Numer. Math. (2017) 136:1035–1070
DOI 10.1007/s00211-016-0858-2

Pricing early-exercise and discrete barrier options by Shannon wavelet expansions

S. C. Maree[1,2] · L. Ortiz-Gracia[3] · C. W. Oosterlee[2,4]

Received: 29 November 2015 / Revised: 7 December 2016 / Published online: 2 January 2017

Abstract We present a pricing method based on Shannon wavelet expansions for early-exercise and discretely-monitored barrier options under exponential Lévy asset dynamics. Shannon wavelets are smooth, and thus approximate the densities that occur in finance well, resulting in exponential convergence. Application of the Fast Fourier Transform yields an efficient implementation and since wavelets give local approximations, the domain boundary errors can be naturally resolved, which is the main improvement over existing methods.

Mathematics Subject Classification 65D30 · 91B24 · 65T60

1 Introduction

Early-exercise options and discrete barrier options are important options for which no analytic valuation formulas exist. Robust and efficient pricing of these options outside the Black–Scholes–Merton framework is a challenging problem.

✉ S. C. Maree
maree@cwi.nl

L. Ortiz-Gracia
luis.ortiz-gracia@ub.edu

C. W. Oosterlee
c.w.oosterlee@cwi.nl

1 Academic Medical Center, University of Amsterdam, Amsterdam, The Netherlands

2 Centrum Wiskunde & Informatica, Amsterdam, The Netherlands

3 Department of Econometrics, University of Barcelona, Diagonal 690, 08034 Barcelona, Spain

4 Delft Institute of Applied Mathematics, Delft University of Technology, Delft, The Netherlands

Bermudan options are basically options that can be exercised at a finite set of dates prior to maturity. This path-dependency and the requirement of the optimal early-exercise strategy make efficient pricing of Bermudan options challenging.

A broad class of option pricing methods is the class of so-called transform methods, where computations take place in the Fourier domain, often utilizing the Fast Fourier Transform (FFT) for fast computations [3,4,8–10,16]. Option pricing methods based on wavelet expansions have been discussed in [12,14,15] for European options. A method for pricing discrete barrier options with a wavelet expansion method is introduced in [11].

The Shannon Wavelet Inverse Fourier Technique (SWIFT) method [15] is an option pricing method for European options based on a Shannon wavelet expansion of the underlying density function. Shannon wavelets are smooth wavelets generated from the cardinal sine function [5]. Shannon wavelets have been used before in the pricing of discrete barrier options in [10], but to approximate Hilbert transforms. In the SWIFT method, the Shannon wavelet expansion is used to directly approximate the underlying density function.

In this paper, we extend the theory of the SWIFT method for European options, and we derive a complete error bound, proving that it exhibits exponential convergence with respect to the wavelet approximation scale.

We furthermore show that the SWIFT method can be reduced to the state of the art COS method [8] under specific parameter choices. The main difference between the two methods is that for the COS method, one chooses a finite computational domain, and recovers the underlying density function by a Fourier series expansion. The accuracy is then controlled by adding more Fourier terms, but the computational domain cannot be increased without recomputing all coefficients. In many situations however, for example in stochastic control problems, backward stochastic differential equations (BSDEs) or recursive pricing problems like Bermudan option pricing, it is unclear how to select a proper computational domain a priori. Then due to the recursion in time, errors caused by an insufficient domain propagate, resulting in incorrect option prices.

With wavelet series expansions the procedure goes differently. First, one determines a required accuracy, and then the size of the domain can be controlled by adding more wavelet terms. An important result is that for Shannon wavelets, the required wavelet approximation scale can be determined a priori using analytic properties of the characteristic function, and the computational domain can be determined recursively, making the method parameter-free.

Finally, we extend the SWIFT method to the pricing of path-dependent and discrete barrier options under exponential Lévy dynamics. We can speed up the computations by benefiting from the FFT, but the main advantage being that we have a natural solution to prevent domain boundary errors. We show numerically that the method exhibits exponential convergence with respect to the wavelet scale when the underlying density is smooth, and algebraic convergence otherwise.

This paper is organized as follows. In Sect. 2, the basics of wavelet approximation theory are discussed in the context of the Multi Resolution Analysis (MRA) framework, together with an analysis of the Shannon MRA. Then, we describe the

SWIFT method for European options in Sect. 3 and prove exponential convergence with respect to the wavelet scale.

An efficient algorithm for Bermudan option pricing with the SWIFT method is presented in Sect. 4. Moreover, we present a second approach of approximating the wavelet coefficients in Sect. 4.2, which is beneficial for short maturity options. In Sect. 4.3, we show how to price discretely monitored barrier options according to the same principle. Numerical results showing exponential convergence and improved boundary behavior are presented in Sect. 5 and we conclude in Sect. 6.

2 Multi resolution analysis

Point of departure for a wavelet analysis is the function space $\mathrm{L}^2(\mathbb{R})$. A Multi Resolution Analysis (MRA) consists of a sequence of nested successive approximation spaces V_m in $\mathrm{L}^2(\mathbb{R})$, being closed subspaces that satisfy,

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

with the properties,

$$\overline{\bigcup_{m\in\mathbb{Z}} V_m} = \mathrm{L}^2(\mathbb{R}), \quad \text{and} \quad \bigcap_{m\in\mathbb{Z}} V_m = \{0\}. \tag{2}$$

There are many subspaces that satisfy the two properties above that have nothing to do with multi resolution. Multi resolution is a consequence of an additional requirement,

$$f(x) \in V_m \Leftrightarrow f(2x) \in V_{m+1}, \tag{3}$$

or equivalently, $f(x) \in V_0 \Leftrightarrow f(2^m x) \in V_m$, that is, all the spaces V_m are scaled versions of the central space V_0.

The second feature we require from an MRA is invariance of V_0 under integer translations,

$$f(x) \in V_0 \Rightarrow f(x-k) \in V_0, \text{ for all } k \in \mathbb{Z}. \tag{4}$$

Requirement (3) implies a similar translation for the spaces V_m, i.e., if $f(x) \in V_m \Rightarrow f(x - 2^m k) \in V_m$ for all $k \in \mathbb{Z}$. We are now ready to define MRA.

Definition 1 (*MRA*) Let $\phi \in \mathrm{L}^2(\mathbb{R})$ be the generator of a wavelet family $\{\phi_{m,k}\}_{m,k\in\mathbb{Z}}$ with $\phi_{m,k}(x) := 2^{\frac{m}{2}}\phi(2^m x - k)$, and define the spaces $V_m \subset \mathrm{L}^2(\mathbb{R})$ as,

$$V_m := \underset{\mathrm{L}^2(\mathbb{R})}{\mathrm{closure}} \left\langle \{\phi_{m,k}\}_{k\in\mathbb{Z}} \right\rangle, \quad m \in \mathbb{Z}. \tag{5}$$

If V_m satisfies the properties (1)–(4), and $\{\phi_{0,k}\}$ forms an orthogonal basis[1] of V_0, then we say that ϕ generates an MRA, and ϕ is called a **scaling function**, or **father wavelet**.

[1] This definition can be relaxed by requiring that the set $\{\phi_{0,k}\}$ forms a Riesz-basis of V_0, see [7].

In words, Definition 1 states that an MRA is a special structure of nested spaces generated from a single function, called the scaling function.

Lemma 1 *Let us define $\mathscr{P}_m f$ as the orthogonal projection of a function $f \in L^2(\mathbb{R})$ on the space V_m of (5), which is by construction given by,*

$$\mathscr{P}_m f(x) = \sum_{k \in \mathbb{Z}} \langle f, \phi_{m,k} \rangle \phi_{m,k}(x), \tag{6}$$

where the inner product is $\langle f,\ g\rangle := \int_{\mathbb{R}} f(x)\overline{g(x)}dx$. Then, convergence of the projection $f(x) = \lim_{m\to\infty} \mathscr{P}_m f(x)$ holds in the $L^2(\mathbb{R})$-norm.

For a proof of the $\mathrm{L}^2(\mathbb{R})$-convergence of wavelet approximations of Lemma 1 and more theory on wavelet approximations, see for example [6,7,13].

2.1 Shannon wavelet approximations

Shannon wavelets are named after Claude Shannon, "the father of information theory" and founder of the sampling theory in signal analysis [18]. The key function in that context is the *cardinal sine function* $\mathrm{sinc}(x) := \frac{\sin(\pi x)}{\pi x}$, extended by $\mathrm{sinc}(0) := 1$, as shown in Fig. 1. In an MRA setting, this cardinal sine function will perform the role of scaling function, $\phi(x) := \mathrm{sinc}(x)$, which we refer to as the Shannon scaling function. The Shannon scaling function is particularly useful due to its simplicity in the Fourier domain,

$$\hat{\phi}(\omega) := \int_{\mathbb{R}} \phi(x)e^{-i\omega x}\, dx = \mathrm{rect}\left(\tfrac{\omega}{2\pi}\right),$$

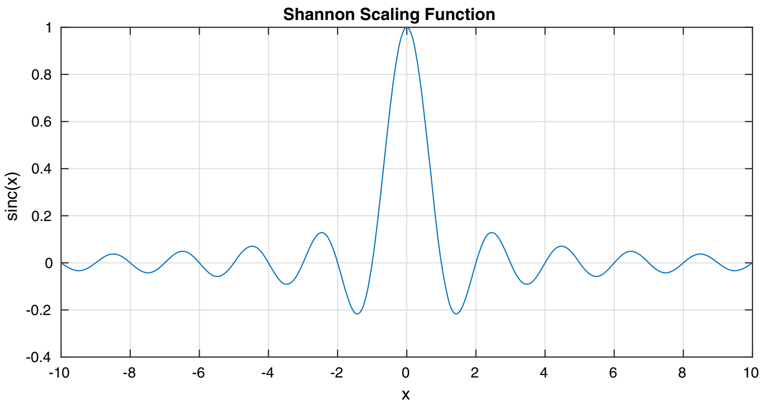

Fig. 1 Shannon Scaling function $\mathrm{sinc}(x) := \sin(\pi x)/(\pi x)$

where rect () is the rectangle function, defined as,

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

Due to this simplicity in the Fourier domain, there is a close connection with band-limited functions.

Definition 2 A function f is called **band-limited** if there exists a positive constant $B < \infty$, such that,

$$f(x) = \frac{1}{2\pi} \int_{-B\pi}^{B\pi} \hat{f}(\omega) e^{i\omega x}\, d\omega,$$

i.e., its Fourier transform $\hat{f}$ is identically zero on $|\omega| > B\pi$. The parameter B is referred to as the bandwidth of f.

We consider an MRA generated from the Shannon scaling function, defined as $\phi(x) := \operatorname{sinc}(x)$, with wavelets defined by $\phi_{m,k}(x) := 2^{\frac{m}{2}} \operatorname{sinc}(2^m x - k)$, and its Fourier transform $\hat{\phi}_{m,k}$ is given by,

$$\hat{\phi}_{m,k}(\omega) = 2^{-\frac{m}{2}} e^{-i\omega \frac{k}{2^m}} \operatorname{rect}\left(\frac{\omega}{2^{m+1}\pi}\right).$$

The relation between the Shannon MRA and band-limited functions is stated in the following lemma from [19].

Lemma 2 *Consider an MRA generated from the Shannon scaling function* $\phi(x) = \operatorname{sinc}(x)$. *The space* V_m *as defined in Definition 1 is precisely the space of all functions* $f \in L^2(\mathbb{R})$ *with bandwidth* $B \leq 2^m$.

Combining Lemma 2 with Lemma 1 yields an alternative formulation for the orthogonal Shannon wavelet projection.

Corollary 1 *The space* V_m *as defined in Definition 1 is the space of all functions* $f \in L^2(\mathbb{R})$ *with bandwidth* $B \leq 2^m$. *Therefore, the orthogonal projection* $\mathscr{P}_m : L^2(\mathbb{R}) \to V_m$ *of* (6) *is equivalent to,*

$$\mathscr{P}_m f(y) = \frac{1}{2\pi} \int_{-2^m\pi}^{2^m\pi} \hat{f}(\omega) e^{i\omega y} d\omega. \quad (7)$$

Proof of Corollary 1 is given in Appendix 3. Another corollary of Lemma 2 is a version of the well-known Whittaker-Shannon interpolation formula, see [19].

Corollary 2 *Let g be a band-limited function with bandwidth B, then g can be recovered exactly by a Shannon wavelet expansion at scale $B < 2^m$, and we have,*

$$g(x) = \sum_{k\in\mathbb{Z}} 2^{-\frac{m}{2}} g(\tfrac{k}{2^m})\phi_{m,k}(x),$$

where the series converges uniformly if $g \in L^2(\mathbb{R})$ or $g \in L^1(\mathbb{R})$.

