

Hedging and Pricing Early-exercise Options with Complex Fourier Series Expansion

Abstract

We introduce a new numerical method called the complex Fourier series (CFS) method proposed by Chan (2017) to price options with an early-exercise feature—American, Bermudan and discretely monitored barrier options—under exponential Lévy asset dynamics. This new method allows us to quickly and accurately compute the values of early-exercise options and their Greeks. We also provide an error analysis to demonstrate that, in many cases, we can achieve an exponential convergence rate in the pricing method as long as we choose the correct truncated computational interval. Our numerical analysis indicates that the CFS method is computationally more comparable or favourable than the methods currently available. Finally, the superiority of the CFS method is illustrated with real financial data by considering Standard & Poor’s depository receipts (SPDR) exchange-traded fund (ETF) on the S&P 500[®] index options, which are American options traded from November 2017 to February 2018 and from 30 January 2019 to 21 June 2019.

JEL classifications: G12; G13

Keywords: American option, Bermuda option, Barrier option, complex Fourier series, early-exercise options, Lévy processes

1. Introduction

Early-exercise stock option valuation has been an important research subject for the past four decades (e.g., Geske and Johnson, 1984; Longstaff and Schwartz, 2001; Fang and Oosterlee, 2009b; Yu and Xie, 2015; Lian et al., 2015; Li et al., 2019). Numerical methods based on fast Fourier transform (FFT) are traditionally very efficient at pricing options due to the existence of the characteristic functions of asset price dynamics. Famous papers (e.g., Carr and Madan, 1999; Itkin, 2005; Lipton, 2002; Gong and Zhuang, 2017) provide good techniques using FFT to price European vanilla options under Lévy processes. The success of these papers has extended the use of FFT or a combination of it and other transformation methods, e.g., the Hilbert transform or Gaussian transform, to pricing exotic options (e.g., Broadie and Yamamoto, 2005; Feng and Linetsky, 2008; Jackson et al., 2008; Lord et al., 2008; Wong and Guan, 2011; Zeng and Kwok, 2014). In these papers, the asset price dynamics are not limited to Lévy processes but extend to more complicated stochastic processes such as time-changing Lévy processes. Although FFT is very efficient at option pricing, a disadvantage is that it requires a sustainable number of terms to reach desirable accuracy

15 (cf. the numerical results of using the fast Fourier time-stepping method (Jackson et al., 2008) and CONV (FFT-based method) (Lord et al., 2008) to price Bermudan options in Table 3.)

Beyond the FFT method, efforts by Oosterlee and his collaborators have attracted considerable attention (Fang and Oosterlee, 2009a,b, 2011; Leentvaar and Oosterlee, 2008; Ruijter and Oosterlee, 2015; Ruijter et al., 2015; Zhang and Oosterlee, 2013). In their work, they adopt the Fourier
20 cosine series (COS) to price options or derivatives that have different contingency claims and are characterised by path dependence and/or early-exercise features. The implementation of these methods is relatively simple but elegant and is capable of pricing options under different stochastic processes as long as their characteristic function exists. The main achievement of these methods is that they can, in many cases, maintain an exponential convergence rate when pricing options, e.g.,
25 European options. Moreover, these methods are also able to accurately price options under infinite variation processes.

Although the COS method is very successful in pricing options, it has a drawback. According to Hurn et al. (2014), the accuracy of the COS method relies on a truncation interval, and the construction of this interval involves the time to maturity of the options (due to the standard
30 derivation or the cumulants of a risky underlying asset). In their paper, when the interval is suitably large (more than, say, approximately ten standard deviations), the accuracy of using the full-range Fourier series (the combination of COS and sine series) is higher than that of using the COS method (the half-range COS method) in pricing European digital and vanilla options. Their finding is crucial because, in the financial industry, especially in the field of insurance and
35 pensions, the maturity of an option issued in the field can range up to 35 or 40 years. If we use the COS method to price the option with a larger truncated interval, the accuracy of the method can decrease and more terms are required to maintain satisfactory accuracy (cf. the last numerical test between the COS and CFS methods in Table 3). Moreover, these researchers also show that because the larger spectrum of the full-range Fourier series guarantees more rapid convergence, it
40 can improve computational time when the full-range Fourier series is used to calibrate financial models with a large number of real data inputs.

In this paper, we extend the theory of the complex Fourier series (CFS) method—the full-range Fourier series—proposed by Chan (2017) to circumvent the aforementioned problems. We use the CFS method to price early-exercise options to demonstrate that the CFS method is better than the
45 FFT method while requiring less computational cost and offering higher accuracy and is a more sensible method than the COS method for options with longer maturity. We also derive a complete error bound to prove that the method exhibits exponential convergence when a probability density function (PDF) is smooth. Finally, we show that the CFS method can be applied to a large amount of real data, the SPDR ETF on the S&P 500[®] American index options traded from November 2017
50 to February 2018 and from January 2019 to June 2019.

This paper proceeds as follows. In Section 2, we briefly introduce Lévy processes and their application to modelling risky underlying assets. In Section 3, we revise the CFS method for pricing

European options. We then present the CFS method for pricing Bermudan and American options, discuss the algorithm for finding the early-exercise points of both American and Bermudan options and show the techniques for accelerating the calculation of their prices in Section 4. The CFS pricing formulae for barrier options are presented in Section 5. We formulate the CFS representation of option Greeks, particularly option Delta and Gamma, in Section 6. Numerical results are presented in Section 7. Section 8 reports the CFS pricing performance over the S&P 500[®] index options. We conclude in Section 9. Finally, Appendix A, Appendix B and Appendix C contain the error analysis, the algorithms of computing Bermudan and American options, and the algorithms of computing discretely monitored barrier options respectively.

2. Lévy processes and their application in financial modelling

In this section, we briefly introduce Lévy processes and their application to modelling risky underlying assets. Standard references for Lévy processes can be found in Cont and Tankov (2004) and Schoutens (2003).

A Lévy process is an adapted real-valued stochastic process X_t , with $X_0 = 0$ that satisfies the following properties.

1. Independent increment: For any $0 \leq t_1 < t_2 < \dots < t_n < \infty$, $X_{t_2} - X_{t_1}, X_{t_3} - X_{t_2}, \dots, X_{t_n} - X_{t_{n-1}}$ are independent.
2. Time-homogeneous: For any $s < t$, $X_t - X_s$, is equal in distribution to X_{t-s} .
3. Stochastically continuous: For any $\epsilon > 0$, $\mathbb{P}[|X_{x+h} - X_t| \geq \epsilon] \rightarrow 0$ as $h \rightarrow 0$.

Here, \mathbb{P} is the probability measure, and ϵ is a very small positive number. Lévy processes consist of a linear drift, a Brownian process, and a jump process¹. The jump process is characterised by the density of jumps, which is called the Lévy density. We denote it as $\nu(d\chi)$. The characteristic function of a Levy process can be described by the Lévy-Khinchine representation given by

$$\phi(u) = \frac{1}{2}\sigma^2 u - i\gamma_c u + \int_{-\infty}^{\infty} (1 - e^{iu\chi} + iu\chi \mathbb{1}_{|\chi| \geq 1}) \nu(d\chi), \quad \chi \in X_{T-t}, \quad (1)$$

where σ^2 is the variance of the Brownian component and $\nu(d\chi)$ satisfies

$$\int_{\mathbb{R} \setminus \{0\}} \min(1, \chi^2) \nu(d\chi) < +\infty.$$

The jump process described above is a process of infinite variation. If the process has finite variation,

¹A jump process is a type of stochastic process that has discrete movements, called jumps, with random arrival times, rather than continuous movement.

$\nu(d\chi)$ will be changed to satisfy

$$\int_{\mathbb{R} \setminus \{0\}} \min(1, |\chi|) \nu(d\chi) < +\infty.$$

Then, the Lévy-Khinchine representation (1) simplifies to

$$\phi(u) = \frac{1}{2}\sigma^2 u^2 - i\gamma u + \int_{-\infty}^{\infty} (1 - e^{iu\chi}) \nu(d\chi), \quad \chi \in X_{T-t}.$$

In Table 1, we present a list of Lévy processes commonly used in financial applications and their characteristic functions.

Table 1: Characteristic functions $\varphi(u)$ of Lévy Processes. γ_c is equal to $r - q + \omega$.

Finite activity models	
Geometric Brownian motion (GBM)	$\exp\left((T-t)\left(iu\gamma_c - \frac{1}{2}\sigma^2 u^2\right)\right)$
The Merton model	$\exp\left((T-t)\left(iu\gamma_c - \frac{\sigma^2 u^2}{2} + \lambda\left(e^{-\sigma^2 u^2/2 + iz\mu_J} - 1\right)\right)\right)$
The Kou model	$\exp\left((T-t)\left(iu\gamma_c - \frac{\sigma^2 u^2}{2} + \lambda\left(\frac{p\alpha_1}{\alpha_1 - iu} + \frac{(1-p)\alpha_2}{\alpha_2 + iu} - 1\right)\right)\right)$
Infinite activity models	
Normal inverse Gaussian	$\exp\left((T-t)\left(iu\gamma_c - \frac{1}{2}\sigma^2 u^2 + \delta\left(\sqrt{\alpha^2 - \beta^2} - \sqrt{\alpha^2 - (\beta + iz)^2}\right)\right)\right)$
Variance Gamma	$\exp\left((T-t)iu\gamma_c\right) \left(1 / \left(1 - i\theta v u + \frac{\sigma^2 v}{2} u^2\right)\right)^{\frac{T-t}{v}}$
Finite-moment log stable (FMLS)	$\exp\left((T-t)\left(iu\gamma_c - (iu\sigma)^\alpha \sec\left(\frac{\pi\alpha}{2}\right)\right)\right)$
CGMY	$\exp\left((T-t)iu\gamma_c + (T-t)\left(C\Gamma(-Y)G^Y\left(\left(1 + \frac{iu}{G}\right)^Y - 1 - \frac{iuY}{G}\right) + C\Gamma(-Y)M^Y\left(\left(1 - \frac{iu}{M}\right)^Y - 1 + \frac{iuY}{M}\right)\right)\right), Y \in (0, 2) \setminus \{1\}$

We now turn our attention to the application of Lévy processes in financial modelling. First, we assume frictionless financial equity markets and no arbitrage and take as given an equivalent martingale measure \mathbb{Q} chosen by the market. All stochastic processes defined in the following are assumed to live on the complete filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t \geq 0}, \mathbb{Q})$. The stock price process $(S_t)_{t \geq 0}$ under \mathbb{Q} driven by a Lévy process X_t can be defined as follows:

$$S_T = S_t e^{(r-q)(T-t) + X_T - X_t + \omega(T-t)} \quad (2)$$

$$= S_t e^{(r-q+\omega)(T-t) + X_{T-t}}. \quad (3)$$

Throughout the paper, $r \geq 0$ and $q \geq 0$ denote the constant risk-free interest rate and the constant dividend yield, respectively; S_t represents the known stock price at time t ; and S_T represents the random stock price at time T . The condition that $(S_T e^{-(r-q)(T-t)})_{t \geq 0}$ is a martingale will be guaranteed by an appropriate choice of the mean-correcting compensator ω as follows:

$$\omega = \frac{1}{T-t} \mathbb{E} (e^{X_{T-t}}), \quad (4)$$

where $\mathbb{E} (e^{X_{T-t}})$ is assumed to be finite for all $0 \leq t \leq T$.

75 3. The complex Fourier series pricing formulae for European options

In this section, we briefly revise the CFS option pricing formula for European options to prepare for pricing early-exercise options in the next section. For further details on applying the CFS method in pricing European options, we refer readers to Chan (2017).

Suppose that we approximate a function $f : [a, b] \rightarrow \mathbb{R}$ with a truncated complex Fourier series
80 given by

$$f_N(x) = \Re \left[\sum_{k=-N}^N b_k e^{i \frac{2\pi}{b-a} kx} \right], \quad \text{with } b_k = \frac{1}{b-a} \int_a^b f(x) e^{-i \frac{2\pi}{b-a} kx} dx. \quad (5)$$

Here, \Re represents the real part of the function. We assume that $|b_k| \leq C$ for any C independent of k and that

$$\lim_{N \rightarrow \infty} (f(x) - f_N(x)) = 0 \quad \text{and} \quad \sum_{k=-\infty}^{\infty} |b_k|^2 < \infty \quad (6)$$

almost everywhere in $x \in [a, b]$.

A European option driven by $(S_t)_{t \geq 0}$ described in (2) and having the strike K can be exercised at the end of its life (maturity) T . With the payoff function of $G(e^{X_T})$, the risk-neutral option value at time t given the state variable of the underlying asset taking the value of the log stock price $x = \log(S_t)$ is given by

$$V(x, t) = e^{-r(T-t)} \mathbb{E}(G(e^{X_T}) | X_t = x), \quad (7)$$

Here, $G(e^{X_T})$ is the payoff of either a vanilla call or put.

If we use a scaled log-price random variable $\tilde{X}_t := X_t - \log(K)$ in (7), we can transform $G(e^{X_T})$ into $g(\tilde{X}_T, T)$ such that

$$g(\tilde{X}_T, T) = \begin{cases} K \max(e^{X_T - \log K} - 1, 0) = K \max(e^{\tilde{X}_T} - 1, 0) & \text{(for a call)} \\ K \max(1 - e^{X_T - \log K}, 0) = K \max(1 - e^{\tilde{X}_T}, 0) & \text{(for a put)} \end{cases}. \quad (8)$$

As the scaled log-price is still a Lévy process and, accordingly, the independent increments condition holds (cf. Condition 1 in Section 2), (7) becomes

$$\begin{aligned}
e^{-r(T-t)}\mathbb{E}(g(\tilde{X}_T, T)|X_t = x) &= e^{-r(T-t)}\mathbb{E}(g(X_t - \log(K) + X_T - X_t, T)) \\
&= e^{-r(T-t)}\mathbb{E}(g(\tilde{X}_t + X_{T-t}, T)) \\
&= e^{-r(T-t)}\int_{-\infty}^{\infty} g(\tilde{x} + \chi, T)f(\chi) d\chi,
\end{aligned} \tag{9}$$

where $\chi \in X_{T-t}$, $\tilde{x} = x - \log K \in \tilde{X}_t$, and $f(\chi)$ is the PDF of X_{T-t} . Then, performing a change in variables and setting $y = \tilde{x} + \chi$, we have

$$V(x, t) := v(\tilde{x}, t) = e^{-r(T-t)}\int_{-\infty}^{\infty} g(y, T)f(y - \tilde{x}) dy. \tag{10}$$

To express the CFS representation of $v(\tilde{x}, t)$, we choose a truncated computational interval $[c, d]$ to replace $[-\infty, \infty]$ in (10). The interval must satisfy the condition that

$$\int_c^d e^{iu\chi} f(\chi) d\chi \approx \mathbb{E}[e^{iuX_{T-t}}] = \varphi(u), \quad u \in \mathbb{R}, \tag{11}$$

where $\varphi(u)$ is the characteristic function of X_{T-t} . Using the Fourier transform shift theorem and the truncated CFS representation of a function described in (5), $f(y - \tilde{x})$ can be approximated as

$$f_N(y - \tilde{x}) = \Re \left[\sum_{k=-N}^N b_{k,T} e^{-i\frac{2\pi}{d-c}k\tilde{x}} e^{i\frac{2\pi}{d-c}ky} \right], \tag{12}$$

where

$$\hat{b}_{k,T} = \frac{1}{d-c}\varphi\left(-\frac{2\pi}{d-c}k\right) \approx \frac{1}{d-c}\int_c^d e^{-iuy} f(y) dy, \quad \hat{b}_{0,T} = \frac{1}{d-c}\varphi(0) \approx \frac{1}{d-c}. \tag{13}$$

We substitute (12) into (10) and apply Fubini's theorem, and $v(\tilde{x}, t)$ can be computed as

$$\begin{aligned}
&e^{-r(T-t)}\int_0^d g(y, T)\Re \left[\sum_{k=-N}^N \hat{b}_{k,T} e^{-i\frac{2\pi}{d-c}k\tilde{x}} e^{i\frac{2\pi}{d-c}ky} \right] dy \\
&= e^{-r(T-t)}\Re \left[\sum_{k=-N}^N \hat{b}_{k,T} e^{-i\frac{2\pi}{d-c}k\tilde{x}} \int_0^d g(y, T) e^{i\frac{2\pi}{d-c}ky} dy \right] \\
&= e^{-r(T-t)}\Re \left[\sum_{k=-N}^N \hat{b}_{k,T} \hat{g}_{k,T} e^{-i\frac{2\pi}{d-c}k\tilde{x}} \right].
\end{aligned} \tag{14}$$

In other words, if we regard $g(y, T)$ as the vanilla call and put payoff functions, we have

$$\begin{aligned}\widehat{g}_{k,T} &= \frac{K(d-c)}{i2\pi k + (d-c)} \left(e^{(i\frac{2\pi}{d-c}k+1)d} - 1 \right) - \frac{K(d-c)}{i2\pi k} \left(e^{i\frac{2\pi}{d-c}kd} - 1 \right), \\ \widehat{g}_{0,T} &= K(e^d - 1) - Kd \quad (\text{for a call}),\end{aligned}\tag{15}$$

$$\begin{aligned}\widehat{g}_{k,T} &= \frac{K(d-c)}{i2\pi k + (d-c)} \left(e^{(i\frac{2\pi}{d-c}k+1)c} - 1 \right) - \frac{K(d-c)}{i2\pi k} \left(e^{i\frac{2\pi}{d-c}kc} - 1 \right), \\ \widehat{g}_{0,T} &= K(e^c - 1) - Kc \quad (\text{for a put}).\end{aligned}\tag{16}$$

Finally, based on (13) and (15), we have the CFS pricing formula for European options defined as

$$v(\tilde{x}, t) = e^{-r(T-t)} \Re \left[2 \sum_{k=1}^N \widehat{b}_{k,T} \widehat{g}_{k,T} e^{-i\frac{2\pi}{d-c}k\tilde{x}} + \widehat{b}_{0,T} \widehat{g}_{0,T} \right].\tag{17}$$

4. The complex Fourier series pricing formulae for Bermudan and American options

In this section, we derive an approximate formula for early-exercise options via the CFS method.

