markets. Over the past few years it has been shown that several exponential Lévy models are, at least to some extent, able to reproduce the skew or

the smile. One particular model is the CGMY model [Car02]. The characteristic function of the log-asset price in the case of the CGMY process reads

$$\begin{aligned}\hat{f}_{\text{CGMY}}(w) =& \exp\left(-iw(r-q+s)(T-t)\right)\\ &\cdot\exp\left(C\Gamma(-Y)\left((M+iw)^Y - M^Y + (G-iw)^Y - G^Y\right)(T-t)\right)\end{aligned}$$

with C, G, M, and Y parameters governing the density and $\Gamma(\cdot)$ represents the Gamma function, and

$$s = -C\Gamma(-Y)\left((M-1)^Y - M^Y + (G+1)^Y - G^Y\right).$$

Under the Heston stochastic volatility model, the volatility, denoted by $\sqrt{u_t}$, is modeled by a stochastic differential equation,

$$\begin{cases} dx_t &= (\mu - \frac{1}{2}u_t)x_t dt + \sqrt{u_t}x_t dW_{1t},\\ du_t &= \lambda(\bar{u}-u_t)dt + \eta\sqrt{u_t}dW_{2t},\end{cases}$$

where x_t denotes the log-asset price and u_t the variance of the asset price process. Parameters $\lambda \geq 0, \bar{u} \geq 0$, and $\eta \geq 0$ are called the speed of mean reversion, the mean level of variance, and the volatility of volatility, respectively. Furthermore, the Brownian motions W_{1t} and W_{2t} are assumed to be correlated with correlation coefficient ρ.

For the Heston model, the pricing equation is also simplified, since

$$\hat{f}(w;x,u_0) = \hat{f}_{\text{Heston}}(w;u_0)\cdot e^{-iwx} \tag{36}$$

with u_0 the volatility of the underlying at initial time. We then find

(37)

$$\begin{aligned}v(\mathbf{x},t,u_0) =& e^{-r(T-t)}\frac{2^m\mathbf{K}}{2^{\jmath+\bar{\jmath}-2}}\\ &\times\sum_{k=k_1}^{k_2}\Re\left[e^{\frac{ik\pi}{2^\jmath}}\sum_{j=0}^{2^\jmath-1}\hat{f}_{\text{Heston}}\left(\frac{(2j+1)\pi 2^m}{2^\jmath};u_0\right)e^{-\frac{i(2j+1)\pi 2^m)}{2^\jmath}}\cdot\mathbf{x}e^{\frac{2\pi ikj}{2^\jmath}}\right]\bar{V}^{\alpha,*}_{m,k},\end{aligned}$$

where the characteristic function of the log-asset price reads

$$\begin{aligned}\hat{f}_{\text{Heston}}(w;u_0) =& \exp\left(-iw\mu(T-t) + \frac{u_0}{\eta^2}\left(\frac{1-e^{-D(T-t)}}{1-Ge^{-D(T-t)}}\right)(\lambda+i\rho\eta w - D)\right)\\ &\cdot\exp\left(\frac{\lambda\bar{u}}{\eta^2}\left((\lambda+i\rho\eta w-D)(T-t) - 2\ln\left(\frac{1-Ge^{-D(T-t)}}{1-G}\right)\right)\right)\end{aligned}$$

with

$$D = \sqrt{(\lambda+i\rho\eta w)^2 + (w^2 - iw)\eta^2} \quad\text{and}\quad G = \frac{\lambda+i\rho\eta w - D}{\lambda+i\rho\eta w + D}.$$

