Quantitative Finance, 2020
Vol. 20, No. 8, 1307–1324, https://doi.org/10.1080/14697688.2020.1736612

Pricing European-type, early-exercise and discrete barrier options using an algorithm for the convolution of Legendre series

TAT LUNG (RON) CHAN *† and NICHOLAS HALE*‡

†School of Business and Law, University of East London, Water Lane, Stratford, London E15 4LZ, UK
‡Department of Mathematical Sciences, Stellenbosch University, Stellenbosch, South Africa

(*Received 22 May 2019; accepted 17 February 2020; published online 7 April 2020*)

This paper applies an algorithm for the convolution of compactly supported Legendre series (the CONLeg method) (cf. Hale and Townsend, An algorithm for the convolution of Legendre series. *SIAM J. Sci. Comput.*, 2014, **36**, A1207–A1220), to pricing European-type, early-exercise and discrete-monitored barrier options under a Lévy process. The paper employs Chebfun (cf. Trefethen *et al.*, *Chebfun Guide*, 2014 (Pafnuty Publications: Oxford), Available online at: http://www.chebfun.org/) in computational finance and provides a quadrature-free approach by applying the Chebyshev series in financial modelling. A significant advantage of using the CONLeg method is to formulate option pricing and option Greek *curves* rather than individual prices/values. Moreover, the CONLeg method can yield high accuracy in option pricing when the risk-free smooth probability density function (PDF) is smooth/non-smooth. Finally, we show that our method can accurately price options deep in/out of the money and with very long/short maturities. Compared with existing techniques, the CONLeg method performs either favourably or comparably in numerical experiments.

Keywords: Convolution; Legendre series; European options; Early-exercise options; Discrete-monitored barrier options; Lévy process

Jel Classification: C6

1. Introduction

Applying robust numerical techniques in option pricing and model calibration provides interesting research questions in financial markets. The techniques must be not only highly accurate, but also efficient.

Suppose we consider the well-known European vanilla option pricing formula driven by a stochastic stock price process $(S_t)_{t\geq 0}$:

$$\begin{aligned} V(x,K,t) &= e^{-r(T-t)}\mathbb{E}(U(S_T,K)|S_t=e^x) \\ &= e^{-r(T-t)}\int_{-\infty}^{+\infty} U(e^{x+\chi},K)g(\chi)\,\mathrm{d}\chi, \end{aligned} \quad (1)$$

where V denotes the option value at an initial date of t and with a strike price of K, U stands for a payoff function at the maturity of T, $\mathbb{E}$ is the expectation operator under the risk-neutral measure, x and χ are the log-price and state variable respectively, S_T can be decomposed into $S_t = e^x$ and e^χ, g is the probability density function of the process, and finally r is a risk-neutral interest rate.

In the aforementioned formula, V can be seen as a convolution integral, more precisely a cross-correlation integral, and the fast Fourier transform method (FFT method), a numerical integration-based method, (e.g. Carr and Madan 1999, Lewis 2001, Lipton 2002, Chourdakis 2004, Jackson *et al.* 2008, Lord *et al.* 2008), is a popular method for pricing European vanilla options as well as more exotic options, such as the American option, under Lévy processes. This is because the characteristic function of the underlying dynamics can be easily transformed into a risk-free probability density function (PDF) via the FFT method. The seminal work of these papers leads to the extension of combining the FFT with other transformation methods, e.g. the Hilbert transform or Gaussian transform, in pricing exotic options under the (time-changed) Lévy process or stochastic volatility models (e.g. Broadie and

*Corresponding author. Emails:
t.l.chan@uel.ac.uk; nickhale@sun.ac.za

Yamamoto 2003, 2005, Feng and Linetsky 2008, Cai and Kou 2011, Wong and Guan 2011, Zeng and Kwok 2014). Within the same numerical integration-based framework, numerical quadrature methods, such as composite Simpson's rule, are proposed mainly by Andricopoulos *et al.* (2003), O'Sullivan (2005), Andricopoulos *et al.* (2007), Chen *et al.* (2014) and Su *et al.* (2017). The authors of these papers abbreviate their quadrature techniques as QUAD methods. The original QUAD method was introduced in Andricopoulos *et al.* (2003), Andricopoulos *et al.* (2007) and requires the transition density to be known in closed form, which is the case in, e.g.the Black-Scholes model and Merton's jump-diffusion model. This requirement is relaxed in O'Sullivan (2005) where the QUAD-FFT or QUAD-CONV method is proposed. The main idea is that the PDF can be recovered by inverting the characteristic function via the FFT method. This helps open up the QUAD method to a much wider range of models. The latest development of the method (Su *et al.* 2017) is to improve the calculation speed, precomputing and caching PDFs and then applying the extrapolation and smoothing techniques to the method. Under the general framework of the QUAD method, the option Greeks, is formulated via the first-order finite difference method (FD) (Andricopoulos *et al.* 2003, Chen *et al.* 2014). However, the accuracy of the first-order FD is debatable in the context. In addition to the QUAD methods, in recent years, Pachón (2018) has introduced the CHEB method, Clenshaw-Curtis quadrature based on an expansion of the integrand in terms of Chebyshev polynomials, for approximate European options with arbitrary payoffs. The method is one of the natural applications of Chebfun (Trefethen *et al.* 2014), an open-source software system for numerical computing with functions.

Beyond the FFT method and the QUAD method, Oosterlee and his collaborators have attracted considerable attention (Leentvaar and Oosterlee 2008, Fang and Oosterlee 2009a, 2009b, 2011, Zhang and Oosterlee 2013, Ruijter *et al.* 2015). In their work, they adopt the Fourier 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 (such as European options), maintain an exponential convergence rate when pricing options. Moreover, these methods are also able to accurately price options under infinite variation processes. The COS method requires an adequate computational domain a priori, and, due to the recursion in time, errors caused by an inadequate domain propagate, resulting in incorrect option prices in the COS method. Moreover, based on the framework of the COS method, Oosterlee and his collaborators further apply the wavelet method called the Shannon wavelet inverse Fourier technique (SWIFT) method to price European, spread, path-dependent and discrete barrier options under exponential Lévy dynamics (e.g. Ortiz-Gracia and Oosterlee 2013, 2016, Colldeforns-Papiol *et al.* 2017). The SWIFT method is used to circumvent the ineffectiveness of using the COS method to

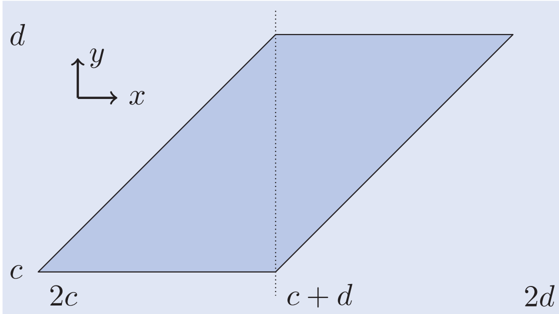

Figure 1. The convolution domain for two Legendre series on $[c, d]$.

price early-exercise options. Nevertheless, comparing with the COS method, a theoretical proof is still lacking to show exponential convergence for the SWIFT method since the wavelet scale m, a parameter to adjust the accuracy of the method, is still chosen heuristically for achieving exponential convergence.

In this manuscript we propose the CONLeg method, which uses Chebyshev and Legendre series to approximate a convolution, to improve some of ineffectiveness of using the aforementioned methods. First, as mentioned in Fang and Oosterlee (2009a), Fang and Oosterlee (2009b) and Chan (2018), the FFT-driven methods, like the CONV method (Lord *et al.* 2008), are computationally expensive to price option prices or to approximate the PDFs because a relatively large number of Fourier terms is required to obtain sufficient accuracy (see table 5 in Section 6). Second, we question whether any kind of quadrature method is an effective method to approximate option prices/Greeks when option pricing/-Greeks hedging formulae are treated as convolution integrals. Hale and Townsend (2014b, Section 2 and Section 6) suggest that since a convolution integral appears as a trapezoid (cf. figure 1), more quadrature weights and abscissae are required to approximate the areas towards the left and right vertices of the trapezoid. Third, we want to provide a method that can work with any stochastic process with or without a closed-form PDF. For example, the COS and SWIFT methods only work very well when the process has a characteristic function. Fourth, comparing against the CONV, COS, QUAD and SWIFT methods, the CONLeg method provides an option pricing/Greek curve rather than an individual point value. Fifth, through our numerical experiments, we show that the CONLeg method can achieve high accuracy in option pricing and hedging when the risk-free smooth probability density function (PDF) is non-smooth, and can be an effective method to price options/hedge their Greeks for long/short maturities. Sixth, to fit in the Chebfun framework, we lay out a closed-form transformation of the complex Fourier expression of a smooth PDF into Chebyshev series when the closed-form PDF is not available (cf. Appendix 1). In the similar fashion, we also provide a solution, the Fourier–Padé method (cf. Appendix 2), for approximating non-smooth PDFs. Seventh, through this paper, we are keen to promote Chebfun (Trefethen *et al.* 2014) in financial modelling. Chebfun is a robust, open source MATLAB package for computing numerically with functions, containing several state of the art algorithms for Chebyshev and other orthogonal polynomials. A distinct feature of Chebfun is that it works with adaptive approximations, in the sense the user specifies a tolerance (`chebfuneps`) and the degree of the discretisation is automatically adjusted during each computation so that this level of accuracy is satisfied. This is in contrast to the methods mentioned above, when the user typically selects a fixed discretisation size which must be manually adjusted to achieve the desired accuracy. We believe the adaptive setting is more suited to practical applications, although as we discuss in Section 6, it does make it difficult to measure convergence and provide quantitative comparisons between CONLeg and the existing methods. We note that there are also Chebfun-like software projects in other programming languages, e.g. ApproxFun (Olver and Townsend 2011) written in Julia and pychebfun (Swierczewski and Verdier 2011) implemented in Python, and it should be relatively straightforward to adapt the CONLeg method to these systems. Finally, the current CONLeg method is different to the previous literature prompted Chebyshev series and interpolants in option pricing (cf. Gaß *et al.* 2018, Pachón 2018). As we have mentioned in the previous point, comparing with the CHEB method (Pachón 2018), the CONLeg method is quadrature-free and not limited in pricing European-type options. Also, unlike Gaß *et al.*'s (2018) tensorized Chebyshev interpolation to computing Parametric Option Prices (POP), our method does not approximate option pricing curves which must be first precomputed by any numerical method, such as the Monte Carlo and the FFT methods, for some fixed parameter configurations, and then compute other option prices for arbitrary parameter constellations. On the contrary, for a set of fixed parameters, the CONLeg method directly generates an option pricing and Greek curve via approximating a convolution integral without applying other numerical methods for computing the integral first.

The remainder of this paper is structured as follows. Section 2 describes the algorithm for the convolution of Legendre series and how Legendre series can be efficiently computed using the relationship with Chebyshev series and the FFT. Section 3 introduces the financial stochastic models we examine in this paper. Section 4 describes the formulation of the CONLeg option pricing/Greek formulae for different styles of European options as well as Bermuda, American, and discrete monitored barrier options. Section 5 describes the choice of truncated integration intervals. Section 6 discusses, analyses, and compares the numerical results of the CONLeg method with those of other numerical methods discussed above. Finally, we conclude and discuss possible future developments in Section 7.

2. Convolution of Legendre series

Convolution is a fundamental operation that arises in many fields, particularly in financial derivatives research (cf. Carr and Madan 1999, Lewis 2001, Lipton 2002, Jackson *et al.* 2008, Lord *et al.* 2008), econometrics (Bondarenko 2003, Liu *et al.* 2016) and statistics (Hogg *et al.* 2004). Given two integrable functions, f and g, their convolution is a third function, h, defined formally by the integral

$$h(x) = (f * g)(x) = \int_{-\infty}^{+\infty} f(y)g(x-y)\,\mathrm{d}y. \tag{2}$$

In general, if both f and g are analytic (smooth) and periodic functions, the FFT method, which utilises the convolution theorem and the fast Fourier transform (FFT), is the best choice for approximating h as the FFT approximations of f and g do not suffer the Gibbs phenomenon.

If we now consider f and $g : [c,d] \to \mathbb{R}$ as two compactly supported outside of $[c,d]$, then the convolution aforementioned $h = f * g$ via the integral is given by

$$h(x) = (f * g)(x) = \int_{\min(d,x-c)}^{\max(c,x-d)} f(y)g(x-y)\,\mathrm{d}y, \quad x \in [2c, 2d], \tag{3}$$

and $h(x) = 0$ for $x \notin [2c, 2d]$† . Without losing any generality, we can visualise each value of x by the diagram in figure 1, and we split h into the two pieces suggested by the diagram:

$$h(x) = \begin{cases} h^L(x) = \displaystyle\int_c^{x-c} f(y)g(y-x)\,\mathrm{d}y & x \in [2c, c+d], \\ h^R(x) = \displaystyle\int_{x-d}^{d} f(y)g(y-x)\,\mathrm{d}y & x \in [c+d, 2d]. \end{cases} \tag{4}$$

Here, we denote, once and for all, L and R as the *left* and *right* hand side of the convolution, respectively.

