Fourier Cosine Expansions and Put–Call Relations for Bermudan Options

Bowen Zhang and Cornelis W. Oosterlee

Abstract In this chapter we describe the pricing of Bermudan options by means of Fourier cosine expansions. We propose a technique to price early-exercise call options with the help of the (European) put-call parity and put–call duality relations. Direct pricing of call options with cosine expansions may give rise to some sensitivity regarding the choice of the size of the domain in which the Fourier expansion is applied. By employing the put–call parity or put–call duality relations, this can be avoided so that call options governed by fat-tailed asset price distributions can be priced as robust and efficiently as put options.

Bowen Zhang
Delft Institute of Applied Mathematics, Delft University of Technology, the Netherlands, e-mail: bowen.zhang@tudelft.nl

Cornelis W. Oosterlee
CWI - Center for Mathematics and Computer Science, Amsterdam, and Delft University of Technology, the Netherlands e-mail: c.w.oosterlee@cwi.nl

1 Introduction

Numerical integration methods are traditionally very efficient for the valuation of single asset European options. They are also referred to as “transform methods” as a transformation, for example to the Fourier domain, is often combined with numerical integration [8, 13, 20]. The transform methods can readily be used with asset price models for which the characteristic function (i.e., the Fourier transform of the probability density function) is available.

Next to Fourier-based transform methods, techniques based on the Gauss or the Hilbert Transform have also been introduced [5, 6, 15, 24]. A contribution of our research group to the development of the transform methods is the COS method [13, 14], which is based on Fourier cosine expansions and converges exponentially in the number of terms in the cosine expansion.

Recently, transform methods have been generalized to pricing options with early-exercise features. The key idea is to set up a time lattice on each early-exercise date and view the option as “European style” between two adjacent lattices. Pricing an early-exercisable option usually involves two steps: recovery of the probability density function and computation of the integral that appears in the risk-neutral valuation formula. Some of existing methods employ quadrature rules in both steps, see for example [19, 11, 2, 3, 17]. We will detail the generalization of the COS method to pricing Bermudan options here.

The purpose of the present chapter is two-fold. First of all, we present the COS method, focusing on options with early-exercise features, like Bermudan and American options. Secondly, we present a novel component for the *robust* pricing of call options, where we use the put–call parity and the put–call duality relations for the valuation of Bermudan call options.

When pricing call options with the COS method, the method’s accuracy may exhibit sensitivity regarding the choice of the domain in which the series expansion is defined. A call payoff grows exponentially with the log-stock price which may introduce significant cancellation errors for *large domain sizes*. Put options do not suffer from this, as their payoff value is bounded by the strike value. For pricing European calls, one can employ the well-known put–call parity or put–call duality and price calls via puts. Here, we generalize this concept, so that we can also apply the put-call parity or put–call duality when pricing Bermudan call options.

The outline of this chapter is as follows: We will introduce the COS method for European options in Section 2, as well as the choice of computational domain. We will focus on Lévy asset price dynamics. In Section 3 the COS method is explained for Bermudan options and error analysis is included for call options. The generalization of the put–call parity and put–call duality is presented in Section 4. Section 5 then presents a variety of numerical results, confirming our robust version of the COS valuation method.

2 Preliminaries

We will discuss asset dynamics in this section and focus on Lévy processes. We also review the COS method for European options and the choice of computational domain for this method.

2.1 Exponential Lévy Asset Dynamics

An asset is modeled here by an exponential Lévy process (e.g. Geometric Brownian Motion, the Variance Gamma (VG) model [21], the CGMY model [7], the Normal Inverse Gaussian model [4], ...).

The asset price can be written as an exponential function of Lévy process, L_t as follows:

$$S_t = S_0 \exp(L_t). \tag{1}$$

For ease of exposure we assume that the asset pays a continuous stream of dividends, measured by the dividend rate, q. In addition, we assume the existence of a bank account, B_t, which evolves according to $dB_t = rB_t dt$, with r being the (deterministic) risk-free rate. Recall that a process L_t on $(\Omega, \mathscr{J}, P)$, with $L_0 = 0$, is a Lévy process if it has independent increments, it has stationary increments, and it is stochastically continuous, i.e., for any $t \geq 0$ and $\varepsilon > 0$ we have

$$\lim_{s \to t} \mathbb{P}(|L_t - L_s| > \varepsilon) = 0. \tag{2}$$

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

$$\int_{\mathbb{R}} \min(1, |x|^2) \nu(dx) < \infty. \tag{3}$$

In terms of this triplet the *characteristic function* of the Lévy process equals:

$$\begin{aligned} \phi(u,t) &= \mathbb{E}[\exp(iuL_t)] \\ &= \exp\left(t\left(i\mu u - \frac{1}{2}\sigma^2 u^2 + \int_{\mathbb{R}} (e^{iux} - 1 - iux\mathbf{1}_{[|x|<1]}\nu(dx)\right)\right), \end{aligned} \tag{4}$$

the celebrated Lévy-Khinchine formula. As is common in most models nowadays we assume that Equation (1) is formulated directly under the risk-neutral measure. To ensure that the reinvested relative price $e^{qt}S_t/B_t$ is a martingale under the risk-neutral measure, we need to ensure that

$$\phi(-i,t) = \mathbb{E}[\exp(L_t)] = e^{(r-q)t}, \tag{5}$$

which is satisfied if the drift μ is chosen as:

$$\mu = r - q - \frac{1}{2}\sigma^2 - \int_{\mathbb{R}} (e^x - 1 - x\mathbf{1}_{[|x|<1]})\nu(dx) \tag{6}$$

Based on Equation (1) we define:

$$\ln(S_t/K) = ln(S_0/K) + L_t := x + L_t.$$

The characteristic function of $\ln(S_t/K)$ is denoted by $\varphi(u,x;t)$ and reads:

$$\varphi(u,x;t) := e^{iux}\phi(u,t) = e^{iux}\mathbb{E}(exp(iuL_t)). \tag{7}$$

Characteristic functions for several exponential Lévy processes are available in [10, 23].

2.2 The Fourier Cosine Method (COS) for European options

The Fourier cosine pricing method "*COS*" is based on the risk-neutral option valuation formula (discounted expected payoff approach):

$$v(x,t_0) = e^{-r\Delta t}\int_{-\infty}^{\infty} v(y,T) f(y|x)dy, \tag{8}$$

where $v(x,t_0)$ is the present option value, r the interest rate, $\Delta t = T - t_0$ and x, y can be any monotone function of the underlying asset at initial time t_0 and the expiration date T. Function $v(y,T)$, which equals payoff function $g(y)$, is known, but the transitional density function, $f(y|x)$ in (8), typically is not.

We approximate the conditional density function on a truncated domain, by a truncated Fourier cosine expansion, which recovers the conditional density function from its characteristic function (see [13]) as follows:

$$f(y|x) \approx \frac{2}{b-a}{\sum_{k=0}^{N-1}}' Re\left(\varphi(\frac{k\pi}{b-a}, x; \Delta t)\exp(-i\frac{ak\pi}{b-a})\right)\cos(k\pi\frac{y-a}{b-a}), \tag{9}$$

with $\varphi(u,x;t)$ the characteristic function of $f(y|x)$; a, b determine the truncated domain and *Re* means taking the real part of the argument. The prime at the sum symbol indicates that the first term in the expansion is multiplied by one-half. The size of the truncated domain can be determined with the help of the cumulants [13] [1], discussed in Section 2.3.

Replacing $f(y|x)$ by its approximation (9) in Equation (8) and interchanging integration and summation gives the COS formula for computing the values of European options:

[1] For example so that $|\int_{\mathbb{R}} f(y|x)dy - \int_a^b f(y|x)dy| < TOL$.

$$v(x,t_0) = e^{-r\Delta t} \sum_{k=0}^{N-1}{}' Re(\varphi(\frac{k\pi}{b-a}, x; \Delta t) e^{-ik\pi \frac{a}{b-a}}) V_k, \tag{10}$$

where:

$$V_k = \frac{2}{b-a} \int_a^b v(y,T) \cos(k\pi \frac{y-a}{b-a}) dy,$$

are the Fourier cosine coefficients of $v(y,T)$, that are available in closed form for several payoff functions, like for plain vanilla puts and calls, but also for example for discontinuous payoffs like for digital options.

It was found by a rigorous analysis in [13], that, with integration interval $[a,b]$ chosen sufficiently wide, the series truncation error dominates the overall error. For conditional density functions $f(y|x) \in C^\infty((a,b) \subset \mathbb{R})$, the method converges exponentially; otherwise convergence is algebraically [14].

Formula (10) also forms the basis for the pricing of Bermudan options [14].

2.3 Truncation Range and Put–Call Relations

The choice of integration range, $[a,b]$, is quite important. An interval which is chosen too small or too wide will lead to significant integration-range errors.

