DELFT UNIVERSITY OF TECHNOLOGY

REPORT 12-03

Efficient Pricing of Asian Options under Lévy Processes based on Fourier Cosine Expansions Part II Early–Exercise Features and GPU Implementation

B. Zhang   J.A.M.van der Weide   C.W. Oosterlee

ISSN 1389-6520

Reports of the Department of Applied Mathematical Analysis

Delft 2012

Efficient Pricing of Asian Options under Lévy Processes based on Fourier Cosine Expansions Part II Early–Exercise Features and GPU Implementation

B. Zhang* J.A.M.van der Weide† C.W. Oosterlee‡

May 23, 2012

Abstract

In this article, we propose an efficient pricing method for Asian options with early–exercise features. It is based on a two–dimensional integration and a backward recursion of the Fourier coefficients, in which several numerical techniques, like Fourier cosine expansions, Clenshaw–Curtis quadrature and the Fast Fourier transform (FFT) are employed. Rapid convergence of the pricing method is illustrated by an error analysis. Its performance is further demonstrated by various numerical examples, where we also show the power of an implementation on the Graphics Processing Unit (GPU).

Keywords: Early–exercise Asian option; arithmetic average; Fourier cosine expansion; chain rule; Clenshaw–Curtis quadrature; exponential convergence; graphics processing unit (GPU) computation.

1 Introduction

An Asian option is a special type of exotic option, introduced in Japan, in 1987. Because the contract description (i.e. the pay-off function) is based on geometric or arithmetic averages of the underlying stock price at monitoring dates, rather than just on the present asset price, this exotic option is also called *path-dependent*. The number of monitoring dates can be finite (so-called discretely–monitored Asian options) or infinite (continuously–monitored Asian options). Asian options are popular, because averages typically move in a more stable way than individual asset prices, and the volatility, inherent in asset prices, is reduced due to the averaging feature, so that Asian option holders may pay lower prices for these contracts, compared to plain European options.

There is not much information on early-exercise Asian option products in the present markets. We may encounter them in the commodity market, and variants in the equity market are so-called American options with an Asian tail (meaning that the final part of the contract time is based on averaged asset prices rather than on plain assets). In the academic literature, important contributions [6, 2] have

*Delft University of Technology, Delft Institute of Applied Mathematics, email: Bowen.Zhang@tudelft.nl

†Delft University of Technology, Delft Institute of Applied Mathematics, email: J.A.M.vanderWeide@tudelft.nl

‡CWI – Centrum Wiskunde & Informatica, Amsterdam, the Netherlands, email: C.W.Oosterlee@cwi.nl, and Delft University of Technology, Delft Institute of Applied Mathematics.

been presented, when pricing these Asian options by partial differential and partial integro-differential equations (PDEs and PIDEs, respectively). In [6], for example, a semi–Lagrangian time-stepping method was used to solve the P(I)DE in a time-stepping procedure. The method worked particularly well for American-style Asian options under a jump–diffusion model.

In [12], European-style Asian options were priced by means of Fourier cosine expansions (as in the COS method [7]) and Clenshaw–Curtis quadrature. The method was named the *Asian cosine method* (ASCOS). This new pricing method can be seen as an efficient alternative to Fast Fourier Transform (FFT) and convolution methods, as in [4, 9, 1, 10], for pricing European–style Asian options under Lévy processes.

In this paper, which is the Part II paper of the European Asian option paper [12], we propose an efficient version of the ASCOS pricing method for early–exercise Asian options, again based on Fourier expansions, Fast Fourier Transform (FFT) and Clenshaw–Curtis quadrature. In the 2D ASCOS method the option price is calculated based on two dimensions of uncertainty, i.e. the uncertainty in the asset process, as well as in the *averaged asset process* over time. The risk–neutral formula then becomes a two–dimensional integration, based on which the continuation value is approximated at each time step. By application of the *chain rule* from probability theory, the joint conditional density function in the risk–neutral formula can be factorized into two marginal conditional density functions that are approximated by Fourier cosine expansions. To calculate the option price, we need to recursively recover the Fourier coefficients with the help of Fourier cosine expansions and Clenshaw–Curtis quadrature. The FFT is used to accelerate the algorithm. The computational complexity of our pricing method for Asian options, with $\mathcal{M}$ early–exercise dates, is $O((\mathcal{M}-1)n_q N_1 N_2 \log_2 N_2)$, with N_1, N_2 the number of Fourier cosine terms in the expansions for the density functions of the asset and the averaged asset price, respectively, and n_q the number of terms in the Clenshaw-Curtis quadrature.

Exponential convergence in the option price, with respect to n_q, N_1, N_2, is obtained for most Lévy processes, for which we give an error analysis, combined with numerical examples. The 2D method is presented in Section 4, followed by an error analysis in Section 5. Numerical results are given in Section 6, where the efficiency and accuracy of the pricing methods are presented. Implementation has taken place on the Graphics Processing Unit (GPU). It may be interesting to see that this computer architecture improves the pricing speed drastically when pricing arithmetic Asian options with early-exercise features.

We start, however, in Section 3, with another, alternative Asian pricing method, which is only accurate in the case of a large number of early–exercise dates. It has a reduced computational cost of $O((\mathcal{M}-1)n_q N^2)$, with N the number of terms in a 1D Fourier cosine expansion. The approximation error appearing from the approximations in this alternative method converges to zero only when the number of early–exercise dates tends to infinity.

2 Early-exercise Asian options under Lévy processes

In this article the underlying asset, S_t, is assumed to be an exponential function of a Lévy process, L_t, i.e. $S_t = S_0 \exp(L_t)$. Lévy process, L_t, with initial condition $L_0 = 0$, has independent and stationary increments and is stochastically continuous. For any $s < t$, and $\forall \epsilon > 0$, we have

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

The (conditional) probability density function is not known for many relevant Lévy asset processes. However, its *Fourier transform*, the (conditional) characteristic function, $\phi_{Y_m|Y_{m-1}}(\cdot)$, is often available, for example, by the Lévy-Khinchine theorem for underlying Lévy processes. Our pricing algorithm is based on this Fourier transform.

In this paper, we deal with early–exercise options in which the contract function at each exercise date is a function of the averaged underlying asset price, up to that date. *Early-exercise* implies that the option may be exercised prior to the expiration date. Let t_0 denote the initial time and $\mathcal{T} = \{t_1, \cdots, t_{\mathcal{M}}\}$ be the collection of all exercise dates with $\Delta t := (t_m - t_{m-1})$, $t_0 < t_1 < \cdots < t_{\mathcal{M}} = T$, and assume that the early–exercise dates and the monitoring dates of the Asian options are the same.

We focus on arithmetic averaging, as it is mathematically more challenging, and on *fixed–strike Asian options*, with payoff functions defined by

$$g(S, t_m) = \begin{cases} \max(\frac{1}{m+1}\sum_{j=0}^{m} S_j - K, 0), & \text{for a call,} \\ \max(K - \frac{1}{m+1}\sum_{j=0}^{m} S_j, 0), & \text{for a put,} \end{cases}$$

These payoff functions change from time step to time step, due to the averaging feature.

3 A first Asian pricing method (for $\mathcal{M} \to \infty$)

We present here a first pricing algorithm for early–exercise arithmetic Asian options, *which is only accurate for a large number of early-exercise dates*, as we will proof in the subsection to follow.

The pricing formula for an early–exercise Asian option with $\mathcal{M}$ exercise dates then reads, for $m = \mathcal{M}, \mathcal{M} - 1, \ldots, 2$:

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

followed by

$$v(y_0, t_0) = e^{-r\Delta t} \int_{\mathbb{R}} v(y_1, t_1) f(y_1|y_0) dy_1. \tag{2}$$

Here, y_m is the state variable at time step t_m, and $v(x, t)$, $c(x, t)$ and $g(x, t)$ are the option value, the continuation value and the payoff at time t, respectively. $v(S, t_{\mathcal{M}}) = g(S, t_{\mathcal{M}})$ is the payoff function at final time, $t_{\mathcal{M}} = T$. Function $f(y_m|y_{m-1})$ is the conditional density of y_m given y_{m-1}. Interest rate r is assumed to be constant here.

In the risk–neutral formula (1) the continuation value is computed at each time step as the discounted expected value of the option price at a next time step. Moreover, to avoid arbitrage opportunities, the option value at each time step cannot be less than the payoff of the option, which is the second equation in (1).

In [7, 8] the COS method was developed for the computation of continuation value $c(y_{m-1}, t_{m-1})$ and option price $v(y_0, t_0)$, for *vanilla* (i.e. non-path dependent) Bermudan options, under the assumption that the characteristic function of the underlying Lévy asset price process is known.

The pricing algorithm for early–exercise arithmetic Asian options in this section can be seen as a generalization of the COS method for Bermudan options (a general 2D pricing algorithm will be presented in Section 4).

Here, we define

$$Y_m := \log(\frac{S_1}{S_0} + \frac{S_2}{S_0} + \cdots + \frac{S_m}{S_0}),\ m = \mathcal{M}, \mathcal{M}-1, \cdots, 1. \quad (3)$$

Based on this variable, the payoff function is given by

$$g(y_m, t_m) = \begin{cases} (\frac{S_0(1+e^{y_m})}{m+1} - K)^+, & \text{for a call,} \\ (K - \frac{S_0(1+e^{y_m})}{m+1})^+, & \text{for a put,} \end{cases} \quad (4)$$

After truncation of the integration range in (1) and (2), from $\mathbb{R}$ to $[a, b]$, we approximate the conditional density function in terms of its characteristic function, via a Fourier cosine expansion. For $m = \mathcal{M}-1, \cdots, 1$ the continuation value is approximated by Fourier cosine expansions, as

$$\hat{c}(y_{m-1}, t_{m-1}) = e^{-r\Delta t} \sum_{k=0}^{N-1}{}' Re\left(\hat{\varphi}_{Y_m|Y_{m-1}}(\frac{k\pi}{b-a}; y_{m-1}, \Delta t) e^{-ik\pi\frac{a}{b-a}}\right) \hat{V}_k(t_m), \quad (5)$$

where $\hat{c}, \hat{V}, \hat{\phi}$ indicate that these are numerical approximations. The prime at the sum symbol indicates that the first term in the summation is weighted by one-half. Conditional characteristic function $\hat{\varphi}_{Y_m|Y_{m-1}}(u; y_{m-1}, \Delta t)$, in (5), will be derived in Subsection 3.1. The Fourier cosine coefficients of the option price at t_m in (5) are defined by

$$\hat{V}_k(t_m) = \int_a^b \hat{v}(y_m, t_m) \cos(k\pi \frac{y_m - a}{b-a}) dy_m, \quad (6)$$

The target option price is obtained by computing

$$\hat{v}(y_0, t_0) = e^{-r\Delta t} \sum_{k=0}^{N-1}{}' Re\left(\phi_{Y_1|Y_0}(\frac{k\pi}{b-a}; y_0, \Delta t) e^{-ik\pi\frac{a}{b-a}}\right) \hat{V}_k(t_1), \quad (7)$$

where for arithmetic Asian options under Lévy processes, $\phi_{Y_1|Y_0}$ is known analytically.

Based on the conditional characteristic function for Y_m in (3), the early–exercise arithmetic Asian option value can be calculated by a *backward recursion* on the Fourier coefficients, $\hat{V}_k(t_m)$, as defined in (6). Then, the option price is obtained by inserting the value of $\hat{V}_k(t_1)$ into (7).