Based in (3), either f or g can be a non-periodic continuous function, the FFT approximations of f and g suffer the Gibbs phenomenon such that there is a permanent oscillatory overshoot in the neighbourhoods of the endpoints c and d. Accordingly, to avoid the Gibbs phenomenon, we adopt the algorithm proposed by Hale and Townsend (2014a) to approximate h. The crucial idea of the algorithm is to approximate f and g with finite Legendre series and then convolve the approximations using the convolution theorem for them. The result is a piecewise polynomial representation that can be evaluated at any x in the domain of h to yield an approximation to $h(x)$. If the polynomials used to approximate f and g have degree at most N, their algorithm produces an approximation to h in $\mathcal{O}(N^2)$ operations. We summarise the approach below.

To illustrate Hale and Townsends' algorithm of a convolution of two Legendre series, we first define the Legendre series on $[-1, 1]$ and then generalise to intervals $[c, d]$. Legendre polynomials, invented by Adrien-Marie Legendre, are the polynomial solutions $P_n(x)$ to Legendre's differential equation

$$\frac{\mathrm{d}}{\mathrm{d}x}\left[\left(1-x^2\right)\frac{\mathrm{d}P_n(x)}{\mathrm{d}x}\right] + n(n+1), \quad x \in [-1, 1], \tag{5}$$

with $P_0(x) = 0, P_1(x) = 0$ and integer parameter $n \geq 0. P_n(x)$ forms a polynomial sequence of orthogonal polynomials of

† Since convolution is a commutative operation, we consider that only f and g are in the same intervals.

degree n and it can be expressed through Rodrigues' formula:

$$P_n(x) = \frac{1}{2^n n!}\frac{\mathrm{d}^n}{\mathrm{d}x^n}\left(x^2-1\right)^n. \tag{6}$$

Given a function $f(x)$ which is continuously differentiable on $[-1,1]$, then it has the uniformly convergent Legendre series expansion (Suetin 1964):

$$f(x) = \sum_{m=0}^{\infty} \alpha P_m(x), \quad \alpha_m = \left(m+\frac{1}{2}\right)\int_{-1}^{1} f(x)P_m(x)\,\mathrm{d}x.$$

If $f_M(x)$ denotes the truncated Legendre series expansion of degree M, i.e.

$$f_M(x) = \sum_{m=0}^{M} \alpha_m P_m(x),$$

then f_M converges geometrically to f as $M \to \infty$ if f is holomorphic in some neighbourhood of $[-1,1]$ and algebraically with order $O(M^{-(k+\frac{1}{2})})$ if $f, f', \ldots f^{(k-1)}$ are absolutely continuous on $[-1,1]$ and $f^{(k)}$ has bounded variation (cf. Wang and Xiang 2012, Theorem 2.5).

Suppose we have f_M and g_N as two finite Legendre series on $[-1,1]$ of degrees M and N with coefficients of $\alpha_0,\ldots,\alpha_M$ and $\beta_0,\ldots\beta_N$, then we may write f_M and g_N as

$$f_M(x) = \sum_{m=0}^{M}\alpha_m P_m(x), \quad g_N(x) = \sum_{n=0}^{N}\beta_n P_n(x). \tag{7}$$

With $[c,d]$ equal to $[-1,1]$, the convolution in (3) becomes

$$h(x) = (f_M * g_N)(x) = \int_{\max(-1,x-1)}^{\min(1,x+1)} f_M(y)g_N(x-y)\,\mathrm{d}y, \quad x\in[-2,2]. \tag{8}$$

From (4), h, consists of two pieces, h^L on the left with $[-2,0]$ and h^R on the right with $[0,2]$, each of degree $N+M+1$. We construct h^L and h^R by computing their Legendre coefficients. We focus on the computation of h^L since that of h^R is similar.

Denote by $\{\gamma_k^L\}_k^{M+N+1}$ the vector of the Legendre coefficients of h^L, such that

$$h^L(x) = \int_{-1}^{x+1} f_M(y)g_N(x-y)\,\mathrm{d}y = \sum_{k=0}^{M+N+1}\gamma_k^L P_k(x+1), \quad x\in[-2,0]. \tag{9}$$

By the orthogonality of Legendre polynomials and the orthonormalisation constant $(k+1/2)^{-1/2}$ for $P_k(x)$ for $k=0,\ldots,M+N+1$, we have for $P_k(x)$ for $k=0,\ldots,M+N+1$, that

$$\gamma_k^L = \frac{2k+1}{2}\int_{-2}^{0} P_k(x+1)\int_{-2}^{0} P_k(x+1) \times \int_{-1}^{x+1} f_M(y)g_N(x-y)\,\mathrm{d}y\,\mathrm{d}x \tag{10}$$

$$= \sum_{n=0}^{N}\beta_n \underbrace{\left[\frac{2k+1}{2}\sum_{m=0}^{M}\alpha_m\int_{-2}^{0}P_k(x+1)\int_{-1}^{x+1} P_m(y)P_n(x-y)\,\mathrm{d}y\,\mathrm{d}x\right]}_{=B_{k,n}^L}. \tag{11}$$

Hale and Townsend (2014a) prove that the above relation can be expressed in matrix form as $\underline{\gamma} = B^L\underline{\beta}$. Importantly (Hale and Townsend 2014a, Theorem 4.1), there is a three-term recurrence relation of $B_{k,n}^L$ such that

$$B_{k,n}^L = -\frac{2n+1}{2k+3}B_{k,n+1}^L + \frac{2n+1}{2k-1}B_{k,n+1}^L + B_{k,n-1}^L, \quad n,k\geq 1, \tag{12}$$

$$B_{k,1}^L = \begin{cases} B_{k-1,0}^L/(2k-1) - B_{k,0}^L - B_{k+1,0}^L/(2k+3), & k\neq 0,\\ -B_{1,0}^L/3, \quad k=0,\end{cases} \tag{13}$$

$$B_{k,0}^L = \begin{cases} \dfrac{\alpha_{k-1}}{2k-1} - \dfrac{\alpha_{k+1}}{2k+3}, & k\neq 0,\\ \alpha_0 - \alpha_1/3 \quad k=0,\end{cases} \tag{14}$$

where $0\leq k\leq M+N+1$ and $0\leq n\leq N$. both $B_{:,0}^L$ and $B_{:,1}^L$ can be computed in $\mathcal{O}(M+N)$ operations, and the whole $(M+N)\times N$ matrix, B^L, in $\mathcal{O}((M+N)N)$ operations. The matrix-vector product $B^L\underline{\beta}$ can be computed with the same cost and, accordingly, the coefficients γ^L of h^L in $\mathcal{O}((M+N)N)$ operations. The coefficients γ^R of h^R can be computed from $B^R\underline{\beta}$, for which a nearly identical recurrence relation can be derived left since the computation for h^R is similar.

Now, we direct our attention to the convolution of two finite Legendre series defined on the same interval $[c,d]$. We can define the composition of $P_k\circ\psi_{[c,d]}$, where $\psi_{[c,d]}(x) = (2x-(d+c))/(d-c)$ is the linear mapping from $[c,d]$ to $[-1,1]$. Apart from this, the Legendre series of f_M and g_N on $[c,d]$, respectively, are formulated by

$$f_M(x) = \sum_{m=0}^{M}\alpha_m P_m\circ\psi_{[c,d]}(x), \quad g_N(x) = \sum_{n=0}^{N}\alpha_n P_n\circ\psi_{[c,d]}(x) \tag{15}$$

Hale and Townsend (2014a, Lemma 4.2). The convolution of $(f*g)(x)$ of two continuous functions of f and g defined on $[c,d]$ can be computed as

$$(f*g)(x) = \int_{\min(d,x-c)}^{\max(c,x-d)} f(y)g(x-y)\,\mathrm{d}y = \frac{d-c}{2}\left((f\circ\psi_{[c,d]}^{-1}) * (g\circ\psi_{[c,d]}^{-1})\right)(y), \tag{16}$$

where $x\in[2c,2d]$ and $y = 2\psi_{[2c,2d]}(x)\in[-2,2]$.

The approach above is not restricted to f and g on the same intervals. In contrast, the algorithm allows f and g which are polynomials of finite degree with zero support outside of $[a, b]$ and $[c, d]$, respectively. In this case (3) becomes

$$h(x) = (f * g)(x) = \int_{\min(b,x-c)}^{\max(a,x-d)} f(y)g(x-y)\,\mathrm{d}y,$$
$$x \in [a+c, b+d]. \tag{17}$$

Depending on the difference between the length of $[a, b]$ and $[c, d]$, these are three different ideas to compute (17) on these subintervals of $[a+c, b+d]$. For the full detail of computing h on these subintervals, we refer interested readers to Hale and Townsend (2014a, Section 5). The algorithm of convolution of Legendre series supported on same/general intervals is fully implemented as `conv` in Chebfun†.

3. Lévy processes

In this section, we briefly introduce the important properties of one-dimensional Lévy processes. Standard references for the stochastic processes can be found in Schoutens (2003) and Cont and Tankov (2004). Since markets are frictionless and have no arbitrage, we assume that an equivalent martingale measure (EMM) $\mathbb{Q}$ is chosen by the market. Moreover, there is a complete filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}\}_{t\geq 0}, \mathbb{Q})$ on which all processes are assumed to live.

3.1. *Lévy processes*

With $r \geq 0$ and $q \geq 0$ as the constant risk-free interest rate and the constant dividend yield, respectively, we describe a stock process $(S_t)_{t\geq 0}$ driven by an exponential Lévy process $(X_t)_{t\geq 0}$ such that

$$S_t = S_0 e^{X_t},$$

where $X_0 = 0$ and X_t has infinitely divisible marginal distributions. Given a random variable X_t, we can define it as the corresponding characteristic function as follows:

$$\varphi(u) = \mathbb{E}[e^{iuX_t}] = e^{t\phi(u)}, \quad u \in \mathbb{R}. \tag{18}$$

If we define a truncation function $h(\chi) = \chi \mathbb{1}_{|\chi|\leq 1}$ which is a measurable function such that for every $u \in \mathbb{R}$, $\int |1 - e^{iu\chi} + iuh(\chi)|\nu(\mathrm{d}\chi) < \infty$, the characteristic function of X_t can be described by the Lévy–Khinchine representation such that

$$\phi(u) = iu(r - q + \omega)t - \frac{1}{2}\sigma^2 u^2 + \int_{-\infty}^{+\infty} \left(e^{iu\chi} - 1 - iuh(\chi)\right)\nu(\mathrm{d}\chi), \quad \chi \in X_{T-t}, \tag{19}$$

Here, $\sigma^2 \geq 0$ and ν are Lévy measures on $[-\infty, \infty]$ which do not depend on the choice of h (but note that $r - q + \omega$ depends on the choice of it). The condition that $(S_t e^{-(r-q)t})_{t\geq 0}$ is a martingale will be guaranteed as long as an appropriate choice of the mean-correcting compensator ω is calculated as follows:

$$\omega = \frac{1}{t}\log\varphi(-i) - (r - q). \tag{20}$$

There are a substantial number of Lévy process examples in financial modelling. In this paper, we focus on geometric Brownian motion (GBM), variance gamma (VG), normal inverse Gaussian (NIG) and the Carr–Geman–Madan–Yor process (CGMY). Their characteristic functions $\varphi(u)$ and closed-form pdf $g(x)$ can be found in Appendix 3 or e.g. Cont and Tankov (2004).

4. Pricing options and hedging Greeks via convolution of Legendre series

In this section, we apply the algorithm for the convolution of Legendre series to formulate option pricing/Greek formulae.

4.1. *Pricing formulae for European type options*

Given the current log price $x := \log S$, the strike price of K and maturity $T \geq t$, and the probability density function (PDF) g of a stochastic process, we can express the option price $V(x, K, t)$ starting at time t with its contingent claim paying out $U(S_T, K)$ as follows:

$$\begin{aligned} V(x, K, t) &= e^{-r(T-t)}\mathbb{E}(U(S_T, K)|S_t = e^x) \\ &= e^{-r(T-t)}\mathbb{E}(U(S_t e^{X_T - X_t}, K)) \\ &= e^{-r(T-t)} \int_{-\infty}^{+\infty} U(e^{x+\chi-\log K}, K)g(\chi)\,\mathrm{d}\chi, \\ \chi &\in X_T - X_t = X_{T-t}. \end{aligned} \tag{21}$$

By replacing $x + \chi - \log K$ with y, we have

$$\begin{aligned} V(x, K, t) &= e^{-r(T-t)} \int_{-\infty}^{+\infty} U(e^y, K)g\left(y - x + \log K\right)\,\mathrm{d}y \\ &= e^{-r(T-t)} K \int_{-\infty}^{+\infty} f(y)g^R\left(\tilde{x} - y\right)\,\mathrm{d}y, \end{aligned} \tag{22}$$

where $\tilde{x} = x - \log K$, $Kf(y) := U(e^y, K)$ is the pay-off in log-price coordinates and $g^R(\tilde{x}) := g(-\tilde{x})$ is the reflected PDF function.

If $f(y)$ is a piecewise continuous function, as is standard for most options, e.g. vanilla put and call, then applying the FFT method for a convolution function‡, $(f * g^R)(\tilde{x}) := \int_{-\infty}^{+\infty} f(y)g^R(\tilde{x} - y)\,\mathrm{d}y$, in $[-\infty, \infty]$ will cause Gibbs phenomenon and as a result, will affect the accuracy of approximating V. To avoid Gibbs phenomenon and allow a good approximation of V, we replace $[-\infty, \infty]$ with an interval

† The algorithm in `conv` actually computes the convolution between two Chebyshev series by using fast Chebyshev-Legendre transform (Townsend *et al.* 2018) implemented in `cheb2leg`.

