Numer. Math. (2009) 114:27–62
DOI 10.1007/s00211-009-0252-4

Numerische Mathematik

Pricing early-exercise and discrete barrier options by fourier-cosine series expansions

F. Fang · C. W. Oosterlee

Received: 19 June 2008 / Revised: 22 May 2009 / Published online: 20 August 2009

Abstract We present a pricing method based on Fourier-cosine expansions for early-exercise and discretely-monitored barrier options. The method works well for exponential Lévy asset price models. The error convergence is exponential for processes characterized by very smooth ($C^\infty[a,b] \in \mathbb{R}$) transitional probability density functions. The computational complexity is $O((M-1)N \log N)$ with N a (small) number of terms from the series expansion, and M, the number of early-exercise/monitoring dates. This paper is the follow-up of (Fang and Oosterlee in SIAM J Sci Comput 31(2):826–848, 2008) in which we presented the impressive performance of the Fourier-cosine series method for European options.

Mathematics Subject Classification (2000) 65D30 · 91B24 · 65T40

1 Introduction

Within stock option pricing applications, interesting numerical mathematics questions can be found in product pricing and in calibration. Whereas the former topic requires especially robust numerical techniques, the latter also relies on efficiency and speed of computation.

Numerical integration methods, based on a transformation to the Fourier domain (the so-called *transform methods*), are traditionally very efficient, due to the availability

F. Fang (✉)
Delft Institute of Applied Mathematics, Delft University of Technology, Delft, The Netherlands
e-mail: f.fang@ewi.tudelft.nl

C. W. Oosterlee
CWI, Centrum Wiskunde and Informatica, Amsterdam, The Netherlands
e-mail: c.w.oosterlee@cwi.nl

of the Fast Fourier Transform (FFT) [11,28], for the pricing of basic European products, and thus for calibration purposes. These methods can readily be applied to solving problems under various asset price dynamics, for which the characteristic function (i.e., the Fourier transform of the probability density function) is available. This is the case for models from the class of regular affine processes of [16], which also includes the exponentially affine jump-diffusion class of [15], and, in particular, the exponential Lévy models.

Recently, transform methods have been generalized to solving somewhat more complicated option contracts, like Bermudan, American or barrier options, see, for example, [3,4,14,17,23,25,31,32]. These exotic options, still with basic features, are used in the financial industry as building blocks for more complicated products. A natural aim for the near future with these transform methods is to calibrate to these exotic products and to price the huge portfolios (at the end of a trading day) very fast.

Next to FFT-based methods, new techniques based on the Fast Gauss or the Hilbert Transform have been introduced for this purpose [7,8,19]. In this paper, we will also generalize a transform method to pricing Bermudan, American and discretely-monitored barrier options. It is the method based on Fourier-cosine series expansions, called the COS method, introduced by us in [18], where we showed that it was highly efficient for pricing European options. The underlying idea is to replace the transitional probability density function by its Fourier-cosine series expansion, which has an elegant relation to the conditional characteristic function. For many underlying asset price models, the method is remarkably fast and the density function can be recovered easily. Since a whole *function of option values* is obtained, the Greeks can be computed at almost no additional computational cost. Here, we will show that the COS method can also price the early-exercise and barrier options with exponential convergence under various Lévy models.

The methods are, for these option contracts, in competition with the methods that require the solution of discrete partial (integro-) differential equation-based operators (PIDE) [9,35]. PIDE-based methods are traditionally used since early-exercise and the exotic features can often be interpreted as special payoffs or boundary conditions. They represent the state of the art for pricing options under the local volatility process. Generally speaking, however, the computational process with PIDE is rather expensive, especially for the infinite activity Lévy processes we are interested in, because they give rise to an integral in the PIDE with a weakly singular kernel [2,21,34].

We will therefore compare our results with other highly efficient transform methods, i.e., with the Convolution (CONV) method [25], based on the FFT, which is one of the state-of-the-art methods for pricing Bermudan and American options. Its computational complexity for pricing a Bermudan option with M exercise dates is $O((M-1)N\log_2(N))$, where N denotes the number of grid points used for numerical integration. Quadrature rule based techniques are, however, not of the highest efficiency when solving Fourier transformed integrals. As these integrands are highly oscillatory, a relatively fine grid has to be used for satisfactory accuracy with the FFT. The COS method presented here requires a substantially smaller value of N.

Especially for barrier options, another highly efficient alternative method from [19] is based on the Hilbert transform. Its error convergence is exponential for models with rapidly decaying characteristic functions, also with a computational complexity of

$O((M-1)N\log_2 N)$ for a barrier option with M monitoring dates. This method is, however, not applicable for Bermudan options.

The paper is organized as follows. In Sect. 2 the COS method for pricing Bermudan and barrier options is presented. The handling of the discretely monitored barrier options is discussed in particular in Sect. 2.4. Error analysis is performed in Sect. 3. Numerical results are presented in Sect. 4, where we focus on option pricing under exponential Lévy processes, in particular under the CGMY [10] and the Normal Inverse Gaussian [5] processes.

2 Pricing Bermudan and barrier options

A Bermudan option can be exercised at pre-specified dates before maturity. The holder receives the exercise payoff when he/she exercises the option. Between two consecutive exercise dates the valuation process can be regarded as that for a European option, priced with the help of the risk-neutral valuation formula.

Let t_0 denote the initial time and $\mathcal{T}\{t_1,\ldots,t_M\}$ be the collection of all exercise dates with $\Delta t := (t_m - t_{m-1})$, $t_0 < t_1 < \cdots < t_M = T$. The pricing formula for a Bermudan option with M exercise dates then reads, for $m = M, M-1,\ldots,2$:

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

followed by

$$v(x,t_0) = e^{-r\Delta t}\int_{\mathbb{R}} v(y,t_1)f(y|x)dy. \tag{2}$$

Here x and y are state variables, defined as the logarithm of the ratio of the asset price S_t over the strike price K,

$$x := \ln(S(t_{m-1})/K) \quad \text{and} \quad y := \ln(S(t_m)/K),$$

$v(x,t)$, $c(x,t)$ and $g(x,t)$ are the option value, the continuation value and the payoff at time t, respectively. Note that for vanilla options, $g(x,t)$ equals $v(x,T)$, with

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

The probability density function of y given x under a risk-neutral measure is denoted by $f(y|x)$ in (2), and r is the (deterministic) risk-neutral interest rate.

Equations (1), and (2) can be efficiently evaluated by the COS method in [18], provided that the Fourier-cosine series coefficients of $v(y,t_m)$ are known.

2.1 The COS method

The COS method is based on the insight that the Fourier-cosine series coefficients of $f(y|x)$ are closely related to its characteristic function.

Since the density function, $f(y|x)$, decays to zero rapidly as $y \to \pm\infty$, we can truncate the infinite integration range in the risk-neutral valuation formula without loosing significant accuracy. Suppose that we have, with $[a, b] \subset \mathbb{R}$,

$$\int_{\mathbb{R}\setminus[a,b]} f(y|x)dy < \text{TOL}, \tag{3}$$

for some given tolerance, TOL, then we can approximate $c(x, t_{m-1})$ in (1) by

$$c_1(x, t_{m-1}) = e^{-r\Delta t} \int_a^b v(y, t_m) f(y|x)dy. \tag{4}$$

(The intermediate terms, c_i, are used in the error analysis in Sect. 3.) As a second step, we replace the density function by its Fourier-cosine series expansion on $[a, b]$,

$$f(y|x) = {\sum_{k=0}^{\infty}}' A_k(x) \cos\left(k\pi \frac{y-a}{b-a}\right), \tag{5}$$

where ${\sum}'$ indicates that the first term in the summation is multiplied by $1/2$. The series coefficients $\{A_k(x)\}_{k=0}^{\infty}$ are defined by

$$A_k(x) := \frac{2}{b-a} \int_a^b f(y|x) \cos\left(k\pi \frac{y-a}{b-a}\right) dy. \tag{6}$$

Interchanging the summation and integration operators yields

$$c_1(x, t_{m-1}) = \frac{1}{2}(b-a)e^{-r\Delta t} {\sum_{k=0}^{\infty}}' A_k(x) V_k(t_m), \tag{7}$$

with $V_k(t_m)$ the Fourier-cosine series coefficients of $v(y, t_m)$ on $[a, b]$, i.e.

$$V_k(t_m) := \frac{2}{b-a} \int_a^b v(y, t_m) \cos\left(k\pi \frac{y-a}{b-a}\right) dy. \tag{8}$$

Truncating the infinite series gives

$$c_2(x, t_{m-1}) = \frac{1}{2}(b-a)e^{-r\Delta t} \sum_{k=0}^{N-1}{}' A_k(x) V_k(t_m). \tag{9}$$

As a third step, we use the relation between $A_k(x)$ and the conditional characteristic function, $\phi(\omega; x)$, defined as

$$\phi(\omega; x) := \int_{\mathbb{R}} f(y|x) e^{i\omega y} dy. \tag{10}$$

Coefficients $A_k(x)$ can be written as

$$A_k(x) = \frac{2}{b-a} \text{Re} \left\{ e^{-ik\pi \frac{a}{b-a}} \int_a^b e^{i\frac{k\pi}{b-a}y} f(y|x) dy \right\}. \tag{11}$$

where $\text{Re}\{\cdot\}$ denotes taking the real part of the input argument. With (3), the finite integration in (11) can be approximated by

$$\int_a^b e^{i\frac{k\pi}{b-a}y} f(y|x) dy \approx \int_{\mathbb{R}} e^{i\frac{k\pi}{b-a}y} f(y|x) dy =: \phi\left(\frac{k\pi}{b-a}; x\right).$$

As a result, $A_k(x)$ can be approximated by $F_k(x)$ with

$$F_k(x) := \frac{2}{b-a} \text{Re} \left\{ \phi\left(\frac{k\pi}{b-a}; x\right) e^{-ik\pi \frac{a}{b-a}} \right\}. \tag{12}$$

Replacing $A_k(x)$ in (9) by $F_k(x)$ gives the COS formula for pricing European options for different underlying processes:

$$\hat{c}(x, t_{m-1}) := e^{-r\Delta t} \sum_{k=0}^{N-1}{}' \text{Re} \left\{ \phi\left(\frac{k\pi}{b-a}; x\right) e^{-ik\pi \frac{a}{b-a}} \right\} V_k(t_m). \tag{13}$$

Here the function $\hat{c}(x, t_{m-1})$ represents the approximation of the continuation value $c(x, t_{m-1})$. An error analysis justifying the different approximations for European options was presented in [18] and is summarized in Sect. 3.

For exponential Lévy processes, formula (13) can be simplified to

$$\hat{c}(x, t_{m-1}) = e^{-r\Delta t} \sum_{k=0}^{N-1}{}' \text{Re} \left\{ \varphi_{\text{levy}}\left(\frac{k\pi}{b-a}\right) e^{ik\pi \frac{x-a}{b-a}} \right\} V_k(t_m), \tag{14}$$

where $\varphi_{\text{levy}}(\omega) := \phi_{\text{levy}}(\omega; 0)$, see [18]. Using this, we can also approximate $v(x, t_0)$ in (2) by

$$\hat{v}(x,t_0) = e^{-r\Delta t}\sum_{k=0}^{N-1}{}' \text{Re}\left\{\varphi_{\text{levy}}\left(\frac{k\pi}{b-a}\right)e^{ik\pi\frac{x-a}{b-a}}\right\}V_k(t_1), \tag{15}$$

provided that the series coefficients, $V_k(t_1)$, are known. We will show that the $V_k(t_m)$, $k = 0, 1, \ldots, N-1$, can be recovered from $V_j(t_{m+1})$, $j = 0, 1, \ldots, N-1$.

2.2 Pricing Bermudan options

The idea of pricing Bermudan options is to compute $V_k(t_1)$, the cosine coefficients of the option value at time point t_1, and insert it into (15), to obtain the value of the option. The main contribution of this section is the derivation of an induction formula for $V_k(t_1)$.

The integral in the definition of $V_k(t_m)$ in (8) can be split into two parts, if we determine the *early-exercise point*, x_m^*, at time t_m, which is the point where the continuation value equals the payoff, i.e., $c(x_m^*, t_m) = g(x_m^*, t_m)$.