We use the definition of the integration (also called truncation) range as given [13] and we center the domain at $x_0 := \ln(S_0/K)$, i.e.

$$[a,b] := \left[(\xi_1 + x_0) - L\sqrt{\xi_2 + \sqrt{\xi_4}}, \quad (\xi_1 + x_0) + L\sqrt{\xi_2 + \sqrt{\xi_4}} \right], \tag{11}$$

with $L \in [6,12]$ depending on a *user-defined tolerance level*, TOL and $\xi_1, \ldots, \xi_4$ being the cumulants of the underlying stochastic process. The error connected to the size of the domain decreases exponentially with L.

Given the characteristic functions, the cumulants, as defined in [10], can be computed via

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

where $t\Psi(u)$ is the logarithm of the characteristic function $\phi(u,t)$, i.e.

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

However, when pricing *call options*, the solution's accuracy exhibits sensitivity regarding the size of this truncated domain. This holds specifically for call options under fat-tailed distributions, like under certain Lévy jump processes, or for options with a very long time to maturity [2]. A call payoff grows exponentially in log–stock price which may introduce cancellation errors for large domain sizes. A put option

[2] This is mainly the case when we consider real options or insurance products with a long life time.

does not suffer from this (see [14]), as their payoff value is bounded by the strike value. In [13], European call options were therefore priced by means of European put option computations, in combination with the *put-call parity*:

$$v^{call}(x,t) = v^{put}(x,t) + S_t e^{-q(T-t)} - K e^{-r(T-t)}, \tag{12}$$

where $v^{call}(x,t)$ and $v^{put}(x,t)$ are the call and put option prices, respectively, and q is again the dividend rate.

Alternatively, one can use the *put–call duality* relation (see also [12]):

$$v^{call}(S,K,r,q,t,\nu) = v^{put}(K,S,q,r,t,e^{-x}\nu(-dx)), \tag{13}$$

where [3] measure $\nu(dx)$ is the same as in (4) and (6). In the case that

$$\nu(dx) = e^{-x}\nu(-dx)$$

is satisfied (for Lévy processes without any jumps, for example), Eqn. (13) simplifies:

$$v^{call}(S,K,r,q) = v^{put}(K,S,q,r).$$

2.3.1 European Option Results

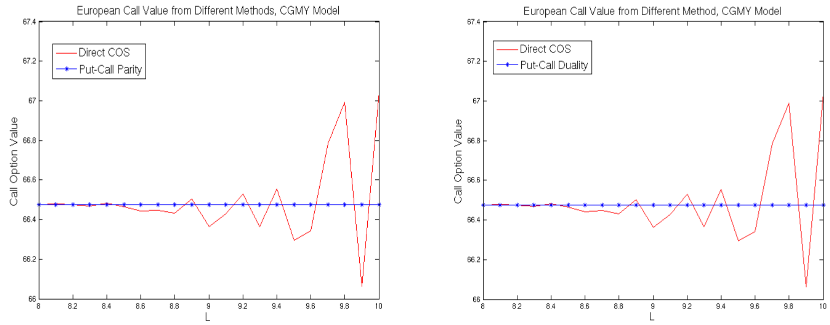
Figures 1 and 2 present European call option values under the infinite activity Lévy CGMY jump model, see [7]. The option values obtained by pricing call options directly by the COS method (solid lines) are compared to the values calculated with the put–call parity and put–call duality relations (dotted lines), for different values of parameter L, which determines the sizes of the truncated domain in (11). Reference solutions are obtained on a very fine grid.

The asset price parameters read $S_0 = 100, K = 110, r = 0.1, q = 0.05$, and the CGMY parameters are chosen as $C = 1, G = 5, M = 5$. For Figure 1, with the remaining CGMY parameter $Y = 1.5$, and with $T = 5$, the reference value for the European option is $66.474333\ldots$ and in Figure 2 we set $Y = 1.98$, and $T = 0.1$ for which the reference value is $86.826264\ldots$.

As shown in Figures 1 and 2, the errors appearing, when call prices are directly computed with the COS method, increase for large Y- and T-values, since then the probability density function of the underlying is governed by fat tails. The errors grow drastically as L, i.e. the size of the computational domain, increases. It seems that the choice $L = 6$ results in accurate values in these tests, but this choice is heuristic.

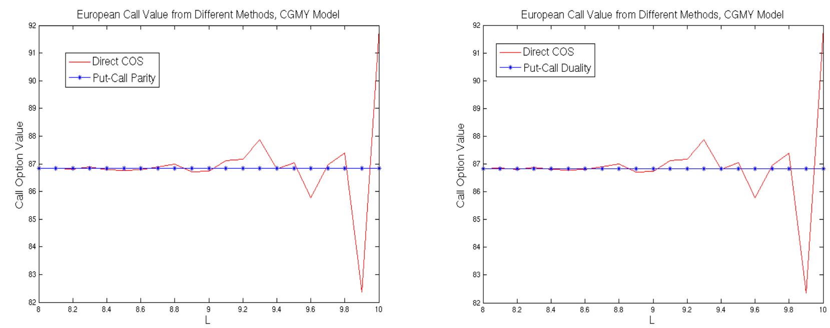
The option prices obtained by the put–call parity or the put–call duality do not deviate from the reference solutions in both test cases, for all integration ranges. The parity and duality lead to robust formulas for pricing European call options by the COS method.

[3] Here we have a long list of arguments, as they are important for the use of the put-call duality.



(a) Put-Call Parity (b) Put-Call Duality

Fig. 1 Comparison of European call option values, directly obtained by the COS method, with those obtained by the put–call parity and the put–call duality, CGMY model, $Y = 1.5, T = 5$, $L \in [8, 10]$.



(a) Put-Call Parity (b) Put-Call Duality

Fig. 2 Comparison of European call option values, directly obtained by the COS method, with those obtained by the put–call parity or put–call duality, CGMY model, $Y = 1.98, T = 0.1$, $L \in [8, 10]$.

3 Pricing Early-Exercise Options

A Bermudan option can be exercised at pre-specified dates before maturity. The holder receives the exercise payoff when she exercises the option. We have again t_0 as initial time and $\{t_1, \cdots, t_M\}$ the collection of 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 $m = M, M-1, \ldots, 2$:

$$\begin{cases} c(x, t_{m-1}) = e^{-r\Delta t} \int_{\mathbb{R}} v(y, t_m) f(y|x) dy, \\ v(x, t_{m-1}) = \max\left(g(x), c(x, t_{m-1})\right), \end{cases} \tag{14}$$

followed by

$$v(x,t_0) = e^{-r\Delta t} \int_{\mathbb{R}} v(y,t_1) f(y|x) dy. \tag{15}$$

Functions $v(x,t)$, $c(x,t)$ and $g(x)$ are the option value, the continuation value and the payoff at time t, respectively. For call and put options, $g(x) \equiv v(x,T)$, with

$$v(x,T) = \max\left[\alpha K(e^x - 1), 0\right], \quad \alpha = \begin{cases} 1 & \text{for a call,} \\ -1 & \text{for a put,} \end{cases} \tag{16}$$

where x and y are state variables, defined as

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

3.1 *Pricing Bermudan Options by the COS Method*

The continuation value in (14) can be defined by means of the COS formula. For exponential Lévy processes it reads:

$$c(x,t_{m-1}) = e^{-r\Delta t} {\sum_{k=0}^{N-1}}' \mathrm{Re}\left\{ \phi\left(\frac{k\pi}{b-a}, \Delta t\right) e^{ik\pi\frac{x-a}{b-a}} \right\} V_k(t_m), \tag{17}$$

where $\phi(u,t) := \varphi(u,0;t)$, as defined in (7).

The technique of pricing Bermudan options by the COS method is based on the computation of the Fourier cosine coefficients of the option value at t_1, $V_k(t_1)$, which are then inserted into (15). The derivation of an induction formula for $V_k(t_1)$, backwards in time, was the basis of the work in [14]. It is briefly explained here.

First, the *early-exercise point*, x_m^*, at time t_m, which is the point where the continuation value equals the payoff, i.e., $c(x_m^*, t_m) = g(x_m^*)$, is determined for example by Newton's method.

Based on x_m^*, we can split $V_k(t_m)$ in Eqn. (17) into two parts: One on the interval $[a, x_m^*]$ and another on $(x_m^*, b]$, i.e.

$$V_k(t_m) = \begin{cases} C_k(a, x_m^*, t_m) + G_k(x_m^*, b), & \text{for a call,} \\ G_k(a, x_m^*) + C_k(x_m^*, b, t_m), & \text{for a put,} \end{cases} \tag{18}$$

for $m = M-1, M-2, \cdots, 1$, where

$$V_k(t_M) = \begin{cases} G_k(0,b), & \text{for a call,} \\ G_k(a,0), & \text{for a put.} \end{cases} \tag{19}$$

We have:

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

and

$$C_k(x_1,x_2,t_m) := \frac{2}{b-a}\int_{x_1}^{x_2} c(x,t_m)\cos\left(k\pi\frac{x-a}{b-a}\right)dx. \tag{21}$$

For $k=0,1,\cdots,N-1$ and $m=1,2,\cdots,M$, the $G_k(x_1,x_2)$ in (20) admit analytic solutions, and the challenge is to compute the C_k-coefficients efficiently.