‡ The expression of $\int_{-\infty}^{+\infty} U(e^y, K)g(y - x + \log K)\,\mathrm{d}y$ in (21) is indeed a cross-correlation integral; however, since we introduce the idea of the reflected function $g^R(\tilde{x}) := g(-\tilde{x})$, we can turn (21) into a convolution integral instead.

$[c,d]$ and employ the CONLeg method. The choice of $[c,d]$ satisfies the condition of

$$\int_c^d g(\chi)e^{iu\chi}\,\mathrm{d}\chi \approx \int_{-\infty}^{+\infty} g(\chi)e^{iu\chi}\,\mathrm{d}\chi = \mathbb{E}[e^{iu(X_T-X_t)}] := \varphi(u), \tag{23}$$

where $\varphi(u)$ is a characteristic function of $X_T - X_t$. Then we can approximate the pricing formula V on $[c,d]$, i.e.

$$\begin{aligned} V(x,K,t) &\approx e^{-r(T-t)}K\int_c^d f(y)g^R(\tilde{x}-y)\,\mathrm{d}y. \\ &= e^{-r(T-t)}K\int_{\min(d,\tilde{x}-c)}^{\max(c,\tilde{x}-d)} f(y)g^R(\tilde{x}-y)\,\mathrm{d}y, \\ &\quad \times \tilde{x}\in[2c,2d] \\ &= e^{-r(T-t)}Kh(\tilde{x}). \end{aligned} \tag{24}$$

where, $h(\tilde{x})$ has compact support outside of $[2c,2d]$. The final form of (24) is ready for approximating via the CONLeg method.

If a stochastic process has a characteristic function, we can adopt the ideas proposed in Chan (2016, 2018) to recover g by inverting the characteristic function in a complex Fourier series (CFS) representation such that g is approximated by:

$$g_N(y) := \mathfrak{Re}\left[2\sum_{k=1}^{N} b_k e^{i\frac{2\pi}{d-c}ky} + b_0\right], \tag{25}$$

where i is a complex number and $\mathfrak{Re}$ is the real part of a complex number, and given the condition of (23),

$$\begin{aligned} b_k &= \int_c^d g(y)e^{-i\frac{2\pi}{d-c}ky}\,\mathrm{d}y \approx \varphi\left(-\frac{2\pi}{d-c}k\right) \quad \text{and} \\ b_0 &= \int_c^d g(y)\,\mathrm{d}y \approx \varphi(0) = 1. \end{aligned} \tag{26}$$

For the expression of g^R, we simply put a negative sign in the basis function $e^{i\frac{2\pi}{d-c}ky}$, i.e.

$$g^R(y) \approx g_N^R(y) := \mathfrak{Re}\left[2\sum_{k=1}^{N} b_k e^{-i\frac{2\pi}{d-c}ky} + b_0\right]. \tag{27}$$

If g^R is smooth throughout on $[c,d]$, we can either directly approximate the CFS representation with a Chebyshev series using Chebfun (cf. Trefethen *et al.* 2014) or transform it into a Chebyshev series using the techniques shown in Appendix 1. If g^R is a piecewise continuous function containing a singularity†, then we use the Fourier–Padé ideas to locate the singularity in g^R and form accurate approximation. The details can be found in the Appendix 2.

Knowing singularities $\tilde{x}_1 \ldots \tilde{x}_{\mathcal{K}+1}$ in $[c,d]$, we divide f and g^R into a set of piecewise continuous functions and then approximate them with Chebyshev series using `chebfun` (cf. Trefethen *et al.* 2014, Chapter 1.4). Accordingly, we have a set of $\mathcal{K}$ polynomials f_M and g_N, each of degree at most M and N on the subintervals $[\tilde{x}_k, \tilde{x}_{k+1}]$, i.e.

$$f_M = \sum_{k=1}^{\mathcal{K}} f_{k,M}\mathbb{1}_{[\tilde{x}_k,\tilde{x}_{k+1}]}, \quad g_N^R = \sum_{k=1}^{\mathcal{K}} g_{k,N}^R\mathbb{1}_{[\tilde{x}_k,\tilde{x}_{k+1}]}. \tag{28}$$

Here, $\mathbb{1}_{[\tilde{x}_k,\tilde{x}_{k+1}]}$ as the indicator function in the interval $[\tilde{x}_k, \tilde{x}_{k+1}]$ and

$$\begin{aligned} f_{k,M}(\tilde{x}) &= \sum_{m=0}^{M} \alpha_{k,m}^{cheb} T_m \circ \psi_{[\tilde{x}_k,\tilde{x}_{k+1}]}(\tilde{x}), \\ g_{k,N}^R(x) &= \sum_{n=0}^{N} \beta_{k,n}^{cheb} T_n \circ \psi_{[\tilde{x}_k,\tilde{x}_{k+1}]}(\tilde{x}). \end{aligned} \tag{29}$$

Then, using the techniques implemented in `cheb2leg` (cf. Townsend *et al.* 2018), we transform both f_N and g_M^R into Legendre series defined by

$$\begin{aligned} f_{k,M}(\tilde{x}) &= \sum_{m=0}^{M} \alpha_{k,m}^{leg} P_m \circ \psi_{[\tilde{x}_k,\tilde{x}_{k+1}]}(\tilde{x}), \\ g_{k,N}^R(\tilde{x}) &= \sum_{n=0}^{N} \beta_{k,n}^{leg} P_n \circ \psi_{[\tilde{x}_k,\tilde{x}_{k+1}]}. \end{aligned} \tag{30}$$

We use the algorithm for the convolution of Legendre series described in Section 2 to approximate their convolution $h(\tilde{x}) = (f_M * g_N)(\tilde{x})$ on the subintervals $[\tilde{x}_k, \tilde{x}_{k+1}]$. Finally, transforming $h(\tilde{x})$ back into Chebyshev series using `leg2cheb`, V can be approximated by

$$e^{-r(T-t)}Kh(\tilde{x}) = e^{-r(T-t)}K\,(f_M * g_N)\,(\tilde{x}) \tag{31}$$

$$= e^{-r(T-t)}K\sum_{k=1}^{M_k} V_{N_k}\mathbb{1}_{[\tilde{x}_k,\tilde{x}_{k+1}]} \tag{32}$$

where,

$$V_{N_k} = \sum_{k=1}^{N_k} \gamma_k T_k \circ \psi_{[\tilde{x}_k,\tilde{x}_{k+1}]}(\tilde{x}) \tag{33}$$

Using (31), we can generate a set of option prices with a value of K and a range of S_t. However, in the financial markets, option price quotes always appear with a value of S_t and a range of K. To fit in this financial phenomenon, we modify (31) using the fact of $K = Se^{-\tilde{x}} = e^{x-\tilde{x}}$ so that we obtain the new pricing formula of

$$V(x,K,t) = e^{-r(T-t)+x-\tilde{x}}h(\tilde{x}). \tag{34}$$

Remark 1 Without applying the adaptive procedure in Chebfun, if the payoff function and the PDF are sufficiently smooth (piecewise analytic) then we expect exponential convergence in the CONLeg method (cf. Section 2).

Remark 2 In Chebfun, there is a built-in algorithm to detect singularities automatically in a piecewise continuous function (cf. Pachón *et al.* 2010). However, since PDFs can be easily expressed in the complex Fourier series and then extend to the Fourier–Padé series (cf. Chan 2016, 2018), we instead use Fourier–Padé ideas to locate singularities or approximate PDFs.

† We refer a singularity as a point at which is not defined, or a point which fails to be well-behaved after differentiability

Table 1. Payoff functions and their transforms for a variety of financial contingency claims.

Financial contingency claim	Payoff function $U(S_T, K)$	Transformed payoff function $U(e^{x+\chi-\log K}, K)$
Call	$\max(S_T - K, 0)$	$K\max(e^{x+\chi-\log K} - 1, 0)$
Put	$\max(K - S_T, 0)$	$K\max(1 - e^{x+\chi-\log K}, 0)$
Covered Call	$\min(S_T, K)$	$K\min(e^{x+\chi-\log K} - 1, 0) + K$
Cash-or-Nothing Call	$\mathbb{1}_{S_T \geq K}$	$\mathbb{1}_{e^{x+\chi-\log K} \geq 1}$
Cash-or-Nothing Put	$\mathbb{1}_{S_T \leq K}$	$\mathbb{1}_{e^{x+\chi-\log K} \leq 1}$
Asset-or-Nothing Call	$S_T \mathbb{1}_{S_T \geq K}$	$e^{x+\chi}\mathbb{1}_{e^{x+\chi-\log K} \geq 1}$
Asset-or-Nothing Put	$S_T \mathbb{1}_{S_T \leq K}$	$e^{x+\chi}\mathbb{1}_{e^{x+\chi-\log K} \leq 1}$
Asymmetric Call	$(S_T^n - K^n)\mathbb{1}_{S_T \geq K}$	$K^n(e^{n(x+\chi-\log K)} - 1)\mathbb{1}_{e^{x+\chi-\log K} \geq 1}$
Asymmetric Put	$(K^n - S_T^n)\mathbb{1}_{S_T \leq K}$	$K^n(1 - e^{n(x+\chi-\log K)})\mathbb{1}_{e^{x+\chi-\log K} \leq 1}$

Notes: $\mathbb{1}$ represents an indicator function and $n < \infty$ is any positive integer. The singularity always exits at $y = 0$ in the transformed payoff function when $y = x + \chi - \log K$.

4.1.1. European vanilla call options as illustration. We now consider pricing a European vanilla call, which can be exercised only at its maturity, defined in (21) with a payoff function of

$$U(S_T, K) = \max(S_T - K). \tag{35}$$

We first transform the payoff into

$$\max\left(e^{x+\chi} - K, 0\right) = K\max\left(e^{x+\chi-\log K} - 1, 0\right). \tag{36}$$

By replacing $x + \chi - \log K$ with y, we have a new form of $V(x, K, t)$ denoted as

$$\begin{aligned} V(x,K,t) &= e^{-r(T-t)}K\int_{-\infty}^{\infty}\max(e^y - 1, 0)g(y - x + \log K)\,\mathrm{d}y \\ &= e^{-r(T-t)}K\int_{-\infty}^{\infty}\max(e^y - 1, 0)g^R(\tilde{x} - y)\,\mathrm{d}y, \end{aligned} \tag{37}$$

where $\tilde{x} = x - \log K$ and $g^R(\tilde{x}) := g(-\tilde{x})$ is a reflecting function. To make the CONLeg more efficient, we define a truncated computational interval $[c, d]$ (cf. Section 5), which satisfies condition (23) to replace $[-\infty, \infty]$. Then, $V(x, K, t)$ is reformulated as

$$\begin{aligned} V(x,K,t) &\approx e^{-r(T-t)}K\int_c^d \max(e^y - 1, 0)g^R(\tilde{x} - y)\,\mathrm{d}y \\ &= e^{-r(T-t)}K\int_{\min(d,x-c)}^{\max(c,x-d)} f(y)g^R(\tilde{x} - y)\,\mathrm{d}y, \\ &\quad \tilde{x} \in [2c, 2d] \\ &= e^{-r(T-t)}Kh(\tilde{x}), \end{aligned} \tag{38}$$

where $f(y) := \max(e^y - 1, 0)$. To easily digest how the CONLeg method approximates $h(\tilde{x})$, we assume that g^R is a piecewise smooth function containing only one jump $y = 0$ appearing in f, the payoff function, on $[c, d]$. We can use `chebfun` to approximate f and g^R on $[c, 0, d]$† and $[c, d]$ respectively. Using the techniques described in Section 4.1, the European call option pricing formula is given by:

$$\begin{aligned} V(x,K,t) &\approx e^{-r(T-t)}K(f_M * g_N)(\tilde{x}) \\ &= e^{-r(T-t)}K\sum_{k=1}^{4} V_{N_k}\mathbb{1}_{[\tilde{x}_k, \tilde{x}_{k+1}]} \quad \tilde{x} \in [2c, 2d], \end{aligned} \tag{39}$$

where, $V_{N_k} = \sum_{k=1}^{N_k}\gamma_k T_k \circ \psi_{[\tilde{x}_k, \tilde{x}_{k+1}]}(\tilde{x})$, and $\tilde{x}_1 = 2c, \tilde{x}_2 = c, \tilde{x}_3 = 0, \tilde{x}_4 = d, \tilde{x}_5 = 2d$.

Moreover, if we only focus on $[c, d]$, we have

$$V(x,K,t) = e^{-r(T-t)}K\left(V_{N_1}\mathbb{1}_{[c,0]} + V_{N_2}\mathbb{1}_{[0,d]}\right) \quad \tilde{x} \in [c, d], \tag{40}$$

where $\tilde{x}_1 = c, \tilde{x}_2 = 0, \tilde{x}_3 = d$. See Section 6.1 for an example.

The CONLeg method is not limited from pricing European vanilla call option (38); it can be readily extended into a put option or other options with different pay-off structures, e.g. Cash-or-Nothing options. In table 1, we list all financial contingency claims we consider in this paper and both their payoff functions and transformed payoff functions.