Once we have x_m^*, we can split the integral that defines $V_k(t_m)$ into two parts: One on the interval $[a, x_m^*]$ and the other on $(x_m^*, b]$, i.e.

$$V_k(t_m) = \begin{cases} C_k(a, x_m^*, t_m) + G_k(x_m^*, b), & \text{for a call,} \\ G_k(a, x_m^*) + C_k(x_m^*, b, t_m), & \text{for a put,} \end{cases} \tag{16}$$

for $m = M-1, M-2, \ldots, 1$, and

$$V_k(t_M) = \begin{cases} G_k(0, b), & \text{for a call} \\ G_k(a, 0), & \text{for a put,} \end{cases} \tag{17}$$

whereby

$$G_k(x_1, x_2) := \frac{2}{b-a}\int_{x_1}^{x_2} g(x, t_m)\cos\left(k\pi\frac{x-a}{b-a}\right)dx. \tag{18}$$

and

$$C_k(x_1, x_2, t_m) := \frac{2}{b-a}\int_{x_1}^{x_2} c(x, t_m)\cos\left(k\pi\frac{x-a}{b-a}\right)dx. \tag{19}$$

Remark 2.1 (Newton's method) Since the numerical approximation for $c(x, t_m)$, denoted by $\hat{c}(x, t_m)$, in (14) is a semi-analytic formula which returns a numerical approximation of $c(x, t_m)$ on the whole support of x, we can easily find the derivatives of $\hat{c}(x, t_m)$ w.r.t. x, and can therefore employ Newton's method to determine x_m^*. Note that, on each time lattice, there is at most one point which satisfies

$\hat{c}(x, t_m) - g(x, t_m) = 0$, for the option problems considered here.[1] If x_m^* is not in $[a, b]$, it is set equal to the nearest boundary point.

Result 2.1 *The $G_k(x_1, x_2)$ in (18) can be determined analytically.*

Proof With $g(x, t_m) \equiv [\pm K(1 - e^x)]^+$, it follows for a put, with $x_2 \leq 0$, that

$$G_k(x_1, x_2) = \frac{2}{b-a} \int_{x_1}^{x_2} K(1 - e^x) \cos\left(k\pi \frac{x-a}{b-a}\right) dx, \tag{20}$$

and for a call, with $x_1 \geq 0$, that

$$G_k(x_1, x_2) = \frac{2}{b-a} \int_{x_1}^{x_2} K(e^x - 1) \cos\left(k\pi \frac{x-a}{b-a}\right) dx, \tag{21}$$

The fact that $x_m^* \leq 0$, for put options, and $x_m^* \geq 0$, for call options, $\forall t \in \mathcal{T}$, gives

$$G_k(x_1, x_2) = \frac{2}{b-a} \alpha K \left[\chi_k(x_1, x_2) - \psi_k(x_1, x_2)\right], \quad \alpha = \begin{cases} 1 & \text{for a call,} \\ -1 & \text{for a put,} \end{cases} \tag{22}$$

with

$$\chi_k(x_1, x_2) := \int_{x_1}^{x_2} e^x \cos\left(k\pi \frac{x-a}{b-a}\right) dx, \tag{23}$$

$$\psi_k(x_1, x_2) := \int_{x_1}^{x_2} \cos\left(k\pi \frac{x-a}{b-a}\right) dx. \tag{24}$$

These integrals admit the following analytic solutions:

$$\begin{aligned} \chi_k(x_1, x_2) = {} & \frac{1}{1 + \left(\frac{k\pi}{b-a}\right)^2} \left[\cos\left(k\pi \frac{x_2 - a}{b-a}\right) e^{x_2} - \cos\left(k\pi \frac{x_1 - a}{b-a}\right) e^{x_1} \right. \\ & \left. + \frac{k\pi}{b-a} \sin\left(k\pi \frac{x_2 - a}{b-a}\right) e^{x_2} - \frac{k\pi}{b-a} \sin\left(k\pi \frac{x_1 - a}{b-a}\right) e^{x_1} \right], \\ \psi_k(x_1, x_2) = {} & \begin{cases} \left[\sin\left(k\pi \frac{x_2 - a}{b-a}\right) - \sin\left(k\pi \frac{x_1 - a}{b-a}\right)\right] \frac{b-a}{k\pi} & k \neq 0, \\ (d - c) & k = 0. \end{cases} \end{aligned} \tag{25}$$

□

[1] Generalizations for more early-exercise points are easily determined.

We now derive the formulas for the Fourier cosine coefficients of the option values, $V_j(t_m)$ with $j = 0, 1, \ldots, N-1$ and $m = 1, 2, \ldots, M$.

At time t_M, these coefficients, $V_j(t_M)$, are exact, see Eq. (17). At time t_{M-1}, from COS formula (14) we obtain approximation $\hat{c}(x, t_{M-1})$, the continuation value at t_{M-1}, which is inserted into (19). Interchanging summation and integration gives the following coefficients, $\hat{C}$:

$$\hat{C}_k(x_1, x_2, t_{M-1}) = e^{-r\Delta t}\text{Re}\left\{ \sum_{j=0}^{N-1}{}' \varphi_{\text{levy}}\left(\frac{j\pi}{b-a}\right) V_j(t_M) \cdot \mathcal{M}_{k,j}(x_1, x_2) \right\}, \quad (26)$$

with the coefficients $\mathcal{M}_{k,j}(x_1, x_2)$ defined as

$$\mathcal{M}_{k,j}(x_1, x_2) := \frac{2}{b-a} \int_{x_1}^{x_2} e^{ij\pi\frac{x-a}{b-a}} \cos\left(k\pi\frac{x-a}{b-a}\right) dx, \quad (27)$$

and $i = \sqrt{-1}$ being the imaginary unit.

For time points t_m, $m = M-2, M-3, \ldots, 1$, we can define

$$\hat{C}_k(x_1, x_2, t_m) := e^{-r\Delta t}\text{Re}\left\{ \sum_{j=0}^{N-1}{}' \varphi_{\text{levy}}\left(\frac{j\pi}{b-a}\right) \hat{V}_j(t_{m+1}) \cdot \mathcal{M}_{k,j}(x_1, x_2) \right\}, \quad (28)$$

which is the result of replacing $V_j(t_{m+1})$ in the definition of $C_k(x_1, x_2, t_m)$ by its numerical approximation $\hat{V}_j(t_{m+1})$.

Replacing C_k in (16) by $\hat{C}_k$ gives us the numerical approximation of the Fourier cosine coefficients of the option values at times t_m, $m = 1, 2, \ldots, M-1$. In vector form, it reads

$$\hat{\mathbf{V}}(t_m) = \begin{cases} \hat{\mathbf{C}}(a, x_m^*, t_m) + \mathbf{G}(x_m^*, b), & \text{for a call,} \\ \hat{\mathbf{C}}(x_m^*, b, t_m) + \mathbf{G}(a, x_m^*), & \text{for a put.} \end{cases} \quad (29)$$

with

$$\hat{\mathbf{C}}(x_1, x_2, t_m) = \begin{cases} e^{-r\Delta t}\text{Re}\left\{\mathcal{M}(x_1, x_2)\ \Lambda\right\}\ \mathbf{V}(t_M), & m = M-1, \\ e^{-r\Delta t}\text{Re}\left\{\mathcal{M}(x_1, x_2)\ \Lambda\right\}\ \hat{\mathbf{V}}(t_{m+1}), & m = 1, 2, \ldots, M-2. \end{cases} \quad (30)$$

where we use bold-faced letters to denote vectors, e.g. $\mathbf{V}(t_M)$ is the vector $(V_0(t_M), V_1(t_M), \ldots, V_{N-1}(t_M))^T$. "$\mathcal{M}\ \Lambda$" denotes a matrix-matrix multiplication with $\mathcal{M}$ being a matrix with elements $\{\mathcal{M}_{k,j}\}_{k,j=0}^{N-1}$ and Λ a diagonal matrix with elements $\{\varphi_{\text{levy}}(\frac{j\pi}{b-a})\}_{j=0}^{N-1}$.

This matrix-vector product representation is useful for analyzing the convergence properties of Bermudan option values to their American counterparts (with $M \to \infty$),

in Appendix A. It should, however, not be employed to determine the coefficients, since these matrix-vector product costs $O(N^2)$ operations and is thus expensive.

In Sect. 2.3, we will present an efficient algorithm for the computation of $\hat{\mathbf{V}}(t_m)$, with complexity $O(N \log_2(N))$, based on the FFT algorithm.

We first summarize the algorithm for pricing Bermudan options:

ALGORITHM 1: Pricing Bermudan options with the COS method.

Initialization: For $k = 0, 1, \ldots, N-1$,

- $V_k(t_M) = G_k(0, b)$ for call options; $V_k(t_M) = G_k(a, 0)$ for put options;

Main Loop to Recover $\hat{V}_k(t_m)$: For $m = M-1$ to 1,

- Determine early-exercise point x_m^* by Newton's method;
- Compute $\hat{V}_k(t_m)$ (with the help of the FFT algorithm).

Final step: Reconstruct $\hat{v}(x, t_0)$ by inserting $\hat{V}_k(t_1)$ into (15).

Remark 2.2 (The Greeks) To compute the Greeks, one only needs to modify the final step in Algorithm 1, from t_1 to t_0, as the Greeks can be approximated by

$$\hat{\Delta} = e^{-r\Delta t} \frac{2}{b-a} \sum_{k=0}^{N-1}{}' \operatorname{Re}\left\{ \left\{ \varphi\left(\frac{k\pi}{b-a}\right) e^{ik\pi \frac{x-a}{b-a}} \frac{ik\pi}{b-a} \right\} \right\} \frac{\hat{V}_k(t_1)}{S_0} \tag{31}$$

and

$$\hat{\Gamma} = e^{-r\Delta t} \frac{2}{b-a} \sum_{k=0}^{N-1}{}' \operatorname{Re}\left\{ \left\{ \varphi\left(\frac{k\pi}{b-a}\right) e^{ik\pi \frac{x-a}{b-a}} \left[-\frac{ik\pi}{b-a} + \left(\frac{ik\pi}{b-a}\right)^2 \right] \right\} \right\} \frac{\hat{V}_k(t_1)}{S_0^2}. \tag{32}$$

2.3 Efficient algorithm

In the following we will develop an FFT-based algorithm for computing the matrix-vector product in (30). The main insight is that matrix $\mathcal{M}$ in (30) is a sum of a Hankel and a Toeplitz matrix.

Theorem 2.1 $\hat{\mathbf{C}}(x_1, x_2, t_m)$ *in* (30) *can be computed in* $O(N \log_2(N))$ *operations with the help of the Fast Fourier Transform (FFT) algorithm.*

Proof Replacing $e^{i\alpha} = \cos(\alpha) + i \sin(\alpha)$ in the definition of $\mathcal{M}_{k,j}(x_1, x_2)$ in (27) gives the following representation:

$$\mathcal{M}_{k,j}(x_1, x_2) = -\frac{i}{\pi} \left(\mathcal{M}^c_{k,j}(x_1, x_2) + \mathcal{M}^s_{k,j}(x_1, x_2) \right), \tag{33}$$

where

$$\mathcal{M}^c_{k,j} := \begin{cases} \dfrac{(x_2 - x_1)\pi i}{(b-a)} & k = j = 0, \\ \dfrac{\exp\left(i(j+k)\dfrac{(x_2-a)\pi}{b-a}\right) - \exp\left(i(j+k)\dfrac{(x_1-a)\pi}{b-a}\right)}{j+k} & \text{otherwise} \end{cases} \tag{34}$$

and

$$\mathcal{M}^s_{k,j} := \begin{cases} \dfrac{(x_2 - x_1)\pi i}{b-a} & k = j, \\ \dfrac{\exp\left(i(j-k)\dfrac{(x_2-a)\pi}{b-a}\right) - \exp\left(i(j-k)\dfrac{(x_1-a)\pi}{b-a}\right)}{j-k} & k \neq j. \end{cases} \tag{35}$$

After inserting (33) into (26) and (28), we obtain a matrix-vector product representation for $\hat{\mathbf{C}}(x_1, x_2, t_m)$, i.e.,

$$\hat{\mathbf{C}}(x_1, x_2, t_m) = \frac{e^{-r\Delta t}}{\pi} \text{Im}\left\{(\mathcal{M}_c + \mathcal{M}_s)\mathbf{u}\right\}, \tag{36}$$

where Im $\{\cdot\}$ denotes taking the imaginary part of the input argument, and

$$\mathbf{u} := \{u_j\}_{j=0}^{N-1}, \quad u_j := \varphi\left(\frac{j\pi}{b-a}\right) V_j(t_{m+1}), \; u_0 = \frac{1}{2}\varphi(0)\, V_0(t_{m+1}). \tag{37}$$

The matrices

$$\mathcal{M}_c := \{\mathcal{M}^c_{k,j}(x_1, x_2)\}_{k,j=0}^{N-1} \quad \text{and} \quad \mathcal{M}_s := \{\mathcal{M}^s_{k,j}(x_1, x_2)\}_{k,j=0}^{N-1}$$

have special structures, so that the FFT algorithm can be employed for the efficient computation of matrix-vector products.