85 To demonstrate our method, we first consider pricing a Bermudan option, a type of early-exercise option that can be exercised only on predetermined dates, typically every month. Subsequently, we extend the method to price American and barrier options. In addition, throughout this section, we illustrate the CFS pricing formula using the vanilla call and put payoff functions.

4.1. The complex Fourier series option pricing formulae for Bermudan options

Considering a Bermudan option with strike K and maturity T that can be exercised only on a given number of exercise dates $t = t_0 < t_1 \leq t_2 \leq \dots \leq t_j \leq t_{j+1} \leq \dots \leq t_{M_1} = T$, we can write the Bermudan option pricing formulae as

$$v(\tilde{x}_{t_j}, t_j) = \begin{cases} g(\tilde{x}_{t_j}, t_j) & j = M_1 \\ \max(c(\tilde{x}_{t_j}, t_j), g(\tilde{x}_{t_j}, t_j)) & j = 1, 2, 3, \dots, M_1 - 1, \\ c(\tilde{x}_{t_j}, t_j) & j = 0 \end{cases}\tag{18}$$

with

$$c(\tilde{x}_{t_j}, t_j) = e^{-r(t_{j+1}-t_j)} \mathbb{E} (v(\tilde{x}_{t_{j+1}}, t_{j+1}) | \tilde{x}_{t_j}).\tag{19}$$

Here, scaled log-price \tilde{x}_t is $x_t - \log K$, and $c(\tilde{x}_{t_j}, t_j)$ and $g(\tilde{x}_{t_j}, t_j)$ are the continuation value and the option payoff value at time t_j , respectively. Moreover, $c(\tilde{x}_{t_j}, t_j)$ is the risk-neutral expectation of $v(\tilde{x}_{t_{j+1}}, t_{j+1})$ given \tilde{x}_{t_j} . Given \tilde{x}_{t_j} at t_j , $\tilde{x}_{t_{j+1}}$ is considered a Lévy process, and as a result, the independent increment condition holds (cf. Condition 1 in Section 2). $c(\tilde{x}_{t_j}, t_j)$ can be further

transformed as follows:

$$e^{-r(t_{j+1}-t_j)} \int_{-\infty}^{\infty} v(\tilde{x}_{t_j} + \chi, t_{j+1}) f(\chi) d\chi, \quad \chi \in X_{t_{j+1}-t_j}, \quad (20)$$

90 where $X_{t_{j+1}-t_j}$ is a Lévy process with PDF f .

There are four main aspects and two assumptions involved when deriving the CFS formula for Bermudan option price $v(\tilde{x}_t, t)$ at $t = t_0$ in (18). First, since there is no closed-form expression in (20), we repetitively approximate $c(\tilde{x}_{t_j}, t_j)$ from t_{M_1-1} to t_0 with the CSF representation. Second, we evaluate the relationship between the complex Fourier transforms of $v(\tilde{x}_{t_j}, t_j)$ and $v(\tilde{x}_{t_{j+1}}, t_{j+1})$.
 95 Eventually, we represent the complex Fourier transform of $v(\tilde{x}_t, t)$ backwardly by that of $g(\tilde{x}_T, T)$. Third, as the option has the early-exercise feature, we show how one can find and implement the early-exercise point $\tilde{x}_{t_j}^*$ in the pricing formula at each t_j . Finally, we have the CFS representation of $v(\tilde{x}_t, t)$ at t . Given the two assumptions, as the requirement of the CFS pricing formula, the number of the terms in the truncated CSF representation of $c(\tilde{x}_{t_j}, t_j)$ is the same at each t_j . The
 100 time difference between two successive time points, e.g., $t_{j+1} - t_j$, is also equivalent.

The procedure for generating the truncated CFS representation $c_N(\tilde{x}_{t_j}, t_j)$ of order N is simply to adapt the ideas of approximating the European option value in (12)–(14). As the formation of $c(\tilde{x}_{t_j}, t_j)$ (20) is similar to that of (9), we apply a definite integration interval $[c, d]$ satisfied with (11), perform a change in variables by setting $y = \tilde{x}_{t_j} + \chi$ and reformulate $c(\tilde{x}_{t_j}, t_j)$ as follows:

$$e^{-r(t_{j+1}-t_j)} \int_c^d v(y, t_{j+1}) f(y - \tilde{x}_{t_j}) dy. \quad (21)$$

Now, approximating $f(y - \tilde{x}_{t_j})$ with a truncated complex Fourier series of order N , we can reach a general truncated Fourier series expansion of $c(\tilde{x}_{t_j}, t_j)$ given by

$$c_N(\tilde{x}_{t_j}, t_j) = e^{-r(t_{j+1}-t_j)} \Re \left[\sum_{k=-N}^N \hat{b}_{k, t_{j+1}} \hat{v}_{k, t_{j+1}} e^{-i \frac{2\pi}{b-a} k \tilde{x}_{t_j}} \right], \quad (22)$$

where

$$\hat{b}_{k, t_{j+1}} = \frac{1}{d-c} \varphi \left(-\frac{2\pi}{d-c} k, t_{j+1} - t_j \right), \quad \hat{b}_{0, t_{j+1}} = \frac{1}{d-c} \varphi(0) = \frac{1}{d-c}. \quad (23)$$

As the time difference of $t_{j+1} - t_j$ is the same, this implies that $\hat{b}_{k, t_{j+1}}$ and $\hat{b}_{0, t_{j+1}}$ are equivalent throughout at each t_{j+1} .

The complex Fourier transform of $v(\tilde{x}_{t_{j+1}}, t_{j+1})$ can be expressed in a function, particularly a matrix-vector product, of the complex Fourier transform of $v(\tilde{x}_{t_{j+2}}, t_{j+2})$. In (22), $\hat{v}_{k, t_{j+1}}$ is the complex Fourier transform of $v(y, t_{j+1})$. It is indeed composed of the Fourier transform of $c(y, t_{j+1})$

and $g(y, t_{j+1})$; therefore, $\widehat{v}_{k, t_{j+1}}$ can be further decomposed into the following forms:

$$\begin{aligned}\widehat{v}_{k, t_{j+1}} &= \int_c^d \max(c(y, t_{j+1}), g(y, t_{j+1})) e^{i \frac{2\pi}{d-c} ky} dy \\ &= \begin{cases} \int_c^{\widetilde{x}_{t_{j+1}}^*} c(y, t_{j+1}) e^{i \frac{2\pi}{d-c} ky} dy + \int_{\widetilde{x}_{t_{j+1}}^*}^d g(y, t_{j+1}) e^{i \frac{2\pi}{d-c} y} dy & \text{(for a call)} \\ \int_{\widetilde{x}_{t_{j+1}}^*}^b c(y, t_{j+1}) e^{i \frac{2\pi}{d-c} ky} dy + \int_c^{\widetilde{x}_{t_{j+1}}^*} g(y, t_{j+1}) e^{i \frac{2\pi}{d-c} y} dy & \text{(for a put)} \end{cases} \\ &= \widehat{c}_{k, t_{j+1}} + \widehat{g}_{k, t_{j+1}}\end{aligned}\quad (24)$$

Here, $\widetilde{x}_{t_{j+1}}^*$ is the early-exercise point at time t_{j+1} , which is the point at which the continuation value equals the payoff function, i.e., $c(\widetilde{x}_{t_{j+1}}^*, t_{j+1}) = g(\widetilde{x}_{t_{j+1}}^*, t_{j+1})$. The value of $\widetilde{x}_{t_{j+1}}^*$, as discussed in Fang and Oosterlee (2009b), can be found numerically via Newton's method and the like. Once $\widetilde{x}_{t_{j+1}}^*$ is obtained, we have

$$\begin{aligned}\widehat{g}_{k, t_{j+1}} &= \frac{K(d-c)}{i2\pi k + (d-c)} \left(e^{(i \frac{2\pi}{d-c} k + 1)d} - e^{(i \frac{2\pi}{d-c} k + 1)\widetilde{x}_{t_{j+1}}^*} \right) - \frac{K(d-c)}{i2\pi k} \left(e^{i \frac{2\pi}{d-c} kd} - e^{(i \frac{2\pi}{d-c} k + 1)\widetilde{x}_{t_{j+1}}^*} \right), \\ \widehat{g}_{0, t_{j+1}} &= K(e^d - e^{\widetilde{x}_{t_{j+1}}^*}) - K(d - \widetilde{x}_{t_{j+1}}^*) \quad \text{(for a call)}, \\ \widehat{g}_{k, t_{j+1}} &= \frac{K(d-c)}{i2\pi k + (d-c)} \left(e^{(i \frac{2\pi}{d-c} k + 1)c} - e^{(i \frac{2\pi}{d-c} k + 1)\widetilde{x}_{t_{j+1}}^*} \right) - \frac{K(d-c)}{i2\pi k} \left(e^{i \frac{2\pi}{d-c} kc} - e^{(i \frac{2\pi}{d-c} k + 1)\widetilde{x}_{t_{j+1}}^*} \right), \\ \widehat{g}_{0, t_{j+1}} &= K(e^c - e^{\widetilde{x}_{t_{j+1}}^*}) - K(c - \widetilde{x}_{t_{j+1}}^*) \quad \text{(for a put)}.\end{aligned}\quad (25)$$

Moreover, in (24), we again approximate $c(k, t_{j+1})$ with a truncated CFS representation of $c_N(k, t_{j+1})$, i.e.,

$$c(\widetilde{x}_{t_{j+1}}, t_{j+1}) \approx c_N(\widetilde{x}_{t_{j+1}}, t_{j+1}) = e^{-r(t_{j+2}-t_{j+1})} \Re \left[\sum_{k=-N}^N \widehat{b}_{k_1, t_{j+2}} \widehat{v}_{k_1, t_{j+2}} e^{-i \frac{2\pi}{b-a} k_1 \widetilde{x}_{t_{j+1}}} \right]. \quad (26)$$

This gives the same formation of the complex Fourier transform of $c(\widetilde{x}_{t_{j+1}}, t_{j+1})$ given by

$$\widehat{c}_{k, t_{j+1}} \approx \widehat{c}_{Nk, t_{j+1}} = e^{-r(t_{j+2}-t_{j+1})} \Re \left[\sum_{k_1=-N}^N \widehat{b}_{k_1, t_{j+2}} \widehat{v}_{k_1, t_{j+2}} \widehat{e}_{-k_1, k} \right], \quad (27)$$

where

$$\widehat{e}_{-k_1, k} := \begin{cases} \int_c^{\widetilde{x}_{t_{j+1}}^*} e^{i \frac{2\pi}{d-c} (-k_1+k)\widetilde{x}_{t_{j+1}}} d\widetilde{x}_{t_{j+1}} = \frac{d-c}{i2\pi(-k_1+k)} e^{i \frac{2\pi}{d-c} (-k_1+k)y} \Big|_c^{\widetilde{x}_{t_{j+1}}^*} & \text{(for a call)} \\ \int_{\widetilde{x}_{t_{j+1}}^*}^b e^{i \frac{2\pi}{d-c} (-k_1+k)\widetilde{x}_{t_{j+1}}} d\widetilde{x}_{t_{j+1}} = \frac{d-c}{i2\pi(-k_1+k)} e^{i \frac{2\pi}{d-c} (-k_1+k)y} \Big|_{\widetilde{x}_{t_{j+1}}^*}^b & \text{(for a put)} \end{cases}, \quad (28)$$

$$\widehat{e}_{0,0} := \begin{cases} \widetilde{x}_{t_{j+1}}^* - c & \text{(for a call)} \\ b - \widetilde{x}_{t_{j+1}}^* & \text{(for a put)} \end{cases}. \quad (29)$$

Based on (24) and (27) and replacing $\widehat{v}_{k_1, t_{j+2}}$ with $\widehat{v}_{N k_1, t_{j+2}}$ in (27), we can determine the relationship between $\widehat{v}_{N k, t_{j+1}}$ and $\widehat{v}_{N k, t_{j+2}}$ in a truncated CFS representation, i.e.,

$$\widehat{v}_{N k, t_{j+1}} \approx e^{-r(t_{j+2}-t_{j+1})} \Re \left[\sum_{k_1=-N}^N \widehat{b}_{k_1, t_{j+2}} \widehat{v}_{N k_1, t_{j+2}} \widehat{e}_{-k_1, k} \right] + \widehat{g}_{k, t_{j+1}}. \quad (30)$$

If we express (30) as a matrix-vector product, we have

$$\widehat{\mathbf{v}}_{N \mathbf{t}_{j+1}} = e^{-r(t_{j+2}-t_{j+1})} \mathbf{E}_{\mathbf{t}_{j+2}} \widehat{\mathbf{b}}_{\mathbf{t}_{j+2}} \widehat{\mathbf{v}}_{N \mathbf{t}_{j+2}} + \widehat{\mathbf{g}}_{\mathbf{t}_{j+1}}. \quad (31)$$

Here, $\widehat{\mathbf{v}}_{N \mathbf{t}_{j+1}}$, $\widehat{\mathbf{b}}_{\mathbf{t}_{j+2}} \widehat{\mathbf{v}}_{N \mathbf{t}_{j+2}}$, and $\widehat{\mathbf{g}}_{\mathbf{t}_{j+1}}$ are $2N \times 1$ vectors:

$$\begin{bmatrix} \widehat{v}_{N-N, t_{j+1}} \\ \widehat{v}_{N-N+1, t_{j+1}} \\ \vdots \\ \widehat{v}_{N, t_{j+1}} \end{bmatrix}, \begin{bmatrix} \widehat{b}_{-N, t_{j+2}} \widehat{v}_{N-N, t_{j+2}} \\ \widehat{b}_{-N+1, t_{j+2}} \widehat{v}_{N-N+1, t_{j+2}} \\ \vdots \\ \widehat{b}_{N, t_{j+2}} \widehat{v}_{N, t_{j+2}} \end{bmatrix}, \text{ and } \begin{bmatrix} \widehat{g}_{-N, t_{j+1}} \\ \widehat{g}_{-N+1, t_{j+1}} \\ \vdots \\ \widehat{g}_{N, t_{j+1}} \end{bmatrix} \quad (32)$$

respectively. Each element of $\widehat{b}_{k, t_{j+2}}$ can be computed by (23). In the same manner, we can compute each element of $\widehat{g}_{k, t_{j+1}}$ using (25). With the result of (28), $\mathbf{E}_{\mathbf{t}_{j+2}}$ can be also constructed as a $2N \times 2N$ matrix:

$$\begin{bmatrix} \widehat{e}_{N_1, -N} & \cdots & \widehat{e}_{N_1, N-1} & \widehat{e}_{N_1, N} \\ \widehat{e}_{N_1-1, -N} & \cdots & \widehat{e}_{N_1-1, N-1} & \widehat{e}_{N_1-1, N} \\ \vdots & \ddots & \ddots & \vdots \\ \widehat{e}_{-N_1, -N} & \cdots & \widehat{e}_{-N_1, N-1} & \widehat{e}_{-N_1, N} \end{bmatrix}. \quad (33)$$