At maturity time, $t_{\mathcal{M}}$, the option value equals the payoff, and Fourier coefficients, $V_k(t_{\mathcal{M}})$, read:

$$V_k(t_{\mathcal{M}}) = \begin{cases} \frac{2}{b-a}(\frac{S_0}{\mathcal{M}+1}\chi_k(y^*_{\mathcal{M}}, b) + (\frac{S_0}{\mathcal{M}+1} - K)\psi_k(y^*_{\mathcal{M}}, b)), & \text{for a call,} \\ \frac{2}{b-a}((K - \frac{S_0}{\mathcal{M}+1})\psi_k(a, y^*_{\mathcal{M}}) - \frac{S_0}{\mathcal{M}+1}\chi_k(a, y^*_{\mathcal{M}})), & \text{for a put,} \end{cases} \quad (8)$$

with $y^*_{\mathcal{M}} = \log(\frac{K(\mathcal{M}+1)}{S_0} - 1)$, the value for which the payoff is nonzero, and

$$\psi_j(y_l, y_u) := \int_{y_l}^{y_u} \cos\left(j\pi \frac{y - \delta_n}{b_2 - \delta_n}\right) dy, \quad (9)$$

and

$$\chi_j(y_l, y_u) := \int_{y_l}^{y_u} e^y \cos\left(j\pi \frac{y - \delta_n}{b_2 - \delta_n}\right) dy. \quad (10)$$

At recursion step $t_m, m = \mathcal{M}-1, \cdots, 1$, as a first step in the algorithm, the so-called *early–exercise point*, y^*_m, for which $c(y^*_m, t_m) = g(y^*_m, t_m)$, is determined by Newton's method, as the derivatives of the continuation value and the payoff function, with respect to y_m, can be easily derived. Based on this, we can split $V_k(t_m)$, as follows

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

where C_k, G_k are Fourier cosine coefficients of the continuation value and payoff at t_m, respectively. Coefficients G_k are of the form

$$G_k(t_m, y_l, y_u) = \frac{2}{b-a} \int_{y_l}^{y_u} g(y, t_m) \cos(k\pi \frac{y-a}{b-a}) dy, \tag{11}$$

where for a call, $[y_l, y_u] = [y_m^*, b]$, and for a put, $[y_l, y_u] = [a, y_m^*]$. Inserting (4) into (11) gives us

$$G_k(t_m) = \frac{2}{b-a} \begin{cases} \dfrac{S_0}{m+1} \chi_k(y_m^*, b) + (\dfrac{S_0}{m+1} - K) \psi_k(y_m^*, b), & \text{for a call,} \\ \left(K - \dfrac{S_0}{m+1}\right) \psi_k(a, y_m^*) - \dfrac{S_0}{m+1} \chi_k(a, y_m^*), & \text{for a put.} \end{cases}$$

Coefficients C_k, approximated by $\hat{C}_k$, defined as

$$\hat{C}_k(y_l, y_u, t_m) = \frac{2}{b-a} \int_{y_l}^{y_u} \hat{c}(y, t_m) \cos(k\pi \frac{y-a}{b-a}) dy, \tag{12}$$

are computed numerically. In the subsection to follow, we will show that, $\forall u \in \mathbb{R}$, the conditional characteristic function $\varphi_{Y_m|Y_{m-1}}(u; y_{m-1}, \Delta t)$ can be approximated by

$$\hat{\varphi}_{Y_m|Y_{m-1}}(u; y_{m-1}, \Delta t) \approx (1 + e^{Y_{m-1}})^{iu} \phi_Z(u; \Delta t), \tag{13}$$

and that the error from approximation (13) converges to zero only when $\mathcal{M}$ goes to infinity. The distribution of Z is identical to that of the logarithm of increment between (any) two consecutive time steps of a Lévy process.

Applying (5) and (13) in (12), gives us

$$\begin{aligned} \hat{C}_k(y_l, y_u, t_m) &= \int_{y_l}^{y_u} \hat{c}(y_m, t_m) \cos(k\pi \frac{y_m - a}{b-a}) dy_m \qquad (14) \\ &= e^{-r\Delta t} Re \left({\sum_{j=0}^{N-1}}' \phi_Z(\frac{j\pi}{b-a}; \Delta t) e^{-ij\pi \frac{a}{b-a}} \hat{V}_j(t_{m+1}) \mathcal{H}_{k,j}(y_l, y_u) \right), \end{aligned}$$

where function $\mathcal{H}_{k,j}(y_l, y_u)$ is given by

$$\mathcal{H}_{k,j}(y_l, y_u) = \frac{2}{b-a} \int_{y_l}^{y_u} (1 + e^y)^{i\frac{j\pi}{b-a}} \cos(k\pi \frac{y-a}{b-a}) dy. \tag{15}$$

With $\hat{C}(y_l, y_u, t_m) := \{\hat{C}_k(y_l, y_u, t_m)\}_{k=0}^{N-1}$, Equation (14) can be written in matrix–vector multiplication form, as

$$\hat{C}(y_l, y_u, t_m) = e^{-r\Delta t} Re(\mathcal{H} \cdot \mathbf{u}), \tag{16}$$

with $\mathcal{H} := \{\mathcal{H}_{k,j}\}_{k,j=0}^{N-1}$, $\mathbf{u} := \{\mathbf{u}_j\}_{j=0}^{N-1}$, and

$$\begin{aligned} \mathbf{u}_0 &= \frac{1}{2} \phi_Z(0; \Delta t) \hat{V}_0(t_{m+1}), \\ \mathbf{u}_j &= \phi_Z(\frac{j\pi}{b-a}; \Delta t) e^{-ij\pi \frac{a}{b-a}} \hat{V}_j(t_{m+1}), \ (j \neq 0). \qquad (17) \end{aligned}$$

Integral $\mathcal{H}_{k,j}(y_l, y_u)$ in (15) can be rewritten in terms of Beta functions, however, the calculation of these Beta functions, with complex-valued arguments for all k, j, is computationally expensive. Therefore, as in [12], we will use the Clenshaw–Curtis quadrature rule to calculate the integrals $\mathcal{H}_{k,j}(y_l, y_u)$ in (15).

By recursion, we get the $\hat{V}_k(t_1)$-coefficients, and the value of an early–exercise arithmetic Asian option is given by

$$\hat{v}(y_0, t_0) = e^{-r\Delta t} \sideset{}{'}\sum_{k=0}^{N-1} Re(\phi_{\log(\frac{S_1}{S_0})}(\frac{k\pi}{b-a}; \Delta t) e^{-ik\pi\frac{a}{b-a}})\hat{V}_k(t_1),$$

where $y_0 = \log(S_0)$ and $\phi_{\log(\frac{S_1}{S_0})}(u; t)$, $\forall u, t$, is known analytically for most Lévy processes.

At each time step the computational complexity is $O(n_q N^2)$ and $O(N)$ to compute the $\hat{C}_k$- and $\hat{G}_k$-terms, respectively, where n_q denotes the number of terms in the discrete cosine expansion of the Clenshaw–Curtis quadrature. In total, $O((\mathcal{M} - 1)n_q N^2)$ computations are required for early-exercise arithmetic Asian options under Lévy processes.

Remark 3.1 (Error Analysis)**.** *With the conditional characteristic function well approximated, the error in the option price propagates basically in the same way as for a plain vanilla Bermudan option, for which we refer to [8], where a detailed error analysis was given.*

3.1 Characteristic function of the first pricing method

The fact that the pricing method explained is only highly accurate for $\mathcal{M} \to \infty$ will be presented in this section, where we derive the conditional characteristic function.

First, we give two lemmas which will be used later in our analysis.

Lemma 3.1. *A conditional characteristic function, $\varphi_{Y|X}(u; x, t)$, satisfies, $\forall X, Y$ with X and $Y - X$ independent, and $\forall u \in \mathbb{R}$,*

$$\varphi_{Y|X}(u; x, t) = e^{iuX} \phi_Z(u; t),$$

where Z and $Y - X$ are identically distributed, that is, $Z \overset{d}{=} Y - X$.

Proof. From the definition, we have

$$\varphi_{Y|X}(u; x, t) = \mathbb{E}(e^{iuY}|X) = e^{iuX}\mathbb{E}(e^{iu(Y-X)}|X).$$

With the notion of independence of X and $Y - X$, it follows that:

$$e^{iuX}\mathbb{E}(e^{iu(Y-X)}|X) = e^{iuX}\mathbb{E}(e^{iu(Y-X)}) = e^{iuX}\phi_{Y-X}(u; t) = e^{iuX}\phi_Z(u; t),$$

which proves the lemma. □

Lemma 3.2. *For random variables X, Y, with a bijective and bi-measurable function $h : \mathbb{R} \to \mathbb{R}$, we have that, $\forall u \in \mathbb{R}$,*

$$\varphi_{Y|X}(u; x, t) = \varphi_{Y|h(X)}(u; h(x), t). \tag{18}$$

Proof. The proof is straightforward as the σ-fields generated by X and $h(X)$ coincide. □

As a consequence of the Lemmas 3.2 and 3.1, we find that, for random variables X, Y and bijective function $h : \mathbb{R} \to \mathbb{R}$, with $h(X)$ and $Y - h(X)$ independent,

$$\varphi_{Y|X}(u; x, t) = e^{iuh(X)}\phi_Z(u; t), \ \forall u \in \mathbb{R} \tag{19}$$

where $Z \overset{d}{=} Y - h(X)$.

The basis for the efficient 1D Asian pricing algorithm is in the following lemma, which describes a relation between the characteristic functions of the state variables at consecutive time steps t_m and t_{m-1}, for Lévy processes.

Lemma 3.3. *If we define*

$$Y_m := \log(\frac{S_1}{S_0} + \frac{S_2}{S_0} + \cdots + \frac{S_m}{S_0}), \ m = \mathcal{M}, \mathcal{M}-1, \cdots, 1,$$

then, at time step t_m, $m = \mathcal{M}, \cdots, 2$, we have, $\forall u \in \mathbb{R}$, in the case of a Lévy process, S_t, that

$$\phi_{Y_m}(u; t) = \phi_{\log(1+e^{Y_{m-1}})}(u; t)\phi_{Z_m}(u; t), \tag{20}$$

where

$$Z_m := \log(\frac{S_m}{S_{m-1}}). \tag{21}$$

Proof. For all $u \in \mathbb{R}$, we have

$$Y_m = \log(1 + e^{W_{m-1}}) + Z_1,$$

with

$$W_{m-1} := \log(\frac{S_2}{S_1} + \frac{S_3}{S_1} + \cdots + \frac{S_m}{S_1}).$$

Note that W_{m-1} and Z_1 are independent, as Lévy processes are defined by the property of independent increments. Therefore, $\forall u \in \mathbb{R}$,

$$\phi_{Y_m}(u; t) = \phi_{\log(1+e^{W_{m-1}})}(u; t)\phi_{Z_1}(u; t).$$

Moreover, a Lévy process has stationary increments, which implies that, $\forall u, m$, $\phi_{Z_1}(u; t) = \phi_{Z_m}(u; t)$, so that we find

$$\phi_{Y_m}(u; t) = \phi_{\log(1+e^{W_{m-1}})}(u; t)\phi_{Z_m}(u; t).$$

Now, we only need to prove that, $\forall u \in \mathbb{R}$,

$$\phi_{\log(1+e^{Y_{m-1}})}(u; t) = \phi_{\log(1+e^{W_{m-1}})}(u; t). \tag{22}$$

Here, $e^{Y_{m-1}}$ and $e^{W_{m-1}}$ can be rewritten as follows

$$\begin{aligned} e^{Y_{m-1}} &= e^{Z_1} + e^{Z_1+Z_2} + \cdots + e^{Z_1+\cdots+Z_{m-1}}, \\ e^{W_{m-1}} &= e^{Z_2} + e^{Z_2+Z_3} + \cdots + e^{Z_2+\cdots+Z_m}, \end{aligned}$$

where Z_m is defined in (21). For a Lévy process all $Z_j, j = 1, \cdots, \mathcal{M}$, are identically and independently distributed, so that $e^{Y_{m-1}} \overset{d}{=} e^{W_{m-1}}$, and $\forall u$, $\phi_{e^{Y_{m-1}}}(u; t) = \phi_{e^{W_{m-1}}}(u; t)$.

Equation (22) can be proved by the fact that for any two random variables, X, Y, if we have $\phi_X(u; t) = \phi_Y(u; t)$, $\forall u \in \mathbb{R}$, then, for any bijective and bi–measurable function $h : \mathbb{R} \to \mathbb{R}$, we have

$$\phi_{h(X)}(u; t) = \phi_{h(Y)}(u; t). \tag{23}$$

This concludes the proof. □

From (20) and since Y_{m-1} and Z_m are independent variables, as we work with Lévy processes, we have

$$\phi_{Y_m}(u;t) = \phi_{\log(1+e^{Y_{m-1}})+Z_m}(u;t), \forall u, t,$$

which implies that Y_m and $\log(1+e^{Y_{m-1}}) + Z_m$ are identically distributed, i.e. $Y_m \stackrel{d}{=} \log(1+e^{Y_{m-1}}) + Z_m$.

Let

$$\bar{Y}_m := \log(1+e^{Y_{m-1}}) + Z_m,$$

so that $Y_m \stackrel{d}{=} \bar{Y}_m$. Conditional characteristic function $\varphi_{\bar{Y}_m|Y_{m-1}}(u; y_{m-1}, \Delta t)$ is known in closed form, that is, we set $h(Y_{m-1}) := \log(1+e^{Y_{m-1}})$, and, from (19), $\forall y_{m-1}, u \in \mathbb{R}$, we find

$$\begin{aligned}
\varphi_{\bar{Y}_m|Y_{m-1}}(u; y_{m-1}, \Delta t) &= \varphi_{\bar{Y}_m|h(Y_{m-1})}(u; h(y_{m-1}), \Delta t) \\
&= e^{iuh(Y_{m-1})}\phi_{\bar{Y}_m - h(Y_{m-1})}(u; \Delta t) \\
&= (1+e^{Y_{m-1}})^{iu}\phi_{Z_m}(u; \Delta t). \qquad (24)
\end{aligned}$$

Our aim is to approximate the conditional characteristic function of Y_m, given Y_{m-1}, in terms of the conditional characteristic function in (24). Both Y_m and $\bar{Y}_m$ can be decomposed in terms of the increments between consecutive time steps, as follows

$$\begin{aligned}
Y_m &= \log(\sum_{j=1}^{m}(\Pi_{l=1}^{j}\frac{S_l}{S_{l-1}})), \\
\bar{Y}_m &= \log(\sum_{j=1}^{m-1}(\Pi_{l=1}^{j}\frac{S_l}{S_{l-1}}) + 1) + \log(\frac{S_m}{S_{m-1}}).
\end{aligned}$$

These increments are identically and independently distributed, depending only on the model parameters and Δt. Therefore, Y_m and $\bar{Y}_m$ are both functions of Δt. In the following lemma we will show that as $\mathcal{M} \to \infty$, that is, as Δt goes to zero, we have $Y_m \to \bar{Y}_m$, $\forall m = 1, \cdots$, and we can use $\varphi_{\bar{Y}_m|Y_{m-1}}(u; y_{m-1}, \Delta t)$ to approximate $\varphi_{Y_m|Y_{m-1}}(u; y_{m-1}, \Delta t)$ at each time step.