The density functions we encounter in finance are not band-limited, so no exact recovery is possible with a Shannon wavelet expansion for a finite scale m. However, these density functions have a fast decay in the Fourier domain, which results in accurate approximations even at low wavelet scales m, as stated in the following lemma.

Lemma 3 *Let $f \in L^2(\mathbb{R})$ and let $H(\xi)$ represent the mass in the tails of Fourier transform $\hat{f}$,*

$$H(\xi) := \frac{1}{2\pi}\int_{|\omega|>\xi} \left|\hat{f}(\omega)\right| d\omega. \tag{8}$$

The pointwise approximation error $\varepsilon_m(y)$ due to the projection of f onto the space V_m in (5) is given by $\varepsilon_m(y) := f(y) - \mathscr{P}_m f(y)$, and can be uniformly bounded by $|\varepsilon_m(y)| \leq H(2^m\pi)$.

Proof We write f as the inverse Fourier transform of $\hat{f}$ and use Corollary 1 to rewrite $\mathscr{P}_m f$. Then, the point-wise error is given by,

$$\varepsilon_m(y; x) := f(y|x) - \mathscr{P}_m f(y|x) = \frac{1}{2\pi}\int_{|\omega|>2^m\pi} \hat{f}(\omega; x)e^{i\omega y} d\omega. \tag{9}$$

The desired bound follows by taking the modulus and noting that $\left|e^{i\omega y}\right| = 1$. □

For the Lévy processes we use, the characteristic function is known, and therefore we can study its decay rate to determine a suitable approximation scale, which we discuss in Sect. 3.2.

An important difficulty to realize when working with the sinc-function, is that no analytic form for its integral $\mathrm{Si}(t) := \int_0^t \mathrm{sinc}(x)\,dx$ is available, which we require in the computation of the wavelet coefficients. This issue is not new in numerical integration, and was adressed in [1,2]. There, a combination of Vieta's formula and a cosine product-to-sum identity was used to approximate the sinc-function by a so-called incomplete cosine expansion. The same approximation can be derived as well by writing the sinc-function as its inverse Fourier transform,

$$\mathrm{sinc}(t) = \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{it\omega}\, d\omega = \frac{1}{\pi}\int_0^{\pi} \cos(t\omega)\, d\omega. \tag{10}$$

We can then discretize the right hand side to obtain the required approximation, as summarized in Lemma 4.

Lemma 4 *We numerically integrate the r.h.s. of (10) using the midpoint rule with J sub-intervals, thus obtaining the approximation,*

$$\operatorname{sinc}(t) \approx \operatorname{sinc}^*(t;J) := \frac{1}{J}\sum_{j=1}^{J} \operatorname{Re}\left\{e^{it\omega_j}\right\}, \tag{11}$$

where $\omega_j := \frac{\pi}{J}(j-\frac{1}{2})$. The approximation error due to the midpoint rule on a finite domain $|t| \le a \le \frac{\pi}{2}J$ is bounded by,

$$\left|\operatorname{sinc}(t) - \operatorname{sinc}^*(t;J)\right| \le \frac{(\pi a)^2}{(4J)^2 - (\pi a)^2}. \tag{12}$$

The Shannon wavelet can be approximated similarly, and we denote its approximation by $\phi^*_{m,k}(y) := 2^{\frac{m}{2}} \operatorname{sinc}^*(2^m y - k; J)$. By error bound (12), the maximum error for any $1-\kappa \le k \le \kappa$ inside the domain $|y| \le c$ for a fixed $J \ge \frac{\pi}{2}(2^m c + \kappa)$ is given by,

$$\max_{|y|\le c,\, 1-\kappa \le k \le \kappa} \left|\phi_{m,k}(y) - \phi^*_{m,k}(y)\right| \le 2^{\frac{m}{2}} \frac{(\pi(2^m c+\kappa))^2}{(4J)^2 - (\pi(2^m c+\kappa))^2}.$$

The approximation of the sinc-function in Lemma 4 can be derived in different ways. Besides the derivation based on Vieta's formula in [2], a derivation based on Parseval's identity, see [15], was given, which is a discretization of the integral that arises in Corollary 1. If that integral is approximated by the midpoint rule with J sub-intervals, it coincides once more with the result in Lemma 4, which was the missing link between the two approaches.

The derivation here generalizes[2] the previous approaches as it is valid for any natural number J, while in previous results, $J = 2^{\eta-1}$ for some $\eta \in \mathbb{N}$ was used. The error bound (12) now follows directly from [15, Lemma 2].

3 European option pricing

Before we discuss Bermudan options, we start by analyzing the SWIFT method for European pricing problems as in [15] in further detail. We derive an analytic error bound and derive the appropriate wavelet approximation scale.

The pricing of a European option under Lévy asset price processes in computational finance is governed by the numerical solution of partial (integro-) differential equations. The corresponding solution, being the option value at time t, can be found by means of the Feynman–Kac formula as the discounted expectation of the option value at final time T. We consider the risk-neutral option valuation formula,

$$v(x,t) = e^{-r\Delta t}\mathbb{E}[\, v(y,T) \mid x] = e^{-r\Delta t}\int_{\mathbb{R}} v(y,T) f(y|x)\, dy, \tag{13}$$

[2] It should be mentioned that the FFT is applied most efficiently when the number of coefficients is a power of two. The generalization here is only of interest from a mathematical point of view.

Table 1 Characteristic exponents and parameters restrictions for Lévy processes occurring in financial applications

Model	$\psi_L(\omega)$	Param. restrictions
GBM	$-\frac{\sigma^2}{2}\omega^2$	$\sigma > 0$
NIG	$\delta\left(\sqrt{\alpha^2-(\beta+i\omega)^2}-\sqrt{\alpha^2-\beta^2}\right)$	$\alpha, \delta > 0$
VG	$-\frac{\sigma^2}{2}\omega^2-\frac{1}{v}\log\left(1-iv\theta\omega+v\frac{\sigma_v^2}{2}\omega^2\right)$	$v, \sigma_v > 0, \sigma \geq 0$
CGMY	$C\Gamma(-Y)\left((M-i\omega)^Y-M^Y+(G+i\omega)^Y-G^Y\right)$	$C, G > 0$

Model details can be found in [17]

where v denotes the option value, T the maturity, t the initial date, $\Delta t := T - t$ the remaining time, $\mathbb{E}$ is the expectation operator under the risk-neutral measure, 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.

Denote by $\{S_t\}_{t=0}^T$ the underlying asset price process and let K be the strike price of the option. We model the asset price process by an exponential Lévy process, so that the log-transformed process $X_t := \log(S_t/K)$ is a Lévy process with drift. Let the state variables be given by $x := X_t = \log(S_t/K)$ and $y = X_T := \log(S_T/K)$. We are interested in the conditional probability density function $f(y|x)$, but this function is rarely known in analytic form. However, due to the celebrated Lévy–Khintchine formula, the Fourier transform $\hat{f}(\omega; x)$ of the density function $f(y|x)$ is known and given by,

$$\hat{f}(\omega; x) := \mathbb{E}\left[e^{-i\omega X_T}\right] = e^{-i\omega x}e^{-i\omega\mu T+T\psi_L(-\omega)} =: e^{-i\omega x}\hat{f}(\omega), \tag{14}$$

where μ is a drift correction term defined as $\mu := r - \psi_L(-i)$, $\psi_L(\omega)$ is the characteristic exponent that uniquely defines the Lévy process, and $\hat{f}(\omega) := \hat{f}(\omega; 0)$. For different models in finance, the characteristic components are listed in Table 1 with their model specific parameters. See [17] for details about these processes.

Remark 1 The SWIFT method can be applied to European pricing problems when the characteristic function of the price process is known. For Bermudan pricing problems, we require that the density function can be written as $f(y|x) = f(y - x|0)$, which is possible for Lévy processes and also for the Heston model.

Under the log-asset transformation, the option value at maturity time T of a European put option can be written as,

$$v(y, T) = g^{put}(y) := K(1-e^y)^+ = K \cdot \max\left(1-e^y, 0\right). \tag{15}$$

Similarly, the payoff function for a call option is $g^{call}(y) = K(e^y - 1)^+$, but this payoff grows exponentially for large values of y which could cause significant round-off errors along the domain boundary. It is therefore highly recommended to price call options via put options by applying the put-call parity relation.

3.1 SWIFT for European options

We assume that the underlying conditional density function $f(y|x)$ is an $L^2(\mathbb{R})$-function so that we can apply the theory of MRA of Sect. 2. The SWIFT method consists of three steps to approximate the density function by recovering it from its characteristic function $\hat{f}(\omega; x)$.

Step 1 (Wavelet projection). In the first step, f is approximated by its Shannon wavelet projection at scale $m \in \mathbb{Z}$. By Lemma 1, this is,

$$f(y|x) \approx f_1(y|x) := \mathscr{P}_m f(y|x) = \sum_{k\in\mathbb{Z}} D_{m,k}(x)\phi_{m,k}(y), \tag{16}$$

where the *density coefficients* depend on the initial asset price x and are defined as $D_{m,k}(x) := \langle f(\cdot|x),\ \phi_{m,k}\rangle$.

Step 2 (Series truncation). To numerically work with the wavelet approximation, the infinite summation in (16) has to be truncated. If the density function vanishes as $y \to \pm\infty$, the wavelet coefficients $D_{m,k}$ vanish as well, which can be seen by noting that for $k \in \mathbb{Z}$,

$$f(2^{-m}k|x) \approx f_1(2^{-m}k|x) = 2^{\frac{m}{2}} D_{m,k}(x). \tag{17}$$

Thus we truncate[3] the summation range for some $\kappa \in \mathbb{N}$, so that we obtain,

$$f_1(y|x) \approx f_2(y|x) := \sum_{k=1-\kappa}^{\kappa} D_{m,k}(x)\phi_{m,k}(y). \tag{18}$$

Step 3 (Coefficient approximation). The final step is to compute the density coefficients $D_{m,k}(x) := \langle f(\cdot|x),\ \phi_{m,k}\rangle$. We do this by replacing $\phi_{m,k}$ by $\phi^*_{m,k}$ as in Lemma 4, so that we obtain,

$$\begin{aligned} D_{m,k}(x) \approx D^*_{m,k}(x) &:= \int_{\mathbb{R}} f(y|x)\overline{\phi^*_{m,k}(y)}\,dy \\ &= \frac{2^{\frac{m}{2}}}{J}\sum_{j=1}^{J} \mathrm{Re}\left\{\int_{\mathbb{R}} f(y|x)e^{-i\omega_j(2^m y-k)}dy\right\} \\ &= \frac{2^{\frac{m}{2}}}{J}\sum_{j=1}^{J} \mathrm{Re}\left\{\hat{f}\left(\omega_j 2^m; x\right)e^{i\omega_j k}\right\}. \end{aligned} \tag{19}$$

In Appendix 2, we show we show how to compute the vector of coefficients $D^*_{m,k}(x)$ efficiently using the Fast Fourier Transform (FFT). We replace the density coefficients by their approximation and obtain the SWIFT series approximation of the density function,

[3] The only reason that we choose a symmetric summation range $1-\kappa \le k \le \kappa$ is for convenience of notation. It is straightforward to work with an arbitrary range $\kappa_1 \le k \le \kappa_2$ for $\kappa_1, \kappa_2 \in \mathbb{Z}$.

$$f_2(y|x) \approx f_3(y|x) := \sum_{k=1-\kappa}^{\kappa} D^*_{m,k}(x)\phi_{m,k}(y). \tag{20}$$

To solve the option pricing integral in (13), we truncate the integration range[4] to $|y| \leq c$ for some positive constant c, so that we obtain,

$$v(x,t) \approx v_0(x,t) := e^{-r\Delta t} \int_{|y|\leq c} f(y|x)v(y,T)\,dy. \tag{21}$$

We substitute the approximation of the density function in (20) that we obtained after the three consecutive approximation steps into the truncated pricing integral (21),

$$\begin{aligned} v(x,t) \approx v_3(x,t) &:= e^{-r\Delta t} \int_{|y|\leq c} f_3(y|x)v(y,T)\,dy \\ &= e^{-r\Delta t} \sum_{k=1-\kappa}^{\kappa} D^*_{m,k}(x) \int_{|y|\leq c} v(y,T)\phi_{m,k}(y)\,dy. \end{aligned} \tag{22}$$

The remaining integrals are closely related to the wavelet coefficients of the value function $v(y,T)$ in y. We therefore define the *value coefficients* $V_{m,k}(T)$ at time T by,

$$V_{m,k}(T) := \int_{|y|\leq c} v(y,T)\phi_{m,k}(y)\,dy. \tag{23}$$

With this definition, the resulting option value at time t can be written as,

$$v(x,t) \approx v_3(x,t) = e^{-r\Delta t} \sum_{k=1-\kappa}^{\kappa} D^*_{m,k}(x)V_{m,k}(T). \tag{24}$$

Remark 2 The truncation of the integration range to $|y| \leq c$ is required for the approximation of the sinc-function by the approach in Lemma 4, as this approximation holds only on a finite domain, which we require in the computation of the value coefficients (23). It is however easy to extend this domain, as discussed in the error analysis in the next section.