At t , we can approximate $c_N(\tilde{x}_t, t)$ via the CFS method to obtain $v(\tilde{x}_t, t)$, as approximating $c_N(\tilde{x}_t, t)$ is equivalent to approximating $v(\tilde{x}_t, t)$ based on (18). This is because $c_N(\tilde{x}_t, t)$ can be expressed as

$$e^{-r(t_1-t)} \Re \left[\sum_{k=-N}^N \widehat{b}_{k, t_1} \widehat{v}_{N k, t_1} e^{-i \frac{2\pi}{b-a} k \tilde{x}_t} \right]. \quad (34)$$

Here, \widehat{v}_{k, t_1} is the element of $\widehat{\mathbf{v}}_{N \mathbf{t}_1}$. Furthermore, $\widehat{\mathbf{v}}_{N \mathbf{t}_{j+1}}$ has a matrix-vector product relationship with $\widehat{\mathbf{g}}_T$ based on (31). Using (31), holding N constant and finding $\tilde{x}_{t_j}^*$ on each early-exercise date, we can recursively determine that

$$\widehat{\mathbf{v}}_{N \mathbf{t}_1} = e^{-r(t_2-t_1)} \mathbf{E}_{\mathbf{t}_2} \widehat{\mathbf{b}}_{\mathbf{t}_2} \left(e^{-r(t_3-t_2)} \mathbf{E}_{\mathbf{t}_3} \widehat{\mathbf{b}}_{\mathbf{t}_3} \left(\dots \right. \right.$$

$$\left(e^{-r(T-t_{M_1-1})} \mathbf{E}_T \widehat{\mathbf{b}}_T \widehat{\mathbf{g}}_T + \widehat{\mathbf{g}}_{t_{M_1-1}} \right) \cdots + \widehat{\mathbf{g}}_{t_2} \Big) + \widehat{\mathbf{g}}_{t_1}. \quad (35)$$

$\widehat{\mathbf{g}}_T$ is the vector of the complex Fourier transform of a terminal payoff function at T . If it is a vanilla put and call, each element of $\widehat{\mathbf{g}}_T$ is as follows:

$$\begin{aligned} \widehat{g}_{k,T} &= \frac{K(d-c)}{i2\pi k + (d-c)} \left(e^{(i\frac{2\pi}{d-c}k+1)d} - 1 \right) - \frac{K(d-c)}{i2\pi k} \left(e^{(i\frac{2\pi}{d-c}k)d} - 1 \right), \\ \widehat{g}_{0,T} &= K(e^d - 1) - Kd \quad (\text{for a call}), \\ \widehat{g}_{k,T} &= \frac{K(d-c)}{i2\pi k + (d-c)} \left(e^{(i\frac{2\pi}{d-c}k+1)c} - 1 \right) - \frac{K(d-c)}{i2\pi k} \left(e^{(i\frac{2\pi}{d-c}k)c} - 1 \right), \\ \widehat{g}_{0,T} &= K(e^c - 1) - Kc \quad (\text{for a put}). \end{aligned} \quad (36)$$

4.2. The complex Fourier series pricing formulae for American options

There are basically two approaches to evaluating American options based on the CFS pricing formula for Bermudan options. As suggested in (Fang and Oosterlee, 2009b), one simple approach is to approximate an American option by a Bermudan option with many exercise opportunities. In other words, we increase the number of exercise opportunities M_1 to a very large value. According to Chang et al. (2007) and Geske and Johnson (1984), the other approach is to use Richardson extrapolation on a series of Bermudan options with an increasing number of exercise opportunities. We adapt the latter approach, which is also implemented in (Fang and Oosterlee, 2009b), to price American options here. The prices of American options can be obtained by applying Richardson extrapolation to the prices of a few Bermudan options with small M_1 . Suppose that we denote by $v(M_1)$ the value of a Bermudan option with M_1 early-exercise dates. By implementing the 4-point Richardson extrapolation scheme (cf. Fang and Oosterlee, 2009b), the American option price is given by

$$v_{Amer}(M_1) = \frac{1}{12} (64v(2^{M_1+3}) - 56v(2^{M_1+2}) + 14v(2^{M_1+1}) - v(2^{M_1})), \quad (37)$$

where $v_{Amer}(M_1)$ denotes the approximated value of the American option and $v(\cdot)$ is the CFS pricing formula for Bermudan options. 105

4.3. Early-exercise point using root finding techniques

In this short section, we combine the CFS method with root finding techniques, mainly Newton's method, to find early-exercise points. Newton's method is first proposed in Fang and Oosterlee (2009b) to find an early-exercise point. This technique can be used when one solves the following equality:

$$c(\widetilde{x}_{t_{j+1}}, t_{j+1}) = g(\widetilde{x}_{t_{j+1}}, t_{j+1}). \quad (38)$$

Therefore, to find $x_{t_j}^*$, we can implement different root finding techniques, such as the secant method. In this paper, as suggested in Fang and Oosterlee (2009b), we implement Newton's method (also known as the Newton-Raphson method) instead. The process of this method is repeated as

$$x_{l+1} = x_l - \frac{g(x_l, t_{j+1}) - c(x_l, t_{j+1})}{\frac{\partial}{\partial x_{t_j}} g(x_l, t_{j+1}) - \frac{\partial}{\partial x_{t_j}} c(x_l, t_{j+1})} \quad (39)$$

over x_l for $l = 1, 2, \dots$ until a sufficiently accurate value is reached. In the equation, we start with x_0 equal to $x_{t_{j+1}}^*$, the exercise point in the exercise date at t_{j+1} , and we also know that at maturity T , x_T^* is equal to 0.

As we know from the previous section, $c(\tilde{x}_{t_j}, t_j)$ in (39) can be approximated by $c_N(\tilde{x}_{t_j}, t_j)$ with the truncated CFS representation of

$$e^{-r(t_{j+1}-t_j)} \Re \left[\sum_{k=-N}^N \hat{b}_{k,t_{j+1}} \widehat{v}_{Nk,t_{j+1}} e^{-i \frac{2\pi}{b-a} k \tilde{x}_{t_j}} \right], \quad (40)$$

and similarly, its first-order differentiation can also be approximated with $\partial c_N(\tilde{x}_{t_j}, t_{j+1}) / \partial \tilde{x}_{t_j}$, which is equal to

$$e^{-r(t_{j+1}-t_j)} \Re \left[\sum_{k=-N}^N \left(-i \frac{2\pi}{b-a} k \right) \hat{b}_{k,t_{j+1}} \widehat{v}_{Nk,t_{j+1}} e^{-i \frac{2\pi}{b-a} k \tilde{x}_{t_j}} \right]. \quad (41)$$

110 4.4. Matrix-vector multiplication using the FFT

In (31), the direct computation of $\widehat{\mathbf{v}}_{\mathbf{N}t_{j+1}}$ is very expensive, i.e., $\mathcal{O}(N^2)$, with the input of $\widehat{\mathbf{v}}_{\mathbf{N}t_{j+2}}$. To make the computation less expensive in (31), we make use of $\mathbf{E}_{t_{j+2}}$, a Toeplitz matrix (each descending diagonal from left to right in the matrix is constant). Therefore, the matrix-vector product (31), with $\mathbf{E}_{t_{j+2}}$ (33), can be transformed into a circular convolution. To do so, we construct

$$\widehat{\mathbf{e}}_{t_{j+2}} = [\widehat{e}_{N_1, -N}, \widehat{e}_{N_1-1, -N}, \dots, \widehat{e}_{-N_1, -N}, 0, \widehat{e}_{N_1, N}, \widehat{e}_{N_1-1, N}, \dots, \widehat{e}_{-N_1+1, N}]^T, \quad (42)$$

$$\widehat{\mathbf{b}}\mathbf{v}_{t_{j+2}} = [\widehat{b}_{-N, t_{j+2}} \widehat{v}_{-N, t_{j+2}}, \widehat{b}_{-N+1, t_{j+2}} \widehat{v}_{-N+1, t_{j+2}}, \dots, \widehat{b}\widehat{v}_{N, t_{j+2}}, 0, \dots, 0]^T, \quad (43)$$

where $\widehat{\mathbf{e}}_{t_{j+2}}$ and $\widehat{\mathbf{b}}\mathbf{v}_{t_{j+2}}$ are $4N \times 1$ vectors. With the help of the FFT algorithm computed in $\mathcal{O}(N \log_2(N))$ operations, a circular convolution of two vectors is equal to the inverse discrete Fourier transform (\mathcal{D}^{-1}) of the products of the forward DFTs, \mathcal{D} , i.e.,

$$\widehat{\mathbf{e}}_{t_{j+2}} * \widehat{\mathbf{b}}\mathbf{v}_{t_{j+2}} = \mathcal{D}^{-1} \left(\mathcal{D}(\widehat{\mathbf{e}}_{t_{j+2}}) \cdot \mathcal{D}(\widehat{\mathbf{b}}\mathbf{v}_{t_{j+2}}) \right). \quad (44)$$

By retrieving the first $2N$ values of the product of (44), multiplying them by a discounted factor $e^{-r(t_{j+2}-t_{j+1})}$ and finally adding them into $\widehat{\mathbf{g}}_{t_{j+1}}$, we can obtain $\widehat{\mathbf{v}}_{\mathbf{N}t_{j+1}}$.

Based on the aforementioned sections, we summarise the computational algorithms for pricing American and Bermudan options in Appendix B.

115 5. The complex Fourier series pricing formulae for barrier options

A barrier option is an early-exercise option whose payoff depends on the stock price crossing a pre-set barrier level during the option's lifetime. We call the option an up-and-out, knock-out, or down-and-out option when the option's existence fades out after crossing the barrier level. By contrast, we call the option an up-and-in, knock-in, or down-and-in option when the option comes
 120 into existence after reaching the barrier level. Like European vanilla options, these options can all be written as either put or call contracts that have a pre-determined strike price on an expiration date. In this paper, we investigate only four basic types of barrier options: the down-and-out barrier (DO) option, down-and-in barrier (DI) option, up-and-out barrier (UO) option and up-and-in barrier (UI) option. If a DO or UO option provides a refund in the event that the knock-out
 125 occurs, we call it a rebate DO or UO option.

After categorising the barrier options we examine in this paper, we turn to investigating their discretely monitored barrier—a subsidiary class of barrier options—and then examine their continuous version. The structure of discretely monitored barrier options is the same as that of Bermudan options. Instead of having a pre-set exercise date and an early-exercise point like Bermudan options,
 130 barrier options have a pre-set monitored date and a barrier level. In the case of Bermudan options, when the stock price goes across the early exercise point, a payoff occurs and the option expires immediately. In the same manner, a barrier option knocks out or in immediately when the barrier level is crossed. The barrier level acts exactly the same as the exercise point in Bermudan options. However, in the case of a barrier option without a rebate, no payoff occurs when the barrier level
 135 is reached; otherwise, a rebate occurs when a barrier option is knocked out.

We use both rebate DO and UO options as illustrations to formulate their singular Fourier-Padé pricing formulae and describe their computational algorithms. Once we obtain the formulae and algorithms for pricing these two options, we extend them to the rest of the options.

Taking the same mathematical notations and formats established in Section 4, we use a scaled log-price $\tilde{x}_{t_j} = x_{t_j} - \log K$ and a scaled barrier $\tilde{B} := \log B - \log K$ to describe the state variable and the barrier level of a discretely monitored barrier option at each t_j , respectively. Suppose that we have a rebate DO (UO) option driven by S_t with a barrier B , a rebate R_b , a strike K and a

series of monitoring dates $M_1: t = t_0 < t_1 < t_j < \dots < t_{M_1} = T$; the option can be described as

$$v(\tilde{x}_{t_j}, t_j) = \begin{cases} gz(\tilde{x}_{t_j}, t_j) & j = M_1, \\ c(\tilde{x}_{t_j}, t_j) \cdot \mathbf{1}_{\tilde{x}_{t_j} > \tilde{B}} + e^{-r(T-t_j)} R_b \cdot \mathbf{1}_{\tilde{x}_{t_j} \leq \tilde{B}} & j = 0, 1, \dots, M_1 - 1, \quad (\text{for a DO}), \\ c(\tilde{x}_{t_j}, t_j) \cdot \mathbf{1}_{\tilde{x}_{t_j} < \tilde{B}} + e^{-r(T-t_j)} R_b \cdot \mathbf{1}_{\tilde{x}_{t_j} \geq \tilde{B}} & j = 0, 1, \dots, M_1 - 1, \quad (\text{for an UO}), \end{cases} \quad (45)$$

with

$$c(\tilde{x}_{t_j}, t_j) = e^{-r(t_{j+1}-t_j)} \mathbb{E} \left(v(\tilde{x}_{t_{j+1}}, t_{j+1}) | \tilde{x}_{t_j} \right). \quad (46)$$

Here, $gz(\tilde{x}_{t_j}, t_j)$ is defined as

$$gz(\tilde{x}_T, T) := \begin{cases} g(\tilde{x}_T, T) \cdot \mathbf{1}_{\tilde{x}_T > \tilde{B}} + R_b \cdot \mathbf{1}_{\tilde{x}_T \leq \tilde{B}} & (\text{for a DO}), \\ g(\tilde{x}_T, T) \cdot \mathbf{1}_{\tilde{x}_T < \tilde{B}} + R_b \cdot \mathbf{1}_{\tilde{x}_T \geq \tilde{B}} & (\text{for an UO}), \end{cases} \quad (47)$$

where $g(\tilde{x}_{t_j}, t_j)$ can be any vanilla put or call, i.e.,

$$g(\tilde{x}_T, T) = \begin{cases} K \max(e^{\tilde{x}_T} - 1, 0) & (\text{for a call}) \\ K \max(1 - e^{\tilde{x}_T}, 0) & (\text{for a put}) \end{cases}. \quad (48)$$

$\mathbf{1}$ is an indicator function that takes a value of 1 when $\tilde{x}_{t_j} > \tilde{B}$ ($\tilde{x}_{t_j} < \tilde{B}$); otherwise, it is equal to zero when its complement $\tilde{x}_{t_j} \leq \tilde{B}$ ($\tilde{x}_{t_j} \geq \tilde{B}$) exists. In addition, at each t_j , both events $\tilde{x}_{t_j} > \tilde{B}$ ($\tilde{x}_{t_j} < \tilde{B}$) and their complement $\tilde{x}_{t_j} \leq \tilde{B}$ ($\tilde{x}_{t_j} \geq \tilde{B}$) cannot occur simultaneously; only one of them can exist at each t_j .

The CFS computational algorithm of DO and UO is the same as that of the Bermudan option in Section 4. First, we approximate $v(\tilde{x}_t, t)$ with a truncated CFS representation. This can be achieved by implementing the CFS representation $c_N(\tilde{x}_t, t)$ of $c(\tilde{x}_t, t)$ in (46). As \hat{v}_{k,t_1} is the coefficient of $c_N(\tilde{x}_t, t)$, we work backwards and recursively from t_1 to T using (45) to identify \hat{v}_{k,t_1} in terms of the complex Fourier transform $\hat{g}_{k,T}$ of $gz(\tilde{x}_T, T)$ at T .