Appendix B. Pay-off formulae for European options. Here we provide details on how the formulae in Proposition 2 can be arranged to apply an FFT algorithm to speed up the computation of the pay-off coefficients $V^{\alpha,*}_{m,k}$. Basically, we have to efficiently calculate the sum,

$$\sum_{j=1}^{2^{\bar{\jmath}-1}}\left[I_1(a,b) - I_2(a,b)\right],$$

where

$$I_1(a,b) = \frac{C_j 2^m}{1+(C_j 2^m)^2}\Bigg[e^b \sin(C_j(2^m b-k)) - e^a \sin(C_j(2^m a-k)) \tag{38}$$

$$+\frac{1}{C_j 2^m}\left(e^b \cos(C_j(2^m b-k)) - e^a \cos(C_j(2^m a-k))\right)\Bigg], \tag{39}$$

$$I_2(a,b) = \frac{1}{C_j 2^m}\left(\sin(C_j(2^m b-k)) - \sin(C_j(2^m a-k))\right), \tag{40}$$

and $C_j = \frac{2j-1}{2^j}\pi$. If we define $A_j = \frac{C_j 2^m}{1+(C_j 2^m)^2}$, $B_j = \frac{1}{C_j 2^m}$, and we expand the sine and cosine terms in expressions (38) and (40), i.e.,

$$\begin{aligned}
I_1(a,b) =& A_j\Bigg[e^b \sin(C_j 2^m b)\cos(C_j k) - e^b \cos(C_j 2^m b)\sin(C_j k)\\
&- e^a \sin(C_j 2^m a)\cos(C_j k) + e^a \cos(C_j 2^m a)\sin(C_j k)\\
&+ B_j\Bigg(e^b \cos(C_j 2^m b)\cos(C_j k) + e^b \sin(C_j 2^m b)\sin(C_j k)\\
&- e^a \cos(C_j 2^m a)\cos(C_j k) - e^a \sin(C_j 2^m a)\sin(C_j k)\Bigg)\Bigg]\\
=& A_j\Bigg[\Bigg(e^b \sin(C_j 2^m b) - e^a \sin(C_j 2^m a) + B_j e^b \cos(C_j 2^m b)\\
&- B_j e^a \cos(C_j 2^m a)\Bigg)\cos(C_j k)\\
&+\Bigg(-e^b \cos(C_j 2^m b) + e^a \cos(C_j 2^m a) + B_j e^b \sin(C_j 2^m b)\\
&- B_j e^a \sin(C_j 2^m a)\Bigg)\sin(C_j k)\Bigg],
\end{aligned}$$

$$\begin{aligned}
I_2(a,b) =& B_j\Bigg[\Bigg(\sin(C_j 2^m b) - \sin(C_j 2^m a)\Bigg)\cos(C_j k)\\
&+\Bigg(\cos(C_j 2^m a) - \cos(C_j 2^m b)\Bigg)\sin(C_j k)\Bigg].
\end{aligned}$$

If we define

$$\begin{aligned}
I_{11}^j &= e^b \sin(C_j 2^m b) - e^a \sin(C_j 2^m a) + B_j e^b \cos(C_j 2^m b) - B_j e^a \cos(C_j 2^m a),\\
I_{12}^j &= -e^b \cos(C_j 2^m b) + e^a \cos(C_j 2^m a) + B_j e^b \sin(C_j 2^m b) - B_j e^a \sin(C_j 2^m a),\\
I_{21}^j &= \sin(C_j 2^m b) - \sin(C_j 2^m a),\\
I_{22}^j &= \cos(C_j 2^m a) - \cos(C_j 2^m b),
\end{aligned}$$

then

$$
\begin{aligned}
&\sum_{j=1}^{2^{\bar{\jmath}-1}} \left[I_1(a,b) - I_2(a,b)\right] \\
&= \sum_{j=1}^{2^{\bar{\jmath}-1}} \left[A_j \left(I_{1,1}^j \cos(C_j k) + I_{1,2}^j \sin(C_j k)\right) - B_j \left(I_{2,1}^j \cos(C_j k) + I_{2,2}^j \sin(C_j k)\right)\right] \\
&= \sum_{j=1}^{2^{\bar{\jmath}-1}} \left[\left(A_j I_{1,1}^j - B_j I_{2,1}^j\right) \cos(C_j k) + \left(A_j I_{1,2}^j - B_j I_{2,2}^j\right) \sin(C_j k)\right] \\
&= \sum_{j=0}^{2^{\bar{\jmath}-1}-1} \left[\left(A_{j+1} I_{1,1}^{j+1} - B_{j+1} I_{2,1}^{j+1}\right) \cos\left(\frac{j+1/2}{2^{\bar{\jmath}-1}} \pi k\right)\right. \\
&\qquad \left. + \left(A_{j+1} I_{1,2}^{j+1} - B_{j+1} I_{2,2}^{j+1}\right) \sin\left(\frac{j+1/2}{2^{\bar{\jmath}-1}} \pi k\right)\right].
\end{aligned}
$$