In particular, matrix $\mathcal{M}_c$ is a *Hankel* matrix,

$$\mathcal{M}_c = \begin{bmatrix} m_0 & m_1 & m_2 & \cdots & m_{N-1} \\ m_1 & m_2 & \cdots & \cdots & m_N \\ \vdots & & & & \vdots \\ m_{N-2} & m_{N-1} & \cdots & & m_{2N-3} \\ m_{N-1} & \cdots & & m_{2N-3} & m_{2N-2} \end{bmatrix}_{N\times N} \tag{38}$$

and $\mathcal{M}_s$ is a *Toeplitz* matrix,

$$\mathcal{M}_s = \begin{bmatrix} m_0 & m_1 & \cdots & m_{N-2} & m_{N-1} \\ m_{-1} & m_0 & m_1 & \cdots & m_{N-2} \\ \vdots & & \ddots & & \vdots \\ m_{2-N} & \cdots & m_{-1} & m_0 & m_1 \\ m_{1-N} & m_{2-N} & \cdots & m_{-1} & m_0 \end{bmatrix}_{N\times N} \tag{39}$$

with

$$m_j := \begin{cases} \dfrac{(x_2 - x_1)\pi}{b-a} i & j = 0 \\ \dfrac{\exp\left(ij\dfrac{(x_2-a)\pi}{b-a}\right) - \exp\left(ij\dfrac{(x_1-a)\pi}{b-a}\right)}{j} & j \neq 0 \end{cases} \tag{40}$$

This concludes the proof. □

The matrix-vector product, with these special matrices, can be transformed into a circular convolution. This is well-known for Toeplitz matrices, described in detail, for example, in [2]. The product $\mathcal{M}_s\mathbf{u}$ is equal to the first N elements of $\mathbf{m}_s \circledast \mathbf{u}_s$ with the $2N$-vectors:

$$\mathbf{m}_s = \left[m_0, m_{-1}, m_{-2}, \ldots, m_{1-N}, 0, m_{N-1}, m_{N-2}, \ldots, m_1\right]^T,$$

and $\mathbf{u}_s = [u_0, u_1, \ldots, u_{N-1}, 0, \ldots, 0]^T$. For the Hankel matrix this is less known, so we formulate it in the following result:

Result 2.2 *The product $\mathcal{M}_c\mathbf{u}$ is equal to the first N elements of $\mathbf{m}_c \circledast \mathbf{u}_c$, in reversed order, with the $2N$-vectors: $\mathbf{m}_c = [m_{2N-1}, m_{2N-2}, \ldots, m_1, m_0]^T$ and $\mathbf{u}_c = [0, \ldots, 0, u_0, u_1, \ldots, u_{N-1}]^T$.*

For the efficient computation of $\mathcal{M}_c\mathbf{u}$, we need to construct the following circulant matrix, $\mathcal{M}_u$,

$$\mathcal{M}_u = \begin{bmatrix} 0 & u_{N-1} & u_{N-2} & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & u_{N-1} & u_{N-2} & \cdots & \cdots & 0 \\ \vdots & & \ddots & & \ddots & & \vdots \\ 0 & \cdots & 0 & u_{N-1} & u_{N-2} & \cdots & u_0 \\ u_0 & 0 & \cdots & 0 & u_{N-1} & \cdots & u_1 \\ u_1 & u_0 & 0 & \cdots & 0 & \cdots & u_2 \\ \vdots & & \ddots & & & \ddots & \vdots \\ u_{N-2} & \cdots & u_0 & 0 & \cdots & 0 & u_{N-1} \\ u_{N-1} & u_{N-2} & \cdots & u_0 & 0 & \cdots & 0 \end{bmatrix}_{(2N)\times(2N)}. \tag{41}$$

Straightforward computation shows that the first N elements of the product of $\mathbf{u}_c$ and $\mathbf{m}_c$ equal $\mathcal{M}_c\mathbf{u}$, in reversed order.

A circular convolution of two vectors is equal to the inverse discrete Fourier transform ($\mathcal{D}^{-1}$) of the products of the forward DFTs, $\mathcal{D}$, i.e.,

$$\mathbf{x} \circledast \mathbf{y} = \mathcal{D}^{-1}\{\mathcal{D}(\mathbf{x}) \cdot \mathcal{D}(\mathbf{y})\}.$$

We now summarize the algorithm of computing $\hat{\mathbf{C}}(x_1, x_2, t_m)$ as follows:

ALGORITHM 2: Computation of $\hat{\mathbf{C}}(x_1, x_2, t_m)$.

1. Compute $m_j(x_1, x_2)$ for $j = 0, 1, \ldots, N-1$ using (40).
2. Construct $\mathbf{m_s}(x_1, x_2)$ and $\mathbf{m_c}(x_1, x_2)$ using the properties of m_j's.
3. Compute $\mathbf{u}(t_m)$ using (37).
4. Construct $\mathbf{u_s}$ by padding N zeros to $\mathbf{u}(t_m)$.
5. **Msu** = the first N elements of $\mathcal{D}^{-1}\{\, \mathcal{D}(\mathbf{m_s}) \cdot \mathcal{D}(\mathbf{u_s}) \,\}$.
6. **Mcu** = reverse{ the first N elements of $\mathcal{D}^{-1}\{\, \mathcal{D}(\mathbf{m_c}) \cdot \mathbf{sgn} \cdot \mathcal{D}(\mathbf{u_s}) \,\}\}$.
7. $\hat{\mathbf{C}}(x_1, x_2, t_m) = e^{-r\Delta t}\mathrm{Im}\{\mathbf{Msu} + \mathbf{Mcu}\}/\pi$.

Note that the operation $\mathcal{D}(\mathbf{u_s})$ is computed only once, and"reverse{$\mathbf{x}$}" denotes an $\mathbf{x}$-generated vector, whose elements are the same as those of $\mathbf{x}$ but sorted in reversed order.

Remark 2.3 (Efficient computation) It is worth mentioning that the computation of the exponentials takes significantly more computer clock cycles than additions or multiplications. One can however benefit from some special properties of the m_j's, like $m_{-j} = -\overline{m_j}$ and, for $j \neq 0$,

$$m_{j+N} = \frac{\exp\left(iN\frac{(x_2-a)\pi}{b-a}\right)\cdot\exp\left(ij\frac{(x_2-a)\pi}{b-a}\right) - \exp\left(iN\frac{(x_1-a)\pi}{b-a}\right)\cdot\exp\left(ij\frac{(x_1-a)\pi}{b-a}\right)}{j+N}.$$

So, in order to construct $\mathbf{m}_s$ and $\mathbf{m}_c$, the factors $\exp(ij\frac{(x_2-a)\pi}{b-a})$ and $\exp(ij\frac{(x_1-a)\pi}{b-a})$, for $j = 0, 1, \ldots, N-1$, should be computed only once.

Also, the DFT of $\mathbf{u}_c$ and of $\mathbf{u}_s$ need not be computed separately, as the shift property of DFTs gives $\mathcal{D}(\mathbf{u}_c) = \mathbf{sgn} \cdot \mathcal{D}(\mathbf{u}_s)$ with $\mathbf{sgn} = [1, -1, 1, -1, \ldots]^T$.

Remark 2.4 (Overall computational complexity) Since the computation of $G_k(x_1, x_2)$ is linear in N, the overall complexity of this recovery procedure is dominated by the computation of $\hat{\mathbf{C}}(x_1, x_2, t_m)$, whose complexity is $O(N \log_2 N)$ with the FFT. As a result, the overall computational complexity for pricing a Bermudan option with M exercise dates is $O((M-1)N \log_2 N)$, as the work needed for the final step, from t_1 to t_0, is $O(N)$.

Remark 2.5 (Use of FFT algorithm) In the main loop of the CONV method from [25], the FFT algorithm is required five times, the same as in the COS method presented above, and the length of the CONV input vectors is halved compared to the COS method. Therefore, the CONV method would be approximately twice as fast, if we did not take the method's accuracy into account. However, for models characterized by density functions in $C^\infty[a, b]$, the COS method exhibits an exponential convergence rate, which is superior to the second order convergence of the CONV method. For

the same level of accuracy, the COS method is therefore significantly faster than the CONV method.

2.4 Discretely-monitored barrier options

Discretely-monitored "out" barrier options are options that cease to exist if the asset price hits a certain barrier level, H, at one of the pre-specified observation dates. If $H > S_0$, they are called "up-and-out" options, and "down-and-out" otherwise. The payoff for an up-and-out option reads

$$v(x,T) = (\max(\alpha(S_T - K), 0) - Rb)\mathbf{1}_{\{S_{t_i} < H\}} + Rb, \tag{42}$$

where $\alpha = 1$ for a call and $\alpha = -1$ for a put, Rb is a rebate, and $\mathbf{1}_A$ is the indicator function,

$$\mathbf{1}_A = \begin{cases} 1 & \text{if } A \text{ is not empty,} \\ 0 & \text{otherwise.} \end{cases}$$

With the set of observation dates, $\mathcal{T} = \{t_1, \ldots, t_M\}$, $t_1 < \cdots < t_{M-1} < t_M = T$, the price of an up-and-out option, monitored M times, satisfies the following recursive formula

$$\begin{cases} c(x, t_{m-1}) = e^{-r(t_m - t_{m-1})} \int_{\mathbb{R}} v(x, t_m) f(y|x) dy, \\ v(x, t_{m-1}) = \begin{cases} e^{-r(T - t_{m-1})} Rb, & x \geq h, \\ c(x, t_{m-1}), & x < h, \end{cases} \end{cases} \tag{43}$$

where $h := \ln(H/K)$ and $m = M, M-1, \ldots, 2$.

Note that the recursive pricing formula (43) is very similar to that for the Bermudan options. What makes barrier pricing easier is that the root-searching algorithm is not needed as the barrier points are known in advance. Thus, similar to Bermudan options, discrete barrier options can be priced in two steps:

1. Recovery of the Fourier-cosine series coefficients of the option value at t_1,
2. The COS formula for European options given by (15).

Based on the derivation for Bermudan options, we have the following lemma:

Lemma 2.1 (Backward induction for discrete barrier options) *By backward recursion we find the following numerical approximation for discretely monitored barrier options: For* $m = M-1, M-2, \ldots, 1$,

$$\hat{V}_k(t_m) = \hat{C}_k(a, h, t_m) + e^{-r(T - t_{m-1})} Rb \frac{2}{b-a} \psi_k(h, b) \tag{44}$$

with $\hat{C}_k(x_1, x_2, t_m)$ *and* $\psi_k(x_1, x_2)$ *given by* (36) *and* (25)*, respectively. If* $h < 0$*, we have*

$$V_k(t_M) = \begin{cases} 2Rb\psi_k(h, b)/(b-a) & \text{for a call,} \\ G_k(a, h) + 2Rb\psi_k(h, b)/(b-a) & \text{for a put.} \end{cases} \tag{45}$$

For $h \geq 0$*, we find*

$$V_k(t_M) = \begin{cases} G_k(0, h) + 2Rb\psi_k(h, b)/(b-a) & \text{for a call,} \\ G_k(a, 0) & \text{for a put.} \end{cases} \tag{46}$$

A similar recursion formula for a down-and-out option can be derived easily.

The proof is straightforward, as it goes along the lines of the derivation for Bermudan options in the previous section.

The computation of $\hat{\mathbf{C}}(a, h, t_m)$ via (36) is less expensive than for Bermudan options, because h is known in advance, and consequently, $\psi_k(h, b)$ in (24), $\mathcal{M}_c$ and $\mathcal{M}_s$ in (36) are known before the recursion step. Therefore, the FFT technique is required only three times.

Barrier options with an "in" barrier, or double barrier options, can be priced as easily with the COS method. Alternatively, one could apply the barrier parity and symmetry results on "out" barrier options [20,33].

We summarize the method by means of the following algorithm:

ALGORITHM 3: Pricing Discrete Barrier Options by the COS Method

Initialization:

- Compute $V_k(t_M)$ using (45) or (46).
- For up-and-out: $x_1 = a$ and $x_2 = h$, and $c = h$ and $d = b$;
  For down-and-out: $x_1 = h$ and $x_2 = b$, and $c = a$ and $d = h$.
- Construct $\mathbf{m_s}(x_1, x_2)$ and $\mathbf{m_c}(x_1, x_2)$ using the properties of m_j's.
- $d_1 = \mathcal{D}\{\mathbf{m_s}(x_1, x_2)\}$, $d_2 = \mathbf{sgn} \cdot \mathcal{D}\{\mathbf{m_c}(x_1, x_2)\}$
- $\mathbf{G} = \frac{2}{b-a} Rb \, \{\psi_k(c, d)\}_{k=0}^{N-1}$.