To illustrate the algorithm, at time t , $c(\tilde{x}_t, t)$ can be approximated with a form of

$$c_N(\tilde{x}_t, t) = e^{-r(t_1-t)} \Re \left[\sum_{k=-N}^N \hat{b}_{k,t_1} \hat{v}_{Nk,t_1} e^{-i \frac{2\pi}{b-a} k \tilde{x}_t} \right] \quad (49)$$

Hence, $v(\tilde{x}_t, t)$ in (45) can be reformulated as

$$\left\{ \begin{array}{l} \left(e^{-r(t_1-t)} \Re \left[\sum_{k=-N}^N \hat{b}_{k,t_1} \widehat{v}_{Nk,t_1} e^{-i \frac{2\pi}{b-a} k \tilde{x}_t} \right] \cdot \mathbb{1}_{\tilde{x}_t > \tilde{B}} + e^{-r(T-t)} R_b \cdot \mathbb{1}_{\tilde{x}_t \leq \tilde{B}} \right) \text{ (for a DO),} \\ \left(e^{-r(t_1-t)} \Re \left[\sum_{k=-N}^N \hat{b}_{k,t_1} \widehat{v}_{Nk,t_1} e^{-i \frac{2\pi}{b-a} k \tilde{x}_t} \right] \cdot \mathbb{1}_{\tilde{x}_t < \tilde{B}} + e^{-r(T-t)} R_b \cdot \mathbb{1}_{\tilde{x}_t \geq \tilde{B}} \right) \text{ (for an UO).} \end{array} \right. \quad (50)$$

Here, \hat{b}_{k,t_1} is the same as (23). Like the assumption of the CFS pricing formula for Bermudan options in Section 4, the time difference between two successive time steps, i.e., $t_{j+1} - t_j$, is the same. We can accordingly conclude that $\hat{b}_{k,t_{j+1}}$ and $\hat{b}_{0,t_{j+1}}$ are equivalent throughout at each t_{j+1} . Furthermore, in (50), \widehat{v}_{Nk,t_1} can be further simplified as

$$\begin{aligned} & \begin{cases} \int_c^d \left(c_N(y, t_1) \cdot \mathbb{1}_{y > \tilde{B}} + e^{-r(T-t_1)} R_b \cdot \mathbb{1}_{y \leq \tilde{B}} \right) e^{i \frac{2\pi}{d-c} ky} dy & \text{(for a DO),} \\ \int_c^d \left(c_N(y, t_1) \cdot \mathbb{1}_{y < \tilde{B}} + e^{-r(T-t_1)} R_b \cdot \mathbb{1}_{y \geq \tilde{B}} \right) e^{i \frac{2\pi}{d-c} ky} dy & \text{(for an UO).} \end{cases} \\ = & \begin{cases} \widehat{c}_{Nk,t_1} + e^{-r(T-t_1)} R_b \cdot \widehat{g\mathbb{1}}_{k,t_1} & \text{(for a DO),} \\ \widehat{c}_{Nk,t_1} + e^{-r(T-t_1)} R_b \cdot \widehat{g\mathbb{1}}_{k,t_1} & \text{(for an UO).} \end{cases} \end{aligned} \quad (51)$$

Here, $\widehat{g\mathbb{1}}_{k,t_1}$ is the complex Fourier transform of $\mathbb{1}$ such that

$$\widehat{g\mathbb{1}}_{k,t_1} = \frac{d-c}{i2\pi k} \left(e^{i \frac{2\pi}{d-c} k \tilde{B}} - e^{i \frac{2\pi}{d-c} kc} \right), \quad \widehat{g\mathbb{1}}_{0,t_1} = \tilde{B} - c, \quad \text{(for a DO),} \quad (52)$$

$$\widehat{g\mathbb{1}}_{k,t_1} = \frac{d-c}{i2\pi k} \left(e^{i \frac{2\pi}{d-c} kd} - e^{i \frac{2\pi}{d-c} k \tilde{B}} \right), \quad \widehat{g\mathbb{1}}_{0,t_1} = d - \tilde{B}, \quad \text{(for an UO).} \quad (53)$$

\widehat{c}_{Nk,t_1} is the complex Fourier transform of $c_N(y, t_1)$ given by

$$\widehat{c}_{Nk,t_1} = e^{-r(t_2-t_1)} \Re \left[\sum_{k_1=-N}^N \hat{b}_{k_1,t_2} \widehat{v}_{Nk_1,t_2} \widehat{e}_{-k_1,k} \right], \quad (54)$$

where

$$\widehat{e}_{-k_1,k} := \begin{cases} \frac{d-c}{i2\pi(-k_1+k)} e^{i \frac{2\pi}{d-c} (-k_1+k)y} \Big|_{\tilde{B}}^d & \text{(for a DO)} \\ \frac{d-c}{i2\pi(-k_1+k)} e^{i \frac{2\pi}{d-c} (-k_1+k)y} \Big|_c^{\tilde{B}} & \text{(for an UO)} \end{cases}, \quad \widehat{e}_{0,0} := \begin{cases} d - \tilde{B} & \text{(for a DO)} \\ \tilde{B} - c & \text{(for an UO)} \end{cases}. \quad (55)$$

To derive the value of \widehat{v}_{Nk,t_1} in (50), we fully utilise the idea of the matrix-vector product in (35) and evaluate backwardly to represent $\widehat{\mathbf{v}}_{\mathbf{N}t_1}$ (\widehat{v}_{Nk,t_1} being its element) in terms of $\widehat{\mathbf{g}\mathbf{z}}_{\mathbf{T}}$. Hence, \widehat{v}_{Nk,t_1}

can be found from $\widehat{\mathbf{v}}_{\mathbf{N}_{t_1}}$, and $\widehat{\mathbf{v}}_{\mathbf{N}_{t_1}}$ is given by

$$e^{-r(t_2-t_1)} \mathbf{E}_{t_2} \widehat{\mathbf{b}}_{t_{j+2}} \left(\dots e^{-r(t_{j+1}-t_j)} \mathbf{E}_{t_j} \widehat{\mathbf{b}}_{t_j} \left(\dots \left(e^{-r(T-t_{M_1-1})} \mathbf{E}_T \widehat{\mathbf{b}}_T \widehat{\mathbf{g}}_{\mathbf{z}_T} \right. \right. \right. \\ \left. \left. \left. + e^{r(T-t_{M_1-1})} R_b \cdot \widehat{\mathbf{g}}_{\mathbf{1}_{t_{M_1-1}}} \right) \dots + e^{-r(T-t_j)} R_b \cdot \widehat{\mathbf{g}}_{\mathbf{1}_{t_j}} \dots \right) + e^{-r(T-t_1)} R_b \cdot \widehat{\mathbf{g}}_{\mathbf{1}_{t_1}}. \quad (56)$$

Here, \mathbf{E}_{t_j} is a $2N \times 2N$ matrix, which is the same as (33) with the elements listed in (55); $\widehat{\mathbf{b}}_{t_j}$ is a $2N \times 1$ vector with the input elements of (23); $\widehat{\mathbf{g}}_{\mathbf{1}_{t_j}}$ is also a $2N \times 1$ vector with the input elements of (52); and finally, $\widehat{\mathbf{g}}_{\mathbf{z}_T}$, a $2N \times 1$ vector, consists of two other $2N \times 1$ vectors— $\widehat{\mathbf{g}}_T$ and $\widehat{\mathbf{g}}_{\mathbf{1}_T}$ —such that

$$\widehat{\mathbf{g}}_{\mathbf{z}_T} = \widehat{\mathbf{g}}_T + \widehat{\mathbf{g}}_{\mathbf{1}_T}. \quad (57)$$

Based on whether the terminal payoff function is a call or put (cf. (48)), each element of $\widehat{\mathbf{g}}_T$ and $\widehat{\mathbf{g}}_{\mathbf{1}_T}$ can be further distinguished in Table 2. For example, if it is a DO call, we have

$$\begin{aligned} \widehat{g}_{k,T} &= \frac{K(d-c)}{i2\pi k + (d-c)} \left(e^{(i\frac{2\pi}{d-c}k+1)d} - e^{(i\frac{2\pi}{d-c}k+1)\tilde{B}} \right) - \frac{K(d-c)}{i2\pi k} \left(e^{i\frac{2\pi}{d-c}kd} - e^{(i\frac{2\pi}{d-c}k+1)\tilde{B}} \right), \\ \widehat{g}_{0,T} &= K(e^d - e^{\tilde{B}}) - K(d - \tilde{B}), \\ \widehat{g}_{\mathbf{1}_{k,T}} &= \frac{d-c}{i2\pi k} \left(e^{i\frac{2\pi}{d-c}k\tilde{B}} - 1 \right), \quad \widehat{g}_{\mathbf{1}_{0,T}} = \tilde{B}. \end{aligned} \quad (58)$$

For both DI and UI barrier options, we modify (45) to allow the feature of knocking-in. Hence, (45) becomes

$$v(\tilde{x}_{t_j}, t_j) = \begin{cases} gz(\tilde{x}_{t_j}, t_j) & j = M_1, \\ c(\tilde{x}_{t_j}, t_j) \cdot \mathbf{1}_{\tilde{x}_{t_j} \leq \tilde{B}} & j = 0, 1, \dots, M_1 - 1, \quad (\text{for a DI}), \\ c(\tilde{x}_{t_j}, t_j) \cdot \mathbf{1}_{\tilde{x}_{t_j} \geq \tilde{B}} & j = 0, 1, \dots, M_1 - 1, \quad (\text{for an UI}), \end{cases} \quad (59)$$

with

$$gz(\tilde{x}_T, T) := \begin{cases} g(\tilde{x}_T, T) \cdot \mathbf{1}_{\tilde{x}_T \leq \tilde{B}} & (\text{for a DI}), \\ g(\tilde{x}_T, T) \cdot \mathbf{1}_{\tilde{x}_T \geq \tilde{B}} & (\text{for an UI}), \end{cases} \quad (60)$$

Here, $g(\tilde{x}_T, T)$ is the call or put payoff function described in (48). In the same manner used to formulate the CFS pricing formula of DO or UO options, the CFS representation of a DI or UI

option can be defined as follows:

$$c_N(\tilde{x}_t, t) = e^{-r(t_1-t)} \Re \left[\sum_{k=1}^N \widehat{b}_{k,t_1} \widehat{v}_{Nk,t_1} e^{-i \frac{2\pi}{b-a} k \tilde{x}_t} + \widehat{b}_{0,t_1} \widehat{v}_{N0,t_1} \right]. \quad (61)$$

Hence, $v(\tilde{x}_t, t)$ in (59) can be reformulated as

$$\left\{ \begin{array}{l} \left(e^{-r(t_1-t)} \Re \left[\sum_{k=-N}^N \widehat{b}_{k,t_1} \widehat{v}_{Nk,t_1} e^{-i \frac{2\pi}{b-a} k \tilde{x}_t} \right] \right) \cdot \mathbf{1}_{\tilde{x}_t \leq \tilde{B}} \text{ (for a DI),} \\ \left(e^{-r(t_1-t)} \Re \left[\sum_{k=-N}^N \widehat{b}_{k,t_1} \widehat{v}_{Nk,t_1} e^{-i \frac{2\pi}{b-a} k \tilde{x}_t} \right] \right) \cdot \mathbf{1}_{\tilde{x}_t \geq \tilde{B}} \text{ (for an UI).} \end{array} \right. \quad (62)$$

The matrix-vector product presentation of \widehat{v}_{Nt_1} in DI and UI options differs from (56). To address their context, $\widehat{\mathbf{v}}_{Nt_1}$ is modified to

$$e^{-r(t_2-t_1)} \mathbf{E}_{t_2} \widehat{\mathbf{b}}_{t_2} \left(\dots e^{-r(t_{j+1}-t_j)} \mathbf{E}_{t_j} \widehat{\mathbf{b}}_{t_j} \left(\dots \left(e^{-r(T-t_{M_1-1})} \mathbf{E}_T \widehat{\mathbf{b}}_T \widehat{\mathbf{g}}_{\mathbf{z}_T} \right) \dots \right) \dots \right). \quad (63)$$

Here,

$$\widehat{\mathbf{g}}_{\mathbf{z}_T} = \widehat{\mathbf{g}}_T. \quad (64)$$

The value of $\widehat{\mathbf{g}}_T$ depends on whether we are considering a call or put and can be found in Table 2. Moreover, each element of \mathbf{E}_{t_j} can be calculated as follows:

$$\widehat{e}_{-k_1, k} := \begin{cases} \frac{d-c}{i2\pi(-k_1+k)} e^{i \frac{2\pi}{d-c} (-k_1+k)y} \Big|_{\tilde{B}}^d & \text{(for an UI)} \\ \frac{d-c}{i2\pi(-k_1+k)} e^{i \frac{2\pi}{d-c} (-k_1+k)y} \Big|_c^{\tilde{B}} & \text{(for a DI)} \end{cases}, \quad \widehat{e}_{0,0} := \begin{cases} d - \tilde{B} & \text{(for an UI)} \\ \tilde{B} - c & \text{(for a DI)} \end{cases}. \quad (65)$$

We summarise the computational algorithms for pricing discretely monitored barrier options in Appendix C.

150 6. Option Greeks

In this paper, we focus on deriving three option Greeks—Delta, Gamma, and Vega. Delta, Δ , is defined as the rate of change in the option value with respect to changes in the underlying asset price; Gamma, Γ , is the rate of change in Δ with respect to changes in the underlying price; and finally, Vega is the measurement of an option's sensitivity to changes in the volatility of the underlying asset price. In general, volatility measures the amount and speed at which the price moves up and down, and it is often based on changes in the recent, historical prices of a trading instrument. Other Greeks, such as Theta, can be derived in a similar fashion; however, depending

Table 2: Complex Fourier transforms for a variety of barrier payoff functions at maturity time T .

Barrier payoff functions	Fourier Transform $\widehat{g}_{k,T}$	Fourier Transform $\widehat{g}_{0,T}$
DO/UI Call	$\frac{K(d-c)}{i2\pi k+(d-c)} \left(e^{(i\frac{2\pi}{d-c}k+1)d} - e^{(i\frac{2\pi}{d-c}k+1)\bar{B}} \right) - \frac{K(d-c)}{i2\pi k} \left(e^{i\frac{2\pi}{d-c}kd} - e^{(i\frac{2\pi}{d-c}k+1)\bar{B}} \right)$	$K(e^d - e^{\bar{B}}) - K(d - \bar{B})$
DO/UI Put	$\frac{K(d-c)}{i2\pi k+(d-c)} \left(e^{(i\frac{2\pi}{d-c}k+1)\bar{B}} - 1 \right) - \frac{K(d-c)}{i2\pi k} \left(e^{i\frac{2\pi}{d-c}k\bar{B}} - 1 \right)$	$K(e^{\bar{B}} - 1) - K\bar{B}$
UO/DI Call	$\frac{K(d-c)}{i2\pi k+(d-c)} \left(e^{(i\frac{2\pi}{d-c}k+1)\bar{B}} - 1 \right) - \frac{K(d-c)}{i2\pi k} \left(e^{i\frac{2\pi}{d-c}k\bar{B}} - 1 \right)$	$K(e^{\bar{B}} - 1) - K\bar{B}$
UO/DI Put	$\frac{K(d-c)}{i2\pi k+(d-c)} \left(e^{(i\frac{2\pi}{d-c}k+1)c} - e^{(i\frac{2\pi}{d-c}k+1)\bar{B}} \right) - \frac{K(d-c)}{i2\pi k} \left(e^{i\frac{2\pi}{d-c}kc} - e^{(i\frac{2\pi}{d-c}k+1)\bar{B}} \right)$	$K(e^c - e^{\bar{B}}) - K(c - \bar{B})$
	$\widehat{g}\mathbb{1}_{k,T}$	$\widehat{g}\mathbb{1}_{0,T}$
DO Call	$\frac{d-c}{i2\pi k} \left(e^{i\frac{2\pi}{d-c}k\bar{B}} - 1 \right)$	\bar{B}
DO Put	$\frac{d-c}{i2\pi k} \left(e^{i\frac{2\pi}{d-c}kc} - e^{i\frac{2\pi}{d-c}k\bar{B}} \right)$	$c - \bar{B}$
UO Call	$\frac{d-c}{i2\pi k} \left(e^{i\frac{2\pi}{d-c}kd} - e^{i\frac{2\pi}{d-c}k\bar{B}} \right)$	$d - \bar{B}$
UO Put	$\frac{d-c}{i2\pi k} \left(e^{i\frac{2\pi}{d-c}k\bar{B}} - 1 \right)$	\bar{B}

on the characteristic function, the derivation expression might be rather lengthy. We omit them here, as many terms are repeated.

Delta is the first derivative of the value of V of the option with respect to the underlying instrument price S . Hence, differentiating the truncated CFS expansion of v defined as the Bermuda option (18), American option (37), and barrier options (45) and (59) with respect to S , we have

$$\begin{aligned} \Delta_t &= \frac{\partial v(\tilde{x}_t, t)}{\partial S} = \frac{\partial v(\tilde{x}_t, t)}{\partial \tilde{x}_t} \frac{\partial \tilde{x}_t}{\partial S} \\ &= e^{-r(t_1-t)-x} \left(\Re \left[2 \sum_{k=1}^N \left(-i \frac{2\pi}{d-c} k \right) \widehat{b}_{k,t_1} \widehat{v}_{Nk,t_1} e^{-i\frac{2\pi}{d-c}k\tilde{x}_t} \right] \right). \end{aligned} \quad (66)$$

In a similar fashion, we can obtain Γ_t by differentiating Δ_t with respect to S such that

$$\Gamma_t = \frac{\partial^2 v(\tilde{x}_t, t)}{\partial S^2} = \frac{\partial \Delta_t}{\partial S} = \frac{\partial \Delta_t}{\partial \tilde{x}_t} \frac{\partial \tilde{x}_t}{\partial S}, \quad (67)$$

and eventually,

$$\Gamma_t = e^{-r(t_1-t)-2x} \left(\Re \left[2 \sum_{k=1}^N \left(i \frac{2\pi}{d-c} k \right) \left(i \frac{2\pi}{d-c} k + 1 \right) \widehat{b}_{k,t_1} \widehat{v}_{Nk,t_1} e^{-i\frac{2\pi}{d-c}k\tilde{x}_t} \right] - \right) \quad (68)$$

$$\Re \left[2 \sum_{k=1}^N \left(-i \frac{2\pi}{d-c} k \right) \widehat{b}_{k,t_1} \widehat{v}_{Nk,t_1} e^{-i \frac{2\pi}{d-c} k \widetilde{x}_t} \right]. \quad (69)$$

It is also easy to obtain the formula for Vega, $\frac{\partial v}{\partial \sigma}$, where σ is the initial value of the volatility at time t . For example, for the finite moment log stable (FMLS) process, as σ is the initial value of the volatility in its characteristic function, we derive Vega as follows:

$$\frac{\partial v(\widetilde{x}_t, \sigma, t)}{\partial \sigma} = e^{-r(t_1-t)} \left(\Re \left[2 \sum_{k=1}^N \frac{\partial \widehat{b}_{k,t_1}}{\partial \sigma} \widehat{v}_{Nk,t_1} e^{-i \frac{2\pi}{d-c} k \widetilde{x}_t} \right] \right), \quad (70)$$

with

$$\frac{\partial \widehat{b}_{k,t_1}}{\partial \sigma} = -\alpha \sigma^{\alpha-1} (t_1 - t) \left(\left(\frac{-2k\pi i}{d-c} \right) \right)^\alpha \sec \left(\frac{\pi\alpha}{2} \right) \widehat{b}_{k,t_1}, \quad (71)$$

160 where φ contains the parameter σ .