We can generally write characteristic functions as:

$$\varphi(u,x;\tau) = e^{iux\beta}\phi(u,\tau), \tag{22}$$

with $\phi(u,\tau)$ not depending on x.

By (22), we can distinguish basically two types of stochastic processes in view of their characteristic functions. The first type, governed by $\beta=1$, which corresponds to a process with independent increments, includes the exponential Lévy processes, for which the characteristic function can thus be written in the form $\varphi(u,x;\tau)=e^{iux}\phi(u,\tau)$. Examples for these are the log-versions of Geometric Brownian Motion, jump-diffusion processes of Kou [18] and Merton [22], infinite activity Lévy processes [10], like Variance-Gamma (VG) [21], Normal Inverse Gaussian (NIG) [4] or CGMY [7].

For the second type of processes, $\phi(u,x;t)$ *cannot* be written as the product of e^{iux} and a function independent of x. An example is the OU mean reverting process, for which $\beta=e^{-\kappa\tau}$ in (22), with κ a mean reversion parameter.

In the lemma to follow we will see that characteristic functions of the first type ($\beta=1$) are beneficial for pricing Bermudan options by the COS method as the Fast Fourier Transform can be applied.

Lemma 3.1 (Efficient Computation)**.** *The terms $C_k(x_1,x_2,t_m)$ can be computed in $O(N\log_2 N)$ operations, if the stochastic process for the underlying is governed by general characteristic function (22) with parameter $\beta=1$.*

Proof. At times t_m, $m=1,2,\cdots,M$, from Equations (14) and (17), we obtain an approximation for $c(x,t_m)$, the continuation value at t_m, which is inserted into (21). Interchanging summation and integration gives the following coefficients, $C_k(x_1,x_2,t_m)$:

$$C_k(x_1,x_2,t_m) := e^{-r\Delta t}{\sum_{j=0}^{N-1}}' Re\left(\phi\left(\frac{j\pi}{b-a},\Delta t\right)V_j(t_{m+1})\cdot H_{k,j}(x_1,x_2)\right), \tag{23}$$

where $\phi(u,\Delta t)$ comes from the general expression for the characteristic function (22). To get $C_k(x_1,x_2,t_m)$, the following integrals need to be computed:

$$H_{k,j}(x_1,x_2) = \frac{2}{b-a}\int_{x_1}^{x_2} e^{ij\pi\frac{\beta x-a}{b-a}}\cos(k\pi\frac{x-a}{b-a})dx,$$

with β defined in (22).

By basic calculus, we can split $H_{k,j}(x_1,x_2)$ into two parts as

$$H_{k,j}(x_1,x_2) = -\frac{i}{\pi}(H^s_{k,j}(x_1,x_2) + H^c_{k,j}(x_1,x_2)),$$

where

$$H^c_{k,j}(x_1,x_2) = \begin{cases} \dfrac{(x_2-x_1)\pi i}{b-a}, & \text{if } k=j=0, \\ \dfrac{1}{(j\beta+k)}\left[\exp\left(\dfrac{((j\beta+k)x_2-(j+k)a)\pi i}{b-a}\right) - \exp\left(\dfrac{((j\beta+k)x_1-(j+k)a)\pi i}{b-a}\right)\right], & \text{otherwise.} \end{cases} \tag{24}$$

and

$$H^s_{k,j}(x_1,x_2) = \begin{cases} \dfrac{(x_2-x_1)\pi i}{b-a}, & \text{if } k=j=0, \\ \dfrac{1}{(j\beta-k)}\left[\exp\left(\dfrac{((j\beta-k)x_2-(j-k)a)\pi i}{b-a}\right) - \exp\left(\dfrac{((j\beta-k)x_1-(j-k)a)\pi i}{b-a}\right)\right], & \text{otherwise.} \end{cases} \tag{25}$$

Matrices H^s and H^c have a Toeplitz and Hankel structure, respectively, *if* $H^s_{k,j}(x_1,x_2) = H^s_{k+1,j+1}(x_1,x_2)$ and $H^c_{k,j}(x_1,x_2) = H^c_{k+1,j-1}(x_1,x_2)$, *which is the case for* $\beta \equiv 1$. In other words, pricing Bermudan options can be done highly efficiently when exponential Lévy asset price models are employed. Then, the Fast Fourier Transform can be applied directly for matrix-vector multiplication [14], and the resulting computational complexity of $C_k(x_1,x_2,t_m)$ is $O(N\log_2 N)$. □

We would obtain terms of the form $j\beta-k, j\beta+k$ in the matrix elements in (24) and (25), instead of terms with $j-k, j+k$ if $\beta \neq 1$ in (22). Terms with β not being an integer hamper an efficient computation of the matrix-vector products, leading to computations with $O(N^2)$ complexity.

3.1.1 American Options

For the valuation of American options by the COS method, there are basically two approaches. One is to approximate an American option by a Bermudan option with many exercise opportunities, the other is to use repeated Richardson extrapolation on a series of Bermudan options with an increasing number of exercise opportunities. Here we will focus on the extrapolation-based method, which has been described in detail in [9], although the approach dates back to [16].

Let here $\hat{v}(M)$ be the price of a Bermudan option with M exercise dates with a maturity of T years where the exercise dates are $\Delta t = T/M$ years apart. It is assumed that $\hat{v}(M)$ can be expanded as:

$$\hat{v}(M) = v_{AM} + \sum_{i=1}^{\infty} a_i (\Delta t)^{\gamma_i}, \tag{26}$$

with $0 < \gamma_i < \gamma_{i+1}$; v_{AM} is the American option value. Classical extrapolation procedures assume that the exponents γ_i are known, which means that we can use $n+1$ Bermudan prices with varying Δt to eliminate the n leading order terms in (26). The prices of American options can be obtained by applying repeated Richardson extrapolation on the values of a few Bermudan options with small M. We use the following 4-point repeated Richardson extrapolation scheme,

$$\hat{v}_{AM}(M) = \frac{1}{21} \left(64\hat{v}(8M) - 56\hat{v}(4M) + 14\hat{v}(2M) - \hat{v}(M) \right), \tag{27}$$

where $\hat{v}_{AM}(M)$ denotes the approximated value of the American option [4].

3.2 Error Analysis

In this subsection we give error analysis for the COS pricing method, *focusing on Bermudan call options*. First, we analyze the local error, i.e., the error in the continuation values at each time step. A similar error analysis has been performed in [13], where, however, the influence of the call payoff function on the global error convergence was omitted.Here, we study the influence of the payoff function and the integration range on the error convergence.

3.2.1 Local Error

It has been shown, [13], that the error of the COS method for the error in the continuation value consists of three parts, denoted by $\varepsilon_1, \varepsilon_2$ and ε_3, respectively.

Error ε_1 is the integration range error

$$|\varepsilon_1(x,[a,b])| = e^{-r\Delta t} \int_{\mathbf{R}\setminus[a,b]} v(y,T) f(y|x) dy,$$

which depends on the payoff function and the integration range.

Error ε_2 is the series truncation error on $[a,b]$, which depends on the smoothness of the probability density function of the underlying processes:

$$\varepsilon_2\left(x;N,[a,b]\right) := e^{-r\Delta t} \sum_{k=N}^{\infty} \mathrm{Re}\left\{ e^{-ik\pi\frac{a}{b-a}} \int_a^b e^{i\frac{k\pi}{b-a}y} f(y|x) dy \right\} \cdot V_k. \tag{28}$$

[4] Without any dividend payments, of course, the American call option value is equal to the European call option value.

For probability density functions $f(y|x) \in \mathbf{C}^{\infty}[a,b]$, we have

$$|\varepsilon_2(x,N,[a,b])| < P\exp(-(N-1)\nu),$$

where N is the number of terms in the Fourier cosine expansions, $\nu > 0$ is a constant and P is a term which varies less than exponentially with respect to N. When the probability density function has a discontinuous derivative, then the Fourier cosine expansions converge algebraically,

$$|\varepsilon_2(x,N,[a,b])| < \frac{P}{(N-1)^{\beta-1}},$$

where P is a constant and $\beta \geq 1$ is the algebraic index of convergence.

Error ε_3 is the error related to the approximation of the Fourier cosine coefficients of the density function in terms of its characteristic function, which reads

$$|\varepsilon_3(x,N,[a,b])| = e^{-r\Delta t}\sum_{k=0}^{N-1}{}' Re\left(\int_{\mathbf{R}\setminus[a,b]} e^{ik\pi\frac{y-a}{b-a}} f(y|x)dy\right)V_k.$$

It can be shown that

$$|\varepsilon_3(x,N,[a,b])| < e^{-r\Delta t}Q_1\int_{\mathbf{R}\setminus[a,b]} f(y|x)dy,$$

where Q_1 is a constant independent of N and Δt.