Main Loop to Recover $\hat{\mathbf{V}}(t_{m-1})$**:** For $m = M$ to 2,

1. Compute $\mathbf{u}(t_m)$ using Eq. (37).
2. Construct $\mathbf{u_s}$ by padding N zeros to $\mathbf{u}(t_m)$.
3. **Msu** = the first N elements of $\mathcal{D}^{-1}\{\, d_1 \cdot \mathcal{D}(\mathbf{u_s}) \,\}$.
4. **Mcu** = reverse{ the first N elements of $\mathcal{D}^{-1}\{\, d_2 \cdot \mathcal{D}(\mathbf{u_s}) \,\}$ }.
5. $\hat{\mathbf{C}}(t_{m-1}) = e^{-r\Delta t}/\pi \, \mathrm{Im}\,\{\mathbf{Msu} + \mathbf{Mcu}\}$.
6. $\hat{\mathbf{V}}(t_{m-1}) = \hat{\mathbf{C}}(t_{m-1}) + e^{-r(T-t_{m-1})}\mathbf{G}$

Finalization: Compute $\hat{v}(t_0, x)$ according to (15); Or Greeks by (31) and (32).

3 Error analysis

In this section, we analyze the rate of convergence as well as the stability of the COS method.

3.1 Convergence for European options

We define ϵ as

$$\epsilon\left(x; N, [a,b]\right) := c(x) - \hat{c}\left(x; N, [a,b]\right), \tag{47}$$

An upper bound for this local error with respect to the truncation range as well as the convergence rate of ϵ in dependence on N, the number of leading terms in the Fourier cosine series, have been derived in [18]. Here we recall the main conclusions.

The COS formula for European options was derived in three steps in Sect. 2.1. Thus, error ϵ is decomposed in three components:

1. The integration range truncation error:

$$\epsilon_1\left(x; [a,b]\right) := c(x) - c_1(x; [a,b]) = \int_{\mathbb{R}\setminus[a,b]} v(y) f(y|x) dy. \tag{48}$$

2. The series truncation error on $[a,b]$:

$$\begin{aligned}\epsilon_2\left(x; N, [a,b]\right) &:= c_1(x; [a,b]) - c_2(x; N, [a,b]) \\ &= \frac{1}{2}(b-a)e^{-r\Delta t} \sum_{k=N}^{\infty} A_k(x) \cdot V_k.\end{aligned} \tag{49}$$

3. The error related to approximating $A_k(x)$ by $F_k(x)$ in (12):

$$\begin{aligned}\epsilon_3\left(x; N, [a,b]\right) &:= c_2\left(x; N, [a,b]\right) - \hat{c}\left(x; N, [a,b]\right) \\ &= e^{-r\Delta t} {\sum_{k=0}^{N-1}}' \mathrm{Re}\left\{ \int_{\mathbb{R}\setminus[a,b]} e^{ik\pi \frac{y-a}{b-a}} f(y|x) dy \right\} V_k.\end{aligned} \tag{50}$$

Since the option value, $v(y)$, is bounded on $[a,b]$, we have

$$\epsilon_1\left(x; [a,b]\right) \sim O\left(\int_{\mathbb{R}\setminus[a,b]} f(y|x) \right) \sim O(\mathrm{TOL}),$$

according to (3). To study the impact of x on ϵ_1, we use the property $f(y|x) = f(y-x)$, which holds for Lévy processes. After a change of variables on (48), we find

$$|\epsilon_1([a,b])| = \left| \int_{\mathbb{R}\setminus[a-x,b-x]} v(x+z) f(z) dz \right| \sim O\left(\int_{\mathbb{R}\setminus[a-x,b-x]} f(z) dz \right). \quad (51)$$

So, when $[a,b]$ is centered around x, or $\min(|a|,|b|) >> x$, the influence of x on ϵ_1 can be ignored, and ϵ_1 only depends on the size of the truncation range: Larger intervals $[b-a]$ result in smaller values of ϵ_1. Numerical experiments supporting this are presented in Fig. 1. The definition of a proper truncation range is given in Sect. 3.3.

The second error component, ϵ_2, converges *exponentially* for probability density functions of class $C^\infty([a,b])$, given a value of x [6,18], i.e.,

$$|\epsilon_2| < P \cdot \exp(-(N-1)\nu), \quad (52)$$

where $\nu > 0$ is a constant and P is a term which varies less than exponentially with N. When the probability density function has a discontinuous derivative, the Fourier-cosine expansion converges *algebraically*, i.e.

$$|\epsilon_2| < \frac{\bar{P}}{(N-1)^{\beta-1}}, \quad (53)$$

where $\bar{P}$ is a constant and $\beta \geq n \geq 1$ (and n is the algebraic index of convergence of the series coefficients).

For Lévy processes, a non-zero x corresponds to a shift, $f(z := y-x)$, and is thus not related to the smoothness of $f(z)$. As a result, the convergence speed, ν in (52) or β in (53), does not depend on x.

The third error component, ϵ_3, consists of the integration range related truncation error [18], and can be bounded by

$$|\epsilon_3| < |\epsilon_1| + Q \left| \int_{\mathbb{R}\setminus[a,b]} f(y|x) dy \right|, \quad (54)$$

where Q is some constant independent of N. Applying a change of variables as for ϵ_1, it is clear that also here the choice of x has no impact on ϵ_3, if $\min(|a|,|b|) >> x$ or if $[a,b]$ is centered around x.

Collecting the three error components and applying the triangle inequality, we can bound the local error, ϵ, as follows:

$$\begin{aligned} |\epsilon(x;N,[a,b])| &\leq |c-c_1| + |c_1-c_2| + \left|c_2-\hat{c}\right| \\ &\leq \bar{Q} \cdot \left| \int_{\mathbb{R}\setminus[a,b]} f(y|x) dy \right| + |\epsilon_2(x;N)|, \end{aligned} \quad (55)$$

with $\bar{Q}$ some constant not depending on $[a, b]$ and N. With integration interval $[a, b]$ chosen sufficiently wide, the series truncation error, $\epsilon_2(N)$, dominates the overall error, which implies that for smooth density functions, ϵ converges exponentially; otherwise it goes algebraically.

3.2 Error propagation in the backward recursion

In this section we study the error in the Fourier coefficients,

$$\varepsilon(k, t_m) := V_k(t_m) - \hat{V}_k(t_m), \tag{56}$$

and its propagation in the backward recursion, which is directly related to the error in the Bermudan option values. We focus on put options here and assume that the error resulting from applying Newton's method is not significant, i.e., the early-exercise points are determined exactly. For ease of presentation, we analyze the case that the underlying density function is infinitely differentiable. Similar analysis can be done for other cases.

Theorem 3.1 *With $[a, b] \subset \mathbb{R}$ sufficiently large and a probability density function in $\mathrm{C}^\infty([a, b])$, error $\varepsilon(k, t_m)$ converges exponentially in N.*

Proof The proof is obtained by an induction argument. At time t_{M-1}, we compare (29) and (16), and find

$$\begin{aligned}\varepsilon(k, t_{M-1}) &= C_k(x^*_{M-1}, b, t_{M-1}) - \hat{C}_k(x^*_{M-1}, b, t_{M-1}),\\ &= \int_{x^*_{M-1}}^{b} \big(c(x, t_{M-1}) - \hat{c}(x, t_{M-1})\big) \cos\left(k\pi \frac{x-a}{b-a}\right) dx.\end{aligned} \tag{57}$$

Since $V_k(t_M)$ is exact, $\hat{c}(x, t_{M-1})$ resulting from the COS formula only consists of local error $\epsilon(x; N, [a, b])$. So,

$$\varepsilon(k, t_{M-1}) = \int_{x^*_{M-1}}^{b} \epsilon\,(x; N, [a, b]) \cos\left(k\pi \frac{x-a}{b-a}\right) dx. \tag{58}$$

This equation can be seen as an inner product of two square-integrable functions. With the Cauchy-Schwarz inequality, we bound error $\varepsilon(k, t_{M-1})$ as follows:

$$|\varepsilon(k, t_{M-1})| \leq \sqrt{\int_{x^*_{M-1}}^{b} \epsilon^2\,(x; N, [a, b])\, dx} \cdot \sqrt{\int_{x^*_{M-1}}^{b} \cos^2\left(k\pi \frac{x-a}{b-a}\right) dx}. \tag{59}$$

We assume that the integration interval $[a, b]$ is chosen sufficiently large, so that the local error, ϵ, is dominated by the series truncation error ϵ_2. Based on the analysis in Sect. 3.1, it then follows that, for density functions belonging to $C^\infty([a, b])$, error $\epsilon(x; N)$ converges exponentially w.r.t. N, i.e.,

$$|\epsilon(x; N, [a, b])| \leq P(x, N) \exp\left(-(N-1)\nu\right),$$

where $\nu > 0$ is a constant not depending on N and x, and $P(x, N) > 0$ is a function which varies less than exponentially in N. With

$$p(N) := \max_{x\in[a,b]} P(x, N), \tag{60}$$

it then holds that

$$\int_{x^*_{M-1}}^{b} \epsilon^2\left(x; N, [a, b]\right) dx \leq (b - x^*_{M-1}) \cdot \left(p(N) \cdot \exp(-(N-1)\nu)\right)^2.$$

Since $\cos^2(\alpha) \leq 1$, we have

$$\int_{x^*_{M-1}}^{b} \cos^2\left(k\pi \frac{x-a}{b-a}\right) dx \leq (b - x^*_{M-1}).$$

After inserting these parts, Eq. (59) can be written as:

$$|\varepsilon(k, t_{M-1})| \leq (b - x^*_{M-1}) \cdot p(N) \cdot \exp(-(N-1)\nu), \tag{61}$$

for $k = 0, 1, \ldots, N-1$.

This indicates that the convergence behaviour of $\varepsilon(k, t_{M-1})$ in N is as the local error for pricing European options. Written in vector form, with

$$\boldsymbol{\varepsilon}(t_{M-1}) := \left(\varepsilon(0, t_{M-1}), \varepsilon(1, t_{M-1}), \ldots, \varepsilon(N-1, t_{M-1})\right)^T,$$

it follows that

$$|\boldsymbol{\varepsilon}(t_{M-1})|_\infty \leq (b - x^*_{M-1}) \cdot p(N) \cdot \exp(-(N-1)\nu). \tag{62}$$

As a second step, we prove that if the theorem holds for time t_{m+1}, i.e.

$$|\boldsymbol{\varepsilon}(t_{m+1})|_\infty \leq p(N) \exp\left(-(N-1)\nu\right), \tag{63}$$

with $p(N)$ as in (60), then it follows that

$$|\boldsymbol{\varepsilon}(t_m)|_\infty \sim O\left(\exp(-(N-1)\nu)\right),$$

for $m = M-2, M-3, \ldots, 1$.