7. Numerical results

In this section, we demonstrate the performance of the CFS method through various numerical tests. The purpose of this section is, first, to test whether the error convergence analysis presented in Section Appendix A is in line with the numerical findings in this section. A number of popular numerical methods are implemented to test the CFS method in terms of the error convergence, convergence rate and computational time. These methods include the COS method (a Fourier COS series method, Fang and Oosterlee, 2009b), the CONV method (an FFT method, Lord et al., 2008) the multinomial method (Wong and Guan, 2011) and the Fourier space time-stepping (FSTS) method (Jackson et al., 2008). When we implement the CONV, we use Simpson's rule for the Fourier integrals to achieve fourth-order accuracy. We also set the damping factors of the CONV to 0 and any value greater than zero, respectively.

In all numerical experiments, applying a minimum and substantial interval $[c, d]$ is crucial to capture most of the mass of a PDF such that the CFS method can maintain a sensible global spectral convergence rate. To do so, we construct an interval related to the closed-form formulas of stochastic process cumulants. The idea of using the cumulants was first proposed by Fang and Oosterlee (2009a) to construct the definite interval $[c, d]$. Based on their ideas, we have the following expression for $[c, d]$:

$$d = \left| c_1 + L \sqrt{c_2 + \sqrt{c_4}} + \left| \log \left(\frac{S_0}{K} \right) \right| \right| \quad (72)$$

$$c = -d, \quad (73)$$

where c_1 , c_2 , and c_4 are the first, second and fourth cumulants, respectively, of the stochastic

process. For simple, less-complicated financial models, we also obtain closed-form formulas for c_1 , c_2 , and c_4 , which can be found in Chan (2017, 2018) and Fang and Oosterlee (2009a,b). As Fang and Oosterlee (2009b) suggest, any maturity time longer than 0.1 years is acceptable, and thus we use $L = 8$ as an appropriate value for the Lévy processes considered. If any maturity time is less than 0.1 years, we add a constant value of 0.1 into (72). Moreover, BSM stands for the Black-Scholes model (Geometric Brownian Motion); VG denotes the variance gamma model (Madan et al., 1998; Madan and Milne, 1991; Madan and Seneta, 1990); CGMY stands for the Carr-German-Maddan-Yor model (Carr et al., 2002); and NIG is short for the normal inverse Gaussian process (Barndorff-Nielsen, 1991). We use the parameter N to denote the number of terms in the CFS and COS methods and the grid points of the others. When we measure the approximation errors of the numerical methods, we use absolute errors, the infinity norm error R_∞ , as the measurement units. Following Fang and Oosterlee (2009b), to observe the spectral error convergence, we define the following ratio:

$$\text{ratio} = \frac{\ln \|err(2^{d+1})\|_\infty}{\ln \|err(2^d)\|_\infty} = \frac{\ln R_\infty(2^{d+1})}{\ln R_\infty(2^d)},$$

where $\ln R_\infty(2^d)$ denotes the infinity norm error between the reference solution and approximation obtained with $N = 2^d$ (for other methods, we use N instead). ratio should be equal to or above 2 if $R_\infty(N) = C_1 \exp(-P_1 N)$ with C_1 and P_1 not depending on N ; if the error convergence is algebraic, i.e., $R_\infty(N) = C_2 \exp(-P_2 N)$ with C_2 and P_2 not depending on N , ratio should equal $(d + 1)/d$.

Finally, all CPU times presented (in seconds) are determined after averaging the computational time over 120 experiments. A MacBook Pro with a 2.8 GHz Intel Core i7 CPU and two 8 GB DDR SDRAM (cache memory) is used for all experiments. The code is written in MATLAB R2011b.

In Table 3, we price Bermudan options under different Lévy processes and compare the CFS method with the CONV, COS, multinomial and FSTS methods in terms of error convergence and computational time. There are 15 predetermined dates of the options in total, and the reference values are generated via the COS method with $N = 2^{15} = 32768$. Since the PDF is smooth, compared with other methods, we can see that both the COS and CFS methods not only have higher accuracy with fewer numbers terms of N but also achieve the spectral convergence rate. In addition, in each numerical test, the computational time is very reasonable for the CFS method. In other words, it requires only approximately 0.06 seconds to generate a Bermudan option price with $N = 256$ and an approximately $1.00e - 13$ error difference from the reference price. Furthermore, in the last numerical test under the NIG model in Table 3, we can see that the CFS method can yield quicker convergence with fewer terms required than COS when $T = 40$. This is in line with the finding of Hurn et al. (2014). As $T = 40$, the truncated interval (72) is larger than approximately ten standard derivations, and as a result, as Hurn et al. (2014) suggest, the CFS method outperforms the COS method. We compare both the COS and CFS methods for pricing the American call

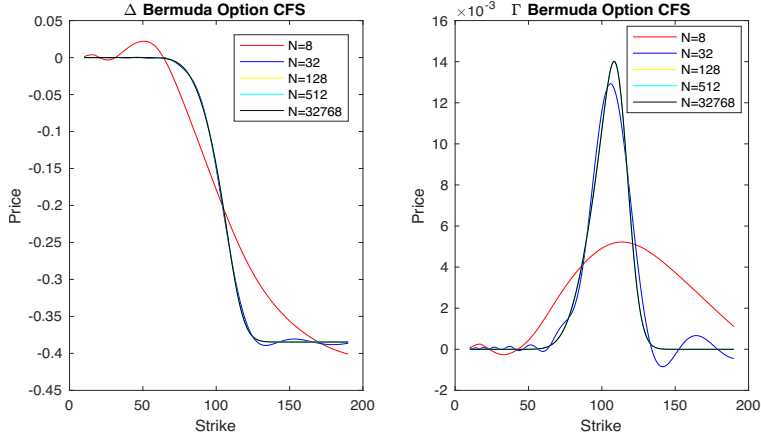


Figure 1: Recovered Bermudan put Delta Δ (top left) and Gamma Γ (top right) via the CFS method using the BSM model with $S_0 = 100$, $r = 0.1$, $q = 0.0$, $\sigma = 0.2$, $T = 1$ and K increasing from 10 to 190. There are a total of 10 predetermined dates for the put. The number of terms N increases in a sequence of 8 (red line), 32 (blue line) and 128 (yellow line), 512 (cyan line) and 32,768 (black line). The parameters are taken from Fang and Oosterlee (2009b).

option under the BSM model and American put option under the CGMY model, respectively, in Tables 4 and 5. In the tables, we apply the extrapolation method using (37). As M_1 increases gradually from 1 to 3, the error converges to a $8.456e - 04$ difference from the true value for a call option and a $3.045e - 07$ difference from the true value for a put option. In Figs. 1 and 2, we recover the Delta Δ and Gamma Γ of a Bermudan put option under the BSM and CGMY models. As long as we increase the value of N , the value difference of Δ and Γ in each N is diminished.

When we turn our attention to using the CFS method to price monthly monitored ($M_1 = 12$) up-and-out call and put options, (UO Call) and (UO Put), and down-and-out call and put options, (DO Call) and (DO Put), we solve the same problems as in Feng and Linetsky (2008) and Fang and Oosterlee (2009b) with barrier level $H = 120$ for the up-and-out and $H = 80$ for the down-and-out options. In Tables 6, 7 and 8, the CPU times are again measured in seconds, and the reference values are obtained by the COS method, with $N = 2^{15}$. Tables 6 and 7 show spectral convergence, as the ratio is almost equal to two when N increases twofold. This convergence is attributed to the PDF in both tables being smooth. When we have more monthly monitored M_1 , as in Table 8, the time intervals tend to be smaller, which means that the transitional PDF tends to become highly peaked. As a result, we have an algebraic convergence rate in the CFS method in Table 8, as the PDF is more highly peaked than those in Tables 6 and 7.

8. Empirical study: Standard & Poor's depositary receipts exchange-traded fund case

In the next numerical experiments, we test our new method to determine whether it is relevant to real market data. We consider American options based on Standard & Poor's Depositary Receipts (known as "SPDRs", with the ticker symbol "SPY") between 17 November 2017 and 16 February

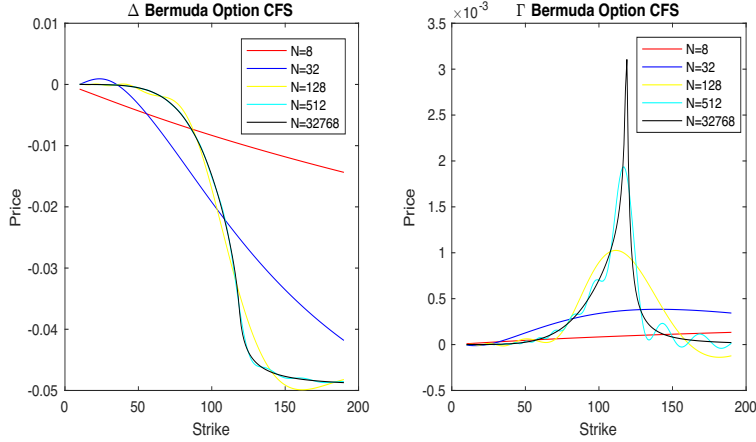


Figure 2: Recovered Bermudan put Delta Δ (top left) and Gamma Γ (top right) via the CFS method using the CGMY model with $S_0 = 100$, $r = 0.1$, $q = 0.0$, $C = 0.5$, $G = 3$, $M = 3$, $Y = 0.5$, $T = 1$ and K increasing from 10 to 190. There are a total of 10 predetermined dates for the put. The number of terms N increases in a sequence of 8 (red line), 32 (blue line) and 128 (yellow line), 512 (cyan line) and 32,768 (black line).

215 2018, covering 112 days, as well as between 30 January 2019 and 21 June 2019, covering 142 days. The SPDR exchange-traded fund (ETF) is designed to track the performance of the S&P 500[®] Index. We collect ask and bid prices, open interest, and volume together with the contract specifications (strike and maturity date) of both calls and puts. Additionally, we use the implied risk-free rates and dividends obtained from Bloomberg. Since the days-to-expiration information is not annualised, we divide it by 365 days to obtain an annualised value. Finally, we summarise the statistics of the retained options in Table 9.

220 In this paper, we only consider calibrating the parameters of the VG and CGMY models. To obtain higher accuracy when estimating model parameters, we discard options with zero volume and zero open interest in the data set, and as a result, we only have 332 observations for the period from 17 November 2017 to 16 February 2018 and 124 observations for the period from 30 January 2019 to 21 June 2019 in total. Moreover, we apply a local optimisation method (sequential quadratic programming [SQP]) suggested by Kienitz and Wetterau (2012) to estimate the model parameters. The input cost function of the SQP method is the root mean square error (RMSE) function, which is defined as follows:

$$RMSE = \sqrt{\frac{\sum_{j=1}^N (C_{mid,j} - C_{mod,j})^2}{N}} \quad (74)$$

where C_{mid} is the observed mid-price, which is the average between a bid price and an ask price, and C_{mod} is the model price at the current time, which is computed via the CFS method with the 4-point Richardson extrapolation scheme (cf. Eq. (37)). In terms of achieving higher accuracy in approximating the model parameters, we find that there is little difference between setting M_1 equal to 3 or 4 in Eq. (37). To reduce the computational time, we set M_1 equal to 3 rather than 4.

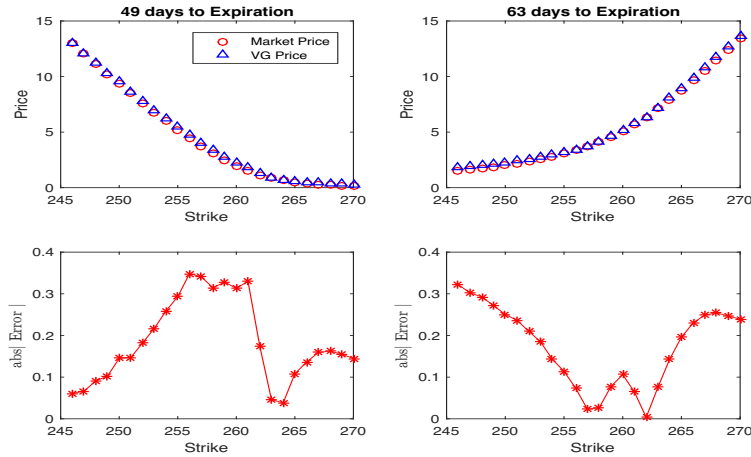


Figure 3: Graphical representations of the market prices and the VG model prices of the call (top left) and put (top right) options and their absolute differences (bottom left for the calls and bottom right for the puts). The data of set 2 and set 3 and the parameters of the VG models are retrieved from Tables 9 and 10, respectively.

As we use all the call and put mid-prices to calibrate the parameters of the models, we report the approximate model parameters and their RMSE values in Table 10. Moreover, based on the parameters of the VG and CGMY models, we approximate American call and put options via the CFS method with the 4-point Richardson extrapolation scheme and compare the approximate model option prices with the market prices in Table 9. We summarise the absolute maximum error difference between them in Table 11 from 17 November 2017 to 16 February 2018. In Table 11, from 17 November 2017 to 16 February 2018, we can restrict the absolute error difference between the model prices and the market prices within less than 0.380 in all data sets. We also graphically represent the market prices and the model prices and their comparisons in terms of absolute error differences in Figs. 3 and 4. Comparing Fig. 3 with Fig. 4, we can see that the CFS approximate option prices under the CGMY model have a smaller absolute error difference when the strike prices are closely grouped around the current stock price. Finally, in Table 12, we compare the accuracy of the CONV, COS and CFS approximate option prices against the market prices between 30 January 2019 and 21 June 2019 listed in Table 9. As we can see in the table, both COS and CFS methods are slightly better than the CONV method in terms of accuracy, but there is not much difference in accuracy between them.

9. Conclusions

In this paper, we show how to price options with an early-exercise feature using the CFS method when asset price dynamics are modelled as Lévy processes. According to all numerical experiments, when the option maturity is less than 2 years, the CFS method can achieve the same accuracy as the COS method. However, if the option maturity is considerably longer, such as 35 years or more, the CFS method can outperform the COS method with fewer terms required for higher accuracy. This is in line with the findings of Hurn et al. (2014). Moreover, if the PDF is smooth,

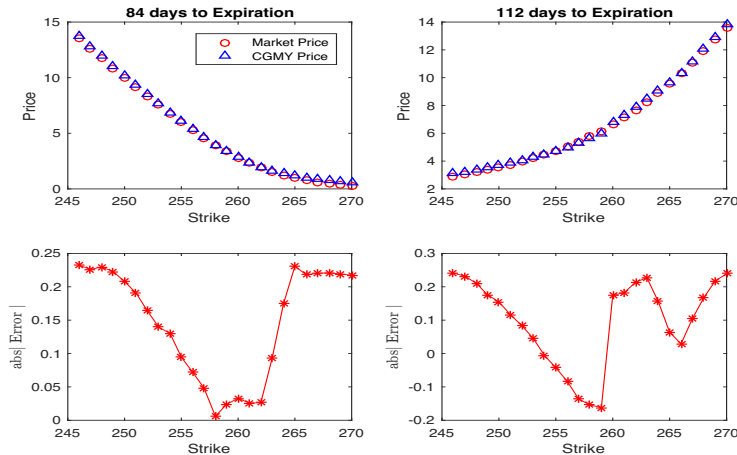


Figure 4: Graphical representations of the market prices and the CGMY model prices of the call (top left) and put (top right) options and their absolute differences (bottom left for the calls and bottom right for the puts). The data of set 4 and set 5 and the parameters of the CGMY models are retrieved from Tables 9 and 10, respectively.

we can obtain an exponential convergence rate; however, if the PDF is non-smooth, we obtain an algebraic convergence rate. In addition, we provide a theoretical proof of our method to show that the computational truncated interval plays an important role in determining the accuracy of our method, and the convergence rate of our method is subject to the smoothness of the input PDFs. Finally, we also evaluate the CFS method applied to the SPDR ETF on the S&P500[®] Index options with an American-style exercise. Through the empirical study, we see that the CFS method can be fairly accurate in modelling real financial option data.