Lemma 3.4. *As $\Delta t \to 0$, that is, as $t_k - t_{k-1} \to 0$, $\forall k = 1, \cdots, \mathcal{M}$, we have that, $\forall m = 1, \cdots, \mathcal{M}$,*

$$\log(\sum_{j=1}^{m}(\Pi_{l=1}^{j}\frac{S_l}{S_{l-1}})) \to \log(\sum_{j=1}^{m-1}(\Pi_{l=1}^{j}\frac{S_l}{S_{l-1}}) + 1) + \log(\frac{S_m}{S_{m-1}}).$$

In other words, as $\mathcal{M} \to \infty$, then, $Y_m \to \bar{Y}_m$, $\forall m = 1, \cdots, \mathcal{M}$.

Proof.

$$\begin{aligned}
\exp(Y_m) - \exp(\bar{Y}_m) &= (\frac{S_1}{S_0} - \frac{S_m}{S_{m-1}}) + (\frac{S_2}{S_1} - \frac{S_m}{S_{m-1}})\frac{S_1}{S_0} \\
&+ (\frac{S_3}{S_2} - \frac{S_m}{S_{m-1}})\frac{S_2}{S_0} + \cdots + (\frac{S_{m-1}}{S_{m-2}} - \frac{S_m}{S_{m-1}})\frac{S_{m-2}}{S_0} \\
&= \sum_{j=1}^{m-1}(\frac{S_j}{S_{j-1}} - \frac{S_m}{S_{m-1}})\frac{S_{j-1}}{S_0}. \qquad (25)
\end{aligned}$$

The mean and variance of $\log(S_j/S_{j-1})$, $j = 1, \cdots, \mathcal{M}$, under a Lévy process, are of the form $J_\mu \Delta t$ and $J_\sigma \Delta t$, where J_μ, J_σ are constants, independent of Δt.

Therefore, as $\mathcal{M} \to \infty$, $\Delta t \to 0$, the mean and variance of $\log(S_j/S_{j-1})$ will tend to zero, so that $\log(S_j/S_{j-1}) \to 0$.

Function $\exp(x), x \in \mathbb{R}$, is a continuous function with respect to x, therefore, we have $S_j/S_{j-1} \to 1$, $\forall j = 1, \cdots, \mathcal{M}$, and thus

$$\frac{S_j}{S_{j-1}} - \frac{S_m}{S_{m-1}} \to 0, \forall j. \tag{26}$$

Moreover, the term $\frac{S_{j-1}}{S_0}$ is independent of the term $\frac{S_j}{S_{j-1}} - \frac{S_m}{S_{m-1}}$, so that, from (25), we find

$$\exp(Y_m) - \exp(\bar{Y}_m) \to 0,$$

or, in other words, $\exp(Y_m) \to \exp(\bar{Y}_m)$ as $\mathcal{M} \to \infty$, and we can conclude that $Y_m \to \bar{Y}_m$, because $\log(x)$ is a continuous function in $x > 0$.

This concludes the proof. $\square$

As an approximation, we can thus use

$$\begin{aligned} \hat{\varphi}_{Y_m|Y_{m-1}}(u; y_{m-1}, \Delta t) &\approx \varphi_{\bar{Y}_m|Y_{m-1}}(u; y_{m-1}, \Delta t) \\ &= (1 + e^{Y_{m-1}})^{iu} \phi_{Z_m}(u; \Delta t). \end{aligned} \tag{27}$$

For Lévy processes all Z_m-terms are identically distributed. Thus

$$\phi_{Z_m}(u; t) =: \phi_Z(u; t),$$

and we replace $\phi_{Z_m}(u; t)$ by $\phi_Z(u; t)$ in (27).

This concludes our discussion of the first pricing method. Numerical experiments, comparing the performance of this method with that of the 2D method presented in the section to follow, for a small and large number of early-exercise dates, will be presented in the section with numerical experiments.

4 The 2D ASCOS method for early-exercise Asian options

In this section we present *a 2D pricing algorithm for early–exercise Asian options*, which can be used for all Lévy processes with any number of early–exercise dates. Calculations of the continuation value and the Fourier coefficients at each time step are discussed, respectively, in Subsections 4.1 and 4.2. The method appears to be more robust than the method from the previous section, but also somewhat more expensive.

4.1 Continuation value

At each time step $m = \mathcal{M}, \cdots, 1$, we use in this case the variables

$$y_m := \frac{S_1}{S_0} + \cdots + \frac{S_m}{S_0}, \quad x_m := \log(\frac{S_m}{S_0}),$$

and we have

$$y_m = y_{m-1} + e^{x_m}. \tag{28}$$

From the risk–neutral evaluation formula, where the continuation value is derived as the discounted expected option price at the next time step, we now use a 2D version, as follows, for $m = \mathcal{M}, \cdots, 1$,

$$\begin{aligned} c(y_{m-1}, x_{m-1}, t_{m-1}) &= e^{-r\Delta t} \mathbb{E}(v(y_m, x_m, t_m)|y_{m-1}, x_{m-1}) \qquad (29) \\ &= e^{-r\Delta t} \int_{\mathbb{R}} \int_{\exp(x_m)}^{+\infty} v(y_m, x_m, t_m) f(y_m, x_m|y_{m-1}, x_{m-1}) dy_m dx_m. \end{aligned}$$

where the integration range of y_m comes from (28) and that $y_{m-1} \geq 0$.

Truncating the integration range, gives us

$$\hat{c}(y_{m-1}, x_{m-1}, t_{m-1}) = e^{-r\Delta t} \int_{a_1}^{b_1} \int_{\exp(x_m)}^{b_2} v(y_m, x_m, t_m) f(y_m, x_m | y_{m-1}, x_{m-1}) dy_m dx_m, \tag{30}$$

where $[a_1, b_1]$ and $[\exp(x_m), b_2]$ are the integration ranges for x_m and y_m, respectively. Integration range $[a_1, b_1]$ is calculated the same way as presented in [8], and the calculation of b_2 will be explained in Subsection 4.4. By applying the *chain rule* to the joint conditional density function in (30), we find

$$\begin{aligned} f(y_m, x_m | y_{m-1}, x_{m-1}) &= f(y_m | x_m, y_{m-1}, x_{m-1}) \cdot f(x_m | y_{m-1}, x_{m-1}) \\ &= f(y_m | x_m, y_{m-1}) \cdot f(x_m | x_{m-1}). \end{aligned} \tag{31}$$

By inserting (31) into (30), the risk–neutral formula becomes

$$\begin{aligned} &\hat{c}(y_{m-1}, x_{m-1}, t_{m-1}) = \\ &e^{-r\Delta t} \int_{a_1}^{b_1} \int_{\exp(x_m)}^{b_2} v(y_m, x_m, t_m) f(y_m | x_m, y_{m-1}) \cdot f(x_m | x_{m-1}) dy_m dx_m \end{aligned} \tag{32}$$

Although the conditional density function is not known analytically for many Lévy processes, the corresponding characteristic function is. Based on this, we approximate the conditional density function by a truncated Fourier cosine expansion based on the characteristic function, as follows,

$$\begin{aligned} \hat{f}(x_m | x_{m-1}) = {} & \frac{2}{b_1 - a_1} {\sum_{k=0}^{N_1-1}}' Re \left(\phi_{x_m - x_{m-1}}(\frac{k\pi}{b_1 - a_1}; \Delta t) \cdot \right. \\ & \left. \exp(ik\pi \frac{x_{m-1} - a_1}{b_1 - a_1}) \right) \cos(k\pi \frac{x_m - a_1}{b_1 - a_1}), \end{aligned} \tag{33}$$

and

$$\begin{aligned} \hat{f}(y_m | x_m, y_{m-1}) = {} & {\sum_{j=0}^{N_2-1}}' \frac{2}{b_2 - \exp(x_m)} Re \left(\exp(i \frac{j\pi}{b_2 - \exp(x_m)} y_{m-1}) \right) \\ & \cos(j\pi \frac{y_m - \exp(x_m)}{b_2 - \exp(x_m)}). \end{aligned} \tag{34}$$

where (34) is based on Equation (28). Note that for Lévy processes, defined by independent and identical increments, the (unconditional) characteristic functions of all increments of consecutive time steps, i.e. $\phi_{x_m - x_{m-1}}(u; \Delta t)$, are the same, for all time steps, and are known analytically [1]. Therefore, we use the notation $\phi(u; \Delta t) := \phi_{x_m - x_{m-1}}(u; \Delta t)$ for all time steps.

By replacing the two density functions in (32) by their approximations in (33)

[1]Compared to the previous section and the 1D pricing method, we have reduced the number of arguments for $\phi(\cdot)$ from three to two. So, for the conditional characteristic function we have used $\phi(u; x, t)$, whereas for the unconditional characteristic function, or if we deal with independent increments, as in the present section, we use $\phi(u; t)$.

and (34), and then interchanging the order of summation and integration, we obtain,

$$\hat{c}(y_{m-1},x_{m-1},t_{m-1}) = e^{-r\Delta t}\frac{2}{b_1-a_1}\sum_{k=0}^{N_1-1}{}'\sum_{j=0}^{N_2-1}{}'\int_{a_1}^{b_1}\int_{\exp(\delta_n)}^{b_2}\hat{v}(y_m,x_m,t_m)\cdot$$
$$Re\left(\phi(\frac{k\pi}{b_1-a_1};\Delta t)\exp(ik\pi\frac{x_{m-1}-a_1}{b_1-a_1})\right)\cos(k\pi\frac{x_m-a_1}{b_1-a_1})\cdot$$
$$\frac{2}{b_2-\exp(x_m)}Re\left(\exp(i\frac{j\pi}{b_2-\exp(\delta_n)}y_{m-1})\right)\cos(j\pi\frac{y_m-\exp(x_m)}{b_2-\exp(x_m)})dy_m dx_m$$
$$= e^{-r\Delta t}\frac{2}{b_1-a_1}\sum_{k=0}^{N_1-1}{}'\sum_{j=0}^{N_2-1}{}' Re\left(\phi(\frac{k\pi}{b_1-a_1};\Delta t)\exp(ik\pi\frac{x_{m-1}-a_1}{b_1-a_1})\right)\cdot$$
$$Re\left(\int_{a_1}^{b_1}\frac{2}{b_2-\exp(x_m)}\exp(i\frac{j\pi}{b_2-\exp(x_m)}y_{m-1})\cos(k\pi\frac{x_m-a_1}{b_1-a_1})\cdot\right.$$
$$\left.\int_{\exp(\delta_n)}^{b_2}\hat{v}(y_m,x_m,t_m)\cos(j\pi\frac{y_m-\exp(x_m)}{b_2-\exp(x_m)})dy_m dx_m\right). \tag{35}$$

For the integration over x_m in (35) numerical approximation is required, for which Clenshaw–Curtis quadrature is employed here. Function

$$\frac{2}{b_2-\exp(x_m)}\exp(i\frac{j\pi}{b_2-\exp(x_m)}y_{m-1})\cos(k\pi\frac{x_m-a_1}{b_1-a_1})\cos(j\pi\frac{y_m-\exp(x_m)}{b_2-\exp(x_m)})$$

is smoothly varying[2] in x_m and the same is true for the option value, $\hat{v}(y_m,x_m,t_m)$, for all $m<\mathcal{M}$. At $t_\mathcal{M}=T$, $v(y_\mathcal{M},x_\mathcal{M},t_\mathcal{M})$ is only a function of $y_\mathcal{M}$, i.e. $v(y_\mathcal{M},x_\mathcal{M},t_\mathcal{M})\equiv g(y_\mathcal{M},t_\mathcal{M})$. Because of these properties, we expect an exponential convergence for this quadrature.

Note that both the Clenshaw–Curtis and Gaussian quadrature rules exhibit exponential convergence for the integral under consideration, however, the Clenshaw–Curtis quadrature appears to be computationally somewhat cheaper. The weights and nodes of the Clenshaw-Curtis quadrature are easy to calculate and they form a nested sequence. We refer the reader to [5] and [3] for more information about Clenshaw–Curtis quadrature.