At time t_m, the definition of error $\varepsilon(k, t_m)$ gives

$$\varepsilon(k, t_m) = \int_{x_m^*}^{b} (c(x, t_m) - \bar{c}(x, t_m)) \cos\left(k\pi \frac{x-a}{b-a}\right) dx, \tag{64}$$

where $\bar{c}(x, t_m)$ is obtained by inserting $\hat{V}_k(t_{m+1})$ into the COS formula. So,

$$\begin{aligned}\bar{c}(x, t_m) &= e^{-r\Delta t} \sum_{j=0}^{N-1}{}' \operatorname{Re}\left\{\varphi\left(\frac{j\pi}{b-a}\right) e^{ij\pi\frac{x-a}{b-a}}\right\} \left(V_j(t_{m+1}) - \varepsilon(j, t_{m+1})\right) \\ &= \hat{c}(x, t_m) - e^{-r\Delta t} \sum_{j=0}^{N-1}{}' \operatorname{Re}\left\{\varphi\left(\frac{j\pi}{b-a}\right) e^{ij\pi\frac{x-a}{b-a}}\right\} \varepsilon(j, t_{m+1}).\end{aligned}$$

Inserted in Eq. (64), we find that $\varepsilon(k, t_m)$ consists of two parts: One related to the local error, as for the European options, and a second related to $\varepsilon(k, t_{m+1})$, i.e.

$$\varepsilon(k, t_m) = \int_{x_m^*}^{b} (\epsilon(x; N, [a, b]) + \bar{\epsilon}(x, t_{m+1})) \cos\left(k\pi \frac{x-a}{b-a}\right) dx, \tag{65}$$

where

$$\bar{\epsilon}(x, t_{m+1}) := e^{-r\Delta t} \sum_{j=0}^{N-1}{}' \operatorname{Re}\left\{\varphi\left(\frac{j\pi}{b-a}\right) e^{ij\pi\frac{x-a}{b-a}}\right\} \varepsilon(j, t_{m+1}). \tag{66}$$

Interchanging summation in (66) with integration in (65) gives the matrix-vector product form for the errors:

$$\boldsymbol{\varepsilon}(t_m) = \boldsymbol{\varepsilon}_1(t_m) + e^{-r\Delta t} \operatorname{Re}\left\{\mathcal{M}(x_m^*, b)\ \Lambda\right\} \boldsymbol{\varepsilon}(t_{m+1}), \tag{67}$$

where the matrices $\mathcal{M}$ and Λ are the same as in (30). Error $\boldsymbol{\varepsilon}_1$ is an N-vector with as k-th element,

$$\int_{x_m^*}^{b} \epsilon(x; N, [a, b]) \cos\left(k\pi \frac{x-a}{b-a}\right) dx, \quad k = 0, 1, \ldots, N-1.$$

Equation (67) explains how $\boldsymbol{\varepsilon}_1(t_m)$ and $\boldsymbol{\varepsilon}(t_{m+1})$ evolve in the backward recursion. To bound $\boldsymbol{\varepsilon}_1$, we can repeat the steps from (58) to (62), to find:

$$|\boldsymbol{\varepsilon}_1(t_m)|_\infty \leq (b - x_m^*) \cdot p(N) \cdot \exp\left(-(N-1)\nu\right). \tag{68}$$

For the term $e^{-r\Delta t}\mathrm{Re}\{\mathcal{M}(x_m^*, b)\ \Lambda\}\boldsymbol{\varepsilon}(t_{m+1})$, whose k-th element reads

$$\int_{x_m^*}^{b} \bar{\epsilon}(x, t_{m+1}) \cos\left(k\pi \frac{x-a}{b-a}\right) dx,$$

we start with the definition of $\bar{\epsilon}(x, t_{m+1})$ in (66) and apply the Cauchy-Schwarz inequality, as follows

$$\left(e^{r\Delta t}\bar{\epsilon}(x, t_{m+1})\right)^2 \leq \sum_{j=0}^{N-1}{}' \left(\mathrm{Re}\left\{\varphi\left(\frac{j\pi}{b-a}\right) e^{ij\pi\frac{x-a}{b-a}}\right\}\right)^2 \sum_{j=0}^{N-1}{}' \varepsilon^2(j, t_{m+1}). \tag{69}$$

From $A_j(x)$ and $F_j(x)$, as defined in (11) and (12), respectively, we have

$$\mathrm{Re}\left\{\varphi\left(\frac{j\pi}{b-a}\right) e^{ij\pi\frac{x-a}{b-a}}\right\} = \frac{1}{2}(b-a)\left[A_j(x) + \int_{\mathbb{R}\setminus[a,b]} f(y|x) \cos\left(j\pi\frac{y-a}{b-a}\right) dy\right].$$

Since $f(y|x) \in \mathbb{R}^+$, it is clear that $f(y|x)\cos(\alpha) \leq f(y|x)$, and thus,

$$\left(\mathrm{Re}\left\{\varphi\left(\frac{j\pi}{b-a}\right) e^{ij\pi\frac{x-a}{b-a}}\right\}\right)^2 \leq \frac{1}{4}(b-a)^2 \left[A_j^2(x) + 2A_j(x) \int_{\mathbb{R}\setminus[a,b]} f(y|x)dy + \left(\int_{\mathbb{R}\setminus[a,b]} f(y|x)dy\right)^2\right]. \tag{70}$$

Assuming now that the interval of integration is set sufficiently large, so that the related truncation error can be neglected, and including the leading term of (70) into (69), one finds

$$\left(e^{r\Delta t}\bar{\epsilon}(x, t_{m+1})\right)^2 \leq \frac{1}{4}(b-a)^2 \sum_{j=0}^{N-1}{}' A_j^2(x) \sum_{j=0}^{N-1}{}' \varepsilon^2(j, t_{m+1}).$$

For density functions belonging to $\mathrm{C}^\infty([a, b])$, the series coefficients $A_j(x)$ converge exponentially in j, see [6], so that $\sum_{j=0}^{N-1}{}' A_j^2(x)$ represents the sum of a geometric

series and is therefore bounded. Define

$$W := \max_{x\in[a,b]} \sum_{j=0}^{N-1}{}' A_j^2(x).$$

With Assumption (63) one obtains:

$$|\bar{\epsilon}(x, t_{m+1})| \leq \frac{1}{2}(b-a)e^{-r\Delta t}\sqrt{NW}\,p(N)e^{-(N-1)\nu}. \tag{71}$$

Application of the Cauchy-Schwarz inequality results in:

$$\left| \int_{x_m^*}^{b} \bar{\epsilon}(x, t_{m+1}) \cos\left(k\pi \frac{x-a}{b-a}\right) dx \right| \leq (b - x_m^*)\,\bar{p}(N)e^{-(N-1)\nu}, \tag{72}$$

or, in vector form:

$$e^{-r\Delta t} \left|\mathrm{Re}\left\{\mathcal{M}(x_m^*, b)\ \Lambda\right\} \boldsymbol{\varepsilon}(t_{m+1})\right|_\infty \leq (b - x_m^*)\,\bar{p}(N)e^{-(N-1)\nu}, \tag{73}$$

where $\bar{p}(N) := \frac{1}{2}(b-a)e^{-r\Delta t}\sqrt{NW}\,p(N)$. Inserting (68) and (73) in (67) completes the proof. □

Summarizing, when the local error evolves through time, via the backward recursion, the method's convergence rate does not change. This is an indication for the method's stability.

Similarly, we can prove that if the local error converges algebraically, so does $\varepsilon(k, t_m)$.

Remark 3.1 The choice of integration range, $[a, b]$, is quite important. An interval which is chosen too small will lead to a significant integration-range truncation error, whereas an interval which is set very large would require a large value for N to achieve a certain level of accuracy, as determined in (68) and (73).

3.3 Choice of truncation range

As a rule of thumb, we propose to use the following formula to define the range of integration in (3):

$$[a, b] := \left[(c_1 + x_0) - L\sqrt{c_2 + \sqrt{c_4}},\quad (c_1 + x_0) + L\sqrt{c_2 + \sqrt{c_4}}\right], \tag{74}$$

where $x_0 := \ln(S_0/K)$ and L depends on the *user-defined tolerance level*, TOL, as given in (3). $c_1, \ldots, c_4$ are the cumulants, based on the characteristic function of the underlying process, and detailed in Appendix B.

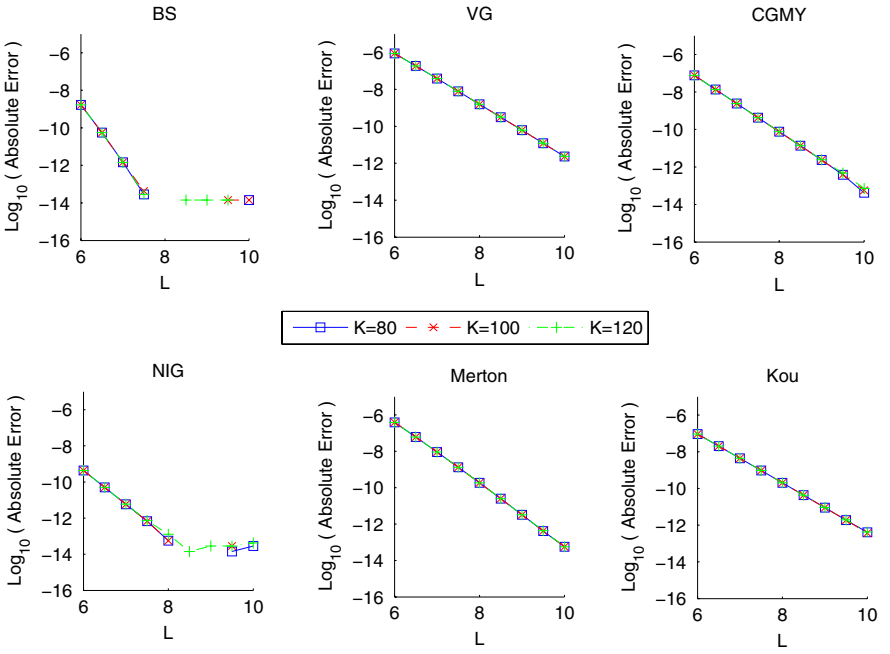

Fig. 1 L versus the logarithm of the absolute errors for pricing calls by the COS method with $N = 2^{14}$, $T = 1$ year and three different strike prices

Cumulant c_4 is included in (74), because, for short maturities, the density functions of many Lévy processes have sharp peaks and flat tails, and this behavior can be well captured by the inclusion of c_4.

Here, we analyze the relation between TOL and L in (74) via numerical experiments, aiming to determine one value of L for different exponential Lévy asset price processes. We present the observed error for different values of L in Fig. 1. With N large, e.g. $N = 2^{14}$, the series truncation error is negligible and the integration range error, which has a direct relation to the user-defined TOL, dominates. The results in Fig. 1 can therefore be used as a guidance for setting parameter L, given a tolerance TOL. In the figure, and throughout this paper, BS denotes the Black-Scholes model (Geometric Brownian Motion), VG stands for the Variance Gamma model [26], CGMY denotes the model from [10], NIG is short for the Normal Inverse Gaussian Lévy process [5], Merton denotes the jump-diffusion model developed in [27], and Kou is the jump-diffusion model from [24].

We see in Fig. 1 that the integration range error decreases exponentially with L.

The use of $L = 8$ seems appropriate for all the Lévy processes considered. This value is used in all numerical experiments to follow. Via experiments, we also found that formula (74), together with a proper choice of L, defines an appropriate truncation range for any maturity time longer than 0.1 years. For even shorter maturities, one can use a larger value of L.

Table 1 Test parameters for pricing Bermudan options

Test no.	Model	S_0	K	T	r	σ	Other parameters
1	BS	100	110	1	0.1	0.2	–
2	CGMY	100	80	1	0.1	0	$C = 1, G = 5, M = 5, Y = 1.5$

4 Numerical results

We will show the method's impressive convergence by pricing Bermudan, American and discretely-monitored barrier options. In the following, we present numerical results for the BS, CGMY and NIG models. Extensive tests (not given here) have demonstrated that the COS method also shows excellent performance for other Lévy processes. The characteristic functions as well as the cumulants for several exponential Lévy asset price processes are listed in Appendix B.

The computer used has an Intel Pentium 4 CPU, 2.80 GHz with cache size 1024 KB; The code is written in Matlab 7.4. The CPU times for all experiments to follow are averaged over 100 repeated tests.

In order to observe the exponential error convergence, we define a ratio,

$$\text{ratio} = \frac{\ln\left(\left|err(2^{d+1})\right|\right)}{\ln\left(\left|err(2^{d})\right|\right)}, \quad d \in \mathbb{Z}^{+}, \tag{75}$$

where $err(2^d)$ denotes the error between reference solution and approximation obtained with $N = 2^d$. If $err(N) = C_1 \exp(-P_1 N)$ with C_1 and P_1 not depending on N, this ratio should be equal to 2; If the error convergence is algebraic, i.e. $err(N) = C_2 N^{-P_2}$ with C_2 and P_2 not depending on N, this ratio should equal $(d+1)/d$.

Next to the series and the integration range truncation error, another error for Bermudan options is related to the stopping criterion of the root-searching algorithm, i.e., Newton's method. With an initial guess $x^*_{m+1} = x^*_m$, $m = M-2, \ldots, 2$ (and $x^*_{M-1} = 0$), this error becomes sufficiently small, of $O(10^{-7})$ in 4 Newton iterations and of $O(10^{-10})$ in 5 iterations. In the experiments to follow, we use 5 iterations.

4.1 Bermudan and American options

Here we price Bermudan put options with 10 exercise dates. Test parameters for two test cases are given in Table 1. These parameters are related to the characteristic functions presented in Table 8 and the cumulants from Table 9.

The CPU times are reported in milli-seconds, and all reference values are obtained by another method, i.e., by the CONV method from [25], setting $N = 2^{20}$.