4.2. Pricing formulae for Bermuda options

Consider now $\log S_t := x_t$ driven by a Lévy process and a Bermudan option with strike K and maturity T that can be exercised only on a given number of exercise dates $t = t_0 < t_1 \leq t_2 \leq \cdots t_l \leq t_{l+1} \leq \cdots \leq t_L = T$. We can write the Bermudan pricing formula for such an option as

$$V(x_{t_l}, K, t_l) = \begin{cases} U(e^{x_{t_l}}, K, t_l) & l = L,\ t_L = T \\ \max\left(C(x_{t_l}, K, t_l), U(e^{x_{t_l}}, K, t_l)\right) & l = 1, 2, 3, \ldots, L-1\,, \\ C(x_{t_l}, K, t_l) & l = 0 \end{cases} \tag{41}$$

where, $U(e^{x_{t_l}}, K, t_l)$ is the payoff function at t_l. That is, if the payoff function is a call, then $U(e^{x_{t_l}}, K, t_l)$ is transformed into $\max(e^{x_{t_l}} - K, 0)$. In (41), $C(x_{t_l}, K, t_l)$ at each t_j can be defined

† One should note that when $y \leq 0$, $\max(e^y - 1, 0) = 0$.

as

$$C(x_{t_j},K,t_j) = e^{-r(t_{j+1}-t_j)}\mathbb{E}\left(V(x_{t_{j+1}},K,t_{j+1})|x_{t_j}\right). \tag{42}$$

To apply the CONLeg method to approximate $C(x_{t_l},K,t_l)$, we first understand that

$$\begin{aligned} S_{t_l} &= e^{-r(t_{l+1}-t_l)}\mathbb{E}\left(S_{t_{l+1}}|S_{t_l}=e^{x_{t_l}}\right)\\ &= e^{-r(t_{l+1}-t_l)}\mathbb{E}\left(e^{x_{t_l}+X_{t_{l+1}}-X_{t_l}}\right) = e^{x_{t_l}} \end{aligned} \tag{43}$$

is a martingale process. We also denote $\tilde{x}_{t_l}$ as $x_{t_l} - \log K$ and follow Section 4.1 to approximate $C(x_{t_l},K,t_l)$ as European option prices at t_l. Then we can transform $C(x_{t_l},K,t_l)$ into

$$\begin{aligned} &= e^{-r(t_{l+1}-t_l)}\mathbb{E}\left(V(x_{t_{l+1}},K,t_{l+1})|x_{t_l}\right)\\ &= e^{-r(t_{l+1}-t_l)}\int_{-\infty}^{+\infty} V(x_{t_l}+\chi-\log K,t_{l+1})g(\chi)\,\mathrm{d}\chi,\\ &\quad \chi\in X_{t_{l+1}}-X_{t_l}\\ &= e^{-r(t_{l+1}-t_l)}K\int_{\min(d,x-c)}^{\max(c,x-d)} f(y)g^R(\tilde{x}_{t_l}-y)\,\mathrm{d}y\\ &= e^{-r(t_{l+1}-t_l)}Kh(\tilde{x}_{t_l}). \end{aligned} \tag{44}$$

Since we approximate $h(\tilde{x}_{t_l})$ with the CONLeg method and use the Chebyshev series to present the Bermuda option prices $C(x_{t_l},K,t_l)$, accordingly, we can further modify (41) with a new form of

$$V(x_{t_l},K,t_l) = \begin{cases} K\tilde{f}(\tilde{x}_{t_l}) & l=L,\ t_L=T\\ K\max\left(e^{-r(t_{l+1}-t_l)}h(\tilde{x}_{t_l}),\tilde{f}(\tilde{x}_{t_l})\right) & l=1,2,3,\dots,L-1\\ Ke^{-r(t_{l+1}-t_l)}h(\tilde{x}_{t_l}) & l=0 \end{cases}, \tag{45}$$

where $K\tilde{f}(\tilde{x}_{t_l}) := U(e^{x_{t_l}},K,t_l)$. Since the no-arbitrage assumption leads to the requirement that $\partial V/\partial x$ is continuous and $V(x_{t_l},K,t_l) = U(e^{x_{t_l}},K,t_l)$ at the early exercise curve, we must determine exercise point $x^*_{t_l}$ appearing in $V(x_{t_l},K,t_l) = U(e^{x_{t_l}},K,t_l)$. One way to do this is to use the Newton method proposed in Fang and Oosterlee (2009b) to find x_{t_l}. However, since V and U are represented by piecewise smooth polynomials (`chebfun`), we can apply a built-in function `roots` in Chebfun to efficiently find these zeros. To do so, we first approximate $\tilde{f}(\tilde{x}_{t_l})$ as a Chebyshev series, then apply the roots function to find $\tilde{x}^*_{t_l}$ in the following equality:

$$e^{-r(t_{l+1}-t_l)}h(\tilde{x}_{t_l}) - \tilde{f}(\tilde{x}_{t_l}) = 0.$$

Once we have $\tilde{x}^*_{t_l}$ and use it as a break point, we approximate

$$\max\left(e^{-r(t_{l+1}-t_l)}h(\tilde{x}_{t_l}),f(\tilde{x}_{t_l})\right) \tag{46}$$

with two different Chebyshev series. Considering $\tilde{x}^*_{t_l}$ as a singularity and combining other singularities, $\tilde{x}_{t_l,1}\dots\tilde{x}_{t_l,\mathcal{K}+1}\in\tilde{x}_{t_l}$, e.g. singularities in a non-smooth PDF and/or in a payoff function, in $V(x_{t_l},K,t_l)$, we approximate $V(x_{t_l},K,t_l)$, with a set of Chebyshev series given by

$$K\sum_{k=0}^{M_k} V_{N_k}\mathbb{1}_{[\tilde{x}_{t_l,k},\tilde{x}_{t_l,k+1}]} \quad\text{and}\quad V_{N_k} = \sum_{k=1}^{N_k}\gamma_k T_k\circ\psi_{[\tilde{x}_k,\tilde{x}_{k+1}]}(\tilde{x}). \tag{47}$$

Finally, summarising the methods above, we present the pseudo-code of our algorithm computing Bermudan option prices in Algorithm 1. A numerical example is also presented in Section 6.2.

Result: Bermuda option price $V(x_t,K,t)$ at time t
initialisation;
discretise $[t,T]$ into timesteps
  $t=t_0,t_1,\dots,t_l,\dots,t_L=T$;
$t_l = t_{L-1}$;
while $t_l \neq t$ **do**
  compute $C(x_{t_l},K,t_l)$ using the CONLeg method;
  $C(x_{t_l},K,t_l) = e^{-r(t_{l+1}-t_l)}h(\tilde{x}_{t_l})$ in (47);
  find $\tilde{x}^*_{t_l}$ in $e^{-r(t_{l+1}-t_l)}h(\tilde{x}_{t_l}) - \tilde{f}(\tilde{x}_{t_l}) = 0$;
  compute $\max\left(e^{-r(t_{l+1}-t_l)}h(\tilde{x}_{t_l}),\tilde{f}(\tilde{x}_{t_l})\right)$ with two Chebyshev series (49);
  $V(x_{t_l},K,t_l) = K\sum_{k=0}^{M_k} V_{N_k}\mathbb{1}_{[\tilde{x}_{t_l,k},\tilde{x}_{t_l,k+1}]}$;
  next t_l;
end
return $V(x_t,K,t)$ equal to $e^{-r(t_1-t)}Kh(\tilde{x}_t)$, where $t_0=t$;

Algorithm 1: Algorithm for computing Bermudan option price $V(x_t,K,t)$ at time t based on (41).

4.3. Pricing formulae for American options

There are two basic approaches to evaluating American options based on our method 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, increase the number of exercise opportunities L to a very large value. An alternative approach is to use Richardson extrapolation on a series of Bermudan options with an increasing number of L (cf. Geske and Johnson 1984, Chang *et al.* 2007). We adapt the latter approach (which is also implemented in Fang and Oosterlee 2009b) to price the American option here. Therefore, implementing the 4-point Richardson extrapolation scheme (cf. Fang and Oosterlee 2009b), we have the American option price given by

$$\begin{aligned} &V_{Amer}(L)\\ &\quad = \frac{1}{21}\left(64V(2^{L+3}) - 56V(2^{L+2}) + 14V(2^{L+1}) - V(2^L)\right), \end{aligned} \tag{48}$$

where $V_{Amer}(L)$ denotes the approximated value of the American option and $V(\cdot)$ is the pricing formulae for Bermudan options in (45).

4.4. Pricing formulae for discretely monitored 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 or down-and-out option when the option's existence fades out after crossing 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 only investigate two basic types of barrier options: down-and-out and up-and-out barrier options.

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, 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 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 is reached; otherwise, a rebate occurs when a barrier option is knocked out.

We use a rebate DO option to illustrate the CONLeg method to approximate discretely monitored barrier option prices. Suppose that we have a rebate DO option driven by S_t with a barrier B, a rebate R_b, a strike K and a series of monitoring dates L: $t = t_0 < \cdots < t_l < \cdots < t_L = T$; the option formulae can be described as

$$V(x_{t_l}, K, t_l) = \begin{cases} U(e^{x_{t_l}}, K, t_l)\mathbb{1}_{\log B > x_{t_l}} \\ \quad + R_b \mathbb{1}_{\log B \le x_{t_l}} & l = L,\ t_L = T \\ C(x_{t_l}, K, t_l)\mathbb{1}_{\log B > x_{t_l}} \\ \quad + e^{-r(T-t_l)} R_b \mathbb{1}_{\log B \le x_{t_l}} & l = 1, \ldots, L-1 \\ C(x_{t_l}, K, t_l) & l = 0 \end{cases}, \tag{49}$$

where $\mathbb{1}$ is an indicator function and $U(e^{x_{t_l}}, K, t_l)$ is again either a call or put payoff. We follow the ideas of (44) and (45) in Section 4.2 to approximate $C(x_{t_l}, K, t_l)$ such that

$$e^{-r(t_{l+1}-t_l)} \mathbb{E}\left(V(x_{t_{l+1}}, K, t_{l+1}) | x_{t_l}\right) = e^{-r(t_{l+1}-t_l)} K h(\tilde{x}_{t_l}). \tag{50}$$

After we apply the CONLeg method, (49) can be transformed into

$$V(x_{t_l}, K, t_l) = \begin{cases} K\left(\tilde{f}(\tilde{x}_{t_l})\mathbb{1}_{\log(B/K) > \tilde{x}_{t_l}} + \dfrac{R_b}{K}\mathbb{1}_{\log(B/K) \le x_{t_l}}\right) & l = L,\ t_L = T \\ K\left(e^{-r(t_{l+1}-t_l)} h(\tilde{x}_{t_l})\mathbb{1}_{\log(\frac{B}{K}) > \tilde{x}_{t_l}} \right. \\ \quad \left. + e^{-r(T-t_l)} \dfrac{R_b}{K} \mathbb{1}_{\log(\frac{B}{K}) \le x_{t_l}}\right) & l = 1, \ldots, L-1 \\ K e^{-r(t_{l+1}-t_l)} h(\tilde{x}_{t_l}) & l = 0. \end{cases} \tag{51}$$

In (51), since there is a jump at $\log(B/K)$, the barrier, at t_l, we use $\log(B/K)$ as a break point and approximate

$$e^{-r(t_{l+1}-t_l)} h(\tilde{x}_{t_l}) \mathbb{1}_{\log(\frac{B}{K}) > \tilde{x}_{t_l}} + e^{-r(T-t_l)} \frac{R_b}{K} \mathbb{1}_{\log(\frac{B}{K}) \le x_{t_l}} \tag{52}$$

with two Chebyshev series. Moreover, combining other singularities $\tilde{x}_{t_l,1} \ldots \tilde{x}_{t_l,\mathcal{K}+1} \in \tilde{x}_{t_l}$ in $V(x_{t_l}, K, t_l)$, we can formulate $V(x_{t_l}, K, t_l)$ with a set of Chebyshev series given in (47). Finally, the pseudo-code of our algorithm calculating discretely monitored DO barrier option prices can be found in Algorithm 2.

Result: discretely monitored barrier option price $V(x_t, K, t)$ at time t

initialisation discretise $[t, T]$ into timesteps $t = t_0, t_1, \ldots, t_l, \ldots, t_L = T$ $t_l = t_{L-1}$ **while** $t_l \neq t$ **do**

compute $C(x_{t_l}, K, t_l)$ using the CONLeg method $C(x_{t_l}, K, t_l) = e^{-r(t_{l+1}-t_l)} h(\tilde{x}_{t_l})$ in (51) compute $\begin{cases} e^{-r(t_{l+1}-t_l)} h(\tilde{x}_{t_l}) & \text{if } \log(B/K) > \tilde{x}_{t_l} \\ e^{-r(T-t_l)} R_b/K & \text{if } \log(B/K) \le \tilde{x}_{t_l} \end{cases}$ with two Chebyshev series (47)

$V(x_{t_l}, K, t_l) = K \sum_{k=0}^{M_k} V_{N_k} \mathbb{1}_{[\tilde{x}_{t_l,k}, \tilde{x}_{t_l,k+1}]}$ next t_l

end

return $V(x_t, K, t)$ equal to $e^{-r(t_1-t)} K h(\tilde{x}_t)$, where $t_0 = t$

Algorithm 2: Algorithm for computing discretely monitored barrier option price $V(x_t, K, t)$ at time t based on (49).