We denote by

$$I_1 = \int_{\mathbf{R}\setminus[a,b]} v(y,T)f(y|x)dy, \quad I_2 = \int_{\mathbf{R}\setminus[a,b]} f(y|x)dy,$$

so that $\varepsilon_1 = e^{-r\Delta t}I_1$, $\varepsilon_3 < e^{-r\Delta t}Q_1 I_2$. Integral I_1 then depends on the payoff function and the integration range, whereas I_2 depends only on the integration range.

We start with a discussion about the influence of the payoff function on the error convergence and then we analyze the influence of L in (11).

For an option with a bounded payoff function, such as a put option or a swing option studied in [25], we have $\forall y$, $v(y,T) \leq Q_2$, so that it follows directly that

$$I_1 \leq Q_2 I_2, \tag{29}$$

and both ε_1 and ε_3 can be controlled by means of parameter L. This was the basis for the detailed error analysis for Bermudan put options in [14].

However, in the case of an unbounded payoff, for instance, a call option, we have:

$$\begin{aligned} I_1 &= \int_{\mathbf{R}\setminus[a,b]} v(y,T)f(y|x)dy \geq \int_b^{\infty} v(y,T)f(y|x)dy \\ &= \int_b^{\infty} (Ke^y - K)^+ f(y|x)dy \geq K(e^b-1)\int_b^{\infty} f(y|x)dy. \end{aligned} \tag{30}$$

Note that we assume that $b \geq 0$, as otherwise for all $y \in [a,b]$, $v(y,T) = 0$ and the option value is also zero.

Function $\int_b^\infty f(y|x)dy$ is bounded by $0 < \int_b^\infty f(y|x)dy < 1$.

Denoting by $Q_3 \triangleq K \int_b^\infty f(y|x)dy$ then

$$I_1 \geq Q_3(e^b - 1).$$

Function $e^b - 1$ will, however, not decrease to zero as N, the number of terms in the Fourier cosine expansion, goes to infinity. Furthermore, the larger the integration range, the larger the value $e^b - 1$, i.e. the error in the option price. Given the fact that $\varepsilon_1 = e^{-r\Delta t} Q_3(e^b - 1)$, the global error in the call option price may increase as the integration range $[a,b]$ (or L) increases. This implies that when we directly use the COS formula for a call option, the value may diverge, depending on the decay rate of $f(y|x)$. This is not the case if a very small integration range (or a very small value of L) is used, but by this error ε_3 may increase. This is the next topic in the error analysis.

To study the influence of truncation on the error convergence, we start the analysis with the Black–Scholes model. From the cumulative density function (which is known analytically) it follows that with $L = 6$, we find $I_2 = 1.9732 \times 10^{-9}$ and with $L = 8$, we have $I_2 = 1.3323 \times 10^{-15}$, so that with $L \in [6,8]$ the errors ε_1 and ε_3 can be controlled. Incorporating jumps in a Lévy model gives rise to a slightly larger value of L. As shown in [14], an integration range with $L \in [8,10]$ is sufficient for most of the Lévy processes with $T > 0.1$ to bound I_2 (but not always for I_1).

In general, from Chebyshev's inequality we know that for any random variable X with expected value μ and finite variance σ and for any real number $k > 0$, $Pr(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}$, which implies

$$\begin{aligned} I_2(x_0) &= \int_{\mathbf{R}\setminus[a,b]} f(y|x_0)dy = Pr(|X_T - (\xi_1 + x_0)| \geq L(\xi_2 + \sqrt{\xi_4})) \\ &\leq Pr(|X_T - (\xi_1 + x_0)| \geq L(\xi_2)) \leq \frac{1}{L^2}. \end{aligned}$$

Therefore for all processes and model parameters, I_2 decays at least algebraically with algebraic index $n \geq 2$.

4 Pricing Bermudan Call Options Using the Put-Call Relations

In this section, we present two techniques to deal efficiently with the inaccurate pricing with the COS method of Bermudan call options. With our new methods, the Fourier cosine coefficients of call options need not to be calculated directly at each time step, which will eliminate the error due to unbounded payoff of call options. In Section 4.1 we discuss the use of the put-call parity relation, and in Section 4.2 we explain the use of the put-call duality relation. These techniques are accompanied by error analysis in Section 4.3.

4.1 The Put–Call Parity

Here we give details of the use of the *European* put-call parity for the robust pricing of *Bermudan* call options by means of the COS method.

At each time step we need to calculate the continuation value based on the Fourier coefficients of the call option payoff. The continuation value is then used to determine the early–exercise points, as well as to recover the Fourier cosine coefficients for a next time step. In these steps, the influence of an exponentially–increasing payoff can be significant as for European call options. Here, we modify the pricing algorithm for Bermudan call options employing put–call parity (12).

We denote the Fourier cosine coefficients for a put and a call option at $t_M = T$ by $V_k^{put}(t_M)$ and $V_k^{call}(t_M)$, respectively. By (12) we then find

$$
\begin{aligned}
& e^{-r\Delta t} \sum_{k=0}^{N-1}{}' Re(\phi(\frac{k\pi}{b-a}, \Delta t) e^{i(x-a)\frac{k\pi}{b-a}}) V_k^{call}(t_M) = \\
& S_t e^{-q\Delta t} - K e^{-r\Delta t} + e^{-r\Delta t} \sum_{k=0}^{N-1}{}' Re(\phi(\frac{k\pi}{b-a}, \Delta t) e^{i(x-a)\frac{k\pi}{b-a}}) V_k^{put}(t_M).
\end{aligned} \tag{31}
$$

We have $V_k^{put}(t_M) = G_k^{put}(a,0)$ and $V_k^{call}(t_M) = G_k^{call}(0,b)$, where G_k^{put} and G_k^{call} are the Fourier cosine coefficients for the respective payoffs. So, we can write (31) as:

$$
\begin{aligned}
e^{-r\Delta t} & \sum_{k=0}^{N-1}{}' Re(\phi(\frac{k\pi}{b-a}, \Delta t) e^{i(x-a)\frac{k\pi}{b-a}}) G_k^{call}(0,b) = S e^{-q\Delta t} - K e^{-r\Delta t} + \\
& e^{-r\Delta t} \sum_{k=0}^{N-1}{}' Re(\phi(\frac{k\pi}{b-a}, \Delta t) e^{i(x-a)\frac{k\pi}{b-a}}) G_k^{put}(a,0).
\end{aligned} \tag{32}
$$

Equation (32) will be used in the backward recursion.

At $t = M-1$, we first determine the early–exercise point, x^*_{M-1}, by Newton's method, for which the functions $c, g, \partial c/\partial x, \partial g/\partial x$ are required. The continuation value for the call option now reads, using (31):

$$
c(x, t_{M-1}) = e^{-r\Delta t} \sum_{k=0}^{N-1}{}' Re(\phi(\frac{k\pi}{b-a}, \Delta t) e^{i(x-a)\frac{k\pi}{b-a}}) V_k^{put}(t_M) + K e^x e^{-q\Delta t} - K e^{-r\Delta t} \tag{33}
$$

with $x = \log(S/K)$, and similarly we find:

$$\frac{\partial c}{\partial x} = e^{-r\Delta t} \sum_{k=0}^{N-1}{}' Re(\phi(\frac{k\pi}{b-a}, \Delta t) e^{i(x-a)\frac{k\pi}{b-a}} (\frac{ik\pi}{b-a})) V_k^{put}(t_M) + Ke^x e^{-q\Delta t}. \quad (34)$$

With $x \geq 0$, we have $g(x) = Ke^x - K$ and $\partial g/\partial x = Ke^x$, whereas for $x < 0$ both the payoff and its derivative are zero, for all time steps.

With the early-exercise point determined, we need to compute the values,

$$V_k^{call}(t_{M-1}) := C_k^{call}(a, x_{M-1}^*, t_{M-1}) + G_k^{call}(x_{M-1}^*, b). \quad (35)$$

Application of (33) gives us:

$$\begin{aligned} C_k^{call}(a, x_{M-1}^*, t_{M-1}) &= \frac{2}{b-a} \int_a^{x_{M-1}^*} c(x, t_{M-1}) \cos(k\pi \frac{x-a}{b-a}) dx \\ &= \frac{e^{-r\Delta t}}{\pi} Im(H^c + H^s) u + \frac{2}{b-a} K e^{-q\Delta t} \chi(a, x_{M-1}^*) \\ &- \frac{2}{b-a} K e^{-r\Delta t} \psi(a, x_{M-1}^*) \end{aligned} \quad (36)$$

where Im means taking the imaginary part, vector u consists of values:

$$u_j = \phi(\frac{k\pi}{b-a}, \Delta t) V_j^{put}(t_M), j = 1, \cdots, N-1,$$

and $u_0 = \frac{1}{2}\phi(0, \Delta t) V_0^{put}(t_M)$. Matrices H^c, H^s are as in Equations (24) and (25), with $\beta = 1$. Moreover,

$$\chi(x_1, x_2) = \int_{x_1}^{x_2} e^x \cos(\frac{k\pi(x-a)}{b-a}) dx, \quad \psi(x_1, x_2) = \int_{x_1}^{x_2} \cos(\frac{k\pi(x-a)}{b-a}) dx, \quad (37)$$

both of which have an analytic solution.