The first test is for the classical BS model with as the reference value 10.479520123. In Fig. 2a it is shown that a highly accurate solution is obtained in less than 20 ms with exponential convergence (the log-error plot displays a straight line). Compared to the

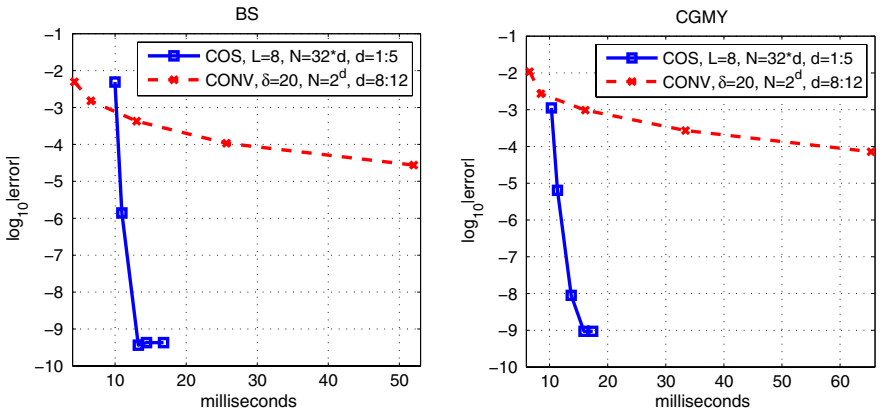

Fig. 2 Error versus CPU time for pricing Bermudan put options under **a** BS (Test no. 1) and **b** CGMY ($Y = 1.5$, Test no. 2) model, comparing the COS and the CONV method

quadrature-rule based CONV method, which exhibits a second-order convergence, we see a significant improvement in the CPU time.

As the second test, we consider a Lévy process of infinite activity, i.e., the CGMY model with $Y > 1$ (Test 2 in Table 1). For this set of CGMY parameters it is now well-known that PIDE-based methods have convergence difficulties [2,34]. The reference value is found to be 28.829781986 . . .. The performance of the COS method for this test, shown in Fig. 2b is highly efficient. Again, in less than 20 ms, the solution is accurate to 9 digits, compared to the reference value. Also here, we observe the exponential error convergence of the COS method.

Remark 4.1 (VG and algebraic convergence) In [18] it was shown that for certain sets of parameters the Variance Gamma (VG) process gives rise to a probability density function which is not in $C^\infty(\mathbb{R})$, and thus option pricing under VG with these parameter sets exhibits only an algebraic convergence. This is observed for contracts with $T < \nu$, where ν denotes the variance of the VG model, see the characteristic function in Appendix B.

When dealing with Bermudan options this also implies that we will encounter algebraic convergence when the time between two exercise dates, $\Delta t < \nu$.

The prices of American options can be obtained by applying a *Richardson extrapolation* on the prices of a few Bermudan options with small M [12], as demonstrated, for example, in [25]. Let $v(M)$ denote the value of a Bermudan option with M early exercise dates. We will use the following 4-point Richardson extrapolation scheme,

$$v_{AM}(d) = \frac{1}{21}\left(64v(2^{d+3}) - 56v(2^{d+2}) + 14v(2^{d+1}) - v(2^d)\right), \tag{76}$$

where $v_{AM}(d)$ denotes the approximated value of the American option.

Now we price an American option using (76) with the 4-point Richardson extrapolation on Bermudan puts and vary the number of exercise dates. The parameters,

presented in Table 2, are taken from [1] and the reference value given was $V(0) = 0.112152$. We deal with the pure Lévy CGMY jump model ($\sigma = 0$) and no dividend payment ($q = 0$) here.

We compare the results of the COS method with those obtained by the CONV method using the same extrapolation. For the COS method, $N = 512$ and the number of Newton iterations is 5; For the CONV method $N = 4{,}096$ to reach a very similar accuracy. The accuracy of the American prices then mainly depends on parameter d in the extrapolation (76). Results are summarized in Table 3. We can see that large values of d give highly accurate results. The COS method in combination with Richardson extrapolation gives, however, a very satisfactory accuracy within 75 ms.

4.2 Barrier options

Now we price monthly-monitored ($M = 12$) up-and-out call and put options, (UOC) and (UOP), down-and-out call and put options, (DOC) and (DOP), by the COS method. The test parameters are in Table 4, again related to the characteristic functions in Table 8. We solve the same problems as in [19] with the barrier level, $H = 120$ for the up-and-out and $H = 80$ for the down-and-out options.

The numerical results under the CGMY model (Test 4) are presented in Table 5. The CPU times are again measured in milli-seconds, and the reference values are obtained by the CONV method [25], with $N = 2^{15}$. Note that "ratio", as presented in the table, is different from the commonly used ratio defining the rate of convergence. In (75), it is the ratio of the *logarithm* of two consecutive errors. This ratio should be equal to two in the case of exponential convergence.

As expected, the COS method is more efficient for discrete barrier options than for Bermudan options, because the barrier levels are known in advance.

Exponential error convergence is observed, as the ratios (75) are around 2, in less than 5 ms with the results accurate up to 8 decimal places.

Table 2 Parameters for American put options under the CGMY model

Test no.	S_0	K	T	r	Other parameters
3	1	1	1	0.1	$C = 1, G = 5, M = 5, Y = 0.5$

Table 3 Errors and CPU times for pricing American puts under CGMY model, Test no. 3

d in Eq. (76)	COS		CONV	
	Error	Time (ms)	Error	Time (ms)
0	4.41e-05	71.41	4.37e-05	134.4
1	7.69e-06	109.2	7.01e-06	198.0
2	9.23e-07	219.3	1.05e-06	336.7
3	3.04e-07	438.9	1.29e-07	610.9

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	$C = 4, G = 50, M = 60, Y = 0.7$
5	NIG	100	100	1	0.05	0.02	$\alpha = 15, \beta = -5, \delta = 0.5$

Table 5 Errors and CPU times for pricing monthly-monitored barrier options under the CGMY model (Test no. 4)

Option type	Ref. val.	N	Time (ms)	Error	Ratio
DOP	2.339381026	2^4	2.8	2.23e-1	–
		2^5	2.7	1.98e-2	2.6
		2^6	3.4	3.23e-4	2.0
		2^7	4.6	7.20e-9	2.3
DOC	9.155070561	2^4	2.7	5.06e-2	–
		2^5	2.9	5.67e-3	1.7
		2^6	3.3	1.99e-4	1.6
		2^7	4.7	5.55e-9	2.2
UOP	6.195603554	2^4	3.0	5.58e-2	–
		2^5	2.9	8.98e-3	1.6
		2^6	3.6	1.96e-4	1.8
		2^7	4.8	2.23e-8	2.1
UOC	1.814827593	2^4	2.8	3.38e-1	–
		2^5	2.8	1.24e-2	4.0
		2^6	3.5	3.45e-6	2.9
		2^7	4.7	1.93e-8	1.4

Next, we focus on the NIG model (Test 5) and repeat the barrier option tests in Table 6. To reach the same level of accuracy as for CGMY, we need a slightly larger value of N under the NIG model. This is because the NIG density function is more peaked with the parameters from Table 4, as shown in Fig. 3a. Consequently, one typically requires some more terms in the series to reconstruct the density function from its Fourier-cosine series expansion. Nevertheless, the performance of the COS method is still excellent: In less than 10 ms, the accuracy is up to the 7-th decimal place.

Note that, the smaller the value of Δt, the larger the value of N needs to be chosen to reach the same level of accuracy. This is because many Lévy processes have highly peaked density functions for very small Δt. An example is presented in Fig. 3b, where the recovered density functions of the NIG model for monthly-, weekly- and daily-monitored barrier options are plotted. We can see that for $\Delta t = 1/252$ the density is highly peaked, compared to $\Delta t = 1/12$. Nevertheless, as long as the density function is in $\mathrm{C}^\infty(\mathbb{R})$, the error convergence rate is exponential.

Table 6 Errors and CPU times for pricing monthly-monitored barrier options under the NIG model (Test no. 5)

Option type	Ref. val.	N	Time (ms)	Error	Ratio
DOP	2.139931117	2^6	3.1	4.25e-2	–
		2^7	3.7	1.28e-3	2.1
		2^8	5.4	4.65e-5	1.5
		2^9	8.4	1.39e-7	1.6
		2^{10}	14.7	1.38e-12	1.7
DOC	8.983106036	2^6	3.1	1.26e-2	–
		2^7	3.7	1.09e-3	1.6
		2^8	5.3	3.99e-5	1.5
		2^9	8.3	9.47e-8	1.6
		2^{10}	14.8	5.61e-13	1.7
UOP	5.995341168	2^6	3.4	4.84e-3	–
		2^7	3.7	1.14e-3	1.3
		2^8	5.3	7.50e-5	1.4
		2^9	8.3	1.52e-7	1.7
		2^{10}	14.7	1.24e-12	1.7
UOC	2.277861597	2^6	3.1	3.83e-2	–
		2^7	3.7	1.10e-3	2.1
		2^8	5.5	8.67e-5	1.4
		2^9	8.6	7.98e-8	1.7
		2^{10}	15.1	7.38e-13	1.7

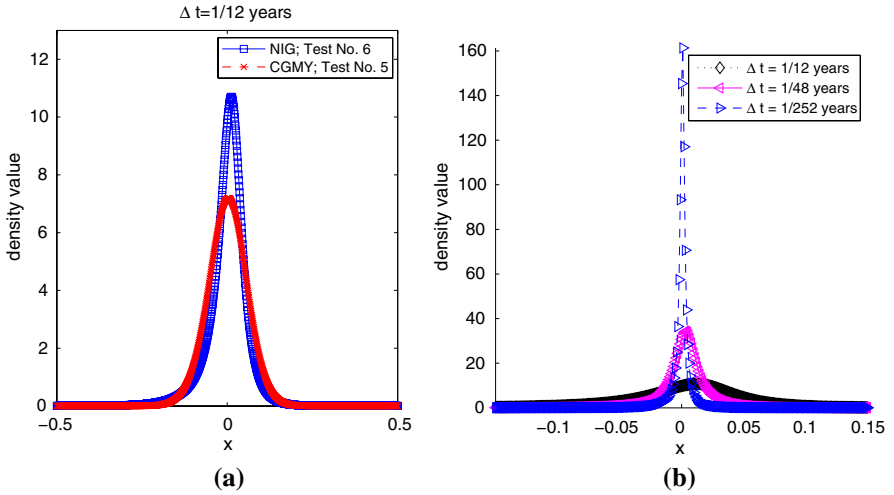

Fig. 3 The recovered density functions for (**a**, the NIG and the CGMY models and *monthly-monitored* barrier options and **b** the NIG model for *monthly-, weekly- and daily-monitored* barrier options. **a** NIG and CGMY with $\Delta t = \frac{1}{12}$, **b** NIG with $\Delta t = \frac{1}{12}, \frac{1}{48}$, and $\frac{1}{252}$

Table 7 Errors and CPU times for pricing daily-monitored ($M = 252$) barrier options under the NIG model (Test 5)

Option type	Ref. val.	N	Time (ms)	Error	Ratio
DOP	1.88148753	2^9	130	1.25e-2	–
		2^{10}	230	2.20e-3	1.4
		2^{11}	460	1.32e-4	1.5
		2^{12}	1170	1.98e-6	1.5
		2^{13}	2560	4.70e-8	1.3
DOC	8.96705248	2^9	140	3.67e-4	–
		2^{10}	230	9.18e-5	1.2
		2^{11}	460	3.14e-5	1.1
		2^{12}	950	2.00e-6	1.3
		2^{13}	2430	5.73e-9	1.4

We now price *daily-monitored* DOP and DOC options under the NIG model with the parameters from Test 5 in Table 4. The reference values are taken from [19]. The results with the COS method are summarized in Table 7. We observe that, as expected, the convergence rate of the COS method is exponential, but the values of N are somewhat larger than in the previous numerical experiments. The almost linear computational complexity of the method can clearly be observed from this table.

For results accurate up to the 4th digit, the COS method needs about 0.2 seconds for the daily-monitored DOP as well as for the DOC.

Remark 4.2 (Comparison to hilbert transform method) The complexity of the COS method is $O((M-1)N\log_2(N))$, as the length of the induction loop (in which the FFT is employed) is $M-1$, and the final step uses N operations. Additionally, its error convergence is exponential for models with density function in the class $C^\infty([a,b])$. By considering both complexity and error convergence, the COS method is as efficient as the Hilbert transform method in [19]. The experiments above show that the COS method is as fast in terms of CPU time (although we have a slower CPU and the code is written in Matlab). That method cannot be used to price Bermudan options, as the information of the early-exercise points is not known in advance. Moreover, the COS method uses more-or-less the same CPU time for different types of barrier options, which is not the case in [19].