Remark 3 From the series truncation argument in (17), when $f(y|x)$ is negligible for $|y| > c$, it follows that we should choose $\kappa \geq 2^m c$. Furthermore, by Lemma 4, we should choose $J \geq \pi\kappa$.

[4] Truncation to a symmetric domain is not required, but chosen only for ease of notation.

3.1.1 Vanilla payoff coefficients

For European options, the option value $v(y, T)$ at maturity equals the payoff function $g(y)$, see (15). Thus, the value coefficients $V_{m,k}(T)$ are given by,

$$V_{m,k}(T) = \int_{|y|\leq c} v(y, T)\phi_{m,k}(y)\, dy = \int_{|y|\leq c} g(y)\phi_{m,k}(y)\, dy =: G_{m,k}(-c, c), \tag{25}$$

and we refer to $G_{m,k}$ as the *payoff coefficients*. These integrals depend on the payoff function g, but for the common payoff functions, the integral cannot be solved analytically. In [15], approximation formulas for the payoff coefficients for put, call and digital option payoffs were derived. For a European put with payoff function g given by (15), the *approximated payoff coefficients* $G^*_{m,k}$ are given by,

$$\begin{aligned} G^*_{m,k}(a, b) &:= K \int_a^b (1 - e^y)^+ \phi^*_{m,k}(y)\, dy \\ &= K \int_a^{\bar{b}} (1 - e^y)\phi^*_{m,k}(y)\, dy \\ &= K \frac{2^{\frac{m}{2}}}{J} \sum_{j=1}^{J} \mathrm{Re}\left\{ e^{-i\omega_j k} \int_a^{\bar{b}} (1 - e^y) e^{i\omega_j 2^m y}\, dy \right\}, \end{aligned} \tag{26}$$

where $\bar{b} := \min(0, b)$. The remaining integral is easily solved analytically, and the whole vector of coefficients can be computed efficiently using the FFT as explained in Appendix 1.

The computation of the payoff coefficients is the final approximation step in the SWIFT method, and by plugging $V_{m,k}(T) = G_{m,k}(-c, c) \approx G^*_{m,k}(-c, c)$ into (22), we obtain the SWIFT pricing formula for European options,

$$v(x, t) \approx v_4(x, t) := e^{-r\Delta t} \sum_{k=1-\kappa}^{\kappa} D^*_{m,k}(x) G^*_{m,k}(-c, c). \tag{27}$$

3.2 European option pricing error analysis

We present an error analysis of the SWIFT method for European options, i.e., the approximation error $\varepsilon(x) := v(x, t) - v_3(x, t)$ with v_3 in (24). We assume that the payoff coefficients $V_{m,k}(T)$ are given explicitly, that the payoff function $g(y)$ is bounded[5] and to simplify notation, we assume $r = 0$.

[5] The assumption of a bounded payoff holds for put and digital options. For options with an unbounded payoff, one has to assume a certain decay rate on the density function in order to bound $\varepsilon_0(c)$ in Lemma 5. This assumption is introduced to derive an error bound, and is not required for the SWIFT pricing formula (27), as the pricing integral is truncated before any approximation is made, and the payoff is finite on a bounded domain.

Key to the existence of the option pricing integral (13) is that the density function decays faster than the growth of the payoff function. We make no assumptions on the decay rate of the density function, as it is unknown in general, but since the mass in the tails of the density function tends to zero, for every $TOL > 0$ there exists a value $c > 0$ such that,

$$\tau(c) := \int_{|y|>c} f(y|x)\, dy \leq TOL. \tag{28}$$

Using this observation, it follows directly that the error due to the integration range truncation to $|y| \leq c$ is bounded, as stated in the following lemma.

Lemma 5 *Let $\varepsilon_0(c) := v(x,t) - v_0(x,t)$ be the error caused by the truncation of the integration range to $|y| \leq c$ in (21). Then,*

$$\varepsilon_0(c) = \int_{|y|>c} f(y|x)g(y)dy \leq \tau(c)\, \|g\|_\infty ,$$

where the infinity norm $\|g\|_\infty := \sup\{|g(y)| : y \in \mathbb{R}\}$. The bound can be made arbitrarily small by increasing the value c.

In the following lemma, we show how the decay rate of the density and its Fourier transform relate to the error in the wavelet projection and series truncation.

Lemma 6 *The error $\varepsilon_2(m,\kappa)$ caused by approximating f by the truncated Shannon wavelet series $f_2(y|x) := \sum_{k=1-\kappa}^{\kappa} D_{m,k}(x)\phi_{m,k}(y)$ as in (18) is given by,*

$$\begin{aligned}|\varepsilon_2(m,\kappa)| &= \left| \int_{|y|\leq c} [f_2(y|x) - f(y|x)]g(y)\, dy \right| \\ &\leq 2c\, \|g\|_\infty \left[(2\kappa+3)H(2^m\pi) + 2^m \tau(\tfrac{\kappa}{2^m})\right].\end{aligned}$$

The proof of Lemma 6 is given in Appendix 3. The remaining step in the SWIFT approximation is the replacement of the density coefficients $D_{m,k}$ by $D^*_{m,k}$ as in (19) which is discussed in [15]. The midpoint rule approximation of the sinc-functions yields a summation of cosines, and is thus periodic. To reduce the impact of the undesired periodic replications of the sinc-function, the density function acts as a windowing function, but only if the period of $\text{sinc}^*(\cdot\,; J)$ is 'large enough'.

Lemma 7 *When the mass in the tails of the density function is represented by $\tau(c)$ as in (28), the error $\varepsilon_3(J) := v_3(x,t) - v_2(x,t)$ is bounded by,*

$$|\varepsilon_3(J)| \leq 2^m(2\kappa+1)\, \|g\|_\infty \left(2\tau(c) + \sqrt{2c}\, \|f\|_2 \frac{(\pi\kappa)^2}{(2J)^2 - (\pi\kappa)^2}\right),$$

when $J \geq \pi\kappa \geq 2^m\pi c$.

The proof of Lemma 7 can be found in Appendix 3. We are now ready to combine all of the above results.

Theorem 1 *The SWIFT pricing formula (24) for European options is bounded by,*

$$\frac{|v(x, t_0) - v_3(x, t_1)|}{\|g\|_\infty e^{-r\Delta t}} = \mathscr{O}\Big(2^m \tau(c) + H(2^m \pi)\Big), \tag{29}$$

whenever $J \geq \pi\kappa \geq 2^m \pi c$, where $\tau(c)$ represents the mass in the tails of the density function (28), and $H(2^m\pi)$ the mass in the tails of the characteristic function (8).

The proof of Theorem 1 can be found in Appendix 3. Theorem 1 states that the error of the SWIFT pricing formula depends the decay of both the density function and its characteristic function. Unfortunately, the uncertainty principle of Fourier transforms states that a fast decay in the Fourier domain implies a slow decay in the time domain and vice-versa. Optimal convergence is obtained when f is Gaussian with variance σ, so that its Fourier transform is a Gaussian with variance σ^{-1}.

In the following sections we discuss how to find a suitable value for the two remaining parameters, the wavelet scale m and the domain truncation parameter c.

3.3 Wavelet scale determination

It is important that the wavelet scale m is chosen a large enough, as it is not possible to alter the scale of approximation afterwards without recomputing all that was done before. We use the Fourier transform $\hat{f}$ of the density function, which is known in analytic form, to find an analytic expression to determine a sufficient wavelet scale m.

All of the processes of interest[6] satisfy,

$$\left|\hat{f}(\omega; x)\right| = \left|e^{\Delta t \psi_L(\omega)}\right| \leq C e^{-d\Delta t|\omega|^\nu}, \tag{30}$$

with constants $C, d > 0$ and $\nu \in (0, 2]$. For any process with a Brownian motion component, as indicated by $-\frac{1}{2}\sigma^2\omega^2$, the bound in (30) is satisfied with $\nu = 2$, which is the ideal case from a computational point of view. For NIG, $\nu = 1$, while $\nu = \bar{Y}$ for CGMY,[7] which can be directly read from the characteristic exponents in Table 1.

With (30), the mass in the tails of the characteristic function $H(2^m\pi)$ is given by,

$$H(2^m\pi) \leq \frac{C}{\pi}\int_{2^m\pi}^{\infty} e^{-d\Delta t|\omega|^\nu} d\omega = \frac{C}{\pi\nu(d\Delta t)^{1/\nu}}\Gamma\left(\frac{1}{\nu}, d\Delta t(2^m\pi)^\nu\right) =: \bar{\varepsilon}_m,$$

where $\Gamma(a, x)$ is the incomplete gamma function, and for large values of $2^m\pi$, the error behaves as,

$$\bar{\varepsilon}_m \sim \frac{C(2^m\pi)^{1-\nu}}{\pi\nu d\Delta t} e^{-d\Delta t(2^m\pi)^\nu}, \tag{31}$$

[6] Geometric Brownian Motion, Normal Inverse Gaussian, Kou, Merton jump model and CGMY. The exception is pure jump Variance Gamma, for which $|\hat{f}(\omega; x)| = \mathscr{O}(C|\omega|^{-2\Delta t/v})$, for which $\hat{f}$ fails to be integrable if $\Delta t \leq v/2$.

[7] We denote the parameters of the CGMY model by $(\bar{C}, \bar{G}, \bar{M}, \bar{Y})$ to avoid confusion with other model parameters.

which holds uniformly for ω and x. From (31), we see that the error converges exponentially with respect to the wavelet scaling factor 2^m. The parameters C and d in (30) are often not readily available, thus we substitute $Ce^{-d\Delta t(2^m\pi)^\nu} \sim |\hat{f}(\pm 2^m\pi; x)|$ back into (31), so that we obtain the approximation,

$$\bar{\varepsilon}_m \sim \frac{(2^m\pi)^{1-\nu}}{2\pi\nu\Delta t}\left(\left|\hat{f}(-2^m\pi; x)\right| + \left|\hat{f}(2^m\pi; x)\right|\right), \tag{32}$$

which we can now cheaply evaluate. A simple iterative procedure can be performed, by setting $m = 0, 1, 2, \ldots$ until $\bar{\varepsilon}_m < TOL$, for some user defined tolerance TOL.

3.4 Domain truncation determination

In general, the mass $\tau(c)$ in the tails of the density function is unknown, but we can determine its rate of decay from the characteristic function, as stated in the following lemma, of which the proof can be found in both [19,20].

Lemma 8 *Consider a function $f \in L^2(\mathbb{R})$ with Fourier transform $\hat{f}$. Define the interval $-\infty \le \lambda_- \le 0 \le \lambda_+ \le \infty$ and let $\lambda := \min(|\lambda_-|, \lambda_+)$. When $\hat{f}$ is analytic in the domain $\mathscr{D}(\lambda_-, \lambda_+) := \{z \in \mathbb{C} : \mathrm{Im}\{z\} \in (\lambda_-, \lambda_+)\}$ then $f(x) = \mathscr{O}(e^{-\lambda|x|})$ for $x \to \pm\infty$.*

For the asset price models we consider, the strip in which $\hat{f}$ is analytic is given in Table 2. It follows from Lemma 8 that all of the models have exponential decay for some decay rate $d > 0$, and thus the mass in the tails $\tau(c)$ decays exponentially in c. However, there are no analytic results on how to find the corresponding c such that $\tau(c) < TOL$ for some user selected tolerance TOL. We therefore use a rule of thumb that was used before in [8].

First, use the cumulants of the density function to heuristically determine an initial guess for the log-asset domain such that $\tau(c) < TOL$, as introduced in [8], by setting,

$$c := |c_1| + L\sqrt{c_2 + \sqrt{c_4}}, \tag{33}$$

where c_i is the ith cumulant, and given for common Lévy processes in Table 2. Numerical results suggest that $L = 6$ is sufficient for the SWIFT method, while $L = 8$ is sufficient for the COS method [9].

Secondly, we set $\kappa := \lceil 2^m c \rceil$, compute the vector of density coefficients using the FFT, and evaluate the test $T_1(\kappa)$,

$$T_1(\kappa) := \left|1 - \int_{\mathbb{R}} f_3(y|x)\,dy\right| = \left|1 - 2^{-\frac{m}{2}} \sum_{k=1-\kappa}^{\kappa} D_{m,k}\right|. \tag{34}$$

If $T_1(\kappa) > TOL$, we repeat the procedure by increasing κ. Previously computed coefficients can still be used, and only the new coefficients have to be computed.