Although the theoretical analysis/numerical results presented here demonstrate the effectiveness of the CFS method, the method might be further developed in three ways. First, it would be an interesting question to extend our method to price options when their underlying price is driven by time-changed Lévy processes (Carr and Wu, 2004). As these processes have been reported to yield higher accuracy when modelling option prices (Kienitz and Wetterau, 2012), such an extension has the potential to significantly improve our method’s ability to model real financial option prices. Second, the exponential convergence rate cannot be maintained if the input PDF is not smooth. To improve this, we apply the singularity Fourier-Pade method (Chan, 2018) or filters (Ruijter et al., 2015; Tadmor and Tanner, 2005; Vandeven, 1991), a numerical method for modifying the Fourier coefficients to maintain an exponential convergence rate of Fourier series expansions, to the CFS method. Finally, there is no theoretical establishment of our choice of the definite interval (72), let alone its value of L . It would be an interesting further research question to develop a theoretical proof for choosing a definite interval.

10. Acknowledgement

We thank Professor Jeremy Levesley, Department of Mathematics, University of Leicester and Professor Shingyu Leung, Department of Mathematics, the Hong Kong University of Science and

Technology for comments that greatly improved the manuscript.

Appendix A. Error analysis

In this section, we first conduct the error analysis of the CFS representation of Bermudan
 275 options to see whether it converges to the true price $v(\tilde{x}, t)$. Thereafter, we extend the result of the
 analysis to the American option and discretely monitored barrier option prices. In our analysis, we
 also show that the choice of $[c, d]$ plays a crucial role in determining the accuracy of the method.
 In addition, as it is the core development of the CFS method, we clearly show that global spectral
 convergence can be maintained despite the PDF being smooth.

280 Based on the context of the CFS representation framework, we can distinguish the following
 four approximation errors for Bermudan options.

1. The integration truncation error at each t_j :

$$\epsilon_{1,t_j} := \left| \left(\int_{-\infty}^{\infty} v(y, t_{j+1}) f(y - \tilde{x}_{t_j}) dy - \int_c^d v(y, t_{j+1}) f(y - \tilde{x}_{t_j}) dy \right) \right| \quad (\text{A.1})$$

2. The summation truncation error at each t_j :

$$\epsilon_{2,t_j} := \left| \sum_{k=-\infty}^{\infty} \left(\frac{1}{d-c} \int_c^d f(y) e^{-i \frac{2\pi}{d-c} ky} dy \right) \widehat{v}_{k,t_{j+1}} e^{-i \frac{2\pi}{d-c} k \tilde{x}_t} - \sum_{k=-N}^N \left(\frac{1}{d-c} \int_c^d f(y) e^{-i \frac{2\pi}{d-c} ky} dy \right) \widehat{v}_{k,t_{j+1}} e^{-i \frac{2\pi}{d-c} k \tilde{x}_t} \right| \quad (\text{A.2})$$

3. Error related to approximating $\frac{1}{(d-c)} \int_c^d f(y) e^{-i \frac{2\pi}{d-c} ky} dy$ with $\widehat{b}_{k,t_{j+1}}$ at each t_j :

$$\epsilon_{3,t_j} := \left| \sum_{k=-N}^N \left(\frac{1}{d-c} \int_c^d f(y) e^{-i \frac{2\pi}{d-c} ky} dy - \widehat{b}_{k,t_{j+1}} \right) \widehat{v}_{k,t_{j+1}} e^{-i \frac{2\pi}{d-c} k \tilde{x}_t} \right| \quad (\text{A.3})$$

4. The error related to approximating $\widehat{v}_{k,t_{j+1}}$ with $\widehat{v}_{Nk,t_{j+1}}$ at each t_j :

$$\epsilon_{4,t_j} := \left| \sum_{k=-N}^N \widehat{b}_{k,t_{j+1}} \left(\widehat{v}_{k,t_{j+1}} - \widehat{v}_{Nk,t_{j+1}} \right) e^{-i \frac{2\pi}{b-a} k \tilde{x}_{t_j}} \right| \quad (\text{A.4})$$

Before we describe the errors above in detail, we first explain how the errors occur in our CFS
 pricing formulae. At each pre-set exercise date t_j in pricing Bermudan options (cf. (18)), we
 approximate $c(\tilde{x}_{t_j}, t_j)$ with a definite integration interval, i.e.,

$$e^{-r(t_{j+1}-t_j)} \int_{-\infty}^{\infty} v(y, t_{j+1}) f(y - \tilde{x}_{t_j}) dy \approx e^{-r(t_{j+1}-t_j)} \int_c^d v(y, t_{j+1}) f(y - \tilde{x}_{t_j}) dy. \quad (\text{A.5})$$

This implies the first approximate error— ϵ_{1,t_j} (A.1). Then, we express $c(\tilde{x}_{t_j}, t_j)$ in a truncated CFS representation of order N of the form

$$e^{-r(t_{j+1}-t_j)} \Re \left[\sum_{k=-N}^N \left(\frac{1}{d-c} \int_c^d f(y) e^{-i \frac{2\pi}{d-c} ky} dy \right) \widehat{v}_{k,t_{j+1}} e^{-i \frac{2\pi}{b-a} k \tilde{x}_{t_j}} \right]. \quad (\text{A.6})$$

This indicates ϵ_{2,t_j} (A.2). In addition, we replace $\frac{1}{(d-c)} \int_c^d f(y) e^{-i \frac{2\pi}{d-c} ky} dy$ with $\widehat{b}_{k,t_{j+1}}$, a function of the characteristic function of (23), in (A.6). This leads to ϵ_{3,t_j} (A.3) and transforms (A.6) into

$$e^{-r(t_{j+1}-t_j)} \Re \left[\sum_{k=-N}^N \widehat{b}_{k,t_{j+1}} \widehat{v}_{k,t_{j+1}} e^{-i \frac{2\pi}{b-a} k \tilde{x}_{t_j}} \right]. \quad (\text{A.7})$$

Due to the lack of a closed-form representation of $\widehat{v}_{k,t_{j+1}}$ in (A.7), we use $\widehat{v}_{Nk,t_{j+1}}$, a complex Fourier transform of $v_N(y, t_{j+1})$ (cf. (27)), to approximate $\widehat{v}_{k,t_{j+1}}$. Accordingly, we have the last error ϵ_{4,t_j} and the CFS pricing formula defined as

$$e^{-r(t_{j+1}-t_j)} \Re \left[\sum_{k=-N}^N \widehat{b}_{k,t_{j+1}} \widehat{v}_{Nk,t_{j+1}} e^{-i \frac{2\pi}{b-a} k \tilde{x}_{t_j}} \right]. \quad (\text{A.8})$$

If we introduce the concept of the cumulative probability density function (CDF) $F(y)$ such that $f(y)dy = dF(y)$, we can simplify the integration truncation error ϵ_{1,t_j} as follows:

$$\begin{aligned} \epsilon_{1,t_j} &= \left| \left(\int_{-\infty}^{\infty} v(y, t_{j+1}) f(y - \tilde{x}_{t_j}) dy - \int_c^d v(y, t_{j+1}) f(y - \tilde{x}_{t_j}) dy \right) \right| \\ &= \left| \left(\int_{-\infty}^c v(y, t_{j+1}) f(y - \tilde{x}_{t_j}) dy + \int_d^{\infty} v(y, t_{j+1}) f(y - \tilde{x}_{t_j}) dy \right) \right| \\ &\leq \left| \left(\int_{-\infty}^c \frac{\partial v(y, t_{j+1})}{\partial y} F(y) dy \right) \right| + \left| \int_d^{\infty} \frac{\partial v(y, t_{j+1})}{\partial y} (1 - F(y)) dy \right| \end{aligned} \quad (\text{A.9})$$

$$\approx 0 : \quad (\text{if } y = c, d, -\infty, \infty). \quad (\text{A.10})$$

We can see that ϵ_{1,t_j} is bounded and approaches zero as long as $[c, d]$ is chosen reasonably such that $1 - F(d) \approx 0$ when $d < \infty$ or $F(c) \approx 0$ when $c > -\infty$. As we use the same interval $[c, d]$ for each time step t_j , we have the same truncated integration error ϵ_{1,t_j} throughout from T to t .

To analyse ϵ_{2,t_j} , we have

$$\begin{aligned} \epsilon_{2,t_j} &:= \left| \sum_{k=-\infty}^{\infty} \left(\frac{1}{d-c} \int_c^d f(y) e^{-i \frac{2\pi}{d-c} ky} dy \right) \widehat{v}_{k,t_{j+1}} e^{-i \frac{2\pi}{d-c} k \tilde{x}_t} - \right. \\ &\quad \left. \sum_{k=-N}^N \left(\frac{1}{d-c} \int_c^d f(y) e^{-i \frac{2\pi}{d-c} ky} dy \right) \widehat{v}_{k,t_{j+1}} e^{-i \frac{2\pi}{d-c} k \tilde{x}_t} \right| \end{aligned}$$

$$= \left| \sum_{|k| \geq N+1} \left(\frac{1}{d-c} \int_c^d f(y) e^{-i \frac{2\pi}{d-c} ky} dy \right) \widehat{v}_{k,t_{j+1}} e^{-i \frac{2\pi}{d-c} k \widetilde{x}_t} \right|$$

285 As we hold N constant from T to t_j , ϵ_{2,t_j} is the same throughout at each t_j .

According to Theorem 4 (Luke, 1969, p., 271), Luke suggests that ϵ_{2,t_j} vanishes at least $(N+1)$ times in $[c, d]$ with $z = \exp\left(-i \frac{2\pi}{d-c} k \widetilde{x}_{t_j}\right)$ and $|z| \leq 1$. Hence, ϵ_{2,t_j} is bounded. Moreover, ϵ_{2,t_j} exhibits an exponential convergence rate because Fang and Oosterlee (cf. Proposition 4.3, Fang and Oosterlee 2009a) suggest that the error following the truncation of the expansion after $N+1$ terms is bound to experience a convergence rate of $P \exp(-N\nu)$, where $P > 0$ and $\nu > 0$ are constant. As a result, we obtain

$$\epsilon_{2,t_j} \leq P \exp(-N\nu) \approx 0 : \quad (\text{if } |N| \rightarrow \infty). \quad (\text{A.11})$$

Sadly, according to Fang and Oosterlee [cf. proposition 4.2 and lemma 4.3], the luxury of having an exponential convergence rate does not remain if the rate becomes algebraic when we apply the complex Fourier expansion series around/at a discontinuity regarding one of its derivatives in a PDF, such as VG and CGMY. In this case, a new bound can be constructed as follows:

$$\epsilon_{2,t_j} \leq \frac{\hat{P}}{(N-1)^{\beta-1}} \approx 0 : \quad (\text{if } |N| \rightarrow \infty). \quad (\text{A.12})$$

Here, \hat{P} is a constant, and $\beta \geq n \geq 1$ (n is the algebraic index of convergence of $e^{i \frac{2\pi}{b-a} k \widetilde{x}_t}$).

Now, we adapt the same idea of investigating ϵ_{1,t_j} to determine the error bound of ϵ_{3,t_j} . Accordingly, taking into account $|\exp(i \frac{2\pi k}{d-c} y)| \leq 1$, we first investigate the error

$$\epsilon_{3,t_{j+1}} := \left| \frac{1}{d-c} \int_c^d f(y) e^{-i \frac{2\pi}{d-c} ky} dy - \widehat{b}_{k,t_{j+1}} \right|$$

in ϵ_{3,t_j} . If we expand the equation above, we obtain

$$\epsilon_{3,t_{j+1}} := \left| \frac{1}{d-c} \int_c^d f(y) e^{-i \frac{2\pi}{d-c} ky} dy - \frac{1}{d-c} \varphi \left(-i \frac{2\pi}{d-c} k, t_{j+1} - t_j \right) \right| \quad (\text{A.13})$$

$$\leq \left| \frac{1}{d-c} \left(\int_{-\infty}^c f(y) dy + \int_d^{\infty} f(y) dy \right) \right| \quad (\text{A.14})$$

$$= \left| \frac{1}{d-c} (F(\infty) - F(d) + F(c) - F(-\infty)) \right| \quad (\text{A.15})$$

$$\approx 0 : \quad (\text{if } y = c, d, -\infty, \infty). \quad (\text{A.16})$$

Based on the result above,

$$\begin{aligned}
\epsilon_{3,t_j} &:= \left| \Re \left[\sum_{k=-N}^N \left(\frac{1}{d-c} \int_c^d f(y) e^{-i \frac{2\pi}{d-c} ky} dy - \widehat{b}_{k,t_{j+1}} \right) \widehat{v}_{k,t_{j+1}} e^{i \frac{2\pi}{d-c} k \widetilde{x} t} \right] \right| \\
&\leq \left| \Re \left[\sum_{k=-N}^N \epsilon_{3,t_{j+1}} \widehat{v}_{k,t_{j+1}} e^{i \frac{2\pi}{d-c} k \widetilde{x} t} \right] \right| \\
&\approx 0 : \quad (\text{if } \epsilon_{3,t_{j+1}} \rightarrow 0).
\end{aligned} \tag{A.17}$$

As the time difference between two successive time steps, e.g., $t_{j+1} - t_j$, is equal to one another and we declare the same $[c, d]$ at each t_j , the value of $\widehat{b}_{k,t_{j+1}}$ is equivalent to that of $\widehat{b}_{k,t_{j+2}}$. Accordingly, we can conclude that ϵ_{3,t_j} is the same for each t_j .