5 Conclusions and discussion

In this paper, we have generalized the COS option pricing method, based on Fourier-cosine expansions, to Bermudan and discretely-monitored barrier options. The method can be used whenever the characteristic function of the underlying price process is available (i.e., for regular affine diffusion processes and, in particular, for exponential Lévy processes).

The main insights in the paper are that the COS formula for European options from [18] can be used for pricing Bermudan and barrier options, if the series coefficients of the option values at the first early-exercise (or monitoring) date are known. These coefficients can be recursively recovered from those of the payoff function. The computational complexity is $O((M-1)N\log_2 N)$, for Bermudan and barrier options with M exercise, or monitoring, dates. The COS method exhibits an exponential convergence in N for density functions in $\mathrm{C}^\infty[a,b]$ and an impressive computational speed. With a small N, it typically produces highly accurate results. For example, with $N=128$, results are accurate up to the 8th digit in less than 20 ms for 10-times exercisable Bermudan options and less than 10 ms for monthly-monitored barrier options. We expect a further factor of three gain in computational speed when replacing the Matlab implementation by an implementation in C.

However, the smaller the time interval between two consecutive dates, the more peaked the underlying density function, and thus larger values of N are required for a similar accuracy. For problems with small time intervals, like daily-monitored barrier options, the COS method shows a similar performance as the Hilbert transform based method [19].

Compared to the CONV method [25], which is one of the fast methods for Bermudan options, the COS method converges significantly faster to the same level of accuracy. Pricing American options can be done by a Richardson extrapolation method on Bermudan options with a varying number of exercise dates.

Appendix A: From Bermudan to American options

In this section, we discuss the behaviour of the error if M, the number of early-exercise dates, goes to infinity. We also check how the Bermudan option prices converge to their American option counterparts.

American options can, in this framework, be priced basically by two approaches. One can either price a Bermudan option with very many exercise dates, or employ extrapolation methods. Whereas the latter approach is much more practical, in terms of CPU time, and has been used in Sect. 4.1, the former approach is interesting from a stability point-of-view. Here we therefore consider the pricing of Bermudan options with many exercise dates, for reasons of stability. It is interesting to consider the limit case, and check whether the method presented is still applicable.

The series truncation error, ϵ_2, may be problematic at first sight, for $\Delta t \to 0$. For small time intervals the transitional probability density function tends to become highly peaked. However, by letting the number of terms in the Fourier-cosine expansion increase, for $\Delta t \to 0$, the method can deal with such highly peaked functions, as long as they are in $\mathrm{C}^\infty[a,b]$. Moreover, the size of the integration range is, by means of the cumulants involved in (74), automatically adapted to the shape of the function.

A.1 Density recovery: influence of adapted truncation range

From the discussion above it is clear that the recovery of the probability density functions of the Lévy processes, for $\Delta t \to 0$, from their Fourier cosine series

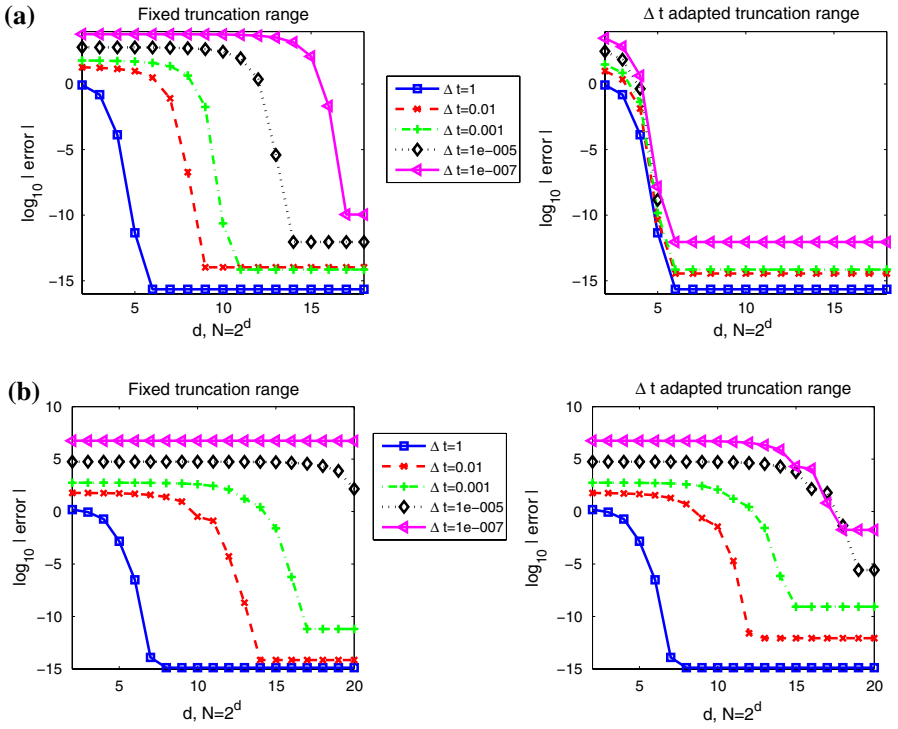

Fig. 4 The recovered density functions for $\Delta t \to 0$; **a** the GBM model with fixed and adapted integration range, **b** the NIG model with fixed and adapted ranges

expansion, is crucial. Figure 4 shows the importance of the proper adaptive choice of the integration range, by means of the cumulants in (74). For several values of Δt, with even $\Delta t = 10^{-7}$, the density recovery with a fixed and an adapted integration range are compared, for GBM as well as for NIG. It is clear that the adaptive choice of integration range is superior for the recovery. Whereas, we see that for GBM the density can be recovered on the adapted integration range without significant difficulties, when $\Delta t \to 0$, it is less trivial for the NIG process. For this latter process, the recovery gets difficult for the smallest time interval, even with the interval adaptation.

A.2 Stability of the method

Based on Theorem 3.1 and its proof in Sect. 3.2, we saw an exponential convergence in N. Inequality (73) indicates that the proportionality constant in the convergence estimate may grow with the number of exercise dates, M, so that with N fixed the error may increase substantially for increasing M.

However, the error convergence is as fast as exponential in N, so that one can achieve very high accuracy by slightly increasing N.

We show here, by means of some numerical experiments that the resulting error in $\hat{\mathbf{V}}(t_1)$ is bounded. In Fig. 5, we present the convergence for Bermudan call options

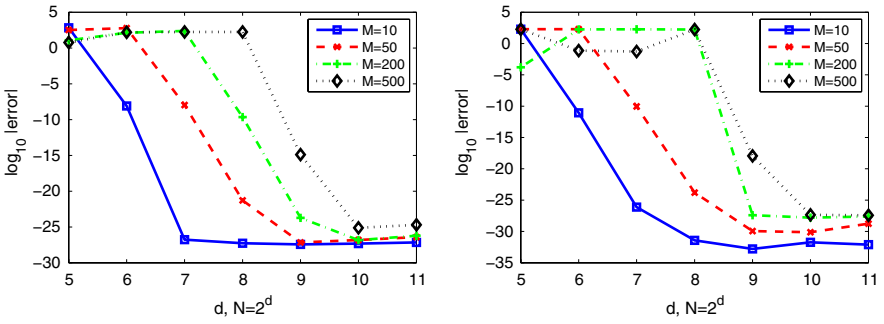

Fig. 5 Error convergence for increasing M and N. **a** $T = 1$ year, **b** $T = 0.1$ year

under GBM with a varying number of exercise dates with respect to the number of terms in the cosine series, N. Shown is the logarithm of the error, $\log_{10}|\text{err}|$, versus N. The dividend rate is set to zero, so that there are no early-exercise opportunities, i.e. $x_m^* = b$ for $m = 0, 1, \ldots, M-1$, and thus, the Bermudan call options have the same values as their European counterparts. The other parameters are as in Table 1. The truncation range is defined according to the description in Sect. 3.3.

We see a convergence of extreme accuracy, because the reference values are also obtained by the COS method for European options, and therefore the error related to the truncation range cannot be observed.

For two maturities, $T = 1$ year and short maturity $T = 0.1$ year, relatively large values of M do not show any significant impact on the error convergence with respect to N. Merely, the start of convergence shifts to larger values of N as M increases, which confirms the intuition that higher values for N can compensate for the higher peakedness of the density function. With N small, however, the error presented remains bounded.

A.3 Convergence of Bermudan to American option prices

It is also interesting to study the convergence of the Bermudan option prices to their American equivalents, by increasing M, fixing N. In Fig. 6, the logarithm of the difference in values of Bermudan put options with 2,000 exercise times and with M (varying) between 0 and 1,000 exercise dates is presented, under the GBM and CGMY models. We set $T = 0.1$ year maturity and use $N = 2^{12}$. Other parameters are given in Table 1.

The figure shows that, when N is sufficiently large, the COS method converges for the problems presented. The convergence of Bermudan to American option prices is evident from the figures. We observe a linear convergence of the Bermudan options, with an increasing number of exercise dates, to the American price by means of the quantity $q(M) = [\bar{v}_{AM} - \bar{v}(M)]/[\bar{v}_{AM} - \bar{v}(2M)]$. We find $q(M) = 2$ for both GBM and the CGMY test case, which indicates linear convergence with respect to Δt. The convergence speed, however, is model-dependent, and is related to the decay rate of

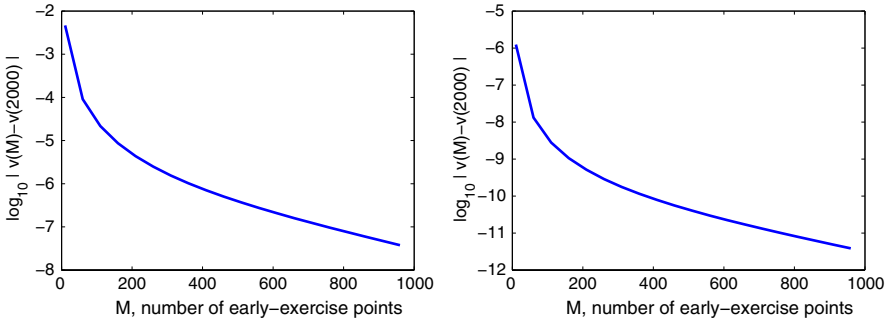

Fig. 6 Convergence of Bermudan options to American options as $M \to \infty$

the characteristic function. For a Bermudan call option without any dividend payment, as an example, (29) reads

$$\hat{\mathbf{V}}(t_m) = \hat{\mathbf{C}}(a, b, t_m) = \begin{cases} \text{Re}\{\mathcal{M}(a,b)\ \Lambda\}\,\mathbf{V}(t_M) & m = M-1 \\ \text{Re}\{\mathcal{M}(a,b)\ \Lambda\}\,\hat{\mathbf{V}}(t_{m+1}) & m = 1, 2, \ldots, M-2. \end{cases}$$

and thus,

$$\hat{\mathbf{V}}(t_1) = (\text{Re}\{\mathcal{M}(a,b)\ \Lambda\})^{M-1}\,\mathbf{V}(t_M). \tag{77}$$

Since neither the matrix $\mathcal{M}$ nor $\mathbf{V}(t_M)$ depends on the type of underlying process, the important matrix here is Λ, the diagonal matrix with the elements being related to the characteristic function.

A.3.1 Extrapolation and American option pricing

In the present article the price of an American option is computed from the Bermudan option prices by means of the repeated Richardson extrapolation scheme by Chang et al. [12]. Let $\hat{v}(M)$ be the price of a Bermudan option with a maturity of T years and M exercise dates, which are $\Delta t = T/M$ years apart. It is assumed that $\hat{v}(M)$ can be expanded as

$$\hat{v}(M) = \hat{v}(0) + \sum_{i=1}^{\infty} a_i (\Delta t)^{\gamma_i}, \tag{78}$$

with $0 < \gamma_i < \gamma_{i+1}$. $\hat{v}(0) \equiv v_{AM}(0)$ represents the price of the American option. Repeated Richardson extrapolation is a well-known technique for improving the accuracy of a solution. It will converge highly satisfactory if $\gamma_i \geq 1$ in Eq. (78). Classical extrapolation procedures assume that the exponents γ_i are known, which means that we can use $n+1$ Bermudan prices with varying Δt in order to eliminate n of the leading order terms in Eq. (78). The only paper considering an expansion of the Bermudan

option price in terms of Δt is [22], finding that $\gamma_1 = 1$ for the Black-Scholes model. Numerical tests also indicate that the assumption $\gamma_i = i$ produces highly satisfactory results for the Lévy models in this paper.