In detail, for the approximation by Clenshaw–Curtis quadrature, we have

$$\int_{a_1}^{b_1}\frac{2}{b_2-\exp(x_m)}\exp(i\frac{j\pi}{b_2-\exp(x_m)}y_{m-1})\cos(k\pi\frac{x_m-a_1}{b_1-a_1})$$
$$\int_{\exp(\delta_n)}^{b_2}\hat{v}(y_m,x_m,t_m)\cos(j\pi\frac{y_m-\exp(x_m)}{b_2-\exp(x_m)})dy_m dx_m$$
$$\approx \frac{b_1-a_1}{2}\sum_{n=1}^{n_q+2}w_n\frac{2}{b_2-\exp(\delta_n)}\exp(i\frac{j\pi}{b_2-\exp(\delta_n)}y_{m-1})\cos(k\pi\frac{\delta_n-a_1}{b_1-a_1})$$
$$\cdot\int_{\exp(\delta_n)}^{b_2}\hat{v}(y_m,\delta_n,t_m)\cos(j\pi\frac{y_m-\exp(\delta_n)}{b_2-\exp(\delta_n)})dy_m,$$

where

$$\delta_n = \begin{cases} \frac{b_1-a_1}{2}\cos(\frac{n\pi}{n_q})+\frac{b_1+a_1}{2}, & n=\{0,\cdots,n_q/2\}, \\ \frac{a_1-b_1}{2}\cos\left(\frac{(n-(n_q/2+1))\pi}{n_q}\right)+\frac{b_1+a_1}{2}, & n=\{n_q/2+1,\cdots,n_q+1\} \end{cases} \tag{36}$$

[2]That is, the function is countinuous in x_m and so its derivatives with resepct to x_m.

and w is an (n_q+2)-vector, defined as $w := \{w_n\}_{n=1}^{n_q+2} = [D^T d; D^T d]$, with D an $(\frac{n_q}{2}+1)\times(\frac{n_q}{2}+1)$-matrix, with elements

$$D(n_1,n_2) = \frac{2}{n_q}\cos\left(\frac{(n_1-1)(n_2-1)\pi}{n_q/2}\right)\cdot\begin{cases} 1/2, & n_2 = \{1, n_q/2+1\}, \\ 1, & \text{otherwise}, \end{cases}$$

and vector d reads

$$d = (1, \frac{2}{1-4}, \frac{2}{1-16}, \cdots, \frac{2}{1-(n_q-2)^2}, \frac{1}{1-n_q^2})^T.$$

Note that, $\forall k, j, m$, the values of δ_n, w_n are the same, in other words, they only need to be calculated once and can be used for all k, j and for all time steps.

Inserting (36) in (35) gives us the formula for the continuation value at each time step, as follows

$$\hat{c}(y_{m-1}, x_{m-1}, t_{m-1}) = e^{-r\Delta t}\sum_{k=0}^{N_1-1}{}' \sum_{j=0}^{N_2-1}{}' Re\left(\phi(\frac{k\pi}{b_1-a_1};\Delta t)\exp(ik\pi\frac{x_{m-1}-a_1}{b_1-a_1})\right)$$

$$\cdot \quad Re\left(\sum_{n=1}^{n_q+2}\frac{2}{b_2-\exp(\delta_n)}w_n\exp(i\frac{j\pi}{b_2-\exp(\delta_n)}y_{m-1})\cos(k\pi\frac{\delta_n-a_1}{b_1-a_1})\right.$$

$$\cdot \quad \left.\int_{\exp(\delta_n)}^{b_2}\hat{v}(y_m,\delta_n,t_m)\cos(j\pi\frac{y_m-\exp(\delta_n)}{b_2-\exp(\delta_n)})dy_m\right).$$

By denoting

$$\hat{V}_{n,j}(t_m) := \int_{\delta_n}^{b_2}\hat{v}(y_m,\delta_n,t_m)\cos(j\pi\frac{y_m-\exp(\delta_n)}{b_2-\exp(\delta_n)})dy_m, \tag{37}$$

with δ_n as in (36), we have

$$\hat{c}(y_{m-1}, x_{m-1}, t_{m-1}) = e^{-r\Delta t}\sum_{k=0}^{N_1-1}{}' \sum_{j=0}^{N_2-1}{}' Re\left(\phi(\frac{k\pi}{b_1-a_1};\Delta t)\exp(ik\pi\frac{x_{m-1}-a_1}{b_1-a_1})\right)\cdot$$
$$Re\left(\sum_{n=1}^{n_q+2}\frac{2}{b_2-\exp(\delta_n)}w_n\exp(i\frac{j\pi}{b_2-\exp(\delta_n)}y_{m-1})\cos(k\pi\frac{\delta_n-a_1}{b_1-a_1})\hat{V}_{n,j}(t_m)\right). \tag{38}$$

From (37) we see that the computational complexity to compute the continuation value at each time step is $O(N_1N_2n_q)$.

The 2D pricing algorithm is based on backward recursion of the Fourier coefficients $\hat{V}_{n,j}(t_m)$, defined in (37). The early–exercise Asian option price, $\hat{v}(x_0,t_0) = \hat{c}(y_0,x_0,t_0)$ is obtained by taking $m=1$ and inserting $\hat{V}_{n,j}(t_1)$ in (38). In the next subsection we will show that the $V_{n,j}(t_{\mathcal{M}})$ are known analytically. For $m = \mathcal{M}-1,\cdots,1$, $\hat{V}_{n,j}(t_m)$ can be recovered from $\hat{V}_{n,j}(t_{m+1})$.

4.2 Fourier coefficients

At maturity time, $t_{\mathcal{M}} = T$, the option value equals the payoff, so that $\forall n, j$,

$$V_{n,j}(t_{\mathcal{M}}) := \int_{\exp(\delta_n)}^{b_2} g(y_{\mathcal{M}},t_{\mathcal{M}})\cos(j\pi\frac{y_{\mathcal{M}}-\exp(\delta_n)}{b_2-\exp(\delta_n)})dy_{\mathcal{M}},$$

where, $\forall m = 1, \cdots, \mathcal{M}$,

$$g(y_m, t_m) = \begin{cases} (\frac{S_0(1+y_m)}{m+1} - K)^+, & \text{for a call,} \\ (K - \frac{S_0(1+y_m)}{m+1})^+, & \text{for a put.} \end{cases} \tag{39}$$

Thus, the Fourier coefficients at maturity read

$$V_{n,j}(t_{\mathcal{M}}) = \begin{cases} \frac{S_0}{\mathcal{M}+1}\varsigma_j(y^*_{\mathcal{M},n}, b_2) + (\frac{S_0}{\mathcal{M}+1} - K)\psi_j(y^*_{\mathcal{M},n}, b_2), & \text{for a call,} \\ (K - \frac{S_0}{\mathcal{M}+1})\psi_j(\exp(\delta_n), y^*_{\mathcal{M},n}) - \frac{S_0}{\mathcal{M}+1}\varsigma_j(\exp(\delta_n), y^*_{\mathcal{M},n}), & \text{for a put,} \end{cases} \tag{40}$$

where $y^*_{\mathcal{M},n} \equiv \frac{K(\mathcal{M}+1)}{S_0} - 1$, $\psi_j(y_l, y_u)$ is as in (9), and

$$\varsigma_j(y_l, y_u) = \int_{y_l}^{y_u} y \cos\left(j\pi \frac{y - \exp(\delta_n)}{b_2 - \exp(\delta_n)}\right) dy. \tag{41}$$

Both $\psi_j(y_l, y_u)$ and $\varsigma_j(y_l, y_u)$ are known analytically, and so is $V_{n,j}(t_{\mathcal{M}})$. The recursive step is presented in the following result.

Result 4.1. *For $t_m, m = \mathcal{M} - 1. \cdots, 1$, the continuation value, $c(y_m, x_m, t_m)$, and the Fourier cosine coefficients, $V_{n,j}(t_m)$, can be obtained from $V_{n,j}(t_{m+1})$. By this approach, the Fourier coefficients $V_{n,j}(t_1)$ are recovered recursively.*

Proof. For $m = \mathcal{M} - 1. \cdots, 1$, first of all, the early–exercise points, $y^*_{m,n}$, for which $c(y^*_{m,n}, \delta_n, t_m) = g(y^*_{m,n}, t_m)$, δ_n as in (36), need to be determined by means of Newton's method. Here, the payoff function is calculated in (39), and the continuation value is derived via

$$\begin{aligned} &\hat{c}(y_m, \delta_n, t_m) = \\ &e^{-r\Delta t} \sum_{k=0}^{N_1-1}{}' \sum_{j=0}^{N_2-1}{}' Re\left(\phi(\frac{k\pi}{b_1 - a_1}; \Delta t) \exp(ik\pi \frac{\delta_n - a_1}{b_1 - a_1})\right) \cdot \\ &Re\left(\sum_{p=1}^{n_q+2} \frac{2}{b_2 - \exp(\delta_p)} w_p \exp(i\frac{j\pi}{b_2 - \exp(\delta_p)} y_m) \cos(k\pi \frac{\delta_p - a_1}{b_1 - a_1}) \hat{V}_{p,j}(t_{m+1})\right). \end{aligned} \tag{42}$$

which is directly obtained from (38). Furthermore, the derivative of the continuation value and that of the payoff function with respect to y_m can be easily computed by (42) and (39). From [8] we know that typically after approximately five Newton iterations, the error in $y^*_{m,n}$ is $O(10^{-10})$.

As a next step, $\hat{V}_{n,j}(t_m)$ is split by means of these early–exercise points, as follows

$$\hat{V}_{n,j}(t_m) = \begin{cases} \hat{C}_{n,j}(\exp(\delta_n), y^*_{m,n}, t_m) + G_{n,j}(y^*_{m,n}, b_2, t_m), & \text{for a call,} \\ G_{n,j}(\exp(\delta_n), y^*_{m,n}, t_m) + \hat{C}_{n,j}(y^*_{m,n}, b_2, t_m), & \text{for a put,} \end{cases} \tag{43}$$

where $\hat{C}_{n,j}, G_{n,j}$ are Fourier cosine coefficients of the continuation value and payoff at t_m, respectively. Coefficient $G_{n,j}$ is of the form,

$$G_{n,j}(t_m) = \begin{cases} \frac{S_0}{m+1}\varsigma_j(y^*_{m,n}, b_2) + (\frac{S_0}{m+1} - K)\psi_j(y^*_{m,n}, b_2), & \text{for a call,} \\ (K - \frac{S_0}{m+1})\psi_j(\exp(\delta_n), y^*_{m,n}) - \frac{S_0}{m+1}\varsigma_j(\exp(\delta_n), y^*_{m,n})), & \text{for a put,} \end{cases} \tag{44}$$

and coefficient $\hat{C}_k$, defined by

$$\hat{C}_{n,j}(y_l, y_u, t_m) = \int_{y_l}^{y_u} \hat{c}(y_m, \delta_n, t_m) \cos(j\pi \frac{y_m - \exp(\delta_n)}{b_2 - \exp(\delta_n)}) dy_m, \tag{45}$$

with integration range $[y_l, y_u] \in [\delta_n, b_2]$, is computed numerically.

By substituting for $\hat{c}(y_m, \delta_n, t_m)$ in (45) its expression in (42) and interchanging integration and summation, we obtain

$$\begin{aligned} &\hat{C}_{n,j}(y_l, y_u, t_m) = \\ &e^{-r\Delta t} \sum_{k=0}^{N_1-1}{}' \sum_{l=0}^{N_2-1}{}' Re\left(\phi(\frac{k\pi}{b_1 - a_1}; \Delta t) \exp(ik\pi \frac{\delta_n - a_1}{b_1 - a_1})\right) \\ \cdot\ & Re\left(\sum_{p=1}^{n_q+2} \Lambda(k, l, p) \int_{y_l}^{y_u} \exp(i \frac{l\pi}{b_2 - \exp(\delta_p)} y_m) \cos(j\pi \frac{y_m - \exp(\delta_p)}{b_2 - \exp(\delta_p)}) dy_m\right), \end{aligned} \tag{46}$$

where

$$\Lambda(k, l, p) := \frac{2}{b_2 - \exp(\delta_p)} w_p \cos(k\pi \frac{\delta_p - a_1}{b_1 - a_1}) \hat{V}_{p,l}(t_{m+1}). \tag{47}$$

The integral in (46) is known analytically. We have, $\forall y_l, y_u, l, j,\ j, l = 0, \cdots, N_2 - 1,\ j \neq l$,

$$\begin{aligned} & \int_{y_l}^{y_u} \exp(i \frac{l\pi}{b_2 - \exp(\delta_p)} y_m) \cos(j\pi \frac{y_m - \exp(\delta_p)}{b_2 - \exp(\delta_p)}) dy_m \\ = \ & \frac{1}{j^2 - l^2} \frac{d - c}{\pi} (\exp(\frac{il\pi}{b_2 - \exp(\delta_p)} y_u) \sin(j\pi \frac{y_u - \exp(\delta_p)}{b_2 - \exp(\delta_p)}) \\ - \ & \exp(\frac{il\pi}{b_2 - \exp(\delta_p)} y_l) \sin(j\pi \frac{y_l - \exp(\delta_p)}{b_2 - \exp(\delta_p)}) \\ + \ & il(\exp(\frac{il\pi}{b_2 - \exp(\delta_p)} y_u) \cos(j\pi \frac{y_u - \exp(\delta_p)}{b_2 - \exp(\delta_p)}) - \exp(\frac{il\pi}{b_2 - \exp(\delta_p)} y_l) \\ \cdot \ & \cos(j\pi \frac{y_l - \exp(\delta_p)}{b_2 - \exp(\delta_p)}))), \end{aligned}$$

and, if $j = l, j \neq 0, l \neq 0$,

$$\begin{aligned} & \int_{y_l}^{y_u} \exp(i \frac{l\pi}{b_2 - \exp(\delta_p)} y_m) \cos(j\pi \frac{y_m - \exp(\delta_p)}{b_2 - \exp(\delta_p)}) dy_m \\ = & \frac{\exp(il\pi \frac{\exp(\delta_p)}{b_2 - \exp(\delta_p)})}{2} (y_u - y_l) + (-\frac{i}{\pi}) \frac{b_2 - \exp(\delta_p)}{2} \exp(il\pi \frac{\exp(\delta_p)}{b_2 - \exp(\delta_p)}) \cdot \\ & \frac{\exp(i(j+l) \frac{y_u - \exp(\delta_p)}{b_2 - \exp(\delta_p)} \pi) - \exp(i(j+l) \frac{y_l - \exp(\delta_n)}{b_2 - \exp(\delta_p)} \pi)}{j + l}, \end{aligned}$$

and, finally, for $l = j = 0$,

$$\int_{y_l}^{y_u} \exp(i \frac{l\pi}{b_2 - \exp(\delta_p)} y_m) \cos(j\pi \frac{y_m - \exp(\delta_p)}{b_2 - \exp(\delta_p)}) dy_m = y_u - y_l.$$

Therefore, Fourier coefficients $\hat{C}_{n,j}(y_l, y_u, t_m)$ can be calculated directly from (46) without additional numerical techniques.