We further have $G_k^{call}(x_{M-1}^*, b) = G_k^{call}(0, b) - G_k^{call}(0, x_{M-1}^*)$, and $\forall x \in (0, x_{M-1}^*)$ the payoff of a call option is less than the continuation value. Therefore, $G_k^{call}(0, x_{M-1}^*)$ can be calculated directly and it will remain accurate, independent of the choice of integration range. Quantity $G_k^{call}(0, b)$ will be replaced by $G_k^{put}(a, 0)$ via (32).

We now arrive at the following lemma:

Lemma 4.1. *Quantities $c(x, t_m)$, x_m^*, and $C_k^{call}(a, x^*, t_m)$ can be recovered from $C_k^{call}(a, x_{m+1}^*, t_{m+1})$ in an accurate way for $m = M-2, \ldots, 1$, with a computational complexity of $O(N \log_2 N)$ at each time step. $C_k^{call}(a, x_1^*, t_1)$ is then finally also recovered in a robust way.*

Proof. At the time steps t_m, $m = M-2, \cdots, 1$, the continuation value reads

$$\begin{aligned} c(x,t_m) &= e^{-r\Delta t}\sum_{k=0}^{N-1}{}' Re(\phi(\frac{k\pi}{b-a},\Delta t)e^{i(x-a)\frac{k\pi}{b-a}})V_k^{call}(t_{m+1}) \\ &= e^{-r\Delta t}\sum_{k=0}^{N-1}{}' Re(\phi(\frac{k\pi}{b-a},\Delta t)e^{i(x-a)\frac{k\pi}{b-a}})(C_k^{call}(a,x^*_{m+1},t_{m+1}) - G_k^{call}(0,x^*_{m+1})) + \\ &\quad e^{-r\Delta t}\sum_{k=0}^{N-1}{}' Re(\phi(\frac{k\pi}{b-a},\Delta t)e^{i(x-a)\frac{k\pi}{b-a}})G_k^{call}(0,b) \\ &= e^{-r\Delta t}\sum_{k=0}^{N-1}{}' Re(\phi(\frac{k\pi}{b-a},\Delta t)e^{i(x-a)\frac{k\pi}{b-a}})(C_k^{call}(a,x^*_{m+1},t_{m+1}) + \\ &\quad G_k^{put}(a,0) - G_k^{call}(0,x^*_{m+1})) + Ke^x e^{-q\Delta t} - Ke^{-r\Delta t}, \end{aligned} \tag{38}$$

where the last step is from (32). Derivative $\partial c/\partial x$ can be obtained similarly. Together with $g(x)$ and $\partial g/\partial x$, they are used to determine early–exercise point x^*_m at t_m.

Furthermore,

$$\begin{aligned} C_k^{call}(a,x^*_m,t_m) &= \frac{e^{-r\Delta t}}{\pi} Im(H^c + H^s)u + \frac{2}{b-a}Ke^{-q\Delta t}\chi(a,x^*_m) \\ &\quad - \frac{2}{b-a}Ke^{-r\Delta t}\psi(a,x^*_m), \end{aligned}$$

where H^c, H^s are as defined earlier in Equations (24) and (25) with $\beta = 1$ and vector u consists of elements:

$$u_j = \phi(\frac{k\pi}{b-a},\Delta t)(C_j^{call}(a,x^*_{m+1},t_{m+1}) + G_j^{put}(a,0) - G_j^{call}(0,x^*_{m+1})), \tag{39}$$

and

$$u_0 = \frac{1}{2}\phi(0,\Delta t)(C_0^{call}(a,x^*_{m+1},t_{m+1}) + G_0^{put}(a,0) - G_0^{call}(0,x^*_{m+1})). \tag{40}$$

Regarding the computational costs, at each time step $C_k^{call}(a,x^*_m,t_m)$ needs to be calculated once. Therefore we have the same computational complexity as the original COS method, which is $O(M-1)N\log_2 N$.

Finally, the two terms $G_k^{put}(a,0)$ and $G_k^{call}(0,x^*_m)$ at t_m admit analytic solutions.

At t_0 we have

$$
\begin{aligned}
v(x,t_0) &= e^{-r\Delta t}\sum_{k=0}^{N-1}{}' Re(\phi(\frac{k\pi}{b-a},\Delta t)e^{i(x-a)\frac{k\pi}{b-a}})V_k(t_1) \\
&= e^{-r\Delta t}\sum_{k=0}^{N-1}{}' Re(\phi(\frac{k\pi}{b-a},\Delta t)e^{i(x-a)\frac{k\pi}{b-a}})(C_k^{call}(a,x_1^*,t_1)+G_k^{put}(a,0)-G_k^{call}(0,x_1^*)) \\
&+ Ke^x e^{-q\Delta t} - Ke^{-r\Delta t},
\end{aligned}
\tag{41}
$$

where the last step follows from (32) and we complete the robust and efficient pricing of Bermudan options via the put-call parity relation. □

4.2 The Put–Call Duality

In this section, we discuss a second possibility to price a Bermudan call with the help of the pricing formula for a put. It is based on the put-call duality from [12].

In the COS pricing formula (10), $r,q,\nu(dx)$ are essential in the definition of the characteristic function ϕ, whereas S and K enter the formula for the Fourier cosine coefficients, V_k. Therefore, we use in this section the notation $\phi := \phi(u,t,r,q,\nu)$. Moreover, we use $V_k^{call}(t_m,S,K)$, $V_k^{put}(t_m,S,K)$, $V_k(t_m)$ to denote the Fourier cosine coefficients of European call options (with stock price S and strike price K), of European put options and the Fourier cosine coefficients of a Bermudan option at t_m, respectively. We also denote $e^{-x}\nu(dx)$ by $\tilde{\nu}(dx)$.

We start at $t_M = T$. From t_M to t_{M-1} the direct application of (13) gives us

$$
\begin{aligned}
c(x,t_{M-1}) &= \mathrm{e}^{-r\Delta t}\sum_{k=0}^{N-1}{}' Re(\phi(\frac{k\pi}{b-a},\Delta t,r,q,\nu)e^{i(x-a)\frac{k\pi}{b-a}})V_k^{call}(t_M,S,K) \\
&= \mathrm{e}^{-q\Delta t}\sum_{k=0}^{N-1}{}' Re(\phi(\frac{k\pi}{b-a},\Delta t,q,r,\tilde{\nu})e^{i(-x-a)\frac{k\pi}{b-a}})V_k^{put}(t_M,K,S)
\end{aligned}
\tag{42}
$$

where $V_k^{call}(t_M,S,K) = G_k^{call}(0,b)$, and

$$
\begin{aligned}
V_k^{put}(t_M,K,S) &= \frac{2}{b-a}\int (S-Se^y)\cos(k\pi\frac{y-a}{b-a})dy \\
&= \frac{2K}{b-a}e^x\int(1-e^y)\cos(k\pi\frac{y-a}{b-a})dy = e^x G_k^{put}(a,0).
\end{aligned}
$$

Note that for both S and K as state variables in the put–call duality formulation, integration ranges need to be defined. We set $a=\min(a_S,a_K)$, $b=\max(b_S,b_K)$. The use of "$-x$" in the second equation in (42) appears because the state variable $\log(K/S) = -\log(S/K) = -x$.

At t_{M-1} the continuation value and its derivative read:

$$c(x,t_{M-1}) = \mathrm{e}^{-q\Delta t} \sum_{k=0}^{N-1}{}' Re(\phi(\frac{k\pi}{b-a},\Delta t,q,r,\tilde{\nu}) e^{i(-x-a)\frac{k\pi}{b-a}}) e^x G_k^{put}(a,0), \tag{43}$$

$$\frac{\partial c(x,t_{M-1})}{\partial x} = \mathrm{e}^{-q\Delta t} \sum_{k=0}^{N-1}{}' Re(\phi(\frac{k\pi}{b-a},\Delta t,q,r,\tilde{\nu}) e^{i(-x-a)\frac{k\pi}{b-a}}(-\frac{ik\pi}{b-a})) e^x G_k^{put}(a,0) + c(x,t_{M-1}),$$

which are used to calculate the early–exercise point x^*_{M-1} by Newton's method, so that

$$\begin{aligned} V_k(t_{M-1}) &= C_k(a,x^*_{M-1}) + G_k^{call}(x^*_{M-1},b) \\ &= C_k(a,x^*_{M-1},t_{M-1}) - G_k^{call}(0,x^*_{M-1}) + G_k^{call}(0,b) \end{aligned} \tag{44}$$

Now, $\forall x \in (0,x^*_{M-1})$ the payoff of the call option is less than the continuation value. Therefore, $G_k^{call}(0,x^*_{M-1})$ can be calculated directly and it will be accurate with respect to the size of the integration range; $G_k^{call}(0,b)$ can be replaced by $G_k^{put}(a,0)$, in a similar way as (42).