For the UO barrier options, we can use (51) and Algorithm 2 to compute their prices, but we consider the condition of the option knocked out when the stock price rises above B, i.e.

$$V(x_{t_l}, K, t_l) = \begin{cases} U(e^{x_{t_l}}, K, t_l)\mathbb{1}_{\log B < x_{t_l}} \\ \quad + R_b \mathbb{1}_{\log B \ge \tilde{x}_{t_l}} & l = L,\ t_L = T \\ C(x_{t_l}, K, t_l)\mathbb{1}_{\log B < x_{t_l}} \\ \quad + e^{-r(T-t_l)} R_b \mathbb{1}_{\log B \ge \tilde{x}_{t_l}} & l = 1, \ldots, L-1 \\ C(x_{t_l}, K, t_l) & l = 0 \end{cases}. \tag{53}$$

4.5. Greek formulae

We now turn our attention to deriving the option Greek values. In particular, we focus on deriving three option Greek values—Delta (Δ), Gamma (Γ), and Vega. Delta is defined as the rate of change in the option value with respect to changes in the underlying asset price; Gamma is the rate of change of Delta 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 is often based on changes in the recent, historical prices of a trading instrument. Other Greek values, such as Theta, can be derived in a similar fashion;

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

As mentioned above, Delta is the first derivative of the value V of the option with respect to the underlying instrument price S. Hence, differentiating the convolution form of V in European options (31), Bermuda options (47), American options (48) and barrier options (47) with respect to S, we have

$$\Delta_t = \frac{\partial V(x,K,t)}{\partial S} = e^{-r(T-t)}K\frac{\partial h(\tilde{x})}{\partial \tilde{x}}\frac{\partial \tilde{x}}{\partial x}\frac{\partial x}{\partial S}, \quad \tilde{x} = x - \log K. \tag{54}$$

Since $\partial x/\partial \tilde{x} = 1$ and $\partial x/\partial S = \exp(-x)$, Δ_t simply becomes

$$e^{-r(T-t)-x}K\frac{\partial h(\tilde{x})}{\partial \tilde{x}} = e^{-r(T-t)-x}K\sum_{k=1}^{M_k}\frac{\partial V_{N_k}}{\partial \tilde{x}}\mathbb{1}_{[\tilde{x}_k,\tilde{x}_{k+1}]}, \tag{55}$$

with

$$\frac{\partial V_{N_k}}{\partial \tilde{x}} = \sum_{k=1}^{N_k}\gamma_k\frac{\partial T_k \circ \psi_{[\tilde{x}_k,\tilde{x}_{k+1}]}(\tilde{x})}{\partial \tilde{x}}. \tag{56}$$

To express the first derivative of the Chebyshev series in (56), we adopt the fact of

$$\frac{\mathrm{d}}{\mathrm{d}x}T_n(x) = \frac{n}{2}\frac{T_{n-1}(x) - T_{n+1}(x)}{1-x^2}$$

(cf. Mason and Handscomb 2002, (2.4.5)) and $\mathrm{d}\psi_{[c,d]}(\tilde{x}) = (2/(d-c))\,\mathrm{d}\tilde{x}$, such that we have

$$\frac{\partial T_k \circ \psi_{[\tilde{x}_k,\tilde{x}_{k+1}]}(\tilde{x})}{\partial \tilde{x}} = \frac{2k}{d-c}\frac{T_{k+1}\circ \psi_{[\tilde{x}_k,\tilde{x}_{k+1}]}(\tilde{x}) - T_{k-1}\circ \psi_{[\tilde{x}_k,\tilde{x}_{k+1}]}(\tilde{x})}{1-\tilde{x}^2}. \tag{57}$$

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

$$\begin{aligned}\Gamma_t &= \frac{\partial^2 V(x,K,t)}{\partial S^2} = \frac{\partial \Delta_t}{\partial S} = \frac{\partial \Delta_t}{\partial x}\frac{\partial x}{\partial S}\\ &= e^{-r(T-t)-2x}K\left(\frac{\partial^2 h(\tilde{x})}{\partial^2 \tilde{x}} - \frac{\partial h(\tilde{x})}{\partial \tilde{x}}\right)\\ &= e^{-r(T-t)-2x}K\left(\sum_{k=1}^{M_k}\frac{\partial^2 V_{N_k}}{\partial^2 \tilde{x}}\mathbb{1}_{[\tilde{x}_k,\tilde{x}_{k+1}]} - \sum_{k=1}^{M_k}\frac{\partial V_{N_k}}{\partial \tilde{x}}\mathbb{1}_{[\tilde{x}_k,\tilde{x}_{k+1}]}\right).\end{aligned} \tag{58}$$

To find $\partial^2 V_{N_k}/\partial^2 \tilde{x}$, we may use

$$\frac{\mathrm{d}^2}{\mathrm{d}x^2}T_n(x) = \frac{n}{4}\frac{(n+1)T_{n-2}(x) - 2nT_n(x) + (n-1)T_{n+2}(x)}{(1-x^2)^2}$$

(cf. Mason and Handscomb 2002, Problem 2.5.17), and thus find

$$\frac{\partial^2 V_{N_k}}{\partial^2 \tilde{x}} = \sum_{k=1}^{N_k}\gamma_k\frac{\partial^2 T_k \circ \psi_{[\tilde{x}_k,\tilde{x}_{k+1}]}(\tilde{x})}{\partial^2 \tilde{x}} \tag{59}$$

and

$$\begin{aligned}&\frac{\partial^2 T_k \circ \psi_{[\tilde{x}_k,\tilde{x}_{k+1}]}(\tilde{x})}{\partial^2 \tilde{x}}\\ &= \frac{k}{(d-c)(1-\tilde{x}^2)^2}\Big((k+1)T_{k-2}\circ \psi_{[\tilde{x}_k,\tilde{x}_{k+1}]}(\tilde{x})\\ &\quad - 2kT_k\circ \psi_{[\tilde{x}_k,\tilde{x}_{k+1}]}(\tilde{x}) + (k-1)T_{k+2}\circ \psi_{[\tilde{x}_k,\tilde{x}_{k+1}]}(\tilde{x})\Big).\end{aligned} \tag{60}$$

Likewise, we can obtain the formula for Vega, $\frac{\partial V}{\partial \sigma_t}$, where σ_t is the initial value of the volatility at time t. For example, for the GBM model with σ_t as the initial value of the volatility, we derive Vega as follows:

$$\frac{\partial V(x,K,\sigma_t,t)}{\partial \sigma_t} = e^{-r(T-t)}K\int_{\min(d,\tilde{x}-c)}^{\max(c,\tilde{x}-d)} f(y)\frac{\partial g^R(\tilde{x}-y)}{\partial \sigma_t}\,\mathrm{d}y, \quad \tilde{x} \in [2c, 2d]. \tag{61}$$

After we differentiate V with respect to σ_t to obtain (61), we can approximate (61) with the CONLeg method.

If the closed-formed PDF g of the stochastic process does not exist, as we mentioned before, we express g^R with the CFS expression such that

$$\frac{\partial g^R(\tilde{x}-y)}{\partial \sigma_t} = \mathfrak{Re}\left[2\sum_{k=1}^{N}\frac{\partial b_k}{\partial \sigma_t}e^{-i\frac{2\pi}{d-c}k(\tilde{x}-y)}\right], \quad \text{and} \quad \frac{\partial b_k}{\partial \sigma_t} = \frac{\partial \varphi(-\frac{2\pi}{d-c}k,\sigma_t)}{\partial \sigma_t}, \tag{62}$$

where φ contains the parameter σ_t.

5. Choice of truncated intervals

In this section, we adopt the ideas of Fang and Oosterlee (2009a) and Chan (2018) to choose the interval $[c, d]$. The choice of the interval $[c, d]$ plays the crucial role in the accuracy of the CONLeg method. If the choice of $[c, d]$ is too big, the CONLeg method can perform inefficiently. On the contrast, if $[c, d]$ is too small, the CONLeg method can produce inaccurate option prices/hedging values. Accordingly, a minimum and substantial interval $[c, d]$ can be chosen to capture most of the mass of a PDF such that our algorithm can in turn yield the highest accuracy. In this short section, we show how to construct an interval related to the closed-form formulas of stochastic process cumulants. The idea of using the cumulants is first proposed by Fang and Oosterlee (2009a) to construct the definite interval $[c, d]$ in (23). Based on their ideas, we have the following expression for $[c, d]$:

$$\begin{aligned} d &= \left|c_1 + L_n\sqrt{c_2 + \sqrt{c_4}}\right| \\ c &= -d, \end{aligned} \tag{63}$$

where c_1, c_2, and c_4 are the first, second and fourth cumulants, respectively, of the stochastic process and $L_n \in [8, 12]$.

Table 2. The first, second, and fourth cumulants of various models.

Lévy models	cumulants
BS	$c_1 = (r - q + \omega)t \quad c_2 = \sigma^2 t, \quad c_4 = 0, \omega = -0.5\sigma^2$
NIG	$c_1 = (r - q + \omega)t + \delta t\beta/\sqrt{\alpha^2 - \beta^2}$ $c_2 = \delta t\alpha^2(\alpha^2 - \beta^2)^{-3/2}$ $c_4 = \delta t\alpha^2(\alpha^2 + 4\beta^2)^{-3/2}(\alpha^2 - \beta^2)^{-7/2}$ $\omega = -0.5\sigma^2 - \delta(\sqrt{\alpha^2 - \beta^2} - \sqrt{\alpha^2 - (\beta+1)^2})$
VG	$c_1 = (r - q + \theta + \omega)t$ $c_2 = (\sigma^2 + \upsilon\theta^2)t$ $c_4 = 3(\sigma^4\upsilon + 2\theta^4\upsilon^3 + 4\sigma^2\theta^2\upsilon^2)t$ $\omega = 1/\upsilon \log(1 - \theta\upsilon - \sigma^2\upsilon/2)$
CGMY	$c_1 = (r - q + \omega)t$ $c_2 = (C\Gamma(2 - Y)(M^{Y-2} + G^{Y-2})t$ $c_4 = (C\Gamma(4 - Y)(M^{Y-4} + G^{Y-4})t$ $\omega = (C\Gamma(-Y)G^Y((1 + \frac{1}{G})^Y - 1 - \frac{Y}{G}) + C\Gamma(-Y)M^Y((1 - \frac{1}{M})^Y - 1 + \frac{Y}{M}))$

For simple, less-complicated financial models, we also obtain closed-form formulas for c_1, c_2, and c_4, which are shown in table 2.

In general, the truncated intervals in (63) work for smooth/non-smooth PDFs with/without singularities. However, as the cumulants in each process in table 2 contain t, we notice that if t is too small, the cumulants shrink and make $[c, d]$ too small for the CONLeg to produce accurate option prices/hedging values. To solve the problem heuristically, as long as t is less than or equal to 0.2, we should add extra 0.5 to d in (63) in order to produce a substantial interval $[c, d]$ for the CONLeg method.

6. Numerical results

The main purpose of this section is to test the accuracy and efficiency of the CONLeg method through various numerical tests. This involves the ability of the method to price any options that are deep in/out of the money and have long/short maturities. Most importantly, we show that the algorithm exhibits good accuracy even when the PDF is smooth/non-smooth. A number of popular numerical methods are implemented to compare the algorithm in terms of the error convergence and computational time. These methods include the COS method (a Fourier COS series method, Fang and Oosterlee 2009a), the filter–COS method (a COS method with an exponential filter to resolve the Gibbs phenomenon; see Ruijter *et al.* 2015), the CONV method (an FFT method, Lord *et al.* 2008), the Lewis–FRFT (a fractional FFT method, Lewis 2001, Chourdakis 2004), the QUAD–CONV (a combination of the quadrature and CONV methods; see O'Sullivan 2005, Chen *et al.* 2014), and the SWIFT methods (a wavelet-based method; see Ortiz-Gracia and Oosterlee 2013, Maree 2015, Ortiz-Gracia and Oosterlee 2016, Maree *et al.* 2017). When we implement the CONV and Lewis–FRFT and QUAD–CONV methods, we use Simpson's rule for the Fourier integrals to achieve fourth-order accuracy. In the filter–COS method, we use an exponential filter and set the accuracy parameter to 10 as Ruijter *et al.* (2015) report that this filter provides better algebraic convergence than the other options. We also set the damping factors of the CONV to 0 for pricing European options. 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. Finally, the code is written in MATLAB and also the codes of implementing the COS method and the FFT method, such as the CONV method and the like, is retrieved from von Sydow *et al.* (2015).

Since we use the built-in Chebfun (Trefethen *et al.* 2014) commands, mainly `chebfun, conv, diff, roots and simplify`, to price and hedge the options in this paper, Chebfun (http://www.chebfun.org/download/) is required for our option pricing algorithm. For a demonstration, we give the MATLAB code for computing European options in Appendix 4. As Chebfun makes use of adaptive procedures that aim to find the right number of points automatically so as to represent each function to roughly machine precision, that is, about 15 digits of relative accuracy, we allow Chebfun to approximate f and g^R automatically. As Chebfun is implemented in MATLAB, we should point out that since Chebfun calls functions, nested functions and subfunctions considerably, the overheads occur and accordingly, this slows the CONLeg method's computation speed down. Hence, as an introductory of the CONLeg method, we weigh more on the convenience of the method over its efficiency at this stage of development.

