An Efficient Transform Method for Asian Option Pricing

J. Lars Kirkby *

Abstract. This paper introduces a novel method to price arithmetic Asian options in Levy-driven models, with discrete and continuous averaging, by expanding on the approach of sequential characteristic function recovery. By utilizing frame duality and a FFT-based implementation of density projection, we obtain rapidly converging value approximations to high precision, consistently resulting in a 10- to 100-fold time reduction compared to state-of-the-art procedures. Theoretical convergence rates are confirmed by an in-depth analysis of error propagation. Formulas for Greeks are provided, in addition to generalized averaging and in-progress option pricing.

Key words. arithmetic Asian options, fast Fourier transform, Levy processes, basis, characteristic function, Carverhill-Clewlow factorization, PROJ, COS, FFT, frame projection, option pricing, B-spline, exotic options.

AMS subject classifications. 62P05, 60E10, 91G20, 91B25, 91G60, 65C20, 65T50, 65D07, 42C15, 65T40, 65T60

1. Introduction.

Since their introduction in 1987, Asian options (known also as average rate or average price options) have provided a popular means of risk management in a variety of markets. For example, Eydeland and Wolyniec (2003) document their importance in mitigating the delivery risks present in gas markets. Since Asian options have payoffs that are contingent on the average price of an underlying asset (index, interest rate, exchange rate, commodity, etc.) over a given time horizon, their prices are less sensitive to price manipulations, and they become easier to hedge towards the option's expiry. By taking an average of the underlying, these options are typically much cheaper than standard European contracts. Moreover, their relative stability has led to the hybridization of exotic options that contain an Asian type specification towards the end of the contract, known as an "Asian tail".

As is generally the case with path-dependent contracts, robust pricing of Asian options is very challenging and computationally demanding. Even in the Black-Scholes-Merton (BSM) framework, no analytical formulas exist for the pricing of arithmetic Asian options. The computational approaches can be categorized as analytical approximations and bounds [1, 2, 31, 34], partial differential equation (PDE) methods [3,4,17,40], lattices [14], Monte Carlo [27,35], and transform methods [6,11,12,16,41], to which our approach belongs. Alternative methods include Taylor expansion [26], perturbation [42], direct iterated integration [23], and maturity randomization [22]. In terms of both speed and accuracy, the transform based approaches are generally superior for models with Levy (log) returns, including BSM.

We develop a fast and highly accurate method for pricing generalized Asian options in exponential Levy models, which we call APROJ[1]. This includes discretely monitored contracts as well as the continuously monitored options that pervade foreign exchange markets. In-progress option prices and Greeks are also determined efficiently. Compared to state-of-the-art-methods, the APROJ method provides a

*Secton 7 devoted to implementing the APROJ method into program platform Premia was prepared by Oleg Kudryavtsev

[1]APROJ is short for Asian PROJection, due to its use of a biorthogonal projection method.

10- to 100-fold improvement in terms of cpu time to reach the same (or better) accuracy. This is confirmed for the methods of [6, 11, 12, 32, 41], most notably the improved convolution method of Cerny and Kyriakou [12], the ASCOS method of Zhang and Oosterlee [41], and the inverse Fourier transform method of Levendorskii and Xie [32], which are (to our knowledge) the fastest available pricing methods for discretely monitored arithmetic Asian options under Levy dynamics.

The paper is organized as follows. Section 2 reviews exponential Levy models and the method of density projection by frame duality. The problem of arithmetic Asian option pricing is formulated in Section 3, along with a derivation of the APROJ method. Section 4 develops extensions to in-progress option pricing and Greeks, generalized averaging, and continuous averaging. An in-depth analysis of error propagation and terminal valuation error is given in Section 5, after which Section 6 demonstrates the accuracy and efficiency of the method with a series of numerical experiments. Comparisons are made to existing methods with parameter sets from the literature. Finally, Section 7 is devoted to implementing the APROJ method into program platform Premia.

2. Density Projection Method.

The projection method described in this section applies whenever the characteristic function of the underlying random variable is known, which is the case for the family of Levy processes. Since the variance gamma (VG) model was introduced in 1990 to price derivatives [33], the versatility and tractability of Levy processes as generalizations of the BSM framework have generated a surge of research and modeling success. While application of the VG model itself has waned, subsequent developments such as the KoBoL [7, 9] model (with CGMY [10] as a special case) as well as the NIG [5] model have proven to be excellent alternatives which calibrate well to market data [10, 24], and the exponential (semi-heavy) decay of their tails engenders a significant computational advantage over the VG model.

2.1. Exponential Levy Models.

Suppose $L(t), t \geq 0$, is a Levy process, which is a stochastically continuous process with stationary and independent increments. We denote its *Levy symbol* by $\psi_L(\xi)$, where by the Levy-Khintchine theorem the characteristic function (ChF) satisfies

$$\phi_{L(t)}(\xi) := \mathbb{E}[e^{iL(t)\xi}] = e^{t\psi_L(\xi)}, \quad t \geq 0.$$

Figure 6 in the appendix provides some of the more popular Levy symbols used in financial modeling, along with any parameter restrictions[2].