Before we completely determine ϵ_{4,t_j} , with the help of (24) and (30), we can see that

$$\begin{aligned}
|\widehat{v}_{k,t_{j+1}} - \widehat{v}_{Nk,t_j}| &= |\widehat{c}_{k,t_{j+1}} + \widehat{g}_{k,t_{j+1}} - \widehat{c}_{Nk,t_{j+1}} - \widehat{g}_{k,t_{j+1}}| \\
&= |\widehat{c}_{k,t_{j+1}} - \widehat{c}_{Nk,t_{j+1}}|
\end{aligned} \tag{A.18}$$

This is attributed to the closed-form expression of $\widehat{g}_{k,t_{j+1}}$ (cf. (25)). According to (30), we know that

$$\widehat{c}_{Nk_1,t_{j+1}} = e^{-r(t_{j+2}-t_{j+1})} \sum_{k_1=-N}^N \widehat{b}_{k_1,t_{j+2}} \widehat{v}_{Nk_1,t_{j+2}} \widehat{e}_{-k_1,k}. \tag{A.19}$$

This naturally implies that $|\widehat{v}_{k,t_{j+1}} - \widehat{v}_{Nk,t_j}|$ is bounded by $\epsilon_{1,t_{j+1}}$, $\epsilon_{2,t_{j+1}}$ and $\epsilon_{3,t_{j+1}}$ from t_{j+1} to t_{j+2} . If we pay attention to $\widehat{v}_{Nk_1,t_{j+2}}$ in (A.19), we can use the same approach to approximate $\widehat{v}_{k_1,t_{j+2}}$ with $\widehat{v}_{Nk_1,t_{j+2}}$. This accordingly indicates the same of error of $\epsilon_{1,t_{j+2}}$, $\epsilon_{2,t_{j+2}}$ and $\epsilon_{3,t_{j+2}}$ from t_{j+2} to t_{j+3} . Furthermore, the matrix-vector relationship of (35) also suggests that if we recursively work until maturity T , where $\widehat{v}_{k,T}$ is equal to the closed-form expression of $\widehat{g}_{k,T}$, we can infer that $|\widehat{v}_{k,t_{j+1}} - \widehat{v}_{Nk,t_j}|$ is bounded by

$$C((M_1 - j) - 1)(\epsilon_{1,t_{j+1}} + \epsilon_{2,t_{j+1}} + \epsilon_{3,t_{j+1}}), \tag{A.20}$$

where C is a constant and M_1 is the total number of exercise dates. This is because $\epsilon_{1,t_{j+1}}$, $\epsilon_{2,t_{j+1}}$ and $\epsilon_{3,t_{j+1}}$ are always the same from t_{j+1} to T when we keep the same N and $[c, d]$ at each t_{j+1} . Based on the result of $|\widehat{v}_{k,t_{j+1}} - \widehat{v}_{Nk,t_j}|$, we can finally determine ϵ_{4,t_j} , i.e.,

$$\epsilon_{4,t_j} := \left| \sum_{k=-N}^N \widehat{b}_{k,t_{j+1}} \left(\widehat{v}_{k,t_{j+1}} - \widehat{v}_{Nk,t_{j+1}} \right) e^{-i \frac{2\pi}{b-a} k \widetilde{x} t_j} \right|$$

$$\begin{aligned}
&\leq \left| \sum_{k=-N}^N \widehat{b}_{k,t_{j+1}} |C((M_1 - j) - 1)(\epsilon_{1,t_{j+1}} + \epsilon_{2,t_{j+1}} + \epsilon_{3,t_{j+1}})| e^{-i \frac{2\pi}{b-a} k \widetilde{x}_{t_j}} \right| \\
&\leq C_{\max}(\epsilon_{1,t_{j+1}} + \epsilon_{3,t_{j+1}} + \epsilon_{3,t_{j+1}}) \\
&\approx 0.
\end{aligned} \tag{A.21}$$

300 Here, $C_{\max} = C((M_1 - j) - 1)N \times \max_{-N \leq k \leq N} (\widehat{b}_{k,t_j})$.

From the analysis of ϵ_{1,t_j} , ϵ_{2,t_j} , ϵ_{3,t_j} , and ϵ_{4,t_j} , the total error of the CFS pricing formula for Bermudan options is

$$\begin{aligned}
\epsilon &= \left| v(\widetilde{x}_t, t) - e^{-r(t_1-t)} \Re \left[\sum_{k=-N}^N \widehat{b}_{k,t_1} \widehat{v}_{Nk,t_1} e^{-i \frac{2\pi}{d-c} k \widetilde{x}_t} \right] \right| \\
&= \left| e^{-r(t_1-t)} \left(\int_{-\infty}^{\infty} v(y, t_1) f(y - \widetilde{x}_t) dz - \Re \left[\sum_{k=-N}^N \widehat{b}_{k,t_1} \widehat{v}_{Nk,t_1} e^{-i \frac{2\pi}{d-c} k \widetilde{x}_t} \right] \right) \right| \\
&\leq \left| e^{-r(t_1-t)} \left(\left| \int_{-\infty}^{\infty} v(y, t_1) f(y - \widetilde{x}_t) dz - \int_c^d v(y, t_1) f(y - \widetilde{x}_t) dz \right| + \right. \right. \\
&\quad \left| \Re \left[\sum_{k=-\infty}^{\infty} \frac{1}{d-c} \int_c^d f(y) e^{-i \frac{2\pi}{d-c} ky} dy \widehat{v}_{k,t_1} e^{-i \frac{2\pi}{d-c} k \widetilde{x}_t} - \right. \right. \\
&\quad \left. \left. \sum_{k=-N}^N \frac{1}{d-c} \int_c^d f(y) e^{-i \frac{2\pi}{d-c} ky} dy \widehat{v}_{k,t_1} e^{-i \frac{2\pi}{d-c} k \widetilde{x}_t} \right] \right| + \\
&\quad \left| \Re \left[\sum_{k=-N}^N \frac{1}{d-c} \int_c^d f(y) e^{-i \frac{2\pi}{d-c} ky} dy \widehat{v}_{k,t_1} e^{-i \frac{2\pi}{d-c} k \widetilde{x}_t} - \sum_{k=-N}^N \widehat{b}_{k,t_1} \widehat{v}_{k,t_1} e^{-i \frac{2\pi}{d-c} k \widetilde{x}_t} \right] \right| + \\
&\quad \left| \Re \left[\sum_{k=-N}^N \widehat{b}_{k,t_1} \widehat{v}_{k,t_1} e^{-i \frac{2\pi}{d-c} k \widetilde{x}_t} - \sum_{k=-N}^N \widehat{b}_{k,t_1} \widehat{v}_{Nk,t_1} e^{-i \frac{2\pi}{d-c} k \widetilde{x}_t} \right] \right| + \\
&< C_{\max}(\epsilon_{1,t} + \epsilon_{3,t} + \epsilon_{2,t}) \\
&< \begin{cases} C_{\max}(\epsilon_{1,t} + \epsilon_{3,t} + P \exp(-N\nu)) & \text{(if a PDF is smooth)} \\ C_{\max} \left(\epsilon_{1,t} + \epsilon_{3,t} + \frac{\hat{P}}{(N-1)^{\beta-1}} \right) & \text{(if a PDF is non-smooth)} \end{cases} \\
&\approx 0.
\end{aligned} \tag{A.23}$$

Here, $C_{\max} = C((M_1 - 1)N \times \max_{-N \leq k \leq N} (\widehat{b}_{k,t_1}))$. The total error of the CFS pricing formula for American options is the same as that of the CFS pricing formula for Bermudan options, as when M_1 approaches a larger number, the CFS pricing formula for a Bermudan option becomes that for an American option.

To determine the total approximate error of discretely monitored barrier options, note that the

CFS pricing formulae for these options are as follows:

$$v(\tilde{x}_t, t) = \begin{cases} c_N(\tilde{x}_t, t)\mathbb{1}_{\tilde{x}_t > \tilde{B}} + Rb \cdot \mathbb{1}_{\tilde{x}_t \leq \tilde{B}} & \text{for an DO} \\ c_N(\tilde{x}_t, t)\mathbb{1}_{\tilde{x}_t < \tilde{B}} + Rb \cdot \mathbb{1}_{\tilde{x}_t \geq \tilde{B}} & \text{for an UO} \\ c_N(\tilde{x}_t, t)\mathbb{1}_{\tilde{x}_t \leq \tilde{B}} & \text{for an DI} \\ c_N(\tilde{x}_t, t)\mathbb{1}_{\tilde{x}_t \geq \tilde{B}} & \text{for an UI} \end{cases} \quad (\text{A.24})$$

305 From the formulae above, we apply only $c_N(\tilde{x}_t, t)$ with the CFS representation of order N . We can accordingly conclude that both discretely monitored barrier options and Bermudan options share the same total errors. This is attributed to the formulation of the CFS pricing formula for discretely monitored barrier options, which is equivalent to that of a Bermudan option, as we replace exercise dates with discretely monitored dates and early-exercise points with barrier levels at each t_j to
 310 obtain the pricing formula for discretely monitored barrier options.

Appendix B. The CFS pricing algorithm for Bermudan options and American options

We are now prepared to formulate the CFS pricing algorithm shown in Section 4 using a vanilla Bermudan call and put as an example. In Algorithm 1, we combine with Newton's method in Section 4.3 of finding the early-exercise points, and with the FFT algorithm in Section 4.4 to
 315 accelerate the calculation. To obtain the American option price, we can either adopt Algorithm 1 while increasing the number of M_1 to a large value or use (37) to apply Algorithm 1 in each $v(\cdot)$ in the equation.

Appendix C. The CFS pricing algorithm for computing discretely monitored barrier options

320 In this section we present the CFS pricing algorithm for discretely monitored barrier options, Algorithm 2. The algorithm summarises the ideas and computational steps shown in Section 5. It is also combined with the FFT algorithm in Section 4.4 to accelerate the calculation.

Result: Bermuda option price $v(\tilde{x}_t, t)$ at time t
initialisation;

$t_j = t_{M_1-1}$;

while $t_j \neq t$ **do**

if $t_j = t_{M_1-1}$ **then**

 compute $\hat{\mathbf{b}}_{\mathbf{T}}$ and $\hat{\mathbf{g}}_{\mathbf{T}}$ using (23) and (36) respectively;

 find $\tilde{x}_{t_j}^*$ via Newton's method shown in Section 4.3;

 compute $\hat{\mathbf{g}}_{t_{M_1-1}}$ with the inputs of (25), and apply (33) to compute $\mathbf{E}_{\mathbf{T}}$ with the inputs of (28) ;

 apply (31) to compute $\widehat{\mathbf{v}}_{\mathbf{N}t_{M_1-1}}$ using FFT shown in Section 4.4 with the inputs of $e^{-r(T-t_{M_1-1})}$, $\mathbf{E}_{\mathbf{T}}$, $\hat{\mathbf{b}}_{\mathbf{T}}$, $\hat{\mathbf{g}}_{\mathbf{T}}$, and $\hat{\mathbf{g}}_{t_{M_1-1}}$;

 store $\widehat{\mathbf{v}}_{\mathbf{N}t_{M_1-1}}$;

$t_j = t_{M_1-2}$;

else

 retrieve $\widehat{\mathbf{v}}_{\mathbf{N}t_{j+1}}$, and compute $\hat{\mathbf{b}}_{t_{j+1}}$ using (23);

 find $\tilde{x}_{t_j}^*$ via Newton's method shown in Section 4.3;

 compute $\hat{\mathbf{g}}_{t_j}$ with the inputs of (25), and apply (33) to compute $\mathbf{E}_{t_{j+1}}$ with the inputs of (28) ;

 apply (31) to compute $\widehat{\mathbf{v}}_{\mathbf{N}t_j}$ using FFT shown in Section 4.4 with the inputs of $e^{-r(t_{j+1}-t_j)}$, $\mathbf{E}_{t_{j+1}}$, $\hat{\mathbf{b}}_{t_{j+1}}$, $\widehat{\mathbf{v}}_{\mathbf{N}t_{j+1}}$, and $\hat{\mathbf{g}}_{t_j}$;

 store $\widehat{\mathbf{v}}_{\mathbf{N}t_j}$;

$t_j = t_{j-1}$;

end

end

retrieve $\widehat{\mathbf{v}}_{\mathbf{N}t_1}$ and compute $\hat{\mathbf{b}}_{t_1}$ using (23);

formulate the CFS expansion of $c_N(\tilde{x}_t, t)$ using (22) with the inputs of $e^{-r(t_1-t)}$, $\hat{\mathbf{b}}_{t_1}$ and $\widehat{\mathbf{v}}_{\mathbf{N}t_1}$ in Section 3;

and apply (34) to approximate $v(\tilde{x}_t, t)$.

Algorithm 1: Algorithm for computing Bermudan option price $v(\tilde{x}_t, t)$ at time t .

References

- Barndorff-Nielsen, O. E. (1991). Normal inverse Gaussian distributions and stochastic volatility
325 modelling. *Scandinavian Journal of Statistics*, 24(1):1–13.
- Broadie, M. and Yamamoto, Y. (2005). A double-exponential fast Gauss transform algorithm for pricing discrete path-dependent options. *Operations Research*, 53(5):764–779.
- Carr, P., Geman, H., Madan, D. B., and Yor, M. (2002). The fine structure of asset returns: An empirical investigation. *Journal of Business*, 75:305–332.
- 330 Carr, P. and Madan, D. (1999). Option valuation using the fast Fourier transform. *Journal of Computational Finance*, 4(2):61–73.

Result: Barrier option price $v(\tilde{x}_t, t)$ at time t
initialisation;

$t_j = t_{M_1-1}$;

while $t_j \neq t$ **do**

if $t_j = t_{M_1-1}$ **then**

 compute $\widehat{\mathbf{b}}_{\mathbf{T}}$ using (23);

 compute $\widehat{\mathbf{g}}\mathbf{z}_{\mathbf{T}}$ using (64) for a DO/UO option or a DI/UI option;

 compute $\mathbf{E}_{\mathbf{T}}$ using (28) with the elements of (55) for a DO/UO option or those of (65) for a DI/UI option;

 choose (56) for a DO/UO option or (63) for a DI/UI option before computing

$\widehat{\mathbf{v}}_{\mathbf{N}t_{M_1-1}}$;

 compute $\widehat{\mathbf{v}}_{\mathbf{N}t_{M_1-1}}$ shown in Section 4.4 with the inputs of $e^{T-t_{M_1-1}}$, $\mathbf{E}_{\mathbf{T}}$, $\widehat{\mathbf{b}}_{\mathbf{T}}$, $\widehat{\mathbf{g}}\mathbf{z}_{\mathbf{T}}$,

 and $\widehat{\mathbf{g}}\mathbf{l}_{t_{M_1-1}}$ for a DO/UO option or those of $e^{T-t_{M_1-1}}$, $\mathbf{E}_{\mathbf{T}}$, $\widehat{\mathbf{b}}_{\mathbf{T}}$ and $\widehat{\mathbf{g}}\mathbf{z}_{\mathbf{T}}$ for a DI/UI option;

 store $\widehat{\mathbf{v}}_{\mathbf{N}t_{M_1-1}}$;

$t_j = t_{M_1-2}$;

else

 retrieve $\widehat{\mathbf{v}}_{\mathbf{N}t_{j+1}}$, and compute $\widehat{\mathbf{b}}_{t_{j+1}}$ using (23);

 compute $\mathbf{E}_{t_{j+1}}$ using (28) with the elements of (55) for a DO/UO option or those of (65) for a DI/UI option;

 choose (56) for a DO/UO option or (63) for a DI/UI option before computing $\widehat{\mathbf{v}}_{\mathbf{N}t_j}$;

 compute $\widehat{\mathbf{v}}_{\mathbf{N}t_j}$ shown in Section 4.4 with the inputs of $e^{t_{j+1}-t_j}$, $\mathbf{E}_{t_{j+1}}$, $\widehat{\mathbf{b}}_{t_{j+1}}$, $\widehat{\mathbf{v}}_{\mathbf{N}t_{j+1}}$

 and $\widehat{\mathbf{g}}\mathbf{l}_{t_j}$ for a DO/UO option or those of $e^{t_{j+1}-t_j}$, $\mathbf{E}_{t_{j+1}}$, $\widehat{\mathbf{b}}_{t_{j+1}}$ and $\widehat{\mathbf{v}}_{\mathbf{N}t_{j+1}}$ for a

 DI/UI option;

 store $\widehat{\mathbf{v}}_{\mathbf{N}t_j}$;

$t_j = t_{j-1}$;

 and

end

end

formulate the CFS expansion of $c(\tilde{x}_t, t)$ using (50) with $\widehat{\mathbf{b}}_{t_1}$ and $\widehat{\mathbf{v}}_{\mathbf{N}t_1}$ in Section 3.;

Algorithm 2: Algorithm for computing the barrier option price $v(\tilde{x}_t, t)$ at time t .

- Carr, P. and Wu, L. (2004). Time-changed Lévy processes and option pricing. *Journal of Financial Economics*, 71(1):113–141.
- 335 Chan, T. L. R. (2017). An orthogonal series expansions method to hedge and price European-type options. Retrieved from https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2852033 (accessed 19 May 2016).
- Chan, T. L. R. (2018). Singular Fourier-Páde series expansion of European option prices. *Quantitative Finance*, 18(7):1149–1171.
- 340 Chang, C.-C., Chung, S.-L., and Stapleton, R. C. (2007). Richardson extrapolation techniques for the pricing of American-style options. *Journal of Futures Markets*, 27(8):791–817.
- Chockalingam, A. and Muthuraman, K. (2015). An approximate moving boundary method for American option pricing. *European Journal of Operational Research*, 240(2):431 – 438.
- Cont, R. and Tankov, P. (2004). *Financial Modelling with Jump Processes*. Chapman & Hall/CRC Financial Mathematics Series. Chapman & Hall/CRC, Boca Raton, FL.
- 345 Fang, F. and Oosterlee, C. W. (2009a). A novel pricing method for European options based on Fourier-cosine series expansions. *SIAM Journal on Scientific Computing*, 31(2):826–848.
- Fang, F. and Oosterlee, C. W. (2009b). Pricing early-exercise and discrete barrier options by Fourier-cosine series expansions. *Numerische Mathematik*, 114(1):27–62.
- Fang, F. and Oosterlee, C. W. (2011). A Fourier-based valuation method for Bermudan and Barrier 350 options under Heston’s model. *SIAM Journal on Financial Mathematics*, 2(1):439–463.
- Feng, L. and Linetsky, V. (2008). Pricing discretely monitored barrier options and defaultable bonds in Lévy process models: A fast heston transform approach. *Mathematical Finance*, 18(3):337–384.
- 355 Geske, R. and Johnson, H. E. (1984). The American put option valued analytically. *Journal of Finance*, 39(5):1511–24.
- Gong, X. and Zhuang, X. (2017). Measuring financial risk and portfolio reversion with time changed tempered stable Lévy processes. *The North American Journal of Economics and Finance*, 40:148–159.
- 360 Hurn, S., Lindsay, K., and McClelland, A. J. (2014). On the efficacy of Fourier series approximations for pricing European and digital options. *Applied Mathematics*, 5:2786–2807.
- Itkin, A. (2005). Pricing options with VG model using FFT. Retrieved from [arXiv:physics/0503137v1](https://arxiv.org/abs/physics/0503137v1).