Appendix B: Characteristic functions and cumulants

The COS method requires from the underlying process the characteristic function to be known. The method fits therefore well to exponential Lévy models, whose characteristic functions are available in closed form. The motivation behind using general Lévy processes for the underlying is the fact that the Black-Scholes model is not able to reproduce the volatility skew or smile present in most financial markets, whereas it has been shown that several exponential Lévy models can, at least to some extent.

In exponential Lévy models the asset price is modeled as an exponential function of a Lévy process $L(t)$:

$$S(t) = S(0)\exp(L(t)). \tag{79}$$

A process $L(t)$, with $L(0) = 0$, is a Lévy process if:

1 it has independent increments;
2 it has stationary increments;
3 it is stochastically continuous, i.e., for any $t \geq 0$ and $\epsilon > 0$ we have

$$\lim_{s \to t} \mathbb{P}(|L(t) - L(s)| > \epsilon) = 0. \tag{80}$$

Each Lévy process can be characterized by a triplet (μ, σ, ζ) with $\mu \in \mathbb{R}, \sigma \geq 0$ and ζ a measure satisfying $\zeta(0) = 0$ and

$$\int_{\mathbb{R}} \min(1, |x|^2)\zeta(dx) < \infty. \tag{81}$$

In terms of this triplet the characteristic function of the Lévy process is available in closed form, due to the celebrated Lévy-Khinchine formula. We recall the formulas for the characteristic function for several exponential Lévy processes in Table 8. For more background information on these processes we point you to [13,30] for the usage of Lévy processes in a financial context and to [29] for a detailed analysis of Lévy processes in general. With respect to the parameters for the various processes in Table 8 we also basically follow the books [13,30].

Given the characteristic functions, the cumulants, defined in [13], can be computed via

$$c_n(X) = \frac{1}{i^n} \frac{\partial^n (t\Psi(\xi))}{\partial \xi^n}\bigg|_{\xi=0},$$

Table 8 Characteristic functions of $\ln(S_t/K)$ for various models

BS	$\varphi(\xi,t)=\exp\left(i\xi\mu t-\frac{1}{2}\sigma^2\xi^2 t\right)$
NIG	$\varphi(\xi,t)=\exp\left(i\xi\mu t-\frac{1}{2}\sigma^2\xi^2 t\right)\phi_{\text{NIG}}(\xi,t;\alpha,\beta,\delta)$
	$\phi_{\text{NIG}}(\xi,t;\alpha,\beta,\delta)=\exp\left[\delta t\left(\sqrt{\alpha^2-\beta^2}-\sqrt{\alpha^2-(\beta+i\xi)^2}\right)\right]$
Kou	$\varphi(\xi,t)=\exp\left(i\xi\mu t-\frac{1}{2}\sigma^2\xi^2 t\right)\phi_{\text{Kou}}(\xi,t;\lambda,p,\eta_1,\eta_2)$
	$\phi_{\text{Kou}}(\xi,t;\lambda,p,\eta_1,\eta_2)=\exp\left[\lambda t\left(\frac{p\eta_1}{\eta_1-i\xi}-\frac{(1-p)\eta_2}{\eta_2+i\xi}-1\right)\right]$
Merton	$\varphi(\xi,t)=\exp\left(i\xi\mu t-\frac{1}{2}\sigma^2\xi^2 t\right)\phi_{\text{Merton}}(\xi,t;\lambda,\bar{\mu},\bar{\sigma})$
	$\phi_{\text{Merton}}(\xi,t)=\exp\left[\lambda t\left(\exp(i\bar{\mu}\xi-\frac{1}{2}\bar{\sigma}^2\xi^2)-1\right)\right]$
VG	$\varphi(\xi,t)=\exp\left(i\xi\mu t\right)\phi_{VG}(\xi,t;\sigma,\nu,\theta)$
	$\phi_{VG}(\xi,t;\sigma,\nu,\theta)=(1-i\xi\theta\nu+\frac{1}{2}\sigma^2\nu\xi^2)^{-t/\nu}$
CGMY	$\varphi_{\ln(S_t/K)}(\xi,t;x)=\exp\left(i\xi\mu t-\frac{1}{2}\sigma^2\xi^2 t\right)\phi_{CGMY}(\xi,t;C,G,M,Y)$
	$\phi_{CGMY}(\xi,t;C,G,M,Y)=\exp(Ct\Gamma(-Y)[(M-i\xi)^Y-M^Y+(G+i\xi)^Y-G^Y])$

Table 9 Cumulants of $\ln(S_t/K)$ for various models

BS	$c_1=(\mu-\frac{1}{2}\sigma^2)t,\quad c_2=\sigma^2 t,\quad c_4=0$	
NIG	$c_1=(\mu-\frac{1}{2}\sigma^2+w)t+\delta t\beta/\sqrt{\alpha^2-\beta^2}$	
	$c_2=\delta t\alpha^2(\alpha^2-\beta^2)^{-3/2}$	
	$c_4=3\delta t\alpha^2(\alpha^2+4\beta^2)(\alpha^2-\beta^2)^{-7/2}$	
	$w=-\delta(\sqrt{\alpha^2-\beta^2}-\sqrt{\alpha^2-(\beta+1)^2})$	
Kou	$c_1=t\left(\mu+\frac{\lambda p}{\eta_1}+\frac{\lambda(1-p)}{\eta_2}\right)$	$c_2=t\left(\sigma^2+2\frac{\lambda p}{\eta_1^2}+2\frac{\lambda(1-p)}{\eta_2^2}\right)$
	$c_4=24t\lambda\left(\frac{p}{\eta_1^4}+\frac{1-p}{\eta_2^4}\right)$	$w=\lambda\left(\frac{p}{\eta_1+1}-\frac{1-p}{\eta_2-1}\right)$
Merton	$c_1=t(\mu+\lambda\bar{\mu})$	$c_2=t\left(\sigma^2+\lambda\bar{\mu}^2+\bar{\sigma}^2\lambda\right)$
	$c_4=t\lambda\left(\bar{\mu}^4+6\bar{\sigma}^2\bar{\mu}^2+3\bar{\sigma}^4\lambda\right)$	
VG	$c_1=(\mu+\theta)t$	$c_2=(\sigma^2+\nu\theta^2)t$
	$c_4=3(\sigma^4\nu+2\theta^4\nu^3+4\sigma^2\theta^2\nu^2)t$	$w=\frac{1}{\nu}\ln(1-\theta\nu-\sigma^2\nu/2)$
CGMY	$c_1=\mu t+Ct\Gamma(1-Y)\left(M^{Y-1}-G^{Y-1}\right)$	
	$c_2=\sigma^2 t+Ct\Gamma(2-Y)\left(M^{Y-2}+G^{Y-2}\right)$	
	$c_4=Ct\Gamma(4-Y)\left(M^{Y-4}+G^{Y-4}\right)$	
	$w=-C\Gamma(-Y)[(M-1)^Y-M^Y+(G+1)^Y-G^Y]$	

where w is the drift correction term that satisfies $\exp(-wt)=\varphi(-i,t)$.

where $t\Psi(\xi)$ is the exponent of the characteristic function $\varphi(\xi,t)$, i.e.

$$\varphi(\xi,t)=e^{t\Psi(\xi)},\quad t\geq 0.$$

The formulas for the cumulants are summarized in Table 9. They have been confirmed with the help of Mathematica.

References

1. Almendral, A., Oosterlee, C.W.: Accurate evaluation of European and American options under the CGMY process. SIAM J. Sci. Comput. **29**, 93–117 (2007)
2. Almendral, A., Oosterlee, C.W.: On American options under the Variance Gamma process. Appl. Math. Finance **14**(2), 131–152 (2007)
3. Andricopoulos, A.D., Widdicks, M., Duck, P.W., Newton, D.P.: Universal option valuation using quadrature methods. J. Fin. Econ. **67**, 447–471 (2003)
4. Andricopoulos, A.D., Widdicks, M., Duck, P.W., Newton, D.P.: Extending quadrature methods to value multi-asset and complex path dependent options. J. Fin. Econ. **83**, 471–499 (2007)
5. Barndorff-Nielsen, O.E.: Normal inverse Gaussian distributions and stochastic volatility modelling. Scand. J. Stat. **24**, 1–13 (1997)
6. Boyd, J.P.: Chebyshev and Fourier Spectral Methods. Springer, Berlin (1989)
7. Broadie, M., Yamamoto, Y.: Application of the fast Gauss transform to option pricing. Manag. Sci. **49**(8), 1071–1088 (2003)
8. Broadie, M., Yamamoto, Y.: A double-exponential fast Gauss transform for pricing discrete path-dependent options. Oper. Res. **53**(5), 764–779 (2005)
9. Cariboni, J., Schoutens, W.: Pricing credit default swaps under Lévy models. J. Comp. Finance **10**(4), 71–91 (2008)
10. Carr, P.P., Geman, H., Madan, D.B., Yor, M.: The fine structure of asset returns: an empirical investigation. J. Bus. **75**, 305–332 (2002)
11. Carr, P.P., Madan, D.B.: Option valuation using the Fast Fourier Transform. J. Comp. Finance **2**, 61–73 (1999)
12. Chang, C.-C., Chung, S.-L., Stapleton, R.C.: Richardson extrapolation technique for pricing American-style options. J. Futures Markets **27**(8), 791–817 (2007)
13. Cont, R., Tankov, P.: Financial Modelling with Jump Processes. Chapman and Hall, Boca Raton, FL (2004)
14. Dempster, M.A.H., Hong, S.S.G.: Spread option valuation and the Fast Fourier transform. Techn. Rep. WP 26/2000, the Judge Inst. Manag. Studies, University Cambridge (2000)
15. Duffie, D., Pan, J., Singleton, K.J.: Transform analysis and asset pricing for affine jump-diffusions. Econometrica **68**, 1343–1376 (2000)
16. Duffie, D., Filipovic, D., Schachermayer, W.: Affine processes and applications in finance. Ann. Appl. Probab. **13**(3), 984–1053 (2003)
17. Eydeland, A.: A fast algorithm for computing integrals in function spaces: financial applications. Comput. Econ. **7**(4), 277–285 (1994)
18. 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)
19. 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)
20. Haug, E.G.: Barrier put-call transformations, see http://www.smartquant.com/references/OptionPricing/option27.pdf
21. Hirsa, A., Madan, D.B.: Pricing American options under Variance Gamma. J. Comp. Finance **7**, 63–80 (2004)
22. Howison, S.: A matched asymptotic expansions approach to continuity corrections for discretely sampled options. Part 2: Bermudanoptions. Appl. Math. Finance **14**(1), 91–104 (2007)
23. Jackson, K., Jaimungal, S., Surkov, V.: Option pricing with regime switching Lévy processes using Fourier space time-stepping. In: Proc. 4th IASTED Intern. Conf. Financial Engin. Applic., pp. 92–97 (2007)
24. Kou, S.G.: A jump diffusion model for option pricing. Manag. Sci. **48**(8), 1086–1101 (2002)
25. Lord, R., Fang, F., Bervoets, F., Oosterlee, C.W.: A fast and accurate FFT-based method for pricing early-exercise options under Lévy processes. SIAM J. Sci. Comput. **30**, 1678–1705 (2008)

26. Madan, D.B., Carr, P.R., Chang, E.C.: The Variance Gamma process and option pricing. Eur. Finance Rev. **2**, 79–105 (1998)
27. Merton, R.: Option pricing when underlying stock returns are discontinuous. J. Financial Econ. **3**, 125–144 (1976)
28. O'Sullivan, C.: Path dependent option pricing under Lévy processes EFA 2005 Moscow Meetings Paper, Available at SSRN: http://ssrn.com/abstract=673424, 2005
29. Sato, K.-I.: Basic results on Lévy processes. In: Lévy processes, pp. 3–37. Birkhäuser Boston, Boston (2001)
30. Schoutens, W.: Lévy Processes in Finance: Pricing Financial Derivatives. Wiley, London (2003)
31. Singleton, K.J., Umantsev, L.: Pricing coupon-bond options and swaptions in affine term structure models. Math. Finance **12**(4), 427–446 (2002)
32. Surkov, V.: Parallel option pricing with Fourier space time-stepping method on graphics processing units. Preprint Univ. of Toronto, 2007. See: http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1020207
33. Taleb, N.: Dynamic Hedging. Wiley, New York (2002)
34. Wang, I., Wan, J.W., Forsyth, P.: Robust numerical valuation of European and American options under the CGMY process. J. Comp. Finance **10**(4), 31–70 (2007)
35. Wilmott, P.: Derivatives: The Theory and Practice of Financial Engineering. Wiley Frontiers in Finance Series, London (1998)