Table 2 Cumulants for Lévy processes occurring in financial applications

Model	Cumulants	Analytic strip (λ_-, λ_+)
GBM	$c_1 = \mu t,\ c_2 = \sigma^2 t,\ c_4 = 0$	$\mathbb{R}$
NIG	$c_1 = \mu t + \delta t\beta/\sqrt{\alpha^2-\beta^2}$	$[\beta \pm \alpha]$
	$c_2 = \delta t \alpha^2(\alpha^2-\beta^2)^{-3/2}$	
	$c_4 = 3\delta t\alpha^2(\alpha^2+4\beta^2)(\alpha^2-\beta^2)^{-7/2}$	
VG	$c_1 = t(\mu+\theta)$	$\left[\frac{\theta}{\sigma^2} \pm \sqrt{\frac{\theta^2}{\sigma_v^4}+\frac{2}{\nu\sigma_v^2}}\right]$
	$c_2 = t\sigma^2 + t(\sigma_v^2 + v\theta^2)$	
	$c_4 = 3t(\sigma_v^2 v + 2\theta^4 v^3 + 4\sigma_v^2\theta^2 v^2)$	
CGMY	$c_1 = \mu t + Ct\Gamma(1-Y)(M^{Y-1} - G^{Y-1})$	$[-M, G]$
	$c_2 = Ct\Gamma(2-Y)(M^{Y-2} - G^{Y-2})$	
	$c_4 = Ct\Gamma(4-Y)(M^{Y-4} - G^{Y-4})$	

The drift parameter is defined as $\mu = r - q - \psi_L(-i)$. Corresponding processes are shown in Table 1. Specific parameters for asset pricing problems can be found in for example [17]

3.5 Domain truncation error

Before we continue to the main contribution of this paper, Bermudan option pricing, we discuss an advantage of the SWIFT method and the motivation for further research.

The SWIFT method exhibits a close relation to the COS method [8]. That method recovers the density function by a Fourier cosine expansion, which is defined on a finite domain and periodically extended outside this domain. This causes price underestimation within the computational domain near to the domain boundary where the payoff function is non-zero, as highlighted in [16]. A ‘workaround’ solution proposed there is by an extrapolation of the payoff coefficients.

The SWIFT method does not suffer from this problem. Wavelet methods in general are approximations on the whole real line. Although we need a truncation of the integration range to evaluate the wavelet coefficients efficiently (Lemma 4), this truncation $|y| \le c$ can be chosen independently of the truncated summation range $1-\kappa \le k \le \kappa$. Let us illustrate this independence by the problem of multiple strike pricing.

From the definition of the payoff coefficients $G^*_{m,k}$ in (26), it follows that we can factor the strike price K out of the computation of the payoff coefficients by defining $U^*_{m,k}(a,b)$ so that $G^*_{m,k}(a,b) = K \cdot U^*_{m,k}(a,b)$, which therefore becomes independent of the strike price.

We insert the definition of the density coefficients $D^*_{m,k}$ in (19) into the SWIFT pricing formula (24), and by interchanging summation and integration, we obtain,

$$v_4(x,t) = e^{-r\Delta t} K \sum_{j=1}^{J} \mathrm{Re}\left\{ \hat{f}\left(\omega_j 2^m; x\right) \left(\frac{2^{\frac{m}{2}}}{J} \sum_{k=1-\kappa}^{\kappa} U^*_{m,k}(-c,c) e^{i\omega_j k} \right) \right\}.$$

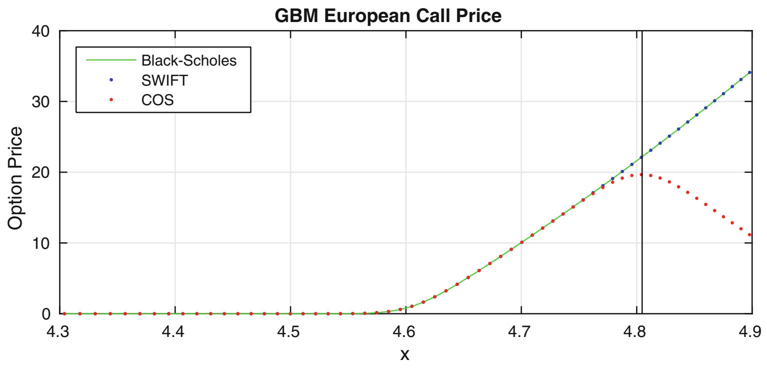

Fig. 2 A European put priced by the COS method on a domain [4.3, 4.8]. We observe price underestimation within the computational domain. The SWIFT method behaves fine, as we can integrate the payoff coefficients independently of the computational domain. Parameters as in [16]

The inner summation, which is independent of x and K, is given by,

$$\tilde{U}_j(-c, c) := \frac{2^{\frac{m}{2}}}{J} \sum_{k=1-\kappa}^{\kappa} U^*_{m,k}(-c, c)e^{i\omega_j k}, \tag{35}$$

which can be computed efficiently using the FFT, see Appendix 2.

Furthermore, for Lévy models and for the Heston stochastic volatility models, the SWIFT pricing formula allows for an efficient formulation. Recall from (14) that for these processes, the Fourier transform $\hat{f}$ of the density function can be factorized as $\hat{f}(\omega; x) = \hat{f}(\omega)e^{-i\omega x}$, where $\hat{f}(\omega) := \hat{f}(\omega; 0)$.

Let us denote vectors with a boldface letter, then we can write the SWIFT option pricing formula for a vector of strike prices $\mathbf{K}$, with corresponding initial asset values $\mathbf{x} := \log(S_0/\mathbf{K})$ as,

$$v_4(\mathbf{x}, t) = e^{-r\Delta t}\mathbf{K} \sum_{j=1}^{J} \text{Re}\left\{ \hat{f}(\omega_j 2^m)\tilde{U}_j(-c, c)e^{-i\omega_j 2^m \mathbf{x}} \right\}, \tag{36}$$

which requires only two times the FFT in the construction of $\tilde{U}_j$, independent of the number of strikes. The main observation here is that the integration range $-c \le y \le c$ within the coefficients $\tilde{U}_j(-c, c)$, see (26), can be chosen independent of the truncation of k. We set $\tilde{U}_j(\min(\mathbf{x}) - c, \max(\mathbf{x}) + c)$, and we observe that the boundary problem can be naturally avoided, as shown in Fig. 2. We highlight that the SWIFT method is also periodic, as the approximation of the sinc function in (11) is periodic. However, now we have the flexibility to cleverly choose the periodicity (in terms of J) far from the computational domain $x \in [a, b]$.

3.6 Relation with the COS method

The close relation between the SWIFT pricing formula (36) and the COS pricing formula [8, Eq. (19)] can be made explicit by a specific choice of parameters. In the SWIFT method, a local wavelet basis is used, however, we approximate the sinc function by a global periodic expansion, see (11).

The parameters for the COS method are the truncated domain $[a, b]$ and the number of Fourier coefficients J_{COS}, and the pricing formula is given by,

$$\begin{aligned}
v^{COS}(x,t) &= e^{-r\Delta t} {\sum_{n=0}^{J_{COS}-1}}' \operatorname{Re}\left\{D_n^{COS}\right\} \operatorname{Re}\left\{V_n^{COS}\right\}, \\
\text{where,} \quad D_n^{COS} &:= \check{f}\left(\frac{n\pi}{b-a}; x\right) e^{-i\frac{n\pi}{b-a}a}, \\
\text{and,} \quad V_n^{COS} &:= \int_a^b g(y) e^{in\pi\frac{y-a}{b-a}}\,dy.
\end{aligned} \tag{37}$$

We match the SWIFT method to this pricing formula by using the same computational domain $[a, b]$ and we approximate the wavelet coefficients $D^*_{m,k}$ and $V^*_{m,k}$ as in (19) and (26) with the same number of discretization points $J := \frac{1}{2}J_{COS}$, where k is in the range $1-J, \ldots, J$, and the wavelet scale m is selected according to $2^m := \frac{J_{COS}}{b-a}$. Then, we insert the definition of the payoff coefficients (26) into the SWIFT pricing formula (27), and by a change of summation and integration we obtain,

$$v^{SWIFT}(x,t) = e^{-r\Delta t} \sum_{j=1}^{J} \operatorname{Re}\left\{\left(\int_a^b g(y) e^{i\omega_j 2^m y}\,dy\right)\left(\sum_{k=1-J}^{J} D^*_{m,k}(x) e^{-i\omega_j k}\right)\right\}, \tag{38}$$

where $D^*_{m,k}$ is the discrete inverse Fourier transform of $\hat{f}$, and when we take the discrete Fourier transform of it, we return to the original vector. Thus we obtain that SWIFT estimate $v_4(x, t)$ of (27) is equivalently written as,

$$v^{SWIFT}(x,t) = e^{-r\Delta t} {\sum_{n=0}^{\frac{1}{2}J_{COS}-1}}' \operatorname{Re}\left\{D_{2n+1}^{COS} \cdot V_{2n+1}^{COS}\right\}, \tag{39}$$

which shows great similarity to (37). The main difference being that the COS method uses only the real part of the coefficients while the SWIFT method only sums the odd coefficients. The result is that the SWIFT approximation $v^{SW}(x, t)$ is replicated oddly and the COS approximation $v^{COS}(x, t)$ is evenly replicated.

Using this representation the SWIFT method loses its flexibility with respect to the boundary as $\kappa = J = 2^{m-1}(b-a)$, and thus the coefficients we use are directly related to the domain truncation. It gives however insight in the rate of convergence of the SWIFT method, which is equivalent to the COS method. Furthermore, it suggests that for the COS method, one should set $J_{COS} := 2^m(b-a)$. Thus, when one wants to retain accuracy while increasing the computational domain (see [15, Fig. 5]), one knows how to change J_{COS}, as m is known analytically from (32).

4 Pricing Bermudan and barrier options

A Bermudan option is a financial contract, which the holder can exercise at a predetermined finite set of exercise moments prior to maturity, and the holder of the option receives a payoff when she exercises the option.

Consider a Bermudan option with strike price K and a set of N exercise moments $t_1, \dots, t_N$, and strictly ordered, $0 = t_0 < t_1 < \cdots < t_N = T$, where T is the option's maturity. When the option is exercised at time t_n, the holder receives a payoff $g(X_{t_n})$, where $\{X_t\}_{t=0}^T$ is the underlying log-asset price process. The value $v(x, t_0)$ of the Bermudan option at time t_0 is then given by,

$$v(x, t_0) = \max_{\tau \in \mathscr{T}} \mathbb{E}\left[e^{-r(\tau - t_0)} g(X_\tau) \,\middle|\, X_0 = x \right], \tag{40}$$

where τ is a stopping time taking values in $\mathscr{T} = \{t_0, t_1, \dots, t_N\}$. We apply Bellman's optimality principle, also known as the dynamic programming principle, stating that if one follows an optimal exercise strategy up to some observation time, then, given this information, it remains optimal to use it after that observation time. By the dynamic programming principle, one can split the optimization problem into two parts. The optimal exercise point may be found at some time θ, given the current state X_θ. Then, the expected value in (40) is maximized over all exercise strategies in $[\theta, T]$. In continuous time, the dynamic programming principle leads to the well known Hamilton–Jacobi–Bellman equation.

In the context of Bermudan option pricing, the dynamic programming principle states that the price of the option at any exercise moment is the maximum of the spot payoff and the so-called continuation value.

Between two exercise moments, the valuation process can be regarded as a European option pricing problem, and can be priced with the help of the risk-neutral option valuation formula (13).

For simplicity of notation, we use an equidistant time grid $\Delta t_n := t_n - t_{n-1} = \Delta t$ and we define, $x := X_{t_{n-1}} = \log(S_{t_{n-1}}/K)$, and $y := X_{t_n} = \log(S_{t_n}/K)$, where S_t is price process of the underlying asset. The payoff of the option is denoted by $g(y)$ and for vanilla options, the value of the option at maturity is given by,

$$v(y, T) = g(y) = [\alpha K(e^y - 1)]^+, \quad \alpha = \begin{cases} 1, & \text{for a call,} \\ -1, & \text{for a put.} \end{cases} \tag{41}$$

We consider again a constant risk-neutral rate r. By the dynamic programming approach, the option value prior to maturity can be expressed recursively for $n = N, N-1, \dots, 2$, by,

$$\begin{cases} v(x, t_{n-1}) = \max(g(x), c(x, t_{n-1})), \\ c(x, t_{n-1}) = e^{-r\Delta t} \int_{\mathbb{R}} v(y, t_n) f(y|x)\, dy, \end{cases} \tag{42}$$

and finally followed by the option value at t_0,

$$v(x, t_0) = e^{-r\Delta t} \int_{\mathbb{R}} v(y, t_1) f(y|x)\, dy, \tag{43}$$

where $c(x, t_{n-1})$ is referred to as the continuation value and the probability function of y given x is denoted by $f(y|x) := f_{\Delta t}(y|x)$.

The integrals in (42) and (43) are of the same form as the European pricing formula (13), and we can apply the SWIFT pricing formula to approximate them.