The last equality allows us to apply an FFT algorithm straightforwardly. The computational complexity associated to a direct computation of the pay-off coefficients by Proposition 2 is of order $\mathcal{O}\left(2^{\bar{\jmath}-1} \cdot (k_2 - k_1 + 1)\right)$, while the complexity during the application of the FFT algorithm explained above is $\mathcal{O}\left(\log(2) \cdot (\bar{\jmath} - 1) \cdot 2^{\bar{\jmath}}\right)$, with k_1 and k_2 fixed indices. In this case, we have to apply an FFT algorithm two times, once for the terms with cosines and once for the terms with sines.

REFERENCES

[Car02] P.P. Carr, H. Geman, D.B. Madan, and M. Yor, *The fine structure of asset returns: An empirical investigation*, J. Business, 75 (2002), pp. 305–333.

[Car99] P.P. Carr and D.B. Madan, *Option valuation using the fast Fourier transform*, J. Comput. Finance, 2 (1999), pp. 61–73.

[Cat08] C. Cattani, *Shannon wavelets theory*, Math. Probl. Eng., 2008 (2008), 164808.

[Con04] R. Cont and P. Tankov, *Financial Modelling with Jump Processes*, Chapman and Hall, London, 2004.

[Dau92] I. Daubechies, *Ten Lectures on Wavelets*, CBMS-NSF Reg. Conf. Ser. Appl. Math., SIAM, Philadelphia, 1992.

[Fan08] F. Fang and C.W. Oosterlee, *A novel pricing method for European options based on Fourier-cosine series expansions*, SIAM J. Sci. Comput., 31 (2008), pp. 826–848.

[Gat06] J. Gatheral, *The Volatility Surface: A Practitioner's Guide,* Wiley Finance, New York, 2006.

[Gea90] W.B. Gearhart and H.S. Shultz, *The function* $\sin(x)/x$, College Math. J., 21 (1990), pp. 90–99.

[Kir14] J.L. Kirkby, *Efficient Option Pricing by Frame Duality with the fast Fourier Transform, working paper*, 2014.

[Lee04] R.W. Lee, *Option pricing by transform methods: Extensions, unification, and error control*, J. Comput. Finance, 7 (2004), pp 51–86.

[Lin08] E. Lindström, J. Ströjby, M. Brodén, M. Wiktorsson, and J. Holst, *Sequential calibration of options*, Computat. Statist. Data Anal., 52 (2008), pp. 2877–2891.

[Ort13] L. Ortiz-Gracia and C.W. Oosterlee, *Robust pricing of European options with wavelets and the characteristic function*, SIAM J. Sci. Comput., 35 (2013), pp. B1055–B1084.

[Ort14] L. Ortiz-Gracia and C.W. Oosterlee, *Efficient VaR and expected shortfall computations for nonlinear portfolios within the delta-gamma approach*, Appl. Math. Comput., 244 (2014), pp. 16–31.

[Qui13] B.M. Quine and S.M. Abrarov, *Application of the spectrally integrated Voigt function to line-by-line radiative transfer modelling*, J. Quantitative Spectroscopy Radiative Transfer, 127 (2013), pp. 37–48.

[Rui12] M. Ruijter and C.W. Oosterlee, *Two-dimensional Fourier cosines series expansion method for pricing financial options*, SIAM J. Sci. Comput., 34 (2012), pp. 642–671.
[Ush99] N.G. Ushakov, *Selected Topics in Characteristic Functions*, De Gruyter, Berlin, 1999.