The computation of C_k represents again the main part of the algorithm. First, we demonstrate how to compute $C_k(x_1,x_2,t_{M-1})$ in (44) with the help of the Fast Fourier Transform (FFT), then we will show that for all $m = M-2,\cdots,1$, $C_k(x_1,x_2,t_m)$ can be recovered from $C_k(x_1,x_2,t_{m+1})$. We denote $D(x_1,x_2) := \{D_k(x_1,x_2)\}_{k=0}^{N-1}$, with

$$D_k(x_1,x_2) = e^{-q\Delta t} Re(\sum_{j=0}^{N-1}{}' \phi(\frac{j\pi}{b-a},\Delta t,q,r,\tilde{\nu}) G_j^{put}(a,0) J_{k,j}(x_1,x_2)) \tag{45}$$

in which

$$J_{k,j}(x_1,x_2) := \frac{2}{b-a}\int_{x_1}^{x_2} e^{ij\pi\frac{\beta x-a}{b-a}} \cos(k\pi\frac{x-a}{b-a})dx.$$

where now $\beta = -1 - \frac{i(b-a)}{j\pi}$, which is different from $\beta = 1$. However, this β-value still results in a sum of a Toeplitz plus Hankel matrix.

Application of (21) and (43) gives $C_k(x_1,x_2,t_{M-1}) = D_k(x_1,x_2)$, $\forall k = 0,\cdots,N-1$.

First we study the structure of $J_{k,j}$ then we compute $D(x_1,x_2)$. From (24) and (25) we find that

$$J_{k,j}(x_1,x_2) = -\frac{i}{\pi}(J^c_{k,j}(x_1,x_2) + J^s_{k,j}(x_1,x_2)),$$

with

$$\begin{aligned}
J^s_{k,j}(x_1,x_2) &= \frac{(-1)}{(j-k)+\frac{i}{\pi}(b-a)}(\exp(x_2)\exp(-\frac{(j-k)x_2\pi i}{b-a}) \\
&- \exp(x_1)\exp(-\frac{(j-k)x_1\pi i}{b-a}))\exp(-\frac{(j+k)a\pi i}{b-a}) \\
&= \frac{(-1)}{(j-k)+\frac{i}{\pi}(b-a)}(\exp(x_2)\frac{1}{\exp(\frac{(j-k)x_2\pi i}{b-a})} \\
&- \exp(x_1)\frac{1}{\exp(\frac{(j-k)x_1\pi i}{b-a})})\exp(\frac{(j-k)a\pi i}{b-a})\frac{1}{\exp(\frac{2ja\pi i}{b-a})}
\end{aligned}$$

and

$$\begin{aligned}
J^c_{k,j}(x_1,x_2) &= \frac{(-1)}{(j+k)+\frac{i}{\pi}(b-a)}(\exp(x_2)\exp(-\frac{(j+k)x_2\pi i}{b-a}) \\
&- \exp(x_1)\exp(-\frac{(j+k)x_1\pi i}{b-a}))\exp(-\frac{(j-k)a\pi i}{b-a}) \\
&= \frac{(-1)}{(j+k)+\frac{i}{\pi}(b-a)}(\exp(x_2)\frac{1}{\exp(\frac{(j+k)x_2\pi i}{b-a})} \\
&- \exp(x_1)\frac{1}{\exp(\frac{(j+k)x_1\pi i}{b-a})})\exp(\frac{(j+k)a\pi i}{b-a})\frac{1}{\exp(\frac{2ja\pi i}{b-a})}
\end{aligned}$$

We denote $\mathbf{u} := \{u_j\}_{j=0}^{N-1}$ with

$$\begin{aligned}
u_j &= \phi(\frac{j\pi}{b-a},\Delta t,q,r,\tilde{\nu})G_j^{put}(a,0)\frac{1}{\exp(\frac{2ja}{b-a}\pi i)}, \\
u_0 &= \frac{1}{2}\phi(0,\Delta t,q,r,\tilde{\nu})G_0^{put}(a,0),
\end{aligned}$$

and we have

$$D = \frac{e^{-q\Delta t}}{\pi} Im\{(J^c+J^s)u\},$$

where J^s is a Toeplitz matrix and J^c is a Hankel matrix.

From [1] and [14] we know that matrix-vector multiplications can be performed highly efficiently then, with the help of the FFT.

With the use of the Fast Fourier and Inverse Fast Fourier Transforms, the computational complexity of $C_k(a,x^*_{M-1},t_{M-1})$ is $O(N\log_2 N)$.

We then have the following lemma:

Lemma 4.2. *For $m = M-2,\cdots,1$, $c(x,t_m)$, $x^*_m, C_k(a,x^*,t_m)$ can all be recovered from $C_k(a,x^*_{m+1},t_{m+1})$ with computational complexity $O(N\log_2 N)$ at each time step. $C_k(a,x^*_1,t_1)$ is recovered at the final step.*

Proof. For any $m = M-2,\cdots,1$, the continuation value reads:

$$\begin{aligned}
c(x,t_m) &= e^{-r\Delta t}\sum_{k=0}^{N-1}{}' Re(\phi(\frac{k\pi}{b-a},\Delta t,r,q,\nu)e^{i(x-a)\frac{k\pi}{b-a}}V_k(t_{m+1})) \\
&= e^{-r\Delta t}\sum_{k=0}^{N-1}{}' Re(\phi(\frac{k\pi}{b-a},\Delta t,r,q,\nu)e^{i(x-a)\frac{k\pi}{b-a}})\cdot \\
&\quad (C_k(a,x^*_{m+1},t_{m+1}) - G_k^{call}(0,x^*_{m+1}) + G_k^{call}(0,b)) \\
&= e^{-r\Delta t}\sum_{k=0}^{N-1}{}' Re(\phi(\frac{k\pi}{b-a},\Delta t,r,q,\nu)e^{i(x-a)\frac{k\pi}{b-a}})(C_k(a,x^*_{m+1},t_{m+1}) - G_k^{call}(0,x^*_{m+1})) \\
&+ \mathrm{e}^{-q\Delta t}\sum_{k=0}^{N-1}{}' Re(\phi(\frac{k\pi}{b-a},\Delta t,q,r,\tilde{\nu})e^{i(-x-a)\frac{k\pi}{b-a}})e^x G_k^{put}(a,0). \qquad (46)
\end{aligned}$$

The last step is from (13) and (42) and the fact that $V_k^{put}(K,S) = e^x G_k^{put}(a,0)$. $G_k^{call}(0,x^*_{m+1})$ and $G_k^{put}(a,0)$ can be calculated directly from their analytic solutions.

By (46) the continuation value $c(x,t_m)$ is recovered from $C_k(a,x^*_{m+1},t_{m+1})$ and $\partial c(x,t_m)/\partial x$ is directly calculated with (46).

The continuation value and its derivative are then used in the Newton method to find early–exercise point x^*_m, which splits $V_k(t_m)$ as follows:

$$V_k(t_m) = C_k(a,x^*_m,t_m) - G_k^{call}(0,x^*_m) + G_k^{call}(0,b).$$

From (46) we now have that

$$\begin{aligned}
C_k(a,x^*_m,t_m) &= \frac{2}{b-a}\int_a^{x^*_m} c(x,t_m)\cos(k\pi\frac{x-a}{b-a})dx \\
&= \frac{e^{-r\Delta t}}{\pi} Im((H^c(a,x^*_m)+H^s(a,x^*_m))u^1) + \\
&\quad \frac{e^{-q\Delta t}}{\pi} Im((J^c(a,x^*_m)+J^s(a,x^*_m))u^2), \qquad (47)
\end{aligned}$$

where we have four matrix-vector multiplications, instead of the usual two.

Matrices H^c and H^s are defined in (24) and (25), respectively, with $\beta = 1$. Moreover, we have in (47):

$$\begin{aligned}
u^1_0 &= \frac{1}{2}\phi(0,\Delta t,r,q,\nu)(C_0(a,x^*_{m+1},t_{m+1}) - G_0^{call}(0,x^*_{m+1})), \\
u^1_j &= \phi(\frac{j\pi}{b-a},\Delta t,r,q,\nu)(C_j(a,x^*_{m+1},t_{m+1}) - G_j^{call}(0,x^*_{m+1})), j=1,\cdots,N-1, \\
u^2_0 &= \frac{1}{2}\phi(0,\Delta t,q,r,\tilde{\nu})G_0^{put}(a,0). \\
u^2_j &= \phi(\frac{j\pi}{b-a},\Delta t,q,r,\tilde{\nu})G_j^{put}(a,0)\frac{1}{\exp(\frac{2ja}{b-a}\pi i)}, j=1,\cdots,N-1,
\end{aligned}$$

H^c and J^c are Hankel matrices, H^s and J^s are Toeplitz matrices. Therefore, the Fast Fourier Transform can be employed to compute $C_k(a, x_m^*, t_m)$, $m = M-2, \cdots, 1$ and the computational complexity at each time step is $O(N \log_2 N)$.