In all numerical experiments, the range of option prices we measure is based on the input range of S or K. To allow our method to have accurate approximation of deep in/out-of-money option prices, the range of S or K lie in the intervals $[K - 20, K + 20]$ or $[S - 20, S + 20]$ respectively. Moreover, as we look at the intervals of $[K - 20, K + 20]$ or $[S - 20, S + 20]$, we set $[c, d]$ rather than $[2c, 2d]$ for approximate option prices and their option Greek values †.

We use N to define the number of terms/grid points of the methods we compare against the CONLeg method. When measuring approximation errors of the numerical methods, we use absolute errors, the L_2 norm errors R_2 and the infinity norm errors R_∞ as the measurement units. Moreover, to improve the accuracy of our method in pricing/hedging European type options, we use the well-known call-put parity relationship,

$$V_{call}(x, K, t) = V_{put}(x, K, t) + Se^{-q(T-t)} - Ke^{-r(T-t)}, \quad (64)$$

to approximate call prices once we have put prices ready.

6.1. European type options

We consider three different test cases based on the following PDFs and other parameters:

$$\textbf{GBM1}: S = 100, \quad K = 80 - 120, \quad \sigma = 0.15, \quad T = 1.0, \quad r = 0.03, \quad q = 0.01. \quad (65)$$

$$\textbf{GBM2}: S = 100, \quad K = 80 - 120, \quad \sigma = 0.25, \quad T = 50 \text{ or } 100, \quad r = 0.1, \quad q = 0. \quad (66)$$

† To achieve this, we can set a flag–'same'–in `conv`. See Appendix 4 for details.

$$\begin{aligned}\mathbf{VG1}: S &= 100, K = 80-90, \quad \sigma = 0.12,\\ \theta &= -0.14, \quad \nu = 0.2, \quad T = 0.1,\\ r &= 0.1, \quad q = 0.\end{aligned} \tag{67}$$

In all three numerical tests, the reference values for the GBM process and the VG process are generated via MATLAB Financial Toolbox$^{\text{TM}}$–`blsprice, blsdelta and blsgamma`–and the Singularity Fourier–Padé (SFP) method (cf. Chan 2018) respectively. We also set $L_n = 10$ in (63) when we create a computational interval $[c, d]$ for pricing/hedging European options under the two processes.

In the first numerical test (**GBM1**)–table 3, we first check for convergence behaviour against a range of strikes K from 80 to 120 for deep in/out-of-the money and at-the-money vanilla put options. Apart from $q = 0.01$, the parameters are retrieved from von Sydow *et al.* (2015). We declare 1000 different option prices within the range of either K. In this test, even without applying the put-call parity (64), the CONLeg can achieve very high accuracy (around $R_\infty = 10^{-14}$) when it is applied to approximate option prices and its Delta Δ and Gamma Γ. Moreover, since our method aims to model option price/Greek curves rather than their values, our method consumes only less than 0.1 seconds to formulate the curves in the test. Using around 0.1 seconds to produce option price and Greek curves is a quite reasonable computational cost for a method to meet the financial standards. The second numerical experiment (**GBM2**) is devoted to the performance of the CONLeg method for long maturity call options, which are often encountered in the insurance and pension industry. The parameters are retrieved from Ortiz-Gracia and Oosterlee (2016) for the test. Table 4 refers to the second test (**GBM2**) and replicates Table 3 in Ortiz-Gracia and Oosterlee (2016). In this test, with the help of (64), the CONLeg method impressively provides high accuracy when we declare 1000 different option prices within the range of K from 80 to 120. The last set of parameters (**VG1**) is chosen in the last numerical test (table 5) because relatively slow convergence was reported for the CONV method for very short maturities in Lord *et al.* (2008). This is attributed to the PDF of the process being sharp-peaked with a difficult logarithmic singularity x_{sing} (see figure 1). Before we approximate the closed-form VG PDF defined in (A17) with Chebyshev series, we apply the Fourier–Padé method (cf. Appendix 2) to locate the singularity in the PDF. We use x_{sing} as a breakpoint and approximate the PDF in two different regions $[c, x_{sing}]$ and $[x_{sing}, d]$ to resolve Gibbs phenomena. Since we use the Fourier–Padé method to approximate the PDF, the number of Fourier terms in the Fouier–Padé approximation is set to be 1024. In table 5, again with the aid of (64), within similar CPU time and 30 options measured within the range of 80 and 90, the CONLeg method yields almost same accuracy to the COS and filter–COS methods, but relatively higher accuracy than the Lewis–FRFT, QUAD–CONV and CONV methods.

REMARK 3 Fang and Oosterlee (2009a) suggest that the VG process with **VG1** gives rise to a probability density function that is not in $C^\infty(\mathbb{R})$, and thus, option pricing under

Table 3. Measuring the CONLeg method in error convergence and CPU time for pricing European at/around-the-money put prices, Delta Δ_t and Gamma Γ_t under the BSM model with parameters taken from **GBM1**.

Price		Delta Δ_t		Gamma Γ_t		
R_∞	R_2	R_∞	R_2	R_∞	R_2	Time
5.329e − 14	3.921e − 13	5.645e − 14	1.846e − 13	8.538e − 13	1.396e − 12	9.11e − 02

Note: 1000 put prices are computed in a range of K from 80 to 120.

Table 4. Measuring the CONLeg method in error convergence and CPU time for pricing European at/around-the-money call prices, Delta Δ_t and Gamma Γ_t under the BSM model with parameters taken from **GBM2**.

	Price		Delta Δ_t		Gamma Γ_t		
	R_∞	R_2	R_∞	R_2	R_∞	R_2	Time
$T = 50$	2.842e − 14	2.821e − 13	2.220e − 16	1.005e − 15	7.170e − 17	5.007e − 16	8.34e − 02
$T = 100$	2.842e − 14	2.874e − 13	3.841e − 16	2.123e − 16	6.126e − 19	3.776e − 18	9.12e − 02

Note: 1000 call prices are computed in a range of K from 80 to 120.

Table 5. Comparison of the Lewis–FRFT, CONV, QUAD–CONV, COS, filter–COS and CONLeg methods for pricing vanilla call options under the VG model with parameters taken from **VG1**.

Lewis–FRFT				CONV				QUAD–CONV			
N	R_∞	R_2	Time	N	R_∞	R_2	Time	N	R_∞	R_2	Time
1024	9.921e − 03	1.121e − 02	0.092	1024	1.217e − 04	8.817e − 03	0.110	1024	2.411e − 04	7.827e − 03	0.121
COS				filter–COS				CONLeg			
N	R_∞	R_2	Time	N	R_∞	R_2	Time		R_∞	R_2	Time
1024	2.271e − 06	7.579e − 05	0.089	1024	6.596e − 07	3.596e − 06	0.0912		5.596e − 07	4.511e − 06	0.102

Note: 30 call prices are computed in a range of K from 80 to 90.

VG with these parameter sets exhibits only an algebraic convergence. Nevertheless, Chan (2018) has recently proposed the SFP method to circumvent the problem and to approximate the VG PDF with the input parameters of **VG1**. Through this method, we can regain global spectral convergence away from singularities. For more details, we refer the readers to Chan (2018).

REMARK 4 In table 5, one may suggest adaptive Simpson's rule (quadrature) be applied into the CONV and Lewis–FRFT and QUAD–CONV methods to have a fair comparison with the CONLeg method. Nevertheless, the VG process with **VG1** gives rise to a probability density function that is not in $C^\infty(\mathbb{R})$. In general, adaptive quadrature can be very inefficient if the integrand has a discontinuity within a subinterval, since repeated subdivision will occur. Accordingly, a more advanced adaptive algorithm should be selected, but it is out of the scope of this paper. Moreover, Chebfun adaptively decides what degree N to use at each stage of the computation, whereas with the other methods we pick an N at the start. It hard to compare convergence rate between the methods.

6.2. American, barrier and Bermuda options

In this section we again focus on three test cases based on the following PDFs and other parameters:

$$\begin{aligned}\textbf{NIG1}: &S=100,\quad K=80-120,\quad \alpha=15,\quad \beta=-5,\\ &\delta=0.5,\quad T=1,\quad r=0.05,\quad q=0.02. \quad (68)\\ \textbf{CGMY1}: &S=0.5-1.5,\quad K=1,\quad C=1,\quad G=5,\\ &M=5,\quad Y=0.5,\quad T=1,\quad r=0.1,\quad q=0.0. \quad (69)\\ \textbf{CGMY2}: &S=90-100,\quad K=100,\quad C=4,\quad G=50,\\ &M=60,\quad Y=0.7,\quad T=1,\quad r=0.05,\\ &q=0.02. \quad (70)\end{aligned}$$

In this section, as suggested by Fang and Oosterlee (2009b), we first set $L_n = 8$ in (63) to create a computational interval $[c, d]$ for approximating the PDFs of the NIG and CGMY processes. In our first test we price Bermudan put options with 10 exercise dates with parameters of **NIG1**. In this test (table 6), a total of 170 option prices are generated in the COS and CONV methods from a range of K between 80 to 120. The CPU times are reported in seconds, and all reference values are obtained by the CONV method with $N = 2^{20}$. Fang and Oosterlee (2009b) report that the NIG PDF (A16) is more peaked at the mean with the parameters of **NIG1**. We then use the mean as a breakpoint to approximate the PDF in two regions $[c, x_{sing}]$ and $[x_{sing}, d]$. From the result of table 6, the accuracy of the COS method stays the same when N increases from 256 to 512. Within the similar CPU time, we can see that the CONLeg has better accuracy than the COS method.

In the next two tests, the CGMY processes with parameters of **CGMY1** and **CGMY2** do not have a closed-form PDF and have singularities. Accordingly, again using the techniques we mentioned above, we use the Fourier–Padé method to approximate the CGMY PDFs and locate their singularities. The number of Fourier terms in the Fouier–Padé approximation is set to be 1024 in the tests.

The prices of American options can be obtained using (48), a 4-point Richardson extrapolation method on the prices of a few Bermudan options with small L. We compare the CONV, COS and CONLeg methods for pricing American option price in table 7. All reference values are obtained by the Fourier time stepping method (Jackson *et al.* 2008) with the number

Table 6. Comparison of the COS and CONLeg methods for pricing a Bermuda ($L = 10$) put option under the NIG model with parameters taken from **NIG1**.

	COS			CONLeg		
N	R_∞	R_2	Time	R_∞	R_2	Time
256	2.411e − 07	5.411e − 06	2.89	1.812e − 13	8.576e − 12	2.93
512	2.411e − 07	5.411e − 06	3.323	1.812e − 13	8.576e − 12	2.93

Note: 170 option prices are generated for both methods in a range of K from 80 to 120 and S equal to 100.

Table 7. Comparison of the CONV, COS and CONLeg methods for pricing an American put option under the CGMY model with parameters taken from **CGMY1**.

L in equation (48)	CONV				CONLeg		
	N	R_∞	R_2	Time (sec.)	R_∞	R_2	Time (sec.)
2	32768	1.052e − 04	8.051e − 03	19.291	7.123e − 05	6.113e − 04	19.311
L in equation (48)	COS				CONLeg		
	N	R_∞	R_2	Time (sec.)	R_∞	R_2	Time (sec.)
2	32768	9.234e − 05	4.012e − 04	19.298	7.123e − 05	6.113e − 04	19.311

Note: 58 and 88 option prices are computed for the CONV method and the COS method respectively in a range of S from 0.5 to 1.5 and K equal to 1.

Table 8. Comparison of the SWIFT and CONLeg methods for pricing daily-monitored ($L = 12$) UO call and UO put under the CGMY model with parameters taken from **CGMY2**.

	SWIFT				CONLeg		
	scale	R_∞	R_2	Time (sec.)	R_∞	R_2	Time (sec.)
UO Call	6	3.181e − 10	7.641e − 09	7.100	5.186e − 09	6.886e − 08	7.201
UO Put	6	1.901e − 11	8.621e − 10	6.901	4.330e − 10	6.131e − 09	6.891

Notes: 725 option prices are computed in a range of S from 90 to 110 and K equal to 100. The barrier level H is equal to 120.

Table 9. Comparison of the COS and CONLeg methods for pricing daily-monitored ($L = 252$) DO Call and DO Put under the NIG model with parameters taken from **NIG1**.

	COS				CONLeg		
	N	R_∞	R_2	Time (sec.)	R_∞	R_2	Time (sec.)
DO Call	8192	6.701e − 08	4.641e − 07	94.231	4.167e − 09	8.651e − 08	93.812
DO Put	8192	7.232e − 09	3.231e − 08	92.532	1.619e − 09	6.886e − 08	92.634

Notes: 38 option prices are computed in the range of S from 90 to 110 and K equal to 100. The barrier level H is equal to 80.

of the Fourier terms equal to 2^{16}. Within the similar CPU time, the CONLeg method achieves the same accuracy with the COS method and is marginally better than the CONV method.