From (42) and (46) we can observe that the continuation value as well as the Fourier coefficients at $t_m, m = \mathcal{M} - 1, \cdots, 1$, can be recovered from the Fourier coefficients at t_{m+1}. This concludes the proof and the $V_{n,j}(t_1)$, $\forall n, j$ are recovered at the end of backward recursion. $\square$

The value of the Asian option with early-exercise features is then obtained by inserting $V_{n,j}(t_1)$ into (38).

4.3 Computational complexity and Fast Fourier Transform

Newton's method is applied to determine the $y^*_{m,n}$-values with $n = 1, \cdots, n_q + 2$. For this purpose the continuation value $\hat{c}(y_m, \delta_n, t_m)$ must be computed by (42). Term

$$Re\left(\sum_{p=1}^{n_q+2} \frac{2}{b_2 - \exp(\delta_p)} w_p \exp\left(i\frac{j\pi}{b_2 - \exp(\delta_p)} y_m\right) \cos\left(k\pi \frac{\delta_p - a_1}{b_1 - a_1}\right) \hat{V}_{p,j}(t_{m+1})\right).$$

in (42) is calculated once and can be reused in all iteration steps and for all δ_n. Therefore, we perform $O(N_1 N_2 n_q)$ computations to determine $y^*_{m,1}$, and to compute $y^*_{m,n}$, $n = 2, \cdots, n_q + 2$, only $O(N_1 N_2)$ computations are needed. We end up with $O(N_1 N_2 n_q)$ compuations to determine all the early–exercise points.

Furthermore, to compute $\hat{C}_{n,j}(y_l, y_u, t_m)$ each time step we perform $O(N_1 N_2 n_q)$ computations, as the integration in (46) has an analytically known solution. We need to calculate $\hat{C}_{n,j}(y_l, y_u, t_m)$ for each value of n and j. Term

$$Re\left(\sum_{p=1}^{n_q+2} \Lambda(k,l,p) \int_{y_l}^{y_u} \exp\left(i\frac{l\pi}{b_2 - \exp(\delta_p)} y_m\right) \cos\left(j\pi \frac{y_m - \exp(\delta_p)}{b_2 - \exp(\delta_p)}\right) dy_m\right) \quad (48)$$

in (46) need not be re–computed for different n, and we have $O(N_1 N_2 n_q)$ computations in total for all values of n, with $n = 1, \cdots, n_q + 2$. To determine all Fourier coefficients, $\hat{C}_{n,j}(y_l, y_u, t_m)$, with $j = 0, \cdots, N_2 - 1$, we require in total $O(N_1 N_2^2 n_q)$ computations, at each time step.

We need to repeat all the computations for time steps, $m = \mathcal{M} - 1, \cdots, 1$, so that the overall computational complexity for the pricing technique is $O((\mathcal{M} - 1) N_1 N_2^2 n_q)$.

The Fast Fourier Transform (FFT) *can however be employed* to reduce this computational complexity. Equation (46) can be rewritten, $\forall k, p$, as

$$\begin{aligned} \hat{C}^{k,p}_{n,j} &= e^{-r\Delta t} {\sum_{l=0}^{N_2-1}}' Re\left(\phi\left(\frac{k\pi}{b_1 - a_1}; \Delta t\right) \exp\left(ik\pi \frac{\delta_n - a_1}{b_1 - a_1}\right)\right) \\ &\cdot\ Re\left(\Lambda(k,l,p) \int_{y_l}^{y_u} \exp\left(i\frac{l\pi}{b_2 - \exp(\delta_p)} y_m\right) \cos\left(j\pi \frac{y_m - \exp(\delta_p)}{b_2 - \exp(\delta_p)}\right) dy_m\right), \end{aligned} \quad (49)$$

If we denote vector $\hat{\mathbf{C}}^{k,p}_n := \{\hat{C}^{k,p}_{n,j}\}_{j=0}^{N_2-1}$, then it is well-known, that,

$$\begin{aligned} \hat{\mathbf{C}}^{k,p}_n &= \frac{e^{-r\Delta t}}{\pi} Re\left(\phi\left(\frac{k\pi}{b_1 - a_1}; \Delta t\right) \exp\left(ik\pi \frac{\exp(\delta_n) - a_1}{b_1 - a_1}\right)\right) \\ &\quad Im((H^c(y_l, y_u) + H^s(y_l, y_u))\mathbf{u}), \end{aligned} \quad (50)$$

where $Im(\cdot)$ denotes taking the imaginary part of the input argument, and

$$\mathbf{u} = \Lambda(k,l,p) \exp\left(\frac{il\pi \exp(\delta_p)}{b_2 - \exp(\delta_p)}\right).$$

Moreover, H^c and H^s have a Hankel and Toeplitz structure, respectively, with elements as follows,

$$H^c_{j,l}(x_1,x_2) = \begin{cases} \dfrac{(x_2-x_1)\pi i}{b_2-\exp(\delta_p)}, & \text{if } j=l=0, \\ \frac{1}{(l+j)} \quad \left[\exp\left(\dfrac{((l+j)x_2-(l+j)\exp(\delta_p))\pi i}{b_2-\exp(\delta_p)}\right)- \right. & \\ \left. \exp\left(\dfrac{((l+j)x_1-(l+j)\exp(\delta_p))\pi i}{b_2-\exp(\delta_p)}\right)\right], & \text{otherwise,} \end{cases} \tag{51}$$

and

$$H^s_{j,l}(x_1,x_2) = \begin{cases} \dfrac{(x_2-x_1)\pi i}{b_2-\exp(\delta_p)}, & \text{if } j=l=0, \\ \frac{1}{(l-j)} \quad \left[\exp\left(\dfrac{((l-j)x_2-(l-j)\exp(\delta_p))\pi i}{b_2-\exp(\delta_p)}\right)- \right. & \\ \left. \exp\left(\dfrac{((l-j)x_1-(l-j)\exp(\delta_p))\pi i}{b_2-\exp(\delta_p)}\right)\right], & \text{otherwise.} \end{cases} \tag{52}$$

From [8] we know that the FFT can be used to calculate matrix–vector multiplications in (50).

To compute vector $\hat{\mathbf{C}}^{k,p}_n$ for each pair of (k,p), with $k=0,\cdots,N_1-1$, $p=1,\cdots,n_q+2$, $O(N_2\log_2 N_2)$ computations are performed. Therefore in total we need $O(N_1N_2\log_2 N_2 n_q)$ computations to compute $\hat{\mathbf{C}}^{k,p}_n$ for all k,p. Furthermore, term $Im((H^c+H^s)\mathbf{u})$ can be reused for all n, with $n=1,n_q+2$, and in total, $O(N_2\log_2 N_2)$ computations are needed for all Fourier coefficients.

At the final stage of the algorithm, we need to add up all $k\times p$ elements, that is

$$\hat{C}_{n,j}(y_l,y_u,t_m) = {\sum_{k=0}^{N_1-1}}' \sum_{p=1}^{n_q+2} \hat{C}^{k,q}_{n,j}(y_l,y_u,t_m), \tag{53}$$

with $\hat{C}^{k,q}_{n,j}(y_l,y_u,t_m)$ defined in (49) and computed by (50).

Define from (50) that

$$\begin{aligned} A_1(k,n) &:= \frac{e^{-r\Delta t}}{\pi} Re(\phi(\frac{k\pi}{b_1-a_1};\Delta t)\exp(ik\pi\frac{\exp(\delta_n)-a_1}{b_1-a_1})) \\ A_2(k,p) &:= Im((H^c(y_l,y_u)+H^s(y_l,y_u))\mathbf{u}) \end{aligned}$$

and (53) can be computed in an efficient way as summarized below.

ALGORITHM: Efficient computation of (53)

For $j=0,\cdots,N_2-1$, compute

- Step 1: Compute
$$A_2(k) := \sum_{p=1}^{n_q+2} A_2(k,p)$$
.

- Step 2: For $n=1,\cdots,n_q+2$, compute
$$\hat{C}_{n,j} := {\sum_{k=0}^{N_1-1}}' A_1(k,n) * A_2(k)$$
.

For each j, with $j = 0, \cdots, N_2 - 1$, we have $O(n_q)$ computations for step 1, and $O(N_1 n_q)$ computations for step 2. Therefore in total, $O(N_1 N_2 n_q)$ computations are needed for summation (53). By the use of the FFT, the computational complexity at each time step is then reduced to $O(N_1 N_2 \log_2 N_2 n_q)$.

The overall 2D ASCOS pricing algorithm is summarized below.

ASCOS ALGORITHM: Pricing early–exercise arithmetic Asian options.

Initialization

- For $n = 1, \cdots, n_q + 2$, $j = 0, \cdots, N_2 - 1$, compute $V_{n,j}(t_{\mathcal{M}})$ from (40).

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

- Determine the early–exercise points, $y^*_{m,n}$, for $n = 1, \cdots, n_q + 2$, with $\hat{c}(y^*_{m,n}, \delta_n, t_m) = g(y^*_{m,n}, t_m)$, by Newton's method. Continuation value and payoff function are given by (42) and (39) , respectively.
- Compute the Fourier coefficients $\hat{V}_{n,j}(t_m)$.
  - For $k = 0, \cdots, N_1 - 1$, compute each column of matrix $\hat{\mathbf{C}}^k_n := \{\hat{C}^k_{n,j}\}_{j=0}^{N_2}$ by (50) with the help of Fast Fourier Transform.
  - Compute $\hat{C}_{n,j}(t_m)$, $\forall n, j$, from (53).
  - Compute $G_{n,j}(t_m)$, $\forall n, j$, from (44).
  - Calculate the Fourier coefficients $\hat{V}_{n,j}(t_m)$ by inserting $\hat{C}_{n,j}(t_m)$ and $G_{n,j}(t_m)$ into (43).

Final step:

- Compute the early–exercise Asian option value, $\hat{v}(x_0, t_0)$, by inserting $\hat{V}_{n,j}(t_1)$ in (38).

4.4 Integration range of Y_m

Here we explain how to determine the upper bound b_2, so that the truncation error in Y_m, with integration range $[\exp(x_m), b_2]$ can be controlled. First of all, we derive the integration range for $\log(Y_m)$ and after that the range for Y_m. From [7, 8], we know that a suitable integration range for $\log(Y_m)$ can be determined by means of the cumulants, as follows

$$\begin{aligned}[][\ell, v] \quad \approx \quad & \Bigg[(\xi_1(\log(Y_m)) - L\sqrt{\xi_2(\log(Y_m)) + \sqrt{\xi_4(\log(Y_m))}}\,, \\ & \xi_1(\log(Y_m)) + L\sqrt{\xi_2(\log(Y_m)) + \sqrt{\xi_4(\log(Y_m))})}\Bigg], \end{aligned} \tag{54}$$

and the integration range of Y_m at t_m can then be set to $[e^{x_m}, e^v]$. By $\xi_n(X)$, we denote the n^{th} cumulant of X, computed via

$$\xi_n(X) := \frac{1}{i^n}\frac{\partial^n(t\Phi(\mathbf{u}))}{\partial \mathbf{u}^n}|\mathbf{u} = 0,$$

where $t\Phi(\mathbf{u})$ is the exponent of the characteristic function, $\phi(\mathbf{u}; t)$, i.e.

$$\phi(\mathbf{u}; t) = e^{t\Phi(\mathbf{u})}.$$

For arithmetic Asian options, it is however expensive to calculate these cumulants, and therefore we propose *another integration range* for the arithmetic case, which is very similar to that in (54). For a Lévy process, the cumulants of any increment, $\log(S_l/S_k)$, $\forall l > k$, are linearly increasing functions of $(l-k)\Delta t$, so that, for all $Y_m,\ m = 1, \cdots, \mathcal{M}$, we have

$$\xi_1(\log(m\frac{S_1}{S_0})) \leq \xi_1(\log(Y_m)) \leq \xi_1(\log(m\frac{S_m}{S_0})),$$
$$0 \leq \xi_2(\log(Y_m)) \leq \xi_2(\log(m\frac{S_m}{S_0})), \quad 0 \leq \xi_4(\log(Y_m)) \leq \xi_4(\log(m\frac{S_m}{S_0})).$$

and we will use the integration boundaries

$$\begin{aligned}\ell &:= \xi_1(\log(m\frac{S_1}{S_0})) - L\sqrt{\xi_2(\log(m\frac{S_m}{S_0})) + \sqrt{\xi_4(\log(m\frac{S_m}{S_0}))}},\\ \upsilon &:= \xi_1(\log(m\frac{S_m}{S_0})) + L\sqrt{\xi_2(\log(m\frac{S_m}{S_0})) + \sqrt{\xi_4(\log(m\frac{S_m}{S_0}))}}.\end{aligned} \tag{55}$$

Interval $[\ell, \upsilon]$ from (55) will be the integration range for $\log(Y_m)$.