4.1 SWIFT Bermudan algorithm

At the initial time t_0, the option value is given by the integral representation in (43), and by application of the SWIFT pricing formula (24), we obtain,

$$v(x, t_0) \approx v^*(x, t_0) = e^{-r\Delta t} \sum_{k=1-\kappa}^{\kappa} D^*_{m,k}(x) V_{m,k}(t_1), \tag{44}$$

where the *density coefficients* $D^*_{m,k}(x)$ are as in (19). Thus, the option value can be determined once the *value coefficients* $V_{m,k}(t_1)$ are known. We propose a backward recursion to recover these coefficients, based on (42). We describe the approach by pricing a vanilla Bermudan put option.

Following (41), the option value at maturity $t_N = T$ equals the payoff of the option, $v(y, T) = g(y)$, and thus we can write the value coefficients for a put option as,

$$\begin{aligned} V_{m,k}(t_N) = V_{m,k}(T) &:= \int_{|y|\le c} v(y, T)\phi_{m,k}(y)\, dy \\ &= \int_{-c}^{0} g(y)\phi_{m,k}(y)\, dy \\ &= G_{m,k}(-c, 0), \end{aligned} \tag{45}$$

where $G_{m,k}(y_1, y_2)$ are the *payoff coefficients* over the exercise region (y_1, y_2), as we saw before in the European case, given in (26), which can be efficiently computed using the FFT.

Now that we have an expression for the value coefficients at maturity, we can compute the coefficients at any time t_n, for $n = N-1, \ldots, 1$, prior to maturity recursively, when we know the *early-exercise point* x_n^*, which is the point where the continuation value equals the payoff, i.e., $c(x_n^*, t_n) = g(x_n^*)$. Once we obtain x_n^*, we can split the integral for the *value coefficients* $V_{m,k}$ in two parts, giving,

$$\begin{aligned}
V_{m,k}(t_{n-1}) &:= \int_{|x|\le c} v(x,t_{n-1})\phi_{m,k}(x)\,dx \\
&= \int_{|x|\le c} \max\Big\{g(x),\ c(x,t_{n-1})\Big\}\phi_{m,k}(x)\,dx \\
&= \int_{-c}^{x_n^*} g(x)\phi_{m,k}(x)\,dx + \int_{x_n^*}^{c} c(x,t_{n-1})\phi_{m,k}(x)\,dx \\
&=: G_{m,k}(-c,x_n^*) + C_{m,k}(x_n^*,c,t_{n-1}),
\end{aligned} \tag{46}$$

where $C_{m,k}(x_1,x_2,t_{n-1})$ are the *continuation coefficients* at time t_{n-1} over the interval (x_1,x_2).

Theorem 2 *The continuation coefficients $C_{m,k}(x_1,x_2,t_{n-1})$ can be efficiently approximated with the use of five times the FFT when the value coefficients at the next time step $\{V_{m,k}(t_n)\}_k$ are known.*

The proof of Theorem 2 is given in Appendix 3.

Remark 4 (*Early-exercise point*) Each time step t_n, we have to determine the early-exercise point x_n^*, which is the x value that solves $g(x) = c(x,t_n)$. We use Newton's method as we know the payoff function $g(x)$ explicitly, and the continuation value in functional form in (65). The coefficients $\tilde{U}_j(t_n)$ here are independent of x, and can be reused for computation of the continuation coefficients, thus Newton's method consists only of a few $\mathscr{O}(J)$ operations per iteration.

The above theorem leads to the SWIFT method for Bermudan options, and is summarized in Algorithm 1. The method uses 5 times the Fast Fourier Transform per time step, which is the same number of times as the COS method [9], but the vectors we use are about two times longer when the same domain is chosen.

Remark 5 One could improve robustness by adding a check to see if the domain truncation c is sufficient. This can be done by evaluating $T_1(c)$ in (34). When using equidistant time steps Δt, this check has to be performed only once at $\mathscr{O}(N\log N)$ cost.

Remark 6 As highlighted in Sect. 3.5, when recovering the option price for x in a range $[-c,c]$, the integration domain of the value coefficients as in (46), should be set larger than this range. However, when the integration range of an integral is extended, while the number of discretization points J is the same, accuracy decreases. Therefore, we slightly extend the integration range to $\tilde{c} := \frac{3}{2}c$, so that we have minimal loss of accuracy, but the error is constant on the whole domain, as confirmed by numerical results in Sect. 5.

Remark 7 (*Error Convergence*) From Sect. 3.6, we know that the SWIFT pricing formula is closely related to the COS method, and the same is true for Bermudan options if we choose the parameters as mentioned in that section. It is proven [9,16] that the COS method exhibits exponential convergence for Bermudan options when the density function is smooth. We show numerically that the same is true for the SWIFT method.

Algorithm 1: SWIFT method for Bermudan options

```
Initialization:
- Select the value for $m$ such that $\bar{\varepsilon}_m < TOL$ in (32);
- Determine $c$ by the cumulants as in (33);
- Set $\kappa := \lceil 2^m c \rceil$, and $J := \lceil \pi\kappa \rceil$;
- Set $\tilde{c} := \frac{3}{2}c$;
At maturity $t_N = T$:
- Compute $V_{m,k}(t_N) = G^*_{m,k}(-\tilde{c}, 0)$, where $G^*_{m,k}$ is computed with the FFT as in (26).
for $n = N-1, \ldots, 1$ do
    Early-exercise point:
    - Construct $\tilde{U}_j(t_{n+1})$ from $\{V_{m,k}(t_{n+1})\}_k$ as in (66) using the FFT;
    - Run 5 iterations of Newton's method to find $x^*_n$, see Remark 4.
    Continuation coefficients:
    - Construct $\mathscr{I}_q(x^*_n, \tilde{c}, t_{n+1})$ as in (71) from $\tilde{U}_j(t_{n+1})$ using the efficient
      Hankel matrix product of Appendix 1 by use of 3 times the FFT;
    - Construct $C^*_{m,k}(x^*_n, \tilde{c}, t_n)$ as in (70) from $\mathscr{I}_q(x^*_n, \tilde{c}, t_{n+1})$ using the FFT;
    Value coefficients:
    - Set $V_{m,k}(t_n) = G^*_{m,k}(-\tilde{c}, x^*_n) + C^*_{m,k}(x^*_n, \tilde{c}, t_n)$.
end
Recover the option value $v(x, t_0)$ at $t = t_0$ by plugging $V_{m,k}(t_1)$ into (44).
```

4.2 Quick SWIFT

When pricing Bermudan options with many exercise moments, a high wavelet scale m is required to recover the peaked density function accurately. However, the payoff function does not need such a high wavelet scale to accurately be recovered, as it is relatively smooth. The main recursion back in time is on the payoff coefficients in the SWIFT method of Algorithm 1. This suggests that a cheaper approximation of the payoff function would be beneficial.

We propose a very cheap approximation of the payoff function here, at cost of some accuracy. From Corollary 2, it follows that when a function g is band-limited, its wavelet coefficients are given by $G_{m,k} := \langle g, \phi_{m,k} \rangle = 2^{-\frac{m}{2}} g(2^{-m}k)$. Equality does not hold for non band-limited functions, as are the payoff functions we encounter, but it serves as a very cheap approximation.[8]

We price Bermudan options using the same recursion on the payoff coefficients as the SWIFT method, but now with the quick approximation to compute coefficients instead of the approximation used before in (11). Similar to (45), we start at maturity t_N, where the option value equals the payoff, which we approximate,

$$V_{m,k}(t_N) = G_{m,k} \approx 2^{-\frac{m}{2}} g\left(2^{-m}k\right) =: V^Q_{m,k}(t_N). \tag{47}$$

[8] Note that this approximation can be seen as the approximation of the sinc-function by the Dirac-delta, which holds in the limit when $m \to \infty$.

Then, at any time t_n prior to maturity, we can recursively recover the payoff coefficients,

$$\begin{aligned} V_{m,k}(t_{n-1}) &= \int_{\mathbb{R}} v(x, t_{n-1})\phi_{m,k}(x)\,dx, \\ &\approx 2^{-\frac{m}{2}} v(2^{-m}k, t_{n-1}) \\ &= 2^{-\frac{m}{2}} \max\left\{ g(2^{-m}k),\ c(2^{-m}k, t_{n-1}) \right\} \\ &=: V^{Q}_{m,k}(t_{n-1}). \end{aligned} \tag{48}$$

Here, the only unknown quantity is the continuation value $c(2^{-m}k, t_{n-1})$, which is a European option pricing problem, and by the application of the SWIFT method for European options (22) we find,

$$\begin{aligned} c(2^{-m}k, t_{n-1}) &\approx e^{-r\Delta t} \sum_{p=1-\kappa}^{\kappa} D^{*}_{m,p} \int_{\mathbb{R}} v(y, t_n)\phi_{m,p}(y - 2^{-m}k)\,dy \\ &\approx e^{-r\Delta t} \sum_{p=1-\kappa}^{\kappa} D^{*}_{m,p} 2^{-\frac{m}{2}} v\Big(2^{-m}(p+k),\ t_n\Big) \\ &= e^{-r\Delta t} \sum_{p=1-\kappa}^{\kappa} D^{*}_{m,p} V^{Q}_{m,p+k}(t_n). \end{aligned} \tag{49}$$

In the first step, we used that wavelets satisfy $\phi_{m,k}(x - 2^{-m}p) = \phi_{m,k+p}(x)$. Furthermore, this last summation can be written as a matrix-vector product, with a Hankel matrix, which we can efficiently compute using three times the FFT, as shown in Appendix 1. In case of equidistant time steps, the payoff coefficients $D^{*}_{m,k}$, computed as before in (19), have to be computed only once, and thus we require only two times the FFT per iteration, as shown in Algorithm 2.

The efficient evaluation of the continuation value makes this method more than twice as fast as the COS method with the same number of coefficients and more than four times faster than the SWIFT method with the same wavelet scale m.

Numerical results in the next section show that for engineering accuracy approximations up to 10^{-5}, the Quick SWIFT method is generally faster, however, due to its linear convergence, it is unsuitable for high-accuracy approximations.

Remark 8 Algorithm 2 returns the option value at time t_0 for a range of initial asset values $x = 2^{-m}k$, with $k \in \mathbb{Z}$. However, often $x := \log(S_0/K)$ is not of this form. Therefore, the alternative log-transform $X_t := \log(S_t/S_0)$ should be used, implying that $x := X_0 = 0$.

Algorithm 2: Quick SWIFT method for Bermudan options

```
Initialization:
- Select the value for $m$ such that $\bar{\varepsilon}_m < TOL$ in (32);
- Determine $c$ by the cumulants as in (33);
- Set $\kappa := \lceil 2^m c \rceil$, and $J := \lceil \pi\kappa \rceil$;
- Set $\tilde{c} := \frac{3}{2}c$;
At maturity $t_N = T$:
- Compute $V^Q_{m,k}(t_N)$ from the payoff function $g$ as in (47);
- Compute the payoff coefficients $D^*_{m,k}$ of (19) using the FFT;
- Construct $\mu_h := \mathrm{DFT}(\mathbf{m}_h)$ from $D^*_{m,k}$ as in (55);
for $n = N-1, \ldots, 1$ do
    - Construct the vector $\mathbf{x}_h$ in (55) from $V^Q_{m,k}(t_{n+1})$;
    - Compute $c(2^{-m}k, t_n)$ with the FFT from $\mathbf{x}_h$ and $\mu_h$ as in Appendix 1;
    - Compute $V^Q_{m,k}(t_n) = 2^{-\frac{m}{2}} \max\left\{g(2^{-m}k),\ c(2^{-m}k, t_n)\right\}$.
end
Recover the option value $v(x, t_0)$ at $t = t_0$ by plugging $V^Q_{m,k}(t_1)$ into (44).
```

4.3 Discretely-monitored barrier options

Discretely-monitored barrier "out" options are options that cease to exist if the asset price hits a certain barrier level, B, at one of the pre-specified observation dates. If $B > S_0$, the option is referred to as "up-and-out", and "down-and-out" otherwise.

The payoff for an up-and-out option reads,

$$v(x, T) = \left[\{\alpha(S_T - K)\}^+ - Rb\right] \mathbb{1}_{\{S_{t_n} < B\}} + Rb, \tag{50}$$

where $\alpha = 1$ for a call and $\alpha = -1$ for a put, Rb is a rebate and $\mathbb{1}_A$ is the indicator function, taking value one whenever A is not empty, and zero otherwise. Let the set of observation dates be $t_1 < \cdots < t_{N-1} < t_N = T$. Then, the price of an up-and-out option, monitored N times, satisfies the following recursion,

$$\begin{cases} c(x, t_{n-1}) &= e^{-r\Delta t} \int_{\mathbb{R}} v(x, t_n) f(y|x)\, dy, \\ v(x, t_{n-1}) &= \begin{cases} e^{-r(T-t_{n-1})} Rb, & x \geq b, \\ c(x, t_{n-1}), & x < b, \end{cases} \end{cases} \tag{51}$$

where $b := \log(H/K)$ and $n = N, N-1, \ldots, 2$.