For the last two tables, all reference values are obtained by the CONV method with $N = 2^{20}$. In table 8, we consider monthly monitored ($L = 12$) up-and-out call and put options, (UO Call) and (UO Put) under the CGMY process with the parameters of **CGMY2**. The barrier level H is set to be 120 for the up-and-out options. As we can see from the table, the CONLeg method is very comparable to the SWIFT method in terms of accuracy. Finally, in table 9, we focus on the NIG process with parameters of **NIG1**, except S = 90–110 and K = 100. Considering monthly monitored ($L = 252$) down-and-out call (DO Call) and put (DO Put) options, we set H equal to 80 for the options. Since $L = 252$ is very large, it causes the time interval difference between t_{l+1} and t_l to be very small. Accordingly, the NIG PDF is very peaked and the COS method requires a larger number of N equal to 8192 to compensate a better convergence in table 9. Comparing the CONLeg and COS methods in the similar CPU time, both methods can achieve similar accuracy.

7. Conclusions

In this paper, we have proposed an algorithm for hedging and pricing various options based on approximating probability density functions by polynomials (in particular, Legendre series) and computing the prices by performing suitable convolutions with the given payoff function. We call this method CONLeg. The main advantages of the CONLeg method are its ability to return the price and Greeks as a function defined on a prescribed interval rather than just point values, its ability to approximate different types options under a process with/without a closed-from PDF, and the simplicity and convenience of its implementation using Chebfun. We presented a proof of concept implementation written in Chebfun, which demonstrates positive results in the accuracy of the method. Since Chebfun uses adaptive approximations, it is difficult at this time to discuss rates of convergence of the CONLeg method or to compare it's efficiency with respect to other methods in the literature (which work with a fixed discretisation size), but initial results are promising. For a fair comparison, one would need to implement a non-adaptive version of CONLeg, which would be time-consuming, and not typically desirable in a practical setting.

Our ultimate goal is to extend the method to price options with path-dependant features under (time-changed) Lévy processes or affine stochastic volatility models with and without jumps. Research in this direction is already underway and will be presented in a forthcoming manuscript.

Disclosure statement

No potential conflict of interest was reported by the author(s).

ORCID

Tat Lung (Ron) Chan http://orcid.org/0000-0002-4048-6370

References

Andricopoulos, A.D., Widdicks, M., Duck, P.W. and Newton, D.P., Universal option valuation using quadrature methods. *J. Financ. Econ.*, 2003, **67**, 447–471.
Andricopoulos, A.D., Widdicks, M., Newton, D.P. and Duck, P.W., Extending quadrature methods to value multi-asset and complex path dependent options. *J. Financ. Econ.*, 2007, **83**, 471–499.
Bondarenko, O., Estimation of risk-neutral densities using positive convolution approximation. *J. Econom.*, 2003, **116**, 85–112.
Broadie, M. and Yamamoto, Y., Application of the fast Gauss transform to option pricing. *Manage. Sci.*, 2003, **49**, 1071–1088.
Broadie, M. and Yamamoto, Y., A double-exponential fast Gauss transform algorithm for pricing discrete path-dependent options. *Oper. Res.*, 2005, **53**, 764–779.
Cai, N. and Kou, S.G., Option pricing under a mixed-exponential jump diffusion model. *Manage. Sci.*, 2011, **57**, 2067–2081.
Carr, P. and Madan, D., Option valuation using the fast Fourier transform. *J. Comput. Finance*, 1999, **4**, 61–73.
Chan, T.L.R., An orthogonal series expansions method to hedge and price European-type options, 2016. Available online at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id = 2852033.
Chan, T.L.R., Singular Fourier-Páde series expansion of European option prices. *Quant. Finance*, 2018, **18**, 1149–1171.

Chang, C.C., Chung, S.L. and Stapleton, R.C., Richardson extrapolation techniques for the pricing of American-style options. *J. Futures Mark.*, 2007, **27**, 791–817.
Chen, D., Härkönen, H.J. and Newton, D.P., Advancing the universality of quadrature methods to any underlying process for option pricing. *J. Financ. Econ.*, 2014, **114**, 600–612.
Chourdakis, K., Option pricing using the fractional FFT. *J. Comput. Finance*, 2004, **8**, 1–18.
Colldeforns-Papiol, G., Ortiz-Gracia, L. and Oosterlee, C., Two-dimensional Shannon wavelet inverse Fourier technique for pricing European options. *Appl. Numer. Math.*, 2017, **117**, 115–138.
Cont, R. and Tankov, P., *Financial Modelling with Jump Processes*, Financial mathematics series, 2004 (Chapman & Hall/CRC: Boca Raton, FL).
Driscoll, T.A. and Fornberg, B., *The Gibbs Phenomenon in Various Representations and Applications*, 2011 (Sampling Publishing: Potsdam, NY).
Fang, F. and Oosterlee, C.W., A novel pricing method for European options based on Fourier-Cosine series expansions. *SIAM J. Sci. Comput.*, 2009a, **31**, 826–848.
Fang, F. and Oosterlee, C.W., Pricing early-exercise and discrete barrier options by Fourier-cosine series expansions. *Numer. Math.*, 2009b, **114**, 27–62.
Fang, F. and Oosterlee, C.W., A Fourier-based valuation method for Bermudan and barrier options under Heston's model. *SIAM J. Financ. Math.*, 2011, **2**, 439–463.
Feng, L. and Linetsky, V., Pricing discretely monitored barrier options and defaultable bonds in Lévy process models: A fast Hilbert transform approach. *Math. Financ.*, 2008, **18**, 337–384.
Gaß, M., Glau, K., Mahlstedt, M. and Mair, M., Chebyshev interpolation for parametric option pricing. *Finance Stoch.*, 2018, **22**, 701–731.
Geske, R. and Johnson, H.E., The American put option valued analytically. *J. Finance*, 1984, **39**, 1511–1524.
Hale, N. and Townsend, A., An algorithm for the convolution of Legendre series. *SIAM J. Sci. Comput.*, 2014a, **36**, A1207–A1220.
Hale, N. and Townsend, A., A fast, simple, and stable Chebyshev–Legendre transform using an asymptotic formula. *SIAM J. Sci. Comput.*, 2014b, **36**, A148–A167.
Hogg, R.V., McKean, J.W. and Craig, A.T., *Introduction to Mathematical Statistics*, Vol. 6, 2004 (Prentice Hall: New Jersey).
Jackson, K.R., Jaimungal, S. and Surkov, V., Fourier space time-stepping for option pricing with Lévy models. *J. Comput. Finance*, 2008, **12**, 1–29.
Leentvaar, C. and Oosterlee, C., Multi-asset option pricing using a parallel Fourier-based technique. *J. Comput. Finance*, 2008, **12**, 1–26.
Lewis, A., A simple option formula for general jump-diffusion and other exponential Lévy processes, 2001. Available online at: http://optioncity.net/pubs/ExpLevy.pdf.
Lipton, A., The vol smile problem. *Risk*, 2002, **15**, 61–66.
Liu, X., Xiao, H. and Chen, R., Convolutional autoregressive models for functional time series. *J. Econom.*, 2016, **194**, 263–282.
Lord, R., Fang, F., Bervoets, F. and Oosterlee, C.W., A fast and accurate FFT-based method for pricing early-exercise options under Lévy processes. *SIAM J. Sci. Comput.*, 2008, **30**, 1678–1705.
Maree, S.C., Numerical pricing of Bermudan options using Shannon wavelet expansions. Master's thesis, Delft Institute of Applied Mathematics, Delft University of Technology, Delft, The Netherlands, 2015.
Maree, S.C., Ortiz-Gracia, L. and Oosterlee, C.W., Pricing early-exercise and discrete barrier options by Shannon wavelet expansions. *Numer. Math.*, 2017, **136**, 1035–1070.
Mason, J.C. and Handscomb, D., *Chebyshev Polynomials*, 2002 (CRC Press: Boca Raton, FL).
Olver, S. and Townsend, A., Julia package for function approximation, random number generation and solving differential equations, 2011. Available online at: https://github.com/JuliaApproximation/ApproxFun.jl.
Ortiz-Gracia, L. and Oosterlee, C.W., Robust pricing of European options with wavelets and the characteristic function. *SIAM J. Sci. Comput.*, 2013, **35**, B1055–B1084.
Ortiz-Gracia, L. and Oosterlee, C.W., A highly efficient Shannon wavelet inverse Fourier technique for pricing European options. *SIAM J. Sci. Comput.*, 2016, **38**, B118–B143.
O'Sullivan, C., Path dependant option pricing under Lévy processes. In *Proceedings of the EFA 2005 Moscow Meetings Paper*, p. 24, 2005.
Pachón, R., Numerical pricing of European options with arbitrary payoffs. *Int. J. Financ. Eng.*, 2018, **5**, 1850015.
Pachón, R., Platte, R.B. and Trefethen, L.N., Piecewise-smooth chebfuns. *IMA J. Numer. Anal.*, 2010, **30**, 898–916.
Ruijter, M., Versteegh, M. and Oosterlee, C., On the application of spectral filters in A Fourier option pricing technique. *J. Comput. Finance*, 2015, **19**, 75–106.
Schoutens, W., *Lévy Processes in Finance: Pricing Financial Derivatives*, Wiley series in probability and mathematical statistics, 2003 (Wiley: Chichester, UK).
Su, H., Chen, D. and Newton, D.P., Option pricing via QUAD: From Black-Scholes-Merton to Heston with jumps. *J. Deriv.*, 2017, **24**, 9–27.
Suetin, P.K., On the representation of continuous and differentiable functions by Fourier series in Legendre polynomials. *Dokl. Akad. Nauk SSSR*, 1964, **158**, 1275–1277.
Swierczewski, C. and Verdier, O., Python Chebyshev functions, 2011. Available online at: https://github.com/olivierverdier/pychebfun.
Townsend, A., Computing with functions in two dimensions. PhD thesis, University of Oxford, Oxford, 2014. Available online at: http://math.mit.edu/ ajt/papers/thesis.pdf.
Townsend, A., Webb, M. and Olver, S., Fast algorithms for Toeplitz and Hankel matrices. *Math. Comput.*, 2018, **87**, 1913–1934.
Trefethen, L.N., Driscoll, T.A. and Hale, N., *Chebfun Guide*, 2014 (Pafnuty Publications: Oxford). Available online at: http://www.chebfun.org/.
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, J., Oosterlee, C.W., Ruijter, M.J., Toropov, A. and Zhao, Y., BENCHOP–The BENCHmarking project in option pricing. *Int. J. Comput. Math.*, 2015, **92**, 2361–2379.
Wang, H. and Xiang, S., On the convergence rates of legendre approximation. *Math. Comput.*, 2012, **81**, 861–877.
Wong, H.Y. and Guan, P., An FFT-network for Lévy option pricing. *J. Bank. Finance*, 2011, **35**, 988–999.
Zeng, P. and Kwok, Y.K., Pricing barrier and Bermudan style options under time-changed Lévy processes: Fast Hilbert transform approach. *SIAM J. Sci. Comput.*, 2014, **36**, B450–B485.
Zhang, B. and Oosterlee, C.W., Efficient pricing of European-style Asian options under exponential Lévy processes based on Fourier cosine expansions. *SIAM J. Financ. Math.*, 2013, **4**, 399–426.

Appendices

Appendix 1. A Closed-form Transformation of Complex Fourier Series into Chebyshev Series

To transform a complex Fourier series (CFS) into a Chebyshev series, we apply the result of Townsend (2014, Lemma A.3), i.e.

$$\int_{-1}^{1} \frac{\exp(ixy\pi)T_q(x)}{sqrt1-x^2}\mathrm{d}x = \pi i^q J_q(y\pi). \tag{A1}$$

Here, J_q is the Bessel functions of the first kind with parameter q and $T_q(x)$ is the Chebyshev polynomial of degree q. We first note that since g, a PDF, is a real function, the CFS representation of g with a form of

$$g(x) \approx g_N(x) = \Re\mathfrak{e}\left[2\sum_{k=1}^{N}\varphi\left(-\frac{2\pi}{d-c}k\right)e^{i\frac{2\pi}{d-c}kx} + \varphi(0)\right]. \tag{A2}$$

can be interchangable into

$$\mathfrak{Re}\left[\sum_{k=-N}^{N}\varphi\left(-\frac{2\pi}{d-c}k\right)e^{i\frac{2\pi}{d-c}kx}\right]. \tag{A3}$$

Using (A1) and (A3), we may define a Chebyshev series on the same interval $[c, d]$ such that

$$\mathfrak{Re}\left[\sum_{k=-N}^{N}\varphi\left(-\frac{2\pi}{d-c}k\right)e^{i\frac{2\pi}{d-c}kx}\right]=\sum_{n=0}^{N}\alpha_n^{cheb}T_n\circ\psi_{[c,d]}(x),$$

where,

$$\alpha_0^{cheb}=\mathfrak{Re}\left[\sum_{\substack{k=-N\\k\neq 0}}^{N}\frac{\Gamma\left(\frac{1}{2}\right)}{\sqrt{\pi}}\varphi\left(-\frac{2\pi}{d-c}k\right)e^{i\frac{(d+c)}{d-c}k}J_0\left(-k\pi\right)+\varphi(0)\right], n=0,$$

$$\alpha_n^{cheb}=\mathfrak{Re}\left[2\sum_{\substack{k=-N\\k\neq 0}}^{N}\varphi\left(-\frac{2\pi}{d-c}k\right)e^{i\frac{(d+c)}{d-c}k}i^nJ_n\left(-k\pi\right)\right], n>0. \tag{A4}$$

Appendix 2. Locating Singularities in Probability Density Functions

Many PDFs of interest are not smooth but piecewise smooth. For example, see figure 1. If the locations of all singularities are not known in advance, we can use Fourier–Padé ideas (cf. Driscoll and Fornberg 2011, Chan 2018) to estimate the locations of singularities well enough to allow good reconstruction nearly everywhere in the interval $[c, d]$.