Note that the cumulants of $\log(m\frac{S_1}{S_0})$ and $\log(m\frac{S_m}{S_0})$ in (55) are known in closed form for Lévy processes, as for $n = 1$, we have $\xi_1(\log(m\frac{S_1}{S_0})) = log(m) + \xi_1(R)$, $\xi_1(\log(m\frac{S_m}{S_0})) = log(m) + m\xi_1(R)$, and for $n \geq 2$, $\xi_n(\log(m\frac{S_1}{S_0})) = \xi_n(R)$, $\xi_n(\log(m\frac{S_m}{S_0})) = m\xi_n(R)$. Here, parameter R denotes the logarithm of the increment between any two consecutive time steps of a Lévy process.

From [7, 8] we know that with $L \approx 10$, the integration range ensures highly accurate option prices for most Lévy processes. With a wider integration range $[\ell, \upsilon]$, the error will be smaller, but an increasing number of Fourier cosine terms may need to be used (which makes it more costly). Adaptation of parameter L for very short term, or very long term options is easily possible.

Integration range of Y_m at t_m is then taken as $[e^{x_m}, e^{\upsilon}]$.

5 Error analysis

Here, we give a detailed error analysis of the 2D ASCOS method for early–exercise arithmetic Asian options from Section 4. We identify three different types of errors, for which we first introduce some notation.

The *truncation error*, ϵ_T, for any random variable, Z, with integration range $[a, b]$, is defined as

$$\epsilon_T(Z; [a, b]) := \int_{\mathbf{R}\backslash[a,b]} f_Z(z)dz, \tag{56}$$

and it decreases as the integration range $[a, b]$ increases. In other words, for a sufficiently large integration range, this error won't dominate the total error in the arithmetic Asian option price.

For Y_m we truncate one side of the integration range, and the truncation error reads

$$\epsilon_T(Y_m; b_2) := \int_{b_2}^{+\infty} f_{Y_m}(y)dy, \tag{57}$$

The *error due to the number of terms used in the Fourier cosine expansion* is denoted by ϵ_F. We know, from [7], that for $f_Z(z) \in \mathbf{C}^\infty[a, b]$, this error can be bounded by

$$|\epsilon_F(Z; N)| \leq P^*(N)\exp(-(N-1)\nu_F), \tag{58}$$

with $\nu_F > 0$ a constant and a term $P^*(N)$, which varies less than exponentially with respect to N. Note that, although the upper bound of ϵ_F is not a function of the underlying state variable Z, the state variable still appears as an input argument, because the smoothness of the density function influences the convergence behavior.

When the probability density function has a discontinuous derivative, the error can be bounded by

$$|\epsilon_F(Z;N)| \leq \frac{\bar{P}^*(N)}{(N-1)^{\beta-1}},$$

where $\bar{P}^*(N)$ is a constant and $\beta \geq 1$.

Error ϵ_F thus decays either exponentially with respect to N, if the density function $f(z) \in \mathbf{C}^\infty[a,b]$, or otherwise algebraically.

We denote the error from the Clenshaw–Curtis quadrature (36) by ϵ_q. From [12] we know that for integrands belonging to $C^\infty[a,b]$, which is the case here, error ϵ_q decays exponentially, i.e.,

$$|\epsilon_q(n_q)| \leq P(n_q)\exp(-(n_q-1)\nu_q), \tag{59}$$

with $\nu_q > 0$ a constant and a term $P(n_q)$, which varies less than exponentially with respect to n_q.

Note that the value of the averaged underlying price, y_m, $\forall t_m$, does not influence the smoothness of the density function of the underlying process. In fact, it can be recursively proved that if $f(x_j), j = 1,\cdots,\mathcal{M}$ is smooth then $f(y_j), j = 1,\cdots,\mathcal{M}$, with $y_j = \sum_{i=1}^{j}\exp(x_i)$ is also smooth, so that it does not influence the convergence speed negatively.

We further denote by $\epsilon(\hat{c}(y_m, x_m, t_m))$, $\epsilon(V_{n,j}(t_m))$ and $\epsilon^*_{m,n}$, the errors in the continuation value, in the Fourier coefficients and in the early–exercise points, $y^*_{m,n}$, at time step t_m, respectively.

Our error analysis is based on backward recursion, i.e. first of all we analyze the error in the continuation value, $\hat{c}(y_{\mathcal{M}-1}, x_{\mathcal{M}-1}, t_{\mathcal{M}-1})$, in Subsection 5.1, after which the error propagation throughout the time steps $t_m, m = \mathcal{M}-2,\cdots,1$ is discussed in Subsection 5.2.

5.1 Initial error

In this subsection, the error from (29) to (35) is discussed. At $t_{\mathcal{M}-1}$, Eqns. (29) and (35) can be rewritten, respectively, as

$$\begin{aligned} &c(y_{\mathcal{M}-1}, x_{\mathcal{M}-1}, t_{\mathcal{M}-1}) = \\ &e^{-r\Delta t}\int_{\mathbb{R}}\int_{\exp(x_{\mathcal{M}})}^{+\infty} v(y_{\mathcal{M}}, x_{\mathcal{M}}, t_{\mathcal{M}}) f(y_{\mathcal{M}}|x_{\mathcal{M}}, y_{\mathcal{M}-1}) f(x_{\mathcal{M}}|x_{\mathcal{M}-1}) dy_{\mathcal{M}} dx_{\mathcal{M}}, \end{aligned} \tag{60}$$

and

$$\begin{aligned} \hat{c}(y_{\mathcal{M}-1}, x_{\mathcal{M}-1}, t_{\mathcal{M}-1}) &= e^{-r\Delta t}\int_{a_1}^{b_1}\int_{\exp(x_{\mathcal{M}})}^{b_2} v(y_{\mathcal{M}}, x_{\mathcal{M}}, t_{\mathcal{M}}) \hat{f}(y_{\mathcal{M}}|x_{\mathcal{M}}, y_{\mathcal{M}-1}) \\ &\cdot \hat{f}(x_{\mathcal{M}}|x_{\mathcal{M}-1}) dy_{\mathcal{M}} dx_{\mathcal{M}}, \end{aligned} \tag{61}$$

where $\hat{f}(x_{\mathcal{M}}|x_{\mathcal{M}-1})$ and $\hat{f}(y_{\mathcal{M}}|x_{\mathcal{M}}, y_{\mathcal{M}-1})$ are defined in (33) and (34), respectively.

Then, the error, which we denote by $\tilde{\epsilon}$, consists of two parts, that is, $\tilde{\epsilon} := \epsilon_I + \epsilon_{II}$, with

$$\begin{aligned} \epsilon_I &:= e^{-r\Delta t}\int_{\mathbb{R}}\int_{\exp(x_{\mathcal{M}})}^{+\infty} v(y_{\mathcal{M}}, x_{\mathcal{M}}, t_{\mathcal{M}}) f(y_{\mathcal{M}}|x_{\mathcal{M}}, y_{\mathcal{M}-1}) dy_{\mathcal{M}} f(x_{\mathcal{M}}|x_{\mathcal{M}-1}) dx_{\mathcal{M}} \\ &- e^{-r\Delta t}\int_{\mathbb{R}}\int_{\exp(x_{\mathcal{M}})}^{b_2} v(y_{\mathcal{M}}, x_{\mathcal{M}}, t_{\mathcal{M}}) \hat{f}(y_{\mathcal{M}}|x_{\mathcal{M}}, y_{\mathcal{M}-1}) dy_{\mathcal{M}} f(x_{\mathcal{M}}|x_{\mathcal{M}-1}) dx_{\mathcal{M}}, \end{aligned} \tag{62}$$

and

$$\begin{aligned}\epsilon_{II} &:= e^{-r\Delta t}\int_{\exp(x_{\mathcal{M}})}^{b_2}\int_{\mathbb{R}} v(y_{\mathcal{M}},x_{\mathcal{M}},t_{\mathcal{M}})\hat{f}(y_{\mathcal{M}}|x_{\mathcal{M}},y_{\mathcal{M}-1})f(x_{\mathcal{M}}|x_{\mathcal{M}-1})dx_{\mathcal{M}}dy_{\mathcal{M}}\\ &- e^{-r\Delta t}\int_{\exp(x_{\mathcal{M}})}^{b_2}\int_{a_1}^{b_1} v(y_{\mathcal{M}},x_{\mathcal{M}},t_{\mathcal{M}})\hat{f}(y_{\mathcal{M}}|x_{\mathcal{M}},y_{\mathcal{M}-1})\hat{f}(x_{\mathcal{M}}|x_{\mathcal{M}-1})dx_{\mathcal{M}}dy_{\mathcal{M}}.\end{aligned} \tag{63}$$

We use the notation ϵ_{cos} to denote the error of one step of the COS method [7],

$$\begin{aligned}\epsilon_{cos}(X_{\mathcal{M}}) &:= \int_{\mathbb{R}} v(y_{\mathcal{M}},x_{\mathcal{M}},t_{\mathcal{M}})f(x_{\mathcal{M}}|x_{\mathcal{M}-1})dx_{\mathcal{M}}\\ &- \int_{a_1}^{b_1} v(y_{\mathcal{M}},x_{\mathcal{M}},t_{\mathcal{M}})\hat{f}(x_{\mathcal{M}}|x_{\mathcal{M}-1})dx_{\mathcal{M}},\end{aligned}$$

and

$$\begin{aligned}\epsilon_{cos}(Y_{\mathcal{M}}) &:= \int_{\exp(x_{\mathcal{M}})}^{+\infty} v(y_{\mathcal{M}},x_{\mathcal{M}},t_{\mathcal{M}})f(y_{\mathcal{M}}|x_{\mathcal{M}},y_{\mathcal{M}-1})dy_{\mathcal{M}}\\ &- \int_{\exp(x_{\mathcal{M}})}^{b_2} v(y_{\mathcal{M}},x_{\mathcal{M}},t_{\mathcal{M}})\hat{f}(y_{\mathcal{M}}|x_{\mathcal{M}},y_{\mathcal{M}-1})dy_{\mathcal{M}}.\end{aligned}$$

The first part of the error (62) then reads

$$\epsilon_I = e^{-r\Delta t}\epsilon_{cos}(Y_{\mathcal{M}})\int_{\mathbb{R}} f(x_{\mathcal{M}}|x_{\mathcal{M}-1})dx_{\mathcal{M}} = e^{-r\Delta t}\epsilon_{cos}(Y_{\mathcal{M}}). \tag{64}$$

To compute the second part of the error, in (63), first of all, from (34) we have that $\forall y_{\mathcal{M}} \in [\exp(x_{\mathcal{M}}), b_2]$,

$$|\hat{f}(y_{\mathcal{M}}|x_{\mathcal{M}},y_{\mathcal{M}-1})| \leq \frac{2}{b_2-\exp(x_{\mathcal{M}})}N_2 \leq \frac{2}{b_2-\exp(a_1)}N_2.$$

Then, ϵ_{II}, in (63), can be written as

$$\begin{aligned}|\epsilon_{II}| &\leq e^{-r\Delta t}\frac{2N_2}{b_2-\exp(a_1)}\left|\int_{\exp(x_{\mathcal{M}})}^{b_2}\left(\int_{\mathbb{R}} v(y_{\mathcal{M}},x_{\mathcal{M}},t_{\mathcal{M}})f(x_{\mathcal{M}}|x_{\mathcal{M}-1})dx_{\mathcal{M}}\right.\right.\\ &- \left.\left.\int_{a_1}^{b_1} v(y_{\mathcal{M}},x_{\mathcal{M}},t_{\mathcal{M}})\hat{f}(x_{\mathcal{M}}|x_{\mathcal{M}-1})dx_{\mathcal{M}}\right)dy_{\mathcal{M}}\right|\\ &\leq e^{-r\Delta t}2N_2|\epsilon_{cos}(X_{\mathcal{M}})|\end{aligned} \tag{65}$$

We will now use the common notation $\epsilon(x) = O(\varsigma)$, $\forall x \in \mathbb{R}$, if $Q > 0$ exists, so that $|\epsilon(x)| \leq Q|\varsigma|$.

We then have $\epsilon_{II} = O(N_2\epsilon_{cos}(X_{\mathcal{M}}))$.