This approach is very similar to the recursion for Bermudan options, with the main difference being that for barrier options the barrier point is known in advance, while the early-exercise point for Bermudans has to be found by a root-searching algorithm.

Theorem 3 (Backward recursion for discrete barrier options) *By the backward recursion, the following numerical approximation is found for discretely monitored up-and-out barrier options. At any monitoring date prior to maturity $n = N, N-1, \ldots, 1$, we obtain,*

$$V^*_{m,k}(t_n) = C^*_{m,k}(-c, h, t_n) + e^{-r(T-t_{n-1})} R^*_{m,k}(h, c), \tag{52}$$

where $C^*_{m,k}(x_1, x_2, t_n)$ is as in (70) and the rebate coefficient $R_{m,k}(h, c)$ defined as, $Rb^*_{m,k}(x_1, x_2) := Rb \int_{x_1}^{x_2} \phi^*_{m,k}(x)\, dx$, which can be computed using the FFT similar to the payoff coefficients in (26). Let $b^+ := \max\{0, b\}$ and $b^- := \min\{b, 0\}$, then we have at maturity $t_N = T$,

$$V^*_{m,k}(t_N) = \begin{cases} G^*_{m,k}(0, b^+) + Rb^*_{m,k}(b, c), & \textit{for a call,} \\ G^*_{m,k}(-c, b^-) + Rb^*_{m,k}(b, c), & \textit{for a put.} \end{cases} \tag{53}$$

In a similar fashion, we can price down-and-out barrier options, barrier "in"-options and double-barrier options with the same ease.

The proof of this theorem goes along the same lines as the recursion we found for Bermudan options. The main difference is that the computation of $C^*_{m,k}(-c, b, t_n)$ is less expensive as b is known in advance, and many computations, like $Rb^*_{m,k}(b, c)$, can be done outside the main loop.

Algorithm 3: SWIFT method for up-and-out barrier options

```
Initialization:
- Select the value for $m$ such that $\bar{\varepsilon}_m < TOL$ in (32);
- Determine $c$ by the cumulants as in (33);
- Set $\kappa := \lceil 2^m c \rceil$, and $J := \lceil \pi\kappa \rceil$;
- Set $\tilde{c} := \frac{3}{2}c$;
At maturity $t_N = T$:
- Compute $Rb^*_{m,k}(b, \tilde{c})$ as in Theorem 3.
- Compute $V^*_{m,k}(t_N)$ as in (53) where $G^*_{m,k}$ is computed with the FFT as in (26).
for $n = N-1, \ldots, 1$ do
    Continuation coefficients:
    - Construct $\mathscr{I}_q(-\tilde{c}, b, t_{n+1})$ as in (71) from $\tilde{U}_j(t_{n+1})$ using the efficient Hankel matrix
      product of Appendix 1 by use of 3 times the FFT;
    - Construct $C^*_{m,k}(-\tilde{c}, b, t_n)$ as in (70) from $\mathscr{I}_q(-\tilde{c}, b, t_{n+1})$ using the FFT;
    Value coefficients:
    - Set $V^*_{m,k}(t_n) = C^*_{m,k}(-\tilde{c}, b, t_n) + e^{-r(T-t_{n-1})} Rb^*_{m,k}(b, \tilde{c})$.
end
Recover the option value $v(x, t_0)$ at $t = t_0$ by plugging the coefficients $V^*_{m,k}(t_1)$ into (44).
```

5 Numerical results

In this section, we illustrate the performance of the SWIFT method for option types with early exercise features. All tests are run with Matlab 2016a on an Intel Core i7-4790 CPU @ 3.60 GHz with 16 GB of memory. We compare the SWIFT method against the COS method and the Quick SWIFT method for different option types

Table 3 Test parameters for pricing Bermudan options

Test no.	Model	S_0	K	T	r	Other parameters
1	GBM	100	110	1	0.1	$\sigma = 0.2$
2	CGMY	100	80	1	0.1	$(\bar{C}, \bar{G}, \bar{M}, \bar{Y}) = (1, 5, 5, 1.5)$
3	CGMY	100	100	1	0.1	$(\bar{C}, \bar{G}, \bar{M}, \bar{Y}) = (1, 5, 5, 0.5)$

Characteristic exponents of these Lévy models can be found in Table 1. For details, we refer the reader to [17]

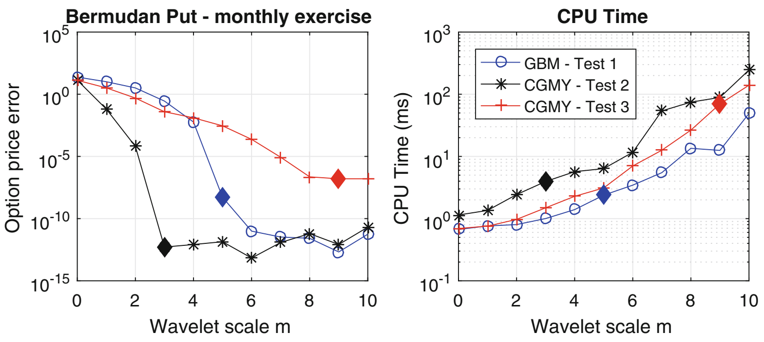

Fig. 3 Convergence of the Bermudan put price with 12 exercise moments under different dynamics with respect to the wavelet scale m. Domain truncated by the cumulants in (33) with $L = 6$. The diamonds denote the recommended wavelet scale by the analytic formula (32)

on different underlying price processes. The computational complexity of all of the methods is $\mathcal{O}(NJ \log J)$, where N is the number of early-exercise moments and J is the number of coefficients, see Sect. 5.2. The reference prices for our tests, if not available analytically, are computed by the COS method with $J_{COS} = 20{,}000$ Fourier terms and domain trucation by cumulants with $L = 50$, see (33).

We start by analyzing convergence behavior in terms of the wavelet scale m, and confirm numerically that the SWIFT method exhibits exponential convergence with respect to 2^m. We price a Bermudan put option with underlying dynamics and parameters given in Table 3. Convergence results are shown in Fig. 3 for a Bermudan put with $N = 12$ (monthly) exercise dates.

We determined *a priori* the required wavelet scale m to obtain an error $<10^{-10}$ by application of the analytic result in (32), which we denoted by a diamond in Fig. 3. This simulation confirms the analytic formula.

For a fixed wavelet scale, Test 1, the GBM model, is optimal in terms of CPU time, which we see in the right sub figure, as it has fast decay in both the asset domain and the Fourier domain. Test 2, the CGMY model with $\bar{Y} = 1.5$ has the fattest tails and thus a wide domain in the asset-space is required, but due to a fast decay in the Fourier domain, $m = 3$ already results in machine accuracy. The contrary is true for Test 3, CGMY with $\bar{Y} = 0.5$, which is a highly peaked density function, and it requires $m = 8$ to reach machine accuracy.

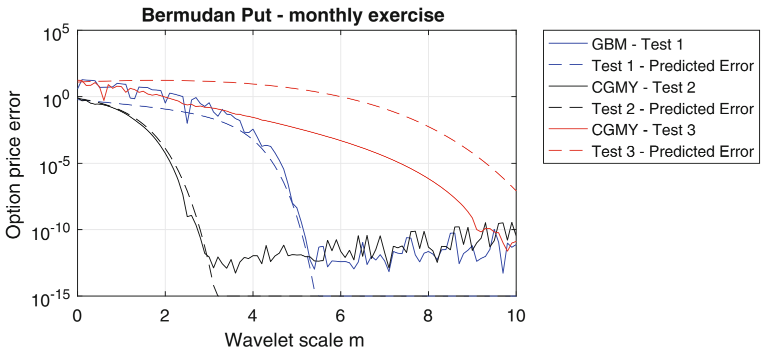

Fig. 4 The three tests from Table 3 with $N = 12$ exercise moments. We plotted the error bound along with the true numerical price error

5.1 Wavelet scale determination

We test the analytic bound that we determined in Sect. 3.3. The bound is not exact as we substitute the upper bound (30) back and forth in the expression for the error (32). In Fig. 4, we see that the bound is very tight for both fat-tailed distributions, the GBM and CGMY with $Y = 1.5$. For the peaked CGMY model of Test No. 3, the bound is an overestimation of the true error. This can be explained by the fact that we neglect a division of d as a divisor in (32), which is larger for Test No. 3.

5.2 Quick SWIFT and CPU time

Convergence of the Quick SWIFT is demonstrated with the two CGMY tests from Table 3 with weekly ($N = 50$) exercise moments. In the left subfigure of Fig. 5, Test No 2. is used, which has a smooth density function which is easy to approximate with the SWIFT and COS methods. We observe that the Quick SWIFT method performs less, and is not able to reach the threshold of an error of 10^{-10}, although it still outperforms the SWIFT and COS methods for low-accuracy estimates.

The right sub figure of Fig. 5 shows the results for the highly peaked Test No. 3. We observe that the Quick SWIFT method is better than SWIFT, although the rate of convergence of the Quick SWIFT method decreases around 10^{-6}. This is where the wavelet scale m is high enough such that the approximation error of the density coefficients is small and the error in the ‘quickly’ approximated payoff coefficients dominate.

The reason is that in Test No. 3, the error made in the approximation of the density function dominates, so the impact of the ‘badly’ approximated payoff coefficients is negligible. The small time between exercise moments and the naturally peaked density function cause that wavelet scale $m = 11$ is required for SWIFT for an option pricing error less than 10^{-5} according to (32). The COS method uses $J_{COS} = 2^m(b - a) = 3 \cdot 2^m$ coefficients for a fair comparison (see Sect. 3.6).

Theoretically, the SWIFT method is twice as slow as the COS method when the same number of coefficients is used. We observe this in the right sub-figure of Fig. 5

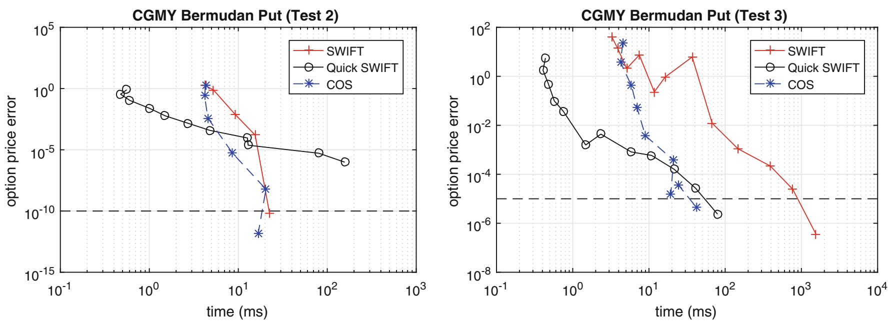

Fig. 5 CGMY Bermudan put with weekly ($N = 50$) exercise moments and parameters as in Test no. 3 in Table 3

for large m (for $m \leq 5$, initialization-time dominates). Furthermore, the Quick SWIFT method is 2.5 times faster than SWIFT, as expected, since Quick SWIFT only uses two times the FFT, compared to five times in the normal SWIFT method.

5.3 Domain boundary error and recursion

In Sect. 3.5, we demonstrated the advantage of the SWIFT method with respect to the boundary. We show a similar example for a Bermudan option with $N = 20$ exercise dates. The boundary error plays a big role in Bermudan option pricing due to the recursive pricing, which has the potential to blow up small errors at the boundary to significant errors within the domain. This is shown in Fig. 6. There, a Bermudan put option is priced under geometric Brownian motion dynamics and parameters as in Test no.1 in Table 3. We choose a relatively small domain using the cumulants method (33) and $L = 3$ for both methods.

We can see that the COS method has a pricing error for negative x, similar to the European case, due to periodicity of the Fourier transform, as the payoff function of a put option is non-zero there. Furthermore, on the positive side of the domain, an error recursion occurs, which becomes significant after about 6 time steps, and increases further at every iteration towards $t = 0$.

The SWIFT method allows us to place the domain truncation outside of the domain of computation, decreasing the error recursion as much as possible. For a fair comparison, we have kept the number of coefficients the same. This results in a loss of accuracy on the positive side of the x domain, and we see that the SWIFT pricing error is no longer machine accuracy, but around 10^{-12}. This error will become more apparent when the domain is chosen larger or the number of exercise moments increases.