- Jackson, K. R., Jaimungal, S., and Surkov, V. (2008). Fourier space time-stepping for option pricing with Lévy models. *Journal of Computational Finance*, 12(2):1–29.
- 365 Kienitz, J. and Wetterau, D. (2012). *Financial Modeling: Theory, Implementation and Practice with MATLAB Source*. John Wiley & Sons, Ltd, Hoboken, NJ.
- Leentvaar, C. and Oosterlee, C. (2008). Multi-asset option pricing using a parallel Fourier-based technique. *Journal of Computational Finance*, 12(1):1–26.
- Li, Z., Zhang, W.-G., Liu, Y.-J., and Zhang, Y. (2019). Pricing discrete barrier options under
370 jump-diffusion model with liquidity risk. *International Review of Economics and Finance*, 59:347 – 368.
- Lian, Y.-M., Liao, S.-L., and Chen, J.-H. (2015). State-dependent jump risks for American gold futures option pricing. *The North American Journal of Economics and Finance*, 33:115 – 133.
- Lipton, A. (2002). The vol smile problem. *Risk*, 15:61–66.
- 375 Longstaff, F. A. and Schwartz, E. S. (2001). Valuing american options by simulation: A simple least-squares approach. *Review of Financial Studies*, pages 113–147.
- Lord, R., Fang, F., Bervoets, F., and Oosterlee, C. W. (2008). A fast and accurate FFT-based method for pricing early-exercise options under Lévy processes. *SIAM Journal on Scientific Computing*, 30(4):1678–1705.
- 380 Luke, Y. L. (1969). *The Special Functions and Their Approximations: VV.1*, volume 53A of *Mathematics in Science and Engineering*. Academic Press, New York.
- Madan, D. B., Carr, P., and Chang, E. C. (1998). The Variance Gamma process and option pricing. *European Finance Review*, 2:79–105.
- Madan, D. B. and Milne, F. (1991). Option pricing with V. G. martingale components. *Mathe-*
385 *matical Finance*, 1:39–55.
- Madan, D. B. and Seneta, E. (1990). The Variance Gamma (V.G.) model for share market returns. *Journal of Business*, 63:511–524.
- Ruijter, M. J. and Oosterlee, C. W. (2015). A Fourier cosine method for an efficient computation of solutions to BSDEs. *Journal of Computational Finance*, 37(2):A859–A889.
- 390 Ruijter, M. J., Versteegh, M., and Oosterlee, C. (2015). On the application of spectral filters in a Fourier option pricing technique. *Journal of Computational Finance*, 19(1):75–106.
- Schoutens, W. (2003). *Lévy Processes in Finance: Pricing Financial Derivatives*. Wiley Series in Probability and Mathematical Statistics. Wiley, Chichester, UK.

- 395 Tadmor, E. and Tanner, J. (2005). Adaptive filters for piecewise smooth spectral data. *IMA Journal of Numerical Analysis*, 25(4):635–647.
- Vandeven, H. (1991). Family of spectral filters for discontinuous problems. *Journal of Scientific Computing*, 6(2):159–192.
- von Sydow, L., Höök, L. J., Larsson, E., Lindström, E., Milovanović, S., Persson, J., Shcherbakov, V., Shpolyanskiy, Y., Sirén, S., Toivanen, J., Waldén, J., Wiktorsson, M., Levesley, J., Li, 400 J., Oosterlee, C. W., Ruijter, M. J., Toropov, A., and Zhao, Y. (2015). BENCHOP—the BENCHmarking project in option pricing. *Int. J. Comput. Math.*, 92(12):2361–2379.
- Wong, H. Y. and Guan, P. (2011). An FFT-network for Lévy option pricing. *Journal of Banking & Finance*, 35(4):988–999.
- Yu, X. and Xie, X. (2015). Pricing American options: RNMs-constrained entropic least-squares 405 approach. *The North American Journal of Economics and Finance*, 31:155 – 173.
- Zeng, P. and Kwok, Y. K. (2014). Pricing barrier and Bermudan style options under time-changed Lévy processes: Fast Hilbert transform approach. *SIAM Journal on Scientific Computing*, 36(3):B450–B485.
- Zhang, B. and Oosterlee, C. W. (2013). Efficient pricing of European-style Asian options under 410 exponential Lévy processes based on Fourier cosine expansions. *SIAM Journal on Financial Mathematics*, 4(1):399–426.

Table 3: Comparison of the CONV, COS, multinomial, FSTS and CFS methods in terms of error convergence and computational time for pricing Bermudan options using Lévy models. The COS and CFS methods are more accurate than the others with fewer summation terms required. The total number of predetermined dates (M_1) is 15. A spectral convergence rate is obtained in both the COS and CFS methods.

Model & Parameters	Type	S_0	K	Multinomial		FSTS		CONV		COS		CFS	
				N	R_∞	Time	R_∞	Time	R_∞	Time	R_∞	Time	R_∞
BSM: $r = 0.04$; Put 100 100 32 1.759e-01 0.01 1.759e-01 0.01 2.258 0.01 3.567e-02 0.01 3.567e-02 0.01													
$q = 0.08$; 64 4.881e-02 0.02 3.231e-02 0.01 8.62e-01 0.02 1.504e-04 0.02 1.504e-04 0.02													
$\sigma = 0.2, T = 0.5$ 128 1.145e-02 0.03 1.091e-02 0.02 3.161e-01 0.04 6.212e-12 0.03 6.089e-12 0.04													
(Chockalingam and Muthuraman, 2015) 256 2.835e-03 0.04 2.891e-03 0.04 3.505e-02 0.04 8.438e-14 0.03 3.138e-13 0.04													
VG: $r = 0.1$, Call 100 80 32 3.129e-01 0.01 4.291e-01 0.01 9.112e-01 0.01 4.511e-02 0.01 4.538e-02 0.01													
$q = 0$, 64 2.082e-01 0.03 2.082e-02 0.03 2.415e-01 0.03 4.871e-05 0.02 8.55e-05 0.02													
$\sigma = 0.12$, 128 2.268e-01 0.04 2.168e-02 0.03 1.327e-01 0.05 1.875e-11 0.05 2.872e-11 0.05													
$\theta = -0.14, \nu = 0.2, T = 1$ 256 2.268e-01 0.05 2.218e-02 0.04 8.036e-04 0.05 -2.842e-14 0.05 8.842e-13 0.05													
(Fang and Oosterlee, 2009a)													
CGMY: $r = 0.1$, Call 100 120 32 1.611e-02 0.01 1.655e-02 0.01 3.213e-01 0.01 1.358 0.01 1.358e-01 0.01													
$q = 0$, 64 3.281e-02 0.02 3.281e-02 0.02 1.756e-01 0.03 5.321e-03 0.03 5.251e-04 0.03													
$C = 1, G = 5, M = 5$, 128 1.511e-03 0.04 1.564e-03 0.03 3.851e-02 0.03 5.234e-10 0.03 5.883e-11 0.04													
$Y = 1.5, T = 1$ 256 1.721e-05 0.04 1.728e-04 0.04 1.58e-03 0.05 2.842e-13 0.04 2.842e-14 0.04													
(Fang and Oosterlee, 2009b)													
NIG: $r = 0.05$; Put 90 100 32 3.784e-02 0.01 4.321e-02 0.05 2.56 0.02 2.176e-01 0.2 1.641e-02 0.01													
$q = 0.02$; 64 6.761e-03 0.02 6.451e-03 0.02 9.378e-02 0.02 1.852e-03 0.05 1.852e-08 0.02													
$T = 40$ 128 2.166e-03 0.04 2.891e-03 0.02 7.215e-03 0.05 5.123e-08 0.06 5.416e-14 0.05													
$\alpha = 15, \beta = -5, \delta = 0.5$ 256 1.946e-03 0.06 6.461e-04 0.01 1.306e-03 0.05 1.234e-13 0.04 5.416e-14 0.05													
(Fang and Oosterlee, 2009b)													

Table 4: CFS vs. COS in terms of error convergence and computational time for pricing American call options using the BSM model with $S_0 = 100$, $K = 100$, $r = 0.3$, $q = 0.3$, $T = 0.5$, $\sigma = 0.15$. The parameters are taken from von Sydow et al. (2015). The reference value is 4.17712.

M_1 in Eq. (37)	COS		CFS	
	R_∞	time	R_∞	time
0	6.23e-01	0.071	4.230e-01	0.081
1	7.34e-02	0.111	6.194e-02	0.126
2	5.23e-03	0.203	5.231e-03	0.214
3	5.04e-04	0.434	8.512e-04	0.521

Table 5: CFS vs. COS in terms of error convergence and computational time for pricing American put options using the CGMY model with $S_0 = 1$, $K = 1$, $r = 0.1$, $q = 0$, $T = 1$, $C = 1$, $G = 5$, $M = 5$, and $Y = 0.5$. The parameters are taken from Fang and Oosterlee (2009b). The reference value is 0.11215.

M_1 in Eq. (37)	COS		CFS	
	R_∞	time	R_∞	time
0	4.41e-05	0.071	5.411e-05	0.081
1	7.69e-06	0.101	5.694e-06	0.123
2	9.23e-07	0.201	1.231e-06	0.211
3	3.04e-07	0.432	3.561e-07	0.511

Table 6: The CFS method in terms of error convergence and computational time for pricing monthly monitored barrier options using the CGMY model with $S_0 = 100$, $K = 100$, $r = 0.05$, $q = 0.02$, $T = 1$, $C = 4$, $G = 50$, $M = 60$, and $Y = 0.7$. The parameters are taken from Fang and Oosterlee (2009b).

Option Type	Ref. Value	N	Time	R_∞	ratio
DO Put	2.339381026	2^4	0.0011	1.121e-01	-
		2^5	0.0024	4.198e-03	2.5
		2^6	0.0035	1.019e-05	2.1
		2^7	0.0044	1.308e-10	1.98
DO Call	9.155070561	2^4	0.0012	6.011e-02	-
		2^5	0.0023	4.912e-03	1.89
		2^6	0.0033	3.8750e-05	1.91
		2^7	0.0041	4.009e-10	2.13
UO Put	6.195603554	2^4	0.0011	6.023e-02	-
		2^5	0.0022	3.91e-03	2.3
		2^6	0.0043	4.321e-06	1.91
		2^7	0.0045	4.433e-11	1.93
UO Call	1.814827593	2^4	0.0013	3.45e-02	-
		2^5	0.0023	1.723e-03	1.89
		2^6	0.0034	2.788e-06	2.01
		2^7	0.0047	2.451e-11	1.91

Table 7: The CFS method in terms of error convergence and computational time for pricing monthly monitored barrier options using the NIG model with $S_0 = 100$, $K = 100$, $r = 0.05$, $q = 0.02$, $T = 1$, $\alpha = 15$, $\beta = -5$, $\delta = 0.5$. The parameters are taken from Fang and Oosterlee (2009b).

Option Type	Ref. Value	N	Time	R_∞	ratio
DO Put	2.139931117	2^7	0.0037	7.112e-03	-
		2^8	0.0047	2.230e-04	1.71
		2^9	0.0071	1.611e-06	1.81
		2^{10}	0.0141	2.0620e-13	1.93
DO Call	8.983106036	2^7	0.0039	5.123e-03	-
		2^8	0.0046	1.662e-04	1.65
		2^9	0.0082	2.433e-07	1.75
		2^{10}	0.0142	3.162e-13	1.89
UO Put	5.995341168	2^7	0.0037	2.345e-02	-
		2^8	0.0047	1.412e-03	1.76
		2^9	0.0081	1.982e-05	1.65
		2^{10}	0.0143	3.819e-09	1.79
UO Call	2.277861597	2^7	0.0035	1.231e-03	-
		2^8	0.0046	1.581e-05	1.65
		2^9	0.0082	5.794e-09	1.71
		2^{10}	0.0141	2.564e-14	1.65

Table 8: The CFS method in terms of error convergence and computational time for pricing monthly monitored ($M_1 = 252$) barrier options using the NIG model with $S_0 = 100$, $K = 100$, $r = 0.05$, $q = 0.02$, $T = 1$, $\alpha = 15$, $\beta = -5$, $\delta = 0.5$. The parameters are taken from Fang and Oosterlee (2009b).

Option Type	Ref. Value	N	Time	R_∞	ratio
DO Put	1.88148753	2^9	0.134	1.251e-02	-
		2^{10}	0.231	3.412e-03	1.31
		2^{11}	0.462	3.306e-04	1.41
		2^{12}	1.234	1.237e-05	1.41
		2^{13}	2.675	4.350e-08	1.52
DO Call	8.96705248	2^9	0.132	3.67e-04	-
		2^{10}	0.243	1.664e-04	1.11
		2^{11}	0.456	2.854e-05	1.21
		2^{12}	1.245	1.003e-06	1.32
		2^{13}	2.654	4.591e-09	1.39

Table 9: The data on the SPDR exchange-traded fund (ETF) retrieved from Bloomberg.

17 November 2017–16 February 2018					
	S_0 (\$)	r	q	Times-to-maturity (days)	K (\$)
Set 1	258.85	0.0123	0.000	21	235.5–263.5
Set 2	258.85	0.0128	0.0121	49	246–270
Set 3	258.85	0.0132	0.0122	63	246–270
Set 4	258.85	0.0136	0.0122	84	246–270
Set 5	258.85	0.0142	0.0122	112	246–270

30 January 2019–21 June 2019					
	S_0 (\$)	r	q	Times-to-maturity (days)	K (\$)
Set 1	265.94	0.0255	0.0128	45	254–278
Set 2	265.94	0.0261	0.0128	58	254–277
Set 3	265.94	0.0269	0.0128	78	257–275
Set 4	265.94	0.0277	0.0277	142	262–270

Table 10: The VG and CGMY model parameters are calibrated using the real market prices in Table 9.

17 November 2017–16 February 2018				
	Call	RMSE	Put	RMSE
VG:	$\sigma = 0.0835,$ $\theta = 0.212, v = 0.000412,$	0.334	$\sigma = 0.126,$ $\theta = 0.331, v = 0.000472,$	0.329
CGMY:	$C = 0.220, G = 8.803,$ $M = 20.000, Y = 0.503$	0.270	$C = 0.485, G = 9.257,$ $M = 12.205, Y = 0.366$	0.278

30 January 2019–21 June 2019				
	Call	RMSE	Put	RMSE
VG:	$\sigma = 0.0731,$ $\theta = 0.232, v = 0.000732,$	0.431	$\sigma = 0.236,$ $\theta = 0.562, v = 0.000372,$	0.478
CGMY:	$C = 0.330, G = 9.023,$ $M = 20.123, Y = 0.645$	0.401	$C = 0.501, G = 10.023,$ $M = 12.341, Y = 0.4112$	0.456

Table 11: The absolute maximum error (abs. err.) between the market prices from Table 9 and the CFS approximate prices using the VG and CGMY models between 17 November 2017 and 16 February 2018.

	VG		CGMY	
	abs. err. (Call)	abs. err. (Put)	abs. err. (Call)	abs. err. (Put)
Set 1	0.375	0.315	0.231	0.211
Set 2	0.341	0.321	0.212	0.234
Set 3	0.325	0.316	0.262	0.234
Set 4	0.311	0.302	0.232	0.211
Set 5	0.323	0.311	0.261	0.242

Table 12: The absolute maximum error (abs. err.) between the market prices from Table 9 and the CONV, COS and CFS approximate prices using the VG (top table) and CGMY (bottom table) models between 30 January 2019 and 21 June 2019.

VG	CONV		COS		CFS	
	abs. err. (Call)	abs. err. (Put)	abs. err. (Call)	abs. err. (Put)	abs. err. (Call)	abs. err. (Put)
Set 1	0.915	0.820	0.415	0.401	0.455	0.432
Set 2	0.875	0.710	0.437	0.421	0.465	0.434
Set 3	0.901	0.801	0.445	0.421	0.434	0.441
Set 4	0.832	0.798	0.423	0.412	0.423	0.434

CGMY	CONV		COS		CFS	
	abs. err. (Call)	abs. err. (Put)	abs. err. (Call)	abs. err. (Put)	abs. err. (Call)	abs. err. (Put)
Set 1	0.955	0.890	0.475	0.425	0.443	0.423
Set 2	0.925	0.789	0.434	0.425	0.410	0.489
Set 3	0.897	0.756	0.421	0.421	0.423	0.414
Set 4	0.867	0.789	0.411	0.402	0.412	0.413