With $C_k(a, x_1^*, t_1)$ known, the call option price then reads:

$$
\begin{aligned}
v(x_0,t_0) &= e^{-r\Delta t} \sum_{k=0}^{N-1}{}' Re(\phi(\frac{k\pi}{b-a}, \Delta t, r, q, \nu) e^{i(x_0-a)\frac{k\pi}{b-a}}) V_k(t_1) \\
&= e^{-r\Delta t} \sum_{k=0}^{N-1}{}' Re(\phi(\frac{k\pi}{b-a}, \Delta t, r, q, \nu) e^{i(x_0-a)\frac{k\pi}{b-a}}) (C_k(a, x_1^*, t_1) - G_k^{call}(0, x_1^*)) \\
&+ e^{-q\Delta t} \sum_{k=0}^{N-1}{}' Re(\phi(\frac{k\pi}{b-a}, \Delta t, q, r, \tilde{\nu}) e^{i(-x_0-a)\frac{k\pi}{b-a}}) e^{x_0} G_k^{put}(a, 0) \qquad (48)
\end{aligned}
$$

From (46) and (47), $\forall m = M-2, \cdots, 1$, $c(x, t_m)$, x^* and $C_k(a, x^*, t_m)$ can be recovered from $C_k(a, x_{m+1}^*, t_{m+1})$ with the help of the Fast Fourier Transform, which finishes the proof. □

4.3 Error analysis with the put-call relations

As shown in the previous sections, put option values, combined with the put–call parity or the put–call duality relations, are used to price call options with the COS method. We denote by v_{call} and v_{put} the exact call and put option values, respectively, and by $\hat{v}_{put}$ the put option value obtained by the COS method. Then, from the put–call parity, we have, $\forall S, t$,

$$
\begin{aligned}
\varepsilon_{call}(x,t) &= v_{call}(x,t) - \hat{v}_{call}(x,t) \\
&= v_{put}(x,t) + Ke^x e^{-q(T-t)} - Ke^{-r(T-t)} - (\hat{v}_{put}(x,t) + Ke^x e^{-q(T-t)} - Ke^{-r(T-t)}) \\
&= v_{put}(x,t) - \hat{v}_{put}(x,t) = \varepsilon_{put}(x,t),
\end{aligned}
$$

whereas for the put–call duality, we find:

$$
\begin{aligned}
\varepsilon_{call} &= v_{call}(S, K, r, q, t, \nu) - \hat{v}_{call}(S, K, r, q, t, \nu) \\
&= v_{put}(K, S, q, r, e^{-x}\nu(-dx)) - \hat{v}_{put}(K, S, q, r, e^{-x}\nu(-dx)) = \varepsilon_{put}.
\end{aligned}
$$

So, by means of the put–call relations, the error of the call options equal that put options. As for put options the payoff is bounded, we have from (29):

$$
|\varepsilon_1(x, [a,b])| = e^{-r\Delta t} I_1 \le e^{-r\Delta t} Q_2 I_2. \qquad (49)
$$

The error can be controlled if the integration range is sufficiently large (which is our next issue). The integration range is defined as in (11) and can be controlled by parameter L.

After discussing the influence of the payoff and integration range on the error convergence separately in the previous section, here we give a remark on the interaction of them on the error convergence of ε_1.

Remark 1 (Interaction of Payoff and Truncation Range on ε_1). From (30) we see that

$$\varepsilon_1 = e^{-r\Delta t} I_1 \geq e^{-r\Delta t} K(e^b - 1) \int_b^\infty f(y|x)dy.$$

For the Black–Scholes model and other underlying processes for which the density function decays very fast both at left and right tails, the fast decay in $\int_b^\infty f(y|x)dy$ can compensate the exponential increase in $e^b - 1$. On the other hand, for underlying processes with fat tails, for instance, the CGMY model with Y close to 2, or with a long maturity, the error decay rate with respect to L is not so high and we require a larger integration range. In these cases the increase in $e^b - 1$ may give rise to divergence of the call value and the put–call parity or the put–call duality *should be used for robust and accurate option values*. This is further illustrated by numerical examples in Section 5.

5 Numerical Examples

In this section we will show the method's accuracy, efficiency and robustness by a series of numerical examples. The CPU used is an Intel(R) Core(TM)2 Duo CPU E6550 (2.33GHz Cache size 4MB) with an implementation in Matlab 7.7.0.

We use as reference values the Bermudan option prices obtained by the robust version of the COS method, with a very fine grid (with $N = 2^{14}$).

In the experiments, we will use the CGMY model, with test parameters $Y = 0.5$, $Y = 1.5$ and $Y = 1.98$; the remaining CGMY parameters are chosen as $[C, M, G] = [1, 5, 5]$. Other parameters include: $r = 0.1, q = 0.02, S_0 = 100, K = 110$. We set again $M = 10$ and maturity $T = 1$. Computational time and the absolute error in the option value are displayed in Tables 1 to 3. From these tables we see that for $Y = 0.5$ $N = 256$ is sufficient while for $Y = 1.5$ and $Y = 1.98$ it is $N = 128$. When $Y > 1$, which implies that the process has infinite activity, the error in the option price is of order 10^{-12}. From the tables we see that the methods with both the put–call parity and the put–call duality converge very well within milliseconds. The CPU time when using the put–call duality is approximately twice the time with put–call parity, because with the put–call duality we need to calculate two matrix-vector products with Hankel and Toeplitz matrices at each time step.

Figure 3 compares Bermudan call option values under the GBM model, obtained directly by the COS method with the values obtained via the put–call parity or the put–call duality, and with reference values. The dividend rate is $q = 0.02$, and the

	N	64	128	256	512
Parity:	abs.err	2.9497e-004	1.0586e-005	8.5622e-007	1.1607e-007
	msec.	4.959	6.819	10.484	18.878
Duality:	abs.err	3.7177e-002	8.5904e-005	5.8262e-005	6.4494e-006
	msec.	8.000	12.105	19.778	35.554

Table 1 Absolute error and CPU time (in milli–seconds) for the CGMY model, $Y = 0.5$. COS pricing with the put-call relations.

	N	32	64	128	256
Parity:	abs.err	7.7799e-003	1.8691e-005	2.2737e-012	5.6843e-014
	msec.	3.735	4.699	6.760	10.527
Duality:	abs.err	2.8937e-002	1.3074e-002	5.8769e-007	7.9581e-013
	msec.	5.839	8.009	12.078	20.016

Table 2 Absolute error and CPU time (in milli–seconds) for the CGMY model, $Y = 1.5$. COS pricing with the put-call relations.

	N	32	64	128	256
Parity:	abs.err	4.0414e-001	3.8936e-004	1.1369e-013	$< 1e-016$
	msec.	3.690	4.831	6.664	10.577
Duality:	abs.err	1.5431e-001	3.4510e-006	1.4495e-011	6.9207e-012
	msec.	7.927	12.034	19.643	35.400

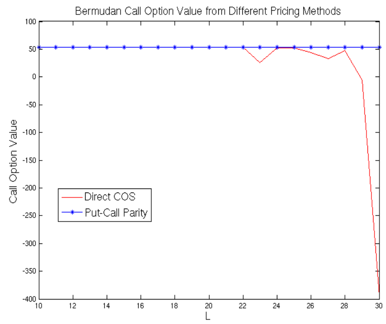
Table 3 Absolute error and CPU time (in milli–seconds) for the CGMY model, $Y = 1.98$. COS pricing with the put-call relations.

reference value is $53.355758\ldots$. For very large values, $L > 20$, the option values obtained by the COS method (without the put-call relations) differ dramatically from the reference values. Pricing is robust, with respect to the size of the integration interval when the put–call parity and the put–call duality are applied, as then accurate call prices are obtained for any value of L, see Figure 3.

We again consider the CGMY model, for which Figure 4 shows Bermudan pricing results for $Y = 0.5$ and $r = 0.1, q = 0.02$. The other parameter values are as in the previous experiments. The reference value is $23.574835\ldots$. Compared to Figure 3, the error in Bermudan call option values under this CGMY parameter set is significantly larger than under the GBM model. However, combined with the put–call parity or the put–call duality, the option prices converge in a robust way to the reference value, for all L.

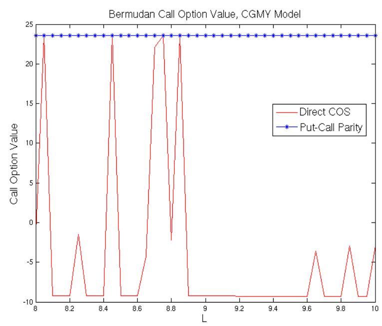
With parameter Y close to 2 in CGMY, the Bermudan call prices, computed directly by the COS method are subject to cancellation errors even for small sizes of the computational domain and small maturity dates, as shown in Figure 5. Here the reference value for the Bermudan call is $99.053582\ldots$. With T and M increasing, the error also increases. The COS method with the put–call parity or the put–call duality remains however robust also for these parameter values.