5.4 Barrier options

We consider monthly-monitored ($N = 12$) up-and-out call and put options (UOC) and (UOP), down-and-out call and put options, (DOC) and (DOP), with up barrier $B^{up} = 120$ and the down barrier $B_{down} = 80$, without rebates. Reference method is

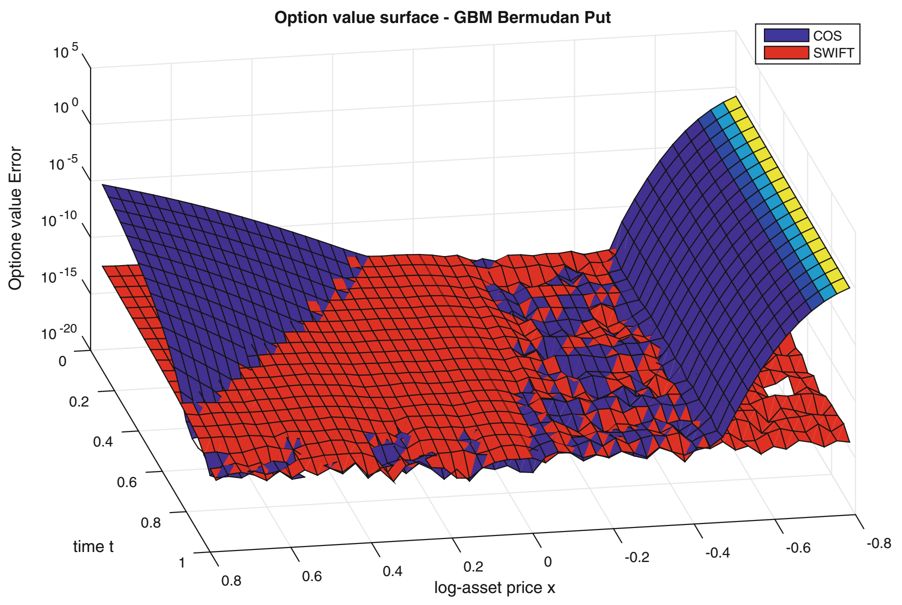

Fig. 6 Option value surface for a Bermudan put with $N = 20$ exercise dates, both priced on a small domain with $L = 3$. We see that the COS method (*blue*) suffers from boundary issues, which are minimal for the SWIFT method (*red*) (color figure online)

Table 4 Test parameters for pricing barrier options

Test no.	Model	S_0	K	T	r	q	Other parameters
4	CGMY	100	100	1	0.05	0.02	$(\bar{C}, \bar{G}, \bar{M}, \bar{Y}) = (4, 50, 60, 0.7)$
5	NIG	100	100	1	0.05	0.02	$\alpha = 15, \beta = -5, \delta = 0.5$

Characteristic exponents of these Lévy models can be found in Table 1. For details, we refer the reader to [17]

the COS method [9]. The test parameters we use are also from [9], and are shown in Table 4, where q is the dividend yield.

The computation of option values for barrier options is faster than for Bermudan options, as the barrier is known in advance, in contrast to the early-exercise point, and we observe in Fig. 7 accurate prices are computed within milliseconds. The down-and-out put (DOP) and the up-and-out call (UOC) have a bounded payoff domain, which results in a higher accuracy as no artificial truncation is required. Furthermore, the NIG model with parameters as in Table 4 has a a high-peaked density function, thus a higher wavelet scale is required.

6 Conclusion and discussion

In this paper, we examined the SWIFT method for pricing European options [15]. We gave a complete proof of exponential convergence with respect to the wavelet

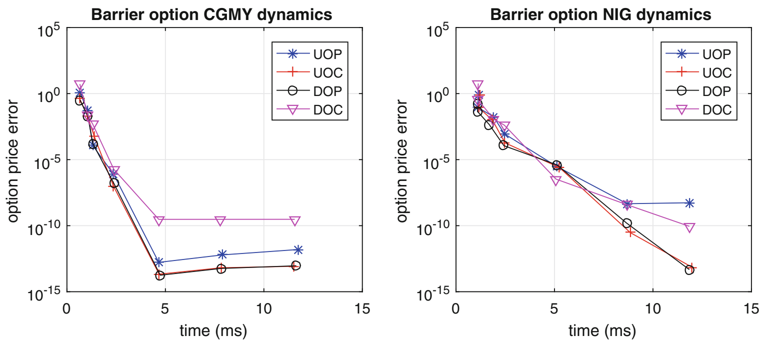

Fig. 7 Barrier options priced with the SWIFT method. Parameters from Table 4. Wavelet scales $m = 2, \ldots, 8$ are shown. Domain truncated by (33) and $L = 6$

scale and gave an analytic argument to determine a suitable approximation scale m. Determination of the appropriate domain truncation can then be carried out recursively. Furthermore, we showed a close relation to the COS method [8] for specific parameter choices and we highlighted the advantage of the SWIFT method with respect to the domain boundary for multiple strike pricing.

The main contribution of this paper is the extension of the SWIFT method for pricing Bermudan options and discretely monitored barrier options under Lévy dynamics. The SWIFT method for options with early-exercise features is competitive to the state-of-the-art methods like the COS method [16], both with a computational complexity of $\mathscr{O}(NJ\log J)$, where J is the number of coefficients and N the number time steps. The advantage of our method is preservation of the exponential convergence of the COS method, while being able to solve boundary issues recursively. This is particularly useful for smooth density functions and options with a long time to maturity.

The difficulty of this method is that the sinc function has to be approximated in order to compute its integral, and a global artificial truncation of the integration range is required if one want to apply the FFT. We introduced a second approach, the Quick SWIFT method, based on the observation that the sinc-function converges to the Dirac-delta when the wavelet scale m goes to infinity. This approach is very quick, but resulting convergence of the error in the option price is linear. If only engineering accuracy is required, this method is preferable over the COS method, especially when the density function contains high-frequency moments, which happens for peaked densities when for example the time between exercise moments is small.

As for possible extensions, it is interesting to apply the SWIFT method to higher dimensions and other option types or exploring different approximations of the sinc integral to balance between accuracy and computational time. We leave these for future research.

Appendix 1: The fast Fourier transform

The SWIFT Bermudan method relies on the application of the Fast Fourier Transform (FFT) for fast computations. Point of departure is the discrete Fourier transform (DFT), which is defined analogously to the continuous Fourier transform, but only on a set of discrete frequencies ω.

Definition 3 (*DFT*) Let $\mathbf{z} \in \mathbb{C}^N$ with entries $\mathbf{z} = \{z_j\}_{j=1}^N$. Then the discrete Fourier transform is defined as,

$$\mathrm{DFT}_k(\mathbf{z}) = \sum_{j=1}^{N} z_j e^{-\frac{2\pi i}{N}(k-1)(j-1)}, \text{ where } k = 1 - N/2, \ldots, N/2. \tag{54}$$

Remark 9 There are different definitions regarding the DFT, but we use the one as implemented in `Matlab` to reduce the differences between notation and implementation. Expression (54) is equivalent to `Z = fftshift(fft(z))` in `Matlab`-code.

We discuss Hankel matrices that have a special structure, which allows us to compute matrix-vector products with a complexity of $\mathcal{O}(N \log_2 N)$ instead of $\mathcal{O}(N^2)$. This is in detail described in [9].

Hankel matrix multiplication

A *Hankel matrix* $\mathcal{M}$ is an $N \times N$ matrix with constant anti-diagonals, i.e.,

$$\mathcal{M} := \begin{bmatrix} m_0 & m_1 & m_2 & \cdots & m_{N-1} \\ m_1 & m_2 & \cdots & m_{N-1} & m_N \\ \vdots & & \iddots & & \vdots \\ m_{N-2} & m_{N-1} & \cdots & \cdots & m_{2N-3} \\ m_{N-1} & \cdots & \cdots & m_{2N-3} & m_{2N-3} \end{bmatrix}.$$

For a vector $\mathbf{x} \in \mathbb{R}^N$, the matrix-vector product $\mathcal{M}\mathbf{x}$ is equal to the first N elements of the circular convolution $\mathbf{m}_h \circledast \mathbf{x}_h$, with the $2N$-vectors,

$$\begin{aligned} \mathbf{m}_h &:= [m_0, m_{-1}, m_{-2}, \cdots, m_{1-N}, 0, m_{N-1}, m_{N-2}, \cdots, m_1]^T, \\ \mathbf{x}_h &:= [x_0, x_1, x_2, \cdots, x_{N_1}, 0, \cdots, 0]^T. \end{aligned} \tag{55}$$

A circular convolution of two vectors is equal to the inverse discrete Fourier transform (DFT^{-1}) of the product of the forward DFTs, DFT, i.e.,

$$\mathbf{x} \circledast \mathbf{y} = \mathrm{DFT}^{-1}\left(\mathrm{DFT}(\mathbf{x}) \cdot \mathrm{DFT}(\mathbf{y})\right).$$

Thus, in total 3 times the FFT algorithm has to be applied on a vector of length $2N$.

Appendix 2: Wavelet coefficient computation with the FFT

We show that we can benefit from the Fast Fourier Transform in the computation of the Shannon wavelet coefficients when we make use of the approximation of the sinc-function as in Lemma 4.

Recall that wavelet coefficients of an arbitrary function g are defined as $G_{m,k} := \left\langle g,\ \phi_{m,k}\right\rangle$, and by approximating the wavelet $\phi_{m,k}$ by $\phi^*_{m,k}$ we obtain,

$$\begin{aligned} G^*_{m,k} &:= \int_{\mathbb{R}} g(y)\overline{\phi^*_{m,k}(y)}\,dy \\ &= \frac{2^{\frac{m}{2}}}{J}\mathrm{Re}\left\{\sum_{j=1}^{J}\left(\int_{\mathbb{R}} g(y)e^{-i\omega_j 2^m y}\,dy\right)e^{i\omega_j k}\right\} \\ &=: \frac{2^{\frac{m}{2}}}{J}\mathrm{Re}\left\{\sum_{j=1}^{J} g_j e^{i\omega_j k}\right\}. \end{aligned} \tag{56}$$

The inner integral g_j in the second step can be solved using the Fourier transform $\hat{g}$ whenever it exists, or it can be solved numerically, depending on the properties of the function g. To apply the FFT, recall that $\omega_j = \frac{\pi}{J}(j - \frac{1}{2})$. We define $N = 2J$ and apply zero padding to $g_j := 0$ for $j = J+1, \ldots, N$, to obtain,

$$G^*_{m,1-k} = \frac{2^{\frac{m}{2}}}{J}\mathrm{Re}\left\{e^{-i\frac{\pi}{N}(k-1)}\sum_{j=1}^{N} g_j e^{-i\frac{2\pi}{N}(j-1)(k-1)}\right\}, \tag{57}$$

which is in the form of the `Matlab` fft-function as in Definition 3 and the whole vector of coefficients $\{G_{m,k}\}_{k=1}^{N}$ can be computed at a computational cost of $\mathscr{O}(N\log N)$.

Remark 10 The FFT is used most efficient when the number of sub intervals J is a power of two, thus by setting $J = 2^\eta$, where $\eta \in \mathbb{N}$.

Appendix 3: Technical details

Proof of Corrolary 1

Fix a wavelet scale $m \in \mathbb{R}$ and recall from (6) that by construction, the projection onto V_m is given by $\mathscr{P}_m f(y) = \sum_{k\in\mathbb{Z}}\left\langle f,\ \phi_{m,k}\right\rangle\phi_{m,k}(y)$, where we can rewrite the coefficient $\left\langle f,\ \phi_{m,k}\right\rangle$ by application of Parseval's identity,

$$\left\langle f,\ \phi_{m,k}\right\rangle = \frac{1}{2\pi}\left\langle \hat{f},\ \hat{\phi}_{m,k}\right\rangle = \frac{2^{-\frac{m}{2}}}{2\pi}\int_{-2^m\pi}^{2^m\pi}\hat{f}(\omega)e^{i\omega k/2^m}\,d\omega.$$

When substituting this expression of the density coefficients in the projection $\mathscr{P}_m f$, we obtain by interchanging integration and summation,

$$\mathscr{P}_m f(y) = \frac{2^{-\frac{m}{2}}}{2\pi}\int_{-2^m\pi}^{2^m\pi}\hat{f}(\omega)\left[\sum_{k\in\mathbb{Z}}\phi_{m,k}(y)e^{i\omega k/2^m}\right]d\omega. \tag{58}$$

It follows now from the Fourier series expansion of $e^{i\omega y}$ in y that,

$$\sum_{k\in\mathbb{Z}} \phi_{m,k}(y)e^{i\omega k/2^m} = 2^{\frac{m}{2}} e^{i\omega y}, \quad \text{when,} \quad \omega \in (-2^m\pi, 2^m\pi), \tag{59}$$

see [19]. Substituting (59) into (58) yields the desired result. □

Proof of Lemma 6

If f is a band-limited function, it follows from Corollary 2 that the density coefficients are given by $D_{m,k}(x) = 2^{-\frac{m}{2}} f(\frac{k}{2^m}|x)$. In our application, density f is not band-limited, but this motivates us to write $D_{m,k}(x) = 2^{-\frac{m}{2}}[f(\frac{k}{2^m}|x) - \varepsilon_m(\frac{k}{2^m}; x)]$, where $\varepsilon_m(y; x)$ is as in (9). Inserting this formulation of the density coefficients into f_2 of (18) results in,

$$f_2(y|x) = 2^{-\frac{m}{2}} \sum_{k=1-\kappa}^{\kappa} \left[f(\tfrac{k}{2^m}|x) - \varepsilon_m(\tfrac{k}{2^m}; x)\right] \phi_{m,k}(y). \tag{60}$$