The Fourier–Padé algorithm proposed in this paper is very simple to implement. If we consider a function g with a power series representation such that

$$g(x)=\sum_{k=0}^{\infty}b_kx^k,$$

and a rational function defined by $R_{N,M}=P_N/Q_M$, where P_N and Q_M are the polynomials of

$$P_N(x)=\sum_{n=0}^{N}p_nx^n \quad\text{and}\quad Q_M(x)=\sum_{m=0}^{M}q_mx^m, \tag{A5}$$

respectively, then we say that $R_{N,M}=P_N/Q_M$ is the *(linear) Padé approximant* of order (N, M) of the formal series satisfying the condition

$$\left(\sum_{n=0}^{N}p_nx^n\right)-\left(\sum_{m=0}^{M}q_mx^m\right)\left(\sum_{k=0}^{M+N}b_kx^k\right)=\mathcal{O}(x^{N+M+1}). \tag{A6}$$

Here, g is approximated by $\sum_{k=0}^{M+N}b_kx^k$, To obtain the approximant $R(N, M)$, we simply calculate the coefficients of polynomials P_N and Q_M by solving a system of linear equations. To obtain $\{q_m\}_{m=0}^{M}$, we first normalise $q_0=1$ to ensure that the system is well determined and has a unique solution in (A6). Then, we consider the coefficients for $x^{N+1},\ldots,x^{M+N}$, and we can yield a Toeplitz*† linear system:

$$\begin{bmatrix} b_{N+1} & b_N & b_{N-1} & \cdots & b_{N+1-M}\\ b_{N+2} & b_{N+1} & b_N & \ddots & b_{N+2-M}\\ \vdots & \ddots & \ddots & \ddots & \vdots\\ b_{N+M} & \cdots & b_{N+2} & b_{N+1} & b_N\end{bmatrix}\begin{bmatrix}q_0\\q_1\\\vdots\\q_M\end{bmatrix}=0. \tag{A7}$$

Once $\{q_m\}_{m=0}^{M}$ is known, $\{p_n\}_{n=0}^{N}$ is found through the terms of order N and less in (A6). This yields $\underline{p}=B\underline{q}$, where $b_{ij}=b_{i-j}$. For example, if $N=M$, one obtains

$$\begin{bmatrix}p_0\\p_1\\\vdots\\p_N\end{bmatrix}=\begin{bmatrix}b_0 & & & \\ b_1 & b_0 & & \\ \vdots & \ddots & \ddots & \\ b_N & \cdots & b_1 & b_0\end{bmatrix}\begin{bmatrix}q_0\\q_1\\\vdots\\q_M\end{bmatrix}. \tag{A8}$$

Now, assuming g is a PDF, to find singularities in g and to express g in a Fourier–Padé series, we first express g with the CFS representation:

$$\mathfrak{Re}\left[2\sum_{k=1}^{\infty}\varphi\left(-\frac{2\pi}{d-c}k\right)e^{i\frac{2\pi}{d-c}kx}+\varphi\left(0\right)\right]. \tag{A9}$$

Then, we can differentiate (A9) with respect to x to obtain

$$\mathfrak{Re}\left[2\sum_{k=1}^{\infty}\left(i\frac{2\pi}{d-c}k\right)\varphi\left(-\frac{2\pi}{d-c}k\right)e^{i\frac{2\pi}{d-c}kx}\right]. \tag{A10}$$

Finally, we let $z=\exp(i\frac{2\pi}{d-c}x)$ in the two equations above, and they are ready for the Fourier–Padé approximation. In general, when the PDF has a singularity, the sharp-peaked singularity point will have an enormously large value after differentiation. In other words, figure 1 is a graphical illustration of the outlooks of the PDF (left) and the first derivative (right) of the VG model after the Fourier–Padé approximation. In the figure, we can see that the non-smooth PDF with a jump can produces a value of 10×10^{11} at the singularity point after the first derivative. See `padeapprox` in Chebfun.

Appendix 3. Characteristic functions and their closed-form risk-neutral probability density functions for various exponential Lévy models

The characteristic functions $\varphi(u)$ of some common exponential Lévy models are investigated in this paper:

$$\varphi_{\mathrm{GBM}}(u):=\exp\left(t\left(iu(r-q+\omega)-\frac{1}{2}\sigma^2u^2\right)\right), \tag{A11}$$

$$\varphi_{\mathrm{NIG}}(u):=\exp\left(t\left(iu(r-q+\omega)-\frac{1}{2}\sigma^2u^2+\delta\left(\sqrt{\alpha^2-\beta^2}-\sqrt{\alpha^2-(\beta+iu)^2}\right)\right)\right), \tag{A12}$$

$$\varphi_{\mathrm{VG}}(u):=\exp\left(iu(r-q+\omega)t\right)\left(\frac{1}{1-i\theta\upsilon u+\frac{\sigma^2\upsilon}{2}u^2}\right)^{\frac{t}{\upsilon}}, \tag{A13}$$

$$\varphi_{\mathrm{CGMY}}(u):=\exp\left(t\left(iu(r-q+\omega)+C\Gamma(-Y)G^Y\left(\left(1+\frac{iz}{G}\right)^Y-1-\frac{izY}{G}\right)+C\Gamma(-Y)M^Y\left(\left(1-\frac{iz}{M}\right)^Y-1+\frac{izY}{M}\right)\right)\right). \tag{A14}$$

† A Toeplitz matrix or diagonal-constant matrix is an invertible matrix in which each descending diagonal from left to right is constant.

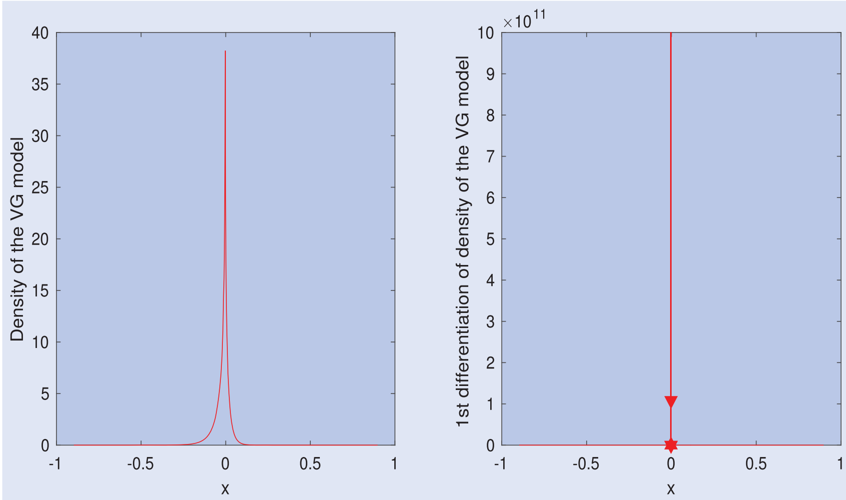

Figure A1. Density functions (left) and its first derivative (right) of the VG model with parameters are taken from **VG1**.

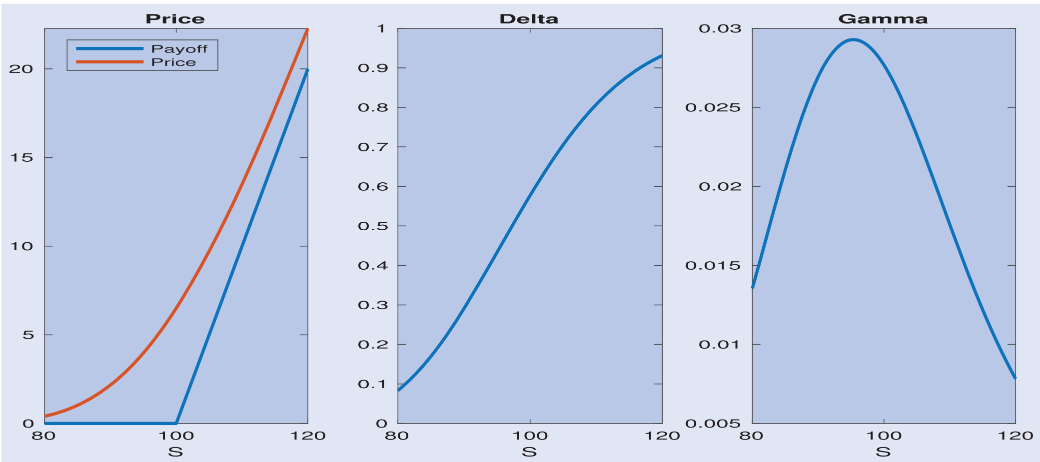

Figure A2. Figure produce by code above.

The closed-form risk-neutral probability density functions (PDF) $g(x)$ of some common exponential Lévy models are investigated in this paper:

$$g_{\mathrm{GBM}}(x) = \frac{1}{\sqrt{2\pi t}\sigma} e^{-\frac{1}{2}\frac{(x-(r-q+\omega_{GBM}))^2}{\sigma^2 t}}, \tag{A15}$$

$$g_{\mathrm{NIG}}(x)(x) = \frac{\alpha\delta t K_1\left(\alpha\sqrt{\delta^2 t^2 + (x-(r-q+\omega_{NIG})t)^2}\right)}{\pi\sqrt{\delta^2 t^2 + (x-(r-q+\omega_{NIG})t)^2}} e^{\delta t\sqrt{\alpha^2-\beta^2} + \beta(x-(r-q+\omega_{NIG})t)}, \tag{A16}$$

$$g_{\mathrm{VG}}(x) = \frac{2e^{\theta Z(x)/\sigma^2}}{\nu^{\frac{t}{\nu}}\sqrt{2\pi}\sigma\Gamma(t/\nu)}\left(\frac{Z(x)^2}{2\sigma^2/\nu + \theta.^2}\right)^{\frac{t}{2\nu}-\frac{1}{4}} \times K_{\frac{t}{\nu}-\frac{1}{2}}\left(\frac{1}{\sigma^2}\sqrt{Z(x)^2\left(\frac{2\sigma^2}{\nu} + \theta^2\right)}\right) \tag{A17}$$

Here, $\omega_{GBM} = -1/2\sigma^2, \omega_{NIG} = \delta(\sqrt{\alpha^2-(\beta+1)^2} - \sqrt{\alpha^2-\beta^2})$, $K(\cdot)$ is a modified Bessel function of a second kind, $\Gamma(\cdot)$ is a gamma function, $Z(x) = x - (r-q+\omega_{VG})t$, and $\omega_{VG} = -1/\nu \log(1-\theta\nu-\frac{\sigma^2\nu}{2})$.

Appendix 4. MATLAB Code

```
% COMPUTE GBM EUROPEAN OPTION PRICE AND GREEKS USING CONLEG METHOD.

K = 100;                                          % Strike price
window = [80, 120];                               % Current stock price window
x = log(linspace(window(1), window(2), 5)/K);     % Example stock prices
r = 0.035; t = 0.5; q = 0; sigma = 0.2;             % Input parameters
call = true;                                      % (call = false ==> put)
mu = (r-q-0.5*sigma^2)*t;
L = 10;
c1 = mu; c2 = sigma^2*t; c4 = 0;        % Cumulants (see section 5)
d = abs(c1+L*sqrt(c2 + sqrt(c4)));      % x-variable domain (see section 5)
c = -d; dom = [c, 0, d];

% Payoff function:
if ( call )
    payoff = @(x) K*max(exp(x)-1, 0); % Call payoff
else
    payoff = @(x) K*max(1-exp(x), 0); % Put payoff
end
payoff = chebfun(payoff, dom);          % Chebfun representation:

% Guassian Density Function:
mypdf = @(u) 1./sqrt(2*pi*sigma^2*t)*exp(-0.5*(-u-mu)^2./(sigma^2*t));
mypdf = chebfun(mypdf, dom);            % Chebfun representation:

% Continuous stock price:
stock = chebfun(@(x) K*exp(x), dom);

% Compute European option price via convolution:
price = exp(-r*t)*conv(payoff, mypdf, 'same');
price = simplify(price);

% Compute Greeks:
dP = diff(price);       % 1st derivative
delta = dP./stock;
dP2T = diff(price, 2); % 2nd derivative
gamma = (dP2T-dP)./(stock.^2);

%Display figures:
subplot(1,3,1), plot(stock, payoff, stock, price, 'LineWidth', 2); title('Price')
legend('Payoff', 'Price', 'Location', 'NW'), xlabel('S'),
axis([window, 0, price(x(end))])
subplot(1,3,2), plot(stock, delta, 'LineWidth', 2); title('Delta');
xlabel('S'), xlim(window)
subplot(1,3,3), plot(stock, gamma, 'LineWidth', 2); title('Gamma')
xlabel('S'), xlim(window), shg

% Display table:
Results = [K+0*x ;  stock(x) ; price(x) ; delta(x) ; gamma(x)];
disp(table(Results, 'RowNames', {'strike', 'stock', 'price', 'delta', 'gamma'}))
```

A. Output

Results

strike	100	100	100	100	100
stock	80	90	100	110	120
price	0.4069	2.1637	6.4983	13.5068	22.2810
delta	0.0833	0.2910	0.5771	0.8074	0.9311
gamma	0.0135	0.0269	0.0277	0.0176	0.0078