Comparing Figures 5 and 4, we see that as Y increases, which implies a fatter tail in the probability density function of the underlying, the error in the call price



(a) Put–Call Parity (b) Put–Call Duality

Fig. 3 Bermudan call option values with varying L-values, GBM model, $r = 0.1, q = 0.02, \sigma = 0.2, T = 10, M = 50$, $L \in [10, 30]$.



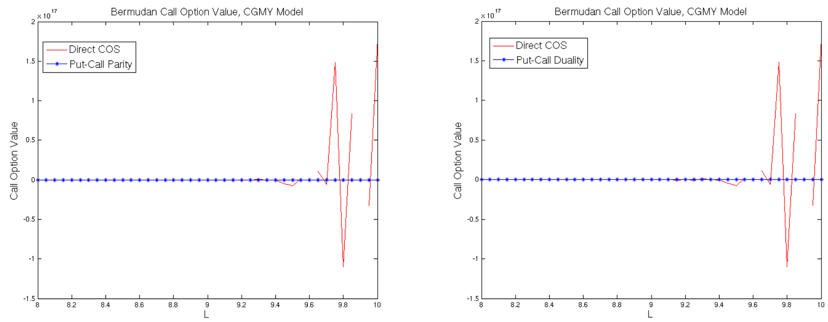
(a) Put–Call Parity (b) Put–Call Duality

Fig. 4 Bermudan call option values with varying L-values, CGMY model with $q = 0.02, Y = 0.5, M = 24$, $L \in [8, 10]$.

obtained by the COS method with respect to large computational domain sizes increases drastically.

5.1 American Options

Finally, we price an American call option by the 4-point Richardson extrapolation using (27) with Bermudan options. We use the CGMY model with $Y = 1.5$ and 1.98, and $q = 0.05$, and compare American calls for which the Bermudan calls in the extrapolation are priced directly by the COS method with those computed using the put–call parity or the put–call duality. In the COS method we used $N = 1024$



(a) Put–Call Parity (b) Put–Call Duality

Fig. 5 Bermudan call option values with varying L-values, CGMY model with $q = 0.05, Y = 1.98, M = 10$, $L \in [8,10]$.

in the case with $Y = 1.98, M = 32$ (M as in (27)); in all other cases, $N = 512$ is sufficient for convergence. The number of Newton iterations is set to 5 (as in [14]).

The accuracy of the American prices depends on parameter M in the extrapolation formula (27). The results obtained are in Tables 4 and 5 with CPU time in seconds. In these table the American option prices are accurate and robust when the put–call parity or the put–call duality was used in the COS pricing procedure.

M in Eq. (27)	Put-Call Parity		Put-Call Duality		direct COS method	
	option value	time (sec.)	option value	time (sec.)	option value	time (sec.)
8	44.0934	0.243	44.0934	0.501	58.3396	0.238
16	44.0933	0.489	44.0933	1.002	56.6221	0.428
32	44.0936	0.998	44.0934	2.014	-5.3915e+02	0.840

Table 4 American call option values and CPU time (in seconds) by Richardson extrapolation, under the CGMY model with, $Y = 1.5, q = 0.05$,

M in Eq. (27)	Put-Call Parity		Put-Call Duality		direct COS method	
	option value	time (sec.)	option value	time (sec.)	option value	time (sec.)
8	99.1739	0.244	99.1739	0.497	-2.2964e+48	0.221
16	99.1739	0.520	99.1739	0.987	5.0141e+46	0.460
32	99.1738	0.976	99.1738	3.761	2.1427e+53	0.820

Table 5 American call option values and CPU times (in seconds) by Richardson extrapolation, under the CGMY model with $Y = 1.98, q = 0.05$,

6 Conclusions and Discussion

In this chapter, we have discussed the generalization of the COS option pricing method, based on Fourier cosine expansions, from European options to Bermudan options. The method can be used whenever the characteristic function of the underlying price process is available. It is especially efficient for exponential Lévy processes.

The COS formula for European options from [13] can be used for pricing Bermudan options, if the series coefficients of the option values at the first early-exercise 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 options under Lévy processes with M exercise dates. The COS method exhibits an exponential convergence in N for density functions in $C^\infty[a,b]$ and an impressive computational speed. With a limited number, N, of Fourier cosine coefficients, it produces highly accurate results. We have also presented error analysis for this method, showing that convergence for put options is easily obtained, whereas the unbounded payoff function for calls may hamper the robust convergence. The convergence of directly applying the COS method to call options depends of the choice of the integration range. Robust pricing, insensitive of the choice of the size of the integration range is achieved for call options, when the put-call parity or the put–call duality relation is applied. The use of these relations for call options with early exercise features has been explained in detail. This is a novel aspect of this work, resulting is a robust pricing technique for Bermudan and American options, independent of the size of the computational domain.

Pricing American options can be done by a Richardson extrapolation method on Bermudan options with a varying number of exercise dates.

References

1. A. Almendral and C. W. Oosterlee. On American options under the Variance Gamma process. *Appl. Math. Finance*, 14(2):131–152, 2007.
2. A. D. Andricopoulos, M. Widdicks, P. W. Duck, and D. P. Newton. Universal option valuation using quadrature methods. *J. Fin. Economics*, 67:447–471, 2003.
3. A. D. Andricopoulos, M. Widdicks, P. W. Duck, and D. P. Newton. Extending quadrature methods to value multi-asset and complex path dependent options. *J. Fin. Economics*, 83(2):471–500, 2007.
4. O. E. Barndorff-Nielsen. Normal inverse Gaussian distributions and stochastic volatility modelling. *Scand. J. Statist.*, 24(1-13), 1997.
5. M. Broadie and Y. Yamamoto. Application of the fast Gauss transform to option pricing. *Management Sci.*, 49:1071–1008, 2003.
6. M. Broadie and Y. Yamamoto. A double-exponential fast Gauss transform for pricing discrete path-dependent options. *Operations Research*, 53(5):764–779, 2005.
7. P. P. Carr, H. Geman, D. B. Madan, and M. Yor. The fine structure of asset returns: An empirical investigation. *J. Business*, 75:305–332, 2002.
8. P. P. Carr and D. B. Madan. Option valuation using the fast Fourier transform. *J. Comp. Finance*, 2:61–73, 1999.

9. C-C Chang, S-L Chung, and R. C. Stapleton. Richardson extrapolation technique for pricing American-style options. *J. Futures Markets*, 27(8):791–817, 2007.
10. R. Cont and P. Tankov. *Financial Modelling with Jump Processes*. Chapman and Hall, Boca Raton, FL, 2004.
11. A. Eydeland. A fast algorithm for computing integrals in function spaces: financial applications. *Computational Economics*, 7(4):277–285, 1994.
12. J. Fajardo, E. Mordecki. Symmetry and duality in Leacutevy markets. *Quantitative Finance*, 6(3): 219–227, 2006.
13. F. Fang and C. W. Oosterlee. A novel option pricing method based on Fourier-cosine series expansions. *SIAM J. Sci. Comput.*, 31(2):826–848, 2008.
14. F. Fang and C. W. Oosterlee. Pricing early-exercise and discrete barrier options by Fourier-cosine series expansions. *Numerische Mathematik*, 114(1):27–62, 2009.
15. L. Feng and V. Linetsky. Pricing discretely monitored barrier options and defaultable bonds in Lévy process models: a fast Heston transform approach. *Math. Finance*, 18(3):337–384, 2008.
16. R. Geske and H. Johnson. The American put valued analytically. *J. of Finance*, 39:1511–1542, 1984.
17. K. Jackson, S. Jaimungal, and V. Surkov. Option pricing with regime switching Lévy processes using Fourier space time-stepping. *Proc. 4th IASTED Intern. Conf. Financial Engin. Applic.*, pages 92–97, 2007.
18. S. G. Kou. A jump diffusion model for option pricing. *Management Science*, 48(8):1086–1101, 2002.
19. R. Lord, F. Fang, F. Bervoets, and C.W. Oosterlee. A fast and accurate FFT-based method for pricing early-exercise options under Lévy processes. *SIAM J. Sci. Comput.*, 30:1678–1705, 2008.
20. R. Lord and C. Kahl. Optimal Fourier inversion in semi-analytical option pricing. *J. Comp. Finance*, 10:1–30, 2007.
21. D. B. Madan, P. R. Carr, and E. C. Chang. The Variance Gamma process and option pricing. *European Finance Review*, 2:79–105, 1998.
22. R. Merton. Option pricing when underlying stock returns are discontinuous. *J. Financial Economics*, 3:125–144, 1976.
23. W. Schoutens. *Lévy processes in finance: Pricing financial derivatives*. Wiley, 2003.
24. Y. Yamamoto. Double-exponential fast Gauss transform algorithms for pricing discrete lookback options. *Publ. Res. Inst. Math. Sci.*, 41:989–1006, 2005.
25. B. Zhang, C.W. Oosterlee. An efficient pricing algorithm for swing options based on Fourier Cosine expansions. *To appear in the Journal of Computational Finance*.