From [7] we know that $\forall Z, a, b, N$, $\epsilon_{cos}(Z) = O(\epsilon_T(Z;[a,b])) + \epsilon_F(Z;N)$, and a similar analysis can be performed for a one–side truncated variable $Y_{\mathcal{M}}$, then, from (64) and (65), we obtain

$$\begin{aligned}\tilde{\epsilon} = \epsilon_I + \epsilon_{II} &= O(N_2(\epsilon_T(X_{\mathcal{M}};[a_1,b_1]) + \epsilon_F(X_{\mathcal{M}};N_1)) +\\ &\quad \epsilon_T(Y_{\mathcal{M}};b_2) + \epsilon_F(Y_{\mathcal{M}};N_2)),\end{aligned} \tag{66}$$

which is the error made up to Eq. (35).

At $t_{\mathcal{M}-1}$, the Fourier coefficients of the option value, $V_{n,j}(t_{\mathcal{M}})$ are known analytically. Therefore, the error from Eq. (35) to Eq. (38) is only due to approximation (36), where the Clenshaw–Curtis quadrature was used. For each j,k the error in the approximated integration is $O(\epsilon_q(n_q))$, with ϵ_q defined in (59). Thus, the error in (38) is found to be $\tilde{\epsilon}+O(N_1N_2\epsilon_q(n_q))$, with $\tilde{\epsilon}$ in (66). Summarizing, the error in the continuation value, $\hat{c}(y_{\mathcal{M}-1},x_{\mathcal{M}-1},t_{\mathcal{M}-1})$, is found to be

$$\begin{aligned}\epsilon(\hat{c}(\quad y_{\mathcal{M}-1} \quad &,x_{\mathcal{M}-1},t_{\mathcal{M}-1}))=O(N_2(\epsilon_T(X_{\mathcal{M}};[a_1,b_1])+\epsilon_F(X_{\mathcal{M}};N_1)) \\ + \quad & \epsilon_T(Y_{\mathcal{M}};b_2)+\epsilon_F(Y_{\mathcal{M}};N_2)+N_1N_2\epsilon_q(n_q)).\end{aligned} \tag{67}$$

With integration ranges $[a_1,b_1]$ and b_2 carefully chosen, truncation errors $\epsilon_T(X_{\mathcal{M}};[a_1,b_1])$ and $\epsilon_T(Y_{\mathcal{M}};b_2)$ will not be the dominant parts of error (67). For a smooth density function of $X_{\mathcal{M}}$ ($f(X_{\mathcal{M}})\in C^\infty$), it can be proved that the density function of $Y_{\mathcal{M}}$ is also smooth, and that the error in the continuation value decays to zero exponentially, with respect to N_1,N_2,n_q. In detail, inserting (58) and (59) into (67) gives us

$$\begin{aligned}|\epsilon(\hat{c}(\quad y_{\mathcal{M}-1} \quad &,x_{\mathcal{M}-1},t_{\mathcal{M}-1}))|\leq P^*(N_1,N_2,n_q)(\exp(-(N_1-1)\nu_1) \\ + \quad & \exp(-(N_2-1)\nu_2)+\exp(-(n_q-1)\nu_q)),\end{aligned} \tag{68}$$

where $P^*(N_1,N_2,n_q)$ is a term which varies less than exponentially with respect to N_1,N_2,n_q.

If the density of $X_{\mathcal{M}}$ is not smooth, then the error converges exponentially to zero with respect to n_q and algebraically with respect to N_1 and N_2.

5.2 Error propagation

Regarding the propagation of the error through time, we state the following lemma:

Lemma 5.1 (Error propagation). *For $m=\mathcal{M}-2,\cdots,0$, assuming that at time step t_{m+1}, $\forall y_{m+1},x_{m+1}$,*

$$\begin{aligned}|\epsilon(\hat{c}(y_{m+1},x_{m+1},t_{m+1}))| \quad \leq \quad & P(N_1,N_2,n_q)(\exp(-(N_1-1)\nu_1) \\ + \quad & \exp(-(N_2-1)\nu_2)+\exp(-(n_q-1)\nu_q)),\end{aligned} \tag{69}$$

where $P(N_1,N_2,n_q)$ is a term which varies less than exponentially with respect to N_1,N_2,n_q, then, at time step t_m, we can show that, $\forall y_m,x_m$,

$$\begin{aligned}|\epsilon(\hat{c}(y_m,x_m,t_m))| \quad \leq \quad & \bar{P}(N_1,N_2,n_q)(\exp(-(N_1-1)\nu_1) \\ + \quad & \exp(-(N_2-1)\nu_2)+\exp(-(n_q-1)\nu_q)),\end{aligned} \tag{70}$$

where $\bar{P}(N_1,N_2,n_q)$ is a term which varies less than exponentially with respect to N_1,N_2,n_q.

Proof. This is a proof based on mathematical induction.

First, we compute the error in the Fourier coefficients, $\hat{V}_{n,j}(t_{m+1})$, after which we analyze the error in $\hat{c}(y_m,x_m,t_m)$.

Error $\epsilon(\hat{V}_{n,j}(t_{m+1}))$ consists of two parts, the error in the Fourier cosine coefficients of the continuation value, and the error due to an incorrect value of the early–exercise point. Without loss of generality, we consider a call option with a positive-valued error in the early–exercise points. The analysis of the error propagation for other cases (negatively-valued error, put option) goes similarly. For a call option, with $\epsilon^*_{m+1,n} > 0$, we have

$$\begin{aligned}
&\epsilon(\hat{V}_{n,j}(t_{m+1})) = (C_{n,j}(\exp(\delta_n), y^*_{m+1,n}, t_{m+1}) - \hat{C}_{n,j}(\exp(\delta_n), y^*_{m+1,n}, t_{m+1})) \\
+ \quad & (G_{n,j}(y^*_{m+1,n}, y^*_{m+1,n} + \epsilon^*_{m+1,n}, t_{m+1}) - \hat{C}_{n,j}(y^*_{m+1,n}, y^*_{m+1,n} + \epsilon^*_{m+1,n}, t_{m+1})) \\
= \quad & \int_{\exp(\delta_n)}^{y^*_{m+1,n}} \epsilon(\hat{c}(y_{m+1}, \delta_n, t_{m+1})) \cos\left(j\pi \frac{y_{m+1} - \exp(\delta_n)}{b_2 - \exp(\delta_n)}\right) dy_{m+1} \\
+ \quad & \int_{y^*_{m+1,n}}^{y^*_{m+1,n}+\epsilon^*_{m+1,n}} (g(y_{m+1}, t_{m+1}) - \hat{c}(y_{m+1}, \delta_n, t_{m+1})) \cos\left(j\pi \frac{y_{m+1} - \exp(\delta_n)}{b_2 - \exp(\delta_n)}\right) dy_{m+1},
\end{aligned} \tag{71}$$

with $g(y_{m+1}, t_{m+1})$ defined in (39).

The error in continuation value $\hat{c}(y_m, x_m, t_m)$ is composed of two parts, $\epsilon(\hat{c}(y_m, x_m, t_m)) := e_I + e_{II}$, where error e_I is the part in which $\epsilon(\hat{V}_{n,j}(t_{m+1}))$ has not yet been considered. It is derived similarly as the error at $t_{\mathcal{M}-1}$ (in Subsection 5.1). We find

$$\begin{aligned}
e_I \quad := \quad & O(N_2(\epsilon_T(X_{m+1}; [a_1, b_1]) + \epsilon_F(X_{m+1}; N_1)) \\
+ \quad & \epsilon_T(Y_{m+1}; b_2) + \epsilon_F(Y_{m+1}; N_2) + N_1 N_2 \epsilon_q(n_q)).
\end{aligned} \tag{72}$$

Error e_{II} is the additional error with $\epsilon(\hat{V}_{n,j}(t_{m+1}))$ taken into consideration,

$$\begin{aligned}
e_{II} \quad := \quad & e^{-r\Delta t} \sum_{k=0}^{N_1-1}{}' \sum_{j=0}^{N_2-1}{}' Re\left(\phi(\frac{k\pi}{b_1 - a_1}; \Delta t) \exp(ik\pi \frac{x_m - a_1}{b_1 - a_1})\right) \cdot \\
& Re\left(\sum_{n=1}^{n_q+2} \frac{2}{b_2 - \exp(\delta_n)} w_n \exp(i \frac{j\pi}{b_2 - \exp(\delta_n)} y_m) \cos(k\pi \frac{\delta_n - a_1}{b_1 - a_1})\right. \\
& \left.\epsilon(\hat{V}_{n,j}(t_{m+1}))\right).
\end{aligned}$$

To analyze these errors, we define a European option, v_α, from t_m to t_{m+1}, with payoff function

$$v_\alpha(y_{m+1}, x_{m+1}, t_{m+1}, L_1, L_2) := \begin{cases} 1, & \text{if} \quad y_{m+1} \in [L_1, L_2], \\ 0, & \text{otherwise}, \end{cases}$$

so that the option value at t_m, $\forall L_1, L_2 \in [\exp(x_{m+1}), +\infty]$, can be written as

$$\begin{aligned}
v_\alpha(y_m, x_m, t_m, L_1, L_2) \quad = \quad & e^{-r\Delta t} \int_{\mathbb{R}} \int_{\exp(x_{m+1})}^{+\infty} v(y_{m+1}, x_{m+1}, t_{m+1}, L_1, L_2) \\
\cdot \quad & f(y_{m+1}, x_{m+1}, t_{m+1} | y_m, x_m) dy_{m+1} dx_{m+1} \\
= \quad & e^{-r\Delta t} \int_{\mathbb{R}} \int_{L_1}^{L_2} f(y_{m+1}, x_{m+1}, t_{m+1} | y_m, x_m) dy_{m+1} dx_{m+1} \\
\leq \quad & e^{-r\Delta t} \int_{\mathbb{R}} \int_{\mathbb{R}} f(y_{m+1}, x_{m+1}, t_{m+1} | y_m, x_m) dy_{m+1} dx_{m+1} \\
= \quad & e^{-r\Delta t}
\end{aligned}$$

and its approximation, by using (38), reads

$$\begin{aligned}
&\hat{v}_\alpha(y_m, x_m, t_m, L_1, L_2) \\
= \quad & e^{-r\Delta t} \sum_{k=0}^{N_1-1}{}' \sum_{j=0}^{N_2-1}{}' Re\left(\phi(\frac{k\pi}{b_1-a_1};\Delta t)\exp(ik\pi\frac{x_m-a_1}{b_1-a_1})\right)\cdot \\
& Re\left(\sum_{n=1}^{n_q+2} \frac{2}{b_2-\exp(\delta_n)} w_n \exp(i\frac{j\pi}{b_2-\exp(\delta_n)} y_m)\cos(k\pi\frac{\delta_n-a_1}{b_1-a_1}) \right.\\
& \left. \int_{L_1}^{L_2} \cos(j\pi\frac{y_{m+1}-\exp(\delta_n)}{b_2-\exp(\delta_n)})\right).,
\end{aligned}$$

from which it follows that, $\forall L_1 \le L_2 \le L_3$,

$$\hat{v}_\alpha(y_m, x_m, t_m, L_1, L_2) + \hat{v}_\alpha(y_m, x_m, t_m, L_2, L_3) = \hat{v}_\alpha(y_m, x_m, t_m, L_1, L_3). \tag{73}$$

The value of $\hat{v}_\alpha$ can bounded, as

$$\begin{aligned}
\hat{v}_\alpha(y_m, x_m, t_m, L_1, L_2) \quad & \le \quad v_\alpha(y_m, x_m, t_m, L_1, L_2) + |\epsilon(v_\alpha(y_m, x_m, t_m, L_1, L_2))| \\
& = \quad e^{-r\Delta t} + O(|e_I|),
\end{aligned} \tag{74}$$

where the last step is because the error from approximation (38) at t_m is of the same order as e_I.

Inserting (71) into (73), then using (69) and (73), gives us

$$\begin{aligned}
|\epsilon(\hat{c}(y_m, x_m, t_m))| \quad & \le \quad |e_I| + P(N_1, N_2, n_q)(\exp(-(N_1-1)\nu_1) + \exp(-(N_2-1)\nu_2) \\
& + \quad \exp(-(n_q-1)\nu_q))\hat{v}_\alpha(y_m, x_m, t_m, \exp(\delta_n), y^*_{m+1,n}) \\
& + \quad \max_n |g(\zeta_n, t_{m+1}) - \hat{c}(\zeta_n, \delta_n, t_{m+1})| \\
& \cdot \quad \hat{v}_\alpha(y_m, x_m, t_m, y^*_{m+1,n}, y^*_{m+1,n} + \epsilon^*_{m+1,n}),
\end{aligned} \tag{75}$$

based on Lagrange's mean value theorem, with $\zeta_n \in (y^*_{m+1,n}, y^*_{m+1,n} + \epsilon^*_{m+1,n})$.