The difference between f_2 and f can be expressed using (60), so that we obtain,

$$f_2(y|x) - f(y|x) = 2^{-\frac{m}{2}} \sum_{k=1-\kappa}^{\kappa} f\left(\tfrac{k}{2^m}|x\right) \phi_{m,k}(y) - f(y|x) - 2^{-\frac{m}{2}} \sum_{k=1-\kappa}^{\kappa} \varepsilon_m(\tfrac{k}{2^m}; x)\phi_{m,k}(y).$$

The right-side sum is a finite summation of which each term can be bounded by noting that $\left|\phi_{m,k}(y)\right| \leq 2^{\frac{m}{2}}$ and $|\varepsilon_m(y; x)| \leq H(2^m\pi)$, see Lemma 3, where $H(\omega)$ is the mass in the tails of the Fourier transform as in (8) Thus we obtain,

$$|f_2(y|x) - f(y|x)| \leq \left|\sum_{k=1-\kappa}^{\kappa} 2^{-\frac{m}{2}} f(\tfrac{k}{2^m}|x)\phi_{m,k}(y) - f(y|x)\right| + (2\kappa + 1)H(2^m\pi). \tag{61}$$

The part in absolute signs is a truncated sinc approximation, and convergence is proven in [19, Theorem 1.3.5, eqn. (1.3.28)], and bounded by,

$$\left|\sum_{k=1-\kappa}^{\kappa} 2^{-\frac{m}{2}} f(\tfrac{k}{2^m}|x)\phi_{m,k}(y) - f(y|x)\right| \leq 2H(2^m\pi) + \sum_{|k|>\kappa} f(\tfrac{k}{2^m}|x). \tag{62}$$

This remaining summation can be interpreted as a Riemann-sum over the tails of the density function. We assume monotonic decay of the density function, and we consider the left and right tails separately. Then, for the left tail, if we interpret the summation as a left Riemann sum, it is bounded by 2^m times the integral of $f(y|x)$ over $(-\infty, \frac{1-\kappa}{2^m})$. Similarly, a right Riemann sum is bounded from above by the integral over the right tail $(\frac{\kappa}{2^m}, \infty)$. Thus, we find,

$$\sum_{|k|>\kappa} f(\tfrac{k}{2^m}|x) \leq 2^m \tau(\tfrac{\kappa}{2^m}). \tag{63}$$

If we summarize the results from (61)–(63), the desired result follows immediately. □

Proof of Lemma 7

We use the definition of f_2 and f_3 in respectively (18) and (20) so that,

$$\begin{aligned}|\varepsilon_3(J)| &= \left| \sum_{k=1-\kappa}^{\kappa} \left(D_{m,k}(x) - D^*_{m,k}(x)\right) V_{m,k} \right| \\ &\leq 2^{\frac{m}{2}} (2\kappa+1) \|g\|_\infty \max_{|k|\leq\kappa} \left|D_{m,k}(x) - D^*_{m,k}(x)\right| \\ &\leq 2^m (2\kappa+1) \|g\|_\infty \left(2\tau(c) + \sqrt{2c}\,\|f\|_2 \frac{(\pi\kappa)^2}{(2J)^2 - (\pi\kappa)^2}\right),\end{aligned}$$

where the last step is a result of Lemma 2 in [15]. □

Proof of Theorem 1

We combine the results the error bounds derived in Lemmas 5–7. That is,

$$\begin{aligned}\frac{|v(x,t_0) - v_3(x,t_1)|}{\|g\|_\infty e^{-r\Delta t}} &= \frac{|\varepsilon_3(J) + \varepsilon_2(m,\kappa) + \varepsilon_0(c)|}{\|g\|_\infty e^{-r\Delta t}} \\ &\leq 2c\left[(2\kappa+3)H(2^m\pi) + 2^m\tau(\tfrac{\kappa}{2^m})\right] + \tau(c) \\ &\quad + 2^m(2\kappa+1)\left(2\tau(c) + \sqrt{2c}\,\|f\|_2 \frac{(\pi\kappa)^2}{(2J)^2 - (\pi\kappa)^2}\right),\end{aligned} \tag{64}$$

and when $J \geq \pi\kappa \geq 2^m\pi c$, the desired result follows. □

Proof of Theorem 2

We show that the continuation coefficients,

$$C_{m,k}(x_1, x_2, t_{n-1}) := \int_{x_1}^{x_2} c(x, t_{n-1})\phi_{m,k}(x)\,dx,$$

can be approximated by the SWIFT method for European options by a repeated use of Lemma 4.

As noted above, the continuation value $c(x, t_{n-1})$ in (42) resembles a European pricing option problem, which we can approximate once more with the SWIFT pricing formula. We use the formulation of the SWIFT method for multiple strikes (36), so that we obtain

$$c(x, t_{n-1}) \approx c^*(x, t_{n-1}) := e^{-r\Delta t} \sum_{j=1}^{J} \mathrm{Re}\left\{ \hat{f}\left(\omega_j 2^m\right) \tilde{U}_j(t_n) e^{-i\omega_j 2^m x} \right\}, \tag{65}$$

where the factor $\tilde{U}_j(t_n)$ is as (35), but due to the time recursion, it now depends on $V_{m,k}(t_n)$ so that it is given by,

$$\tilde{U}_j(t_n) := \frac{2^{\frac{m}{2}}}{J} \sum_{|p|\leq\kappa} V_{m,p}(t_n) e^{i\omega_j p}. \tag{66}$$

This expression can be efficiently constructed using the FFT as explained in Appendix 1. We use the same truncation $|p| \leq \kappa$ at each time step for simplicity of notation and to make optimal use of the FFT.

When substituting this SWIFT approximation of the continuation value (65) in the definition of the continuation coefficients (46), we obtain,

$$\begin{aligned} C_{m,k}(x_1, x_2, t_{n-1}) &:= \int_{x_1}^{x_2} c(x, t_{n-1}) \phi_{m,k}(x)\, dx \\ &\approx e^{-r\Delta t} \int_{x_1}^{x_2} \sum_{j=1}^{J} \mathrm{Re}\left\{ \hat{f}\left(\omega_j 2^m\right) \tilde{U}_j(t_n) e^{-i\omega_j 2^m x} \right\} \phi_{m,k}(x)\, dx \\ &= e^{-r\Delta t} \sum_{j=1}^{J} \mathrm{Re}\left\{ \hat{f}\left(\omega_j 2^m\right) \tilde{U}_j(t_n) \int_{x_1}^{x_2} \phi_{m,k}(x) e^{-i\omega_j 2^m x}\, dx \right\}, \end{aligned} \tag{67}$$

Thus, the continuation coefficients at time t_{n-1} can be recovered from the value coefficients at time t_n. The remaining step is the computation of integrals at the right hand side, which we do by replacing $\phi_{m,k}$ by $\phi^*_{m,k}$ as in Lemma 4, but we use the complex form,

$$\mathrm{sinc}^*(x) = \frac{1}{2J} \sum_{q=1-J}^{J} e^{it\omega_q}, \tag{68}$$

so that we will not be confused by two real-parts functions Re {} in one equation. Then, the integrals at the right hand side of (67) can be approximated by,

$$I^*_{j,k}(x_1, x_2) := \int_{x_1}^{x_2} \phi^*_{m,k}(x) e^{-i\omega_j 2^m x}\, dx = \frac{2^{\frac{m}{2}}}{2J} \sum_{q=1-J}^{J} \mathscr{M}_{j,q}(x_1, x_2) e^{i\omega_q k}, \tag{69}$$

where the integrals $\mathscr{M}_{j,q}(x_1, x_2)$ are defined as,

$$\mathscr{M}_{j,q}(x_1, x_2) := \int_{x_1}^{x_2} e^{-i(\omega_j+\omega_q)2^m x}\, dx.$$

These integrals can be solved analytically, and the solutions are given by,

$$\mathscr{M}_{j,q}(x_1,x_2)=\begin{cases} x_2-x_1, & \text{for } q=1-j,\\ i\dfrac{e^{-i(\omega_j+\omega_q)2^m x_2}-e^{-i(\omega_j+\omega_q)2^m x_1}}{(\omega_j+\omega_q)2^m}, & \text{else.}\end{cases}$$

Substituting the approximation $I^*_{j,k}(x_1,x_2)$ into (67) and changing the order of the summations yields,

$$C^*_{m,k}(x_1,x_2,t_{n-1}):=e^{-r\Delta t}\mathrm{Re}\left\{\sum_{q=1-J}^{J}\mathscr{J}_q(x_1,x_2)e^{i\omega_q k}\right\}, \tag{70}$$

where the coefficients $\mathscr{J}_q(x_1,x_2)$ are defined as,

$$\mathscr{J}_q(x_1,x_2):=\left(\sum_{j=1}^{J}\left[\hat{f}\left(\omega_j 2^m\right)\tilde{U}_j(t_n)\right]\mathscr{M}_{j,q}(x_1,x_2)\right). \tag{71}$$

This represents a matrix-vector product, where $\{\mathscr{M}_{j,q}\}_{j,q}$ is a Hankel matrix, as its values only depend on j and q through $(j+q)$. As described in Appendix 1, we can recover this matrix-vector product with a Hankel matrix by application of three times the FFT. Then finally, the summation over q in (70) can be once more computed using the FFT, see Appendix 2. □

References

1. Abrarov, S.M., Quine, B.M.: A rational approximation for efficient computation of the Voigt function in quantitative spectroscopy. J. Math. Res. **7**(2), 163–174 (2015)
2. Abrarov, S.M., Quine, B.M.: Sampling by incomplete cosine expansion of the sinc function: application to the Voigt/complex error function. Appl. Math. Comput. **258**, 425–435 (2015)
3. Broadie, M., Yamamoto, Y.: Application of the fast Gauss transform algorithm for pricing discrete path-dependent options. Manag. Sci. **8**(49), 1071–1088 (2003)
4. Carr, P., Madan, D.: Option valuation using the fast Fourier transform. J. Comput. Finance **2**, 61–73 (1999)
5. Cattani, C.: Shannon wavelets theory. Math. Probl. Eng. **2008**, 164–808 (2008)
6. Chui, C.K.: An Introduction to Wavelets. Academic Press, Cambridge (1992)
7. Daubechies, I.: Ten Lectures on Wavelets. Society for Industrial and Applied Mathematics (1992)
8. Fang, F., Oosterlee, C.W.: A novel option pricing method based on Fourier-cosine series expansions. SIAM J. Sci. Comput. **31**(2), 826–848 (2008)
9. Fang, F., Oosterlee, C.W.: Pricing early-exercise and discrete barrier options by Fourier-cosine series expansions. Numer. Math. **114**(1), 27–62 (2009)
10. Feng, L., Linetsky, V.: Pricing discretely monitored barrier options and defaultable bonds in Lévy process models: a fast Hilbert transform approach. Math. Finance **18**(3), 337–384 (2008)
11. Kirkby, J.L.: Robust barrier option pricing by frame projection under exponential Lévy dynamics. Working Paper (2014)
12. Kirkby, J.L.: Efficient option pricing by frame duality with the fast Fourier transform. SIAM J. Financ. Math. **6**(1), 713–747 (2015)
13. Mallat, S.: A Wavelet Tour of Signal Processing. Academic Press, Cambridge (2009)

14. Ortiz-Gracia, L., Oosterlee, C.W.: Robust pricing of European options with wavelets and the characteristic function. SIAM J. Sci. Comput. **35**(5), B1055–B1084 (2013)
15. Ortiz-Gracia, L., Oosterlee, C.W.: A highly efficient Shannon wavelet inverse Fourier technique for pricing European options. SIAM J. Sci. Comput. **38**(1), B118–B143 (2016)
16. Ruijter, M.J., Oosterlee, C.W., Aalbers, R.F.T.: On the Fourier cosine series expansion method for stochastic control problems. Numer. Linear Algebra Appl. **20**(4), 598–625 (2013)
17. Schoutens, W.: Levy Processes in Finance: Pricing Financial Derivatives. Wiley, London (2003)
18. Shannon, C.E.: Communication in the presence of noise. Proc. IRE **37**, 10–21 (1949)
19. Stenger, F.: Handbook of Sinc Numerical Methods. CRC PRess, Boca Raton (2011)
20. Ushakov, U.: Selected topics in characteristic functions. De Gruyter, Berlin (1999)