For a call option, with $\zeta_n \in (y^*_{m+1,n}, y^*_{m+1,n} + \epsilon^*_{m+1,n})$, we then have $\forall n$,

$$\begin{aligned}
|g(\zeta_n, t_{m+1}) - \hat{c}(\zeta_n, \delta_n, t_{m+1})| \quad & = \quad \hat{c}(\zeta_n, \delta_n, t_{m+1}) - g(\zeta_n, t_{m+1}) \\
& \le \quad \hat{c}(y^*_{m+1,n}, \delta_n, t_{m+1}) - g(y^*_{m+1,n}, t_{m+1}) \\
& = \quad \hat{c}(y^*_{m+1,n}, \delta_n, t_{m+1}) - c(y^*_{m+1,n}, \delta_n, t_{m+1}) \\
& = \quad \epsilon(\hat{c}(y^*_{m+1,n}, \delta_n, t_{m+1})).
\end{aligned} \tag{76}$$

Inserting (76) in (75), and using (69) and (73), gives us

$$\begin{aligned}
|\epsilon(\hat{c}(y_m, x_m, t_m))| \quad & \le \quad |e_I| + P(N_1, N_2, n_q)(\exp(-(N_1-1)\nu_1) + \exp(-(N_2-1)\nu_2) \\
& + \quad \exp(-(n_q-1)\nu_q))\hat{v}_\alpha(y_m, x_m, t_m, \exp(\delta_n), y^*_{m+1,n} + \epsilon^*_{m+1,n}).
\end{aligned} \tag{77}$$

Finally, by using (74) and (72) in (77), and then inserting (58) and (59), we reach the conclusion that if $[a_1, b_1]$, b_2 are carefully chosen, then the truncation errors $\epsilon_T(X_{m+1}; [a_1, b_1])$ and $\epsilon_T(Y_{m+1}; b_2)$ will not be the dominant parts of error (72), and we obtain

$$\begin{aligned}
|\epsilon(\hat{c}(y_m, x_m, t_m))| \quad & \le \quad \bar{P}(N_1, N_2, n_q)(\exp(-(N_1-1)\nu_1) \\
& + \quad \exp(-(N_2-1)\nu_2) + \exp(-(n_q-1)\nu_q)),
\end{aligned}$$

where $\bar{P}(N_1, N_2, n_q)$ is a term which varies less than exponentially with respect to N_1, N_2, n_q. This concludes the proof. □

In the case of put options or negative-valued errors in the early–exercise points, a similar error expression as in (77) can be derived, by a very similar analysis.

6 Numerical results

In this section we perform experiments with two different Lévy processes, the Black-Scholes (BS) and the Normal Inverse Gaussian (NIG) processes. We will present numerical results for the two methods presented. Reference values are derived by our 2D version of the ASCOS method, with $N_1 = N_2 = (n_q/2) + 1 = 4096$. When increasing the values of $\mathcal{M}, N_1, N_2, n_q$ in the numerical experiments, the 2D ASCOS method gives the same American Asian option values for the BS model as the values in [6] (in the accuracy given in the reference, which is 10^{-4}).

The same model parameters, as used in [9] for pricing European-style Asian options, are also used here:

- *BS:* $r = 0.0367, \sigma = 0.178$;
- *NIG:* $r = 0.0367, \sigma = 0.178, \alpha = 6.188, \beta = -3.894, \delta = 0.1622$.

Two types of processors, a CPU (Central Processing Unit), and a GPU (Graphics Processing Unit) with double precision are used and compared to obtain the numerical ASCOS results. On the CPU, an Intel(R) Core(TM)2 Duo CPU E6550 (@ 2.33GHz Cache size 4MB), the algorithm is implemented in MATLAB 7.7.0. On the GPU, a Tesla C2070 GPU with 6GB memory, we coded in Compute Unified Device Architecture (CUDA) [11]. Computing time is recorded in seconds.

In this section, the notation 'first method' and '2D method' refer to the pricing methods proposed in Section 3 and Section 4, respectively.

Remark 6.1 (Data transfer)**.** *Data transfer between the GPU and the CPU is the bottleneck for most GPU implementations. However, in our GPU implementation, we code in such a way that, no matter the size of the problem, only one number needs to be transferred between the CPU and the GPU, which is the option price and we transfer it back to the CPU at the end of the computations. As the size of the problem increases, there will be no extra burden of data transfer.*

6.1 GPU implementation and acceleration

A GPU is an SIMT (Single Instruction, Multiple Threads) machine. In other words, the same command can be executed simultaneously for each data element on each thread on the GPU. Therefore, GPU processing is advantageous for problems that can be expressed in the form of data–parallel computations.

In both early-exercise ASCOS algorithms we proceed from time step to time step sequentially, however, there are certain parts of the algorithms for which parallelization is possible. For instance, in the first method, the integrals (15), for all k, l, can be computed independently of each other, and in the 2D method, the early–exercise points, $y^*_{m,n}$, can also be calculated independently for each $na-value$. The Fourier coefficients, that represent a vector in the first method and a matrix in the 2D method, are computed simultaneously on the GPU at each time step.

In both methods we need to perform matrix–vector multiplications, that result in $O(N^2)$ computations for the first method, and result in $O(n_q N_2)$ computations with the 2D method. In these operations, the summation in each row must be done sequentially. Two techniques can however be used to accelerate the GPU computation of a summation. First of all, we can sum up each row in a pairwise fashion, that is, we split the vector into two parts and add up the two sub–vectors simultaneously on the GPU. This process is repeated until we reach vector sizes of one element, being the sum of all elements of the original vector. A second way to accelerate the process is to use the shared–memory within each block, which significantly reduces the data–communication time on the GPU.

As an example, Table 1 presents the error and the GPU speedup, compared to the CPU implementation, when pricing an early–exercise arithmetic Asian option with $\mathcal{M} = 2$ using the 2D method. A speedup factor between 30 and 300 is achieved on the GPU. When the problem size increases, an even higher GPU speedup is expected, since then option pricing will be computationally more intensive and the advantages of parallelization are more profound. When the number of early–exercise dates increases, the GPU as well as the CPU times will increase linearly.

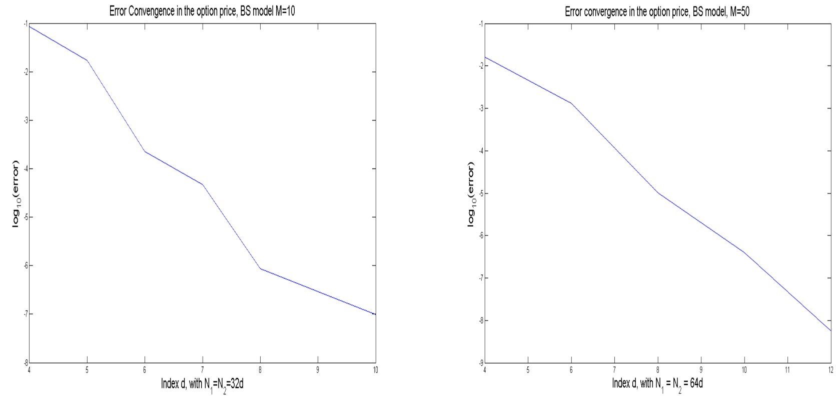
All further numerical experiments will be performed on the GPU.

$N_1 = N_2 = (n_q/2) + 1$	128	256	512
abs.error	1.2134e-01	4.6379e-06	1.3043e-08
GPU speedup	30.4	139.6	341.0

Table 1: GPU speedup for Bermudan Asian options, BS model, $\mathcal{M} = 2, S_0 = 100, K = 100$.

6.2 Arithmetic Asian options on the GPU

Error convergence of early–exercise arithmetic Asian options under the NIG model, with 10 and 50 early–exercise dates, using the 2D ASCOS method, are presented in Figure 1. The horizontal axis presents index d, where in Figure 1(a), $N_1 = N_2 = 32d$, $(n_q/2)+1 = 256$, and in Figure 1(b), we use $N_1 = N_2 = 64d$, $(n_q/2)+1 = 512$. The vertical axis shows the logarithm of the absolute error. For $\mathcal{M} = 10$ as well as $\mathcal{M} = 50$ an exponential convergence is observed: When N_1 and N_2 increase linearly, the logarithm of the error in the option price decreases accordingly.



(a) $\mathcal{M} = 10$ (b) $\mathcal{M} = 50$

Figure 1: 2D ASCOS error convergence for early–exercise arithmetic Asian options with different numbers of early–exercise dates, NIG model, $S_0 = 100, K = 110$.

When comparing the two plots in Figure 1, we see that with an increasing number of early–exercise dates we require larger values for N_1, N_2 and n_q to reach the same level of accuracy. With smaller time steps, Δt, the conditional density function between consecutive time steps tends to be peaked, and an accurate approximation by means of cosine expansions then requires an increasing number of terms. The need for a larger value of n_q comes from the fact that the error of the Clenshaw–Curtis quadrature is observed in each term of (38) and there are N_1N_2-terms in total. Therefore, larger values for N_1 and N_2 give rise to a larger n_q-value to ensure the accuracy.

Tables 2 and 3 present the convergence behavior and computing time for the NIG model with $\mathcal{M} = 10, 50$, respectively, with the performance of both pricing methods presented. From Table 2 we see that when $\mathcal{M} = 10$, due to the error of the first method with a small number of exercise dates, the option price does not converge to the reference value with the first method. On the other hand, the first method is significantly faster than the 2D method. The first method exhibits a reduced computation complexity, by a factor $O(log_2 N_2)$, and the GPU speedup is higher when implementing the first method, as with the 2D method, there is an N_1 by $\mathcal{M}$ loop in the CUDA code.

First method			
$N_1 = N_2 = (n_q/2) + 1$	128	192	256
abs.error	3.3236e-01	3.1511e-01	3.1641e-01
GPU time	0.28	0.53	0.87
2D method			
$N_1 = N_2 = (n_q/2) + 1$	256	384	512
abs.error	1.4213e-04	3.1444e-07	2.2129e-09
GPU time	4.76	9.05	31.25

Table 2: Convergence and computation time of early-exercise arithmetic Asian put options, under the NIG model, with $\mathcal{M} = 10, S_0 = 100$ (time in seconds).

As $\mathcal{M}$ increases, as shown in Table 3, the error in the first method gets much smaller, and the option prices gradually converges to the reference value. This is consistent with our analysis.

First method	$N_1, N_2 = 256$ $(n_q/2) + 1 = 256$	$N_1, N_2 = 512$ $(n_q/2) + 1 = 512$	$N_1, N_2 = 768$
abs.error	1.7165e-02	1.4364e-03	4.4992e-05
GPU time	1.52	9.40	9.64
2D method	$N_1, N_2 = 256$ $(n_q/2) + 1 = 256$	$N_1, N_2 = 512$ $(n_q/2) + 1 = 512$	$N_1, N_2 = 768$
abs.error	3.8746e-03	1.0809e-04	4.8980e-07
GPU time	20.0	160.7	399.5

Table 3: Convergence and computation time of early-exercise arithmetic Asian put options, under the NIG model with $\mathcal{M} = 50, S_0 = 100$ (time in seconds).

7 Conclusions

In this article, we have developed an efficient pricing method for Asian options with early–exercise features for arithmetic averages, based on a two-dimensional risk–neutral formula. As an alternative, especially for a large number of exercise dates, a 1D pricing method based on the approximation of the conditional characteristic functions, is proposed, which can be used for very frequently exercised Asian options at a reduced amount of computations. Both methods are based on Fourier cosine expansions and Clenshaw–Curtis quadrature, and, depending on the smoothness of the density function, may give rise to exponential error convergence. The convergence behavior of the 2D ASCOS method is supported by a detailed error analysis, as well as by various numerical experiments. The flexibility and robustness of the 2D pricing method for different Lévy models and different numbers of early–exercise dates is shown in the numerical experiments. In particular, the Graphics Processing Unit, which supports parallel computing, turns out to be very efficient for the computation of arithmetic Asian option values. The speedup on the GPU is high as there are many "parallel" computations and not much data transfer.

References

[1] Benhamou, E., Fast Fourier Transform for Discrete Asian Options. *J. Computational Finance.* 6, 49–61, 2002.

[2] Bermúdez, A., Nogueiras, M.R., Vázquez, C., Numerical solution of variational inequalities for pricing Asian options by high order Lagrange-Galerkin methods. *Applied Num. Math.* 56: 1256-1270, 2006.

[3] Boyd, J. P., *Chebychev and Fourier Spectral Methods*, 2nd ed., Dover, New York, 2001.

[4] Carverhill, A., and Clewlow L., Flexible Convolution, *From Black Scholes to Black Holes*, 165–171, 1992.

[5] Clenshaw, C. W., Curtis, A. R., A method for numerical integration on an automatic computer, *Numer. Mathematik* 2: 197–205. 1960.

[6] D'Halluin, Y., Forsyth, P. A., Labahn, G., A semi–Lagrangian approach for American Asian options under jump diffusion. *SIAM J. Sci. Comput.* 27, 315–345, 2005.

[7] 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.

[8] Fang, F., Oosterlee, C.W., Pricing early-exercise and discrete barrier options by Fourier cosine series expansions. *Numer. Mathematik* 114: 27-62, 2009.

[9] Fusai, G., Meucci, A., Pricing discretely monitored Asian options under Lévy processes. *J. Banking and Finance.* 32, 2076–2088, 2008.

[10] Lemmens, D., Liang, L. Z. J., Tempere, J., De Schepper, A., Pricing bounds for discrete arithmetic Asian options under Lévy models. *Physica A: Statistical Mechanics and its Applications.* Vol. 389, Issue 22: 5193–5207, 2010.

[11] NVIDIA CUDA Programming Guide, Version 4.0, 2011.

[12] Zhang, B., Oosterlee, C. W., Efficient Pricing of Asian Options under Lévy Processes based on Fourier Cosine Expansions *Part I: European-Style Products*, submitted and available as TU Delft DIAM report 11–11.