To model the underlying randomness on which Asian options are contracted, we consider exponential Levy processes of the form

$$S(t) = S(0)e^{Y(t)} = S(0)e^{(r-q+\omega)t+L(t)}, \quad \omega = -\psi_L(-i),$$

where $r, q \geq 0$ are the interest rate and dividend yield. Here ω is a "convexity correction" that is used to ensure that discounted asset processes (with reinvested dividends) behave as martingales. That is, $\mathbb{E}[S(t+\Delta_t)|S(t)] \equiv S(t)\mathbb{E}[e^{R_{\Delta_t}}] = S(t)e^{(r-q)\Delta_t}$, $\Delta_t, t \geq 0$, where

$$R_{\Delta_t} := \log(S(t+\Delta_t)/S(t)) \stackrel{d}{=} (r-q+\omega)\Delta_t + L(\Delta_t), \quad t, \Delta_t > 0.$$

[2]If no restriction is given, the permissible parameter values are taken to be the real line.

The ChF of R_{Δ_t} is given by

$$\phi_{R_{\Delta_t}}(\xi) = e^{i\xi(r-q+w)\Delta_t} e^{\psi_L(\xi)\Delta_t}, \quad \Delta_t > 0.$$

Note that the underlying Levy processes satisfies an exponential moment condition $\mathbb{E}[e^{-\alpha L(t)}] < \infty$, $\forall t \geq 0$, where $\mathcal{I}_L = (\lambda_-, \lambda_+)$ denotes the set of all such α. Here $-\infty \leq \lambda_- \leq 0 \leq \lambda_+ \leq \infty$ with possible inclusion of the endpoints. As a function of $z = \xi + iw$, $\psi_L(z)$ is analytic in the strip $\mathcal{D}_{(\lambda_-,\lambda_+)} := \{z \in \mathbb{C} : \Im(z) \in (\lambda_-, \lambda_+)\}$. With the exception of the pure jump VG (ie when $\sigma = 0$), the Levy processes of interest in finance satisfy the following bound for some $c, \kappa > 0$ and $\nu \in (0, 2]$

$$|\phi_{R_{\Delta_t}}(\xi)| = |e^{\psi_L(\xi)\Delta_t}| \leq \kappa e^{-\Delta_t c|\xi|^\nu}. \tag{1}$$

2.2. Density Recovery and Option Pricing by Frame Projection. In [28], a method of European option pricing, called PROJ, is derived from the theory of frames and Riesz bases. The insight is to project the risk-neutral log return density, given in terms of its ChF, onto a tractable basis of compactly supported functions. The basis is formed by scaling and shifting a fixed *generator* or *scaling function*. The resulting method produces highly accurate localized approximations at low resolutions, where the number of basis elements grows with the resolution. The reader is referred to [28] for more details on the PROJ method, in particular the derivation of dual bases. We refer the reader to [13, 25] for an introduction to frame theory (also see [30] for applications to static hedging).

The B-spline bases of order p are of particular interest, and can be derived as follows. Starting with the Haar scaling function defined by $\varphi^{[0]}(y) := \mathbb{1}_{[-\frac{1}{2},\frac{1}{2}]}(y)$, the p-th order B-spline scaling functions are derived successively by the convolution

$$\varphi^{[p]}(x) = \varphi^{[0]} \star \varphi^{[p-1]}(x) = \int_{-\infty}^{\infty} \varphi^{[p-1]}(y-x)\mathbb{1}_{[-\frac{1}{2},\frac{1}{2}]}(y)dy. \tag{2}$$

With $p = 1$, the linear B-spline basis is generated by

$$\varphi^{[1]}(x) := (1 - |x|)^+ = (1 - |x|)\mathbb{1}_{[-1,1]}(x),$$

while for $p = 2$ we obtain the quadratic scaling function

$$\varphi^{[2]}(y) = \begin{cases} y^2/2 + 3y/2 + 9/8, & y \in [-3/2, -1/2] \\ 3/4 - y^2, & y \in [-1/2, 1/2] \\ y^2/2 - 3y/2 + 9/8, & y \in [1/2, 3/2]\,. \end{cases}$$

To ease notation, we will write $\varphi = \varphi^{[p]}$ when the context is clear.

Given a *resolution* a, and a grid $x_n = x_1 + (n-1)/a$, the approximation space for a fixed generator φ is given by the span of $\varphi_{a,n}(x) = a^{1/2}\varphi(a(x - x_n))$, which is centered over x_n. To derive finite dimensional approximations in terms of $\{\varphi_{a,n}\}_{n=1}^N$ for N fixed, we will truncate the corresponding projections onto the infinite dimensional space $\mathcal{M}_a := \overline{\text{span}}\{\varphi_{a,n}\}_{n\in\mathbb{Z}}$, using the fact that φ satisfies the *frame bounds*

$$A\|f\|^2 \leq \sum_{n\in\mathbb{Z}} |\langle f, \varphi_{a,n}\rangle|^2 \leq B\|f\|^2, \quad \forall f \in L^2(\mathbb{R}), \tag{3}$$

for some $0 < A \leq B$ (independent of a).

2.2.1. Density Projection by Duality. Given a random variable X, with unknown density[3] f_X, we utilize the frame representation theorem [13, 25] which states that the orthogonal projection $P_{\mathcal{M}_a} f_X$ of f_X onto $\mathcal{M}_a$ is given by

$$P_{\mathcal{M}_a} f_X = \sum_{n\in\mathbb{Z}} \langle f_X, \widetilde{\varphi}_{a,n}\rangle \varphi_{a,n},$$

where $\{\widetilde{\varphi}_{a,n}\}_{n\in\mathbb{Z}}$ is the *dual basis*, which is guaranteed to exist in some form. As shown in [28], if the ChF $\phi_X(\xi) := \mathbb{E}[e^{iX\xi}]$ is known, the projection coefficients satisfy for $1 \le n \le N$

$$\langle f_X, \widetilde{\varphi}_{a,n}\rangle = \mathbb{E}[\widetilde{\varphi}_{a,n}(X)] = \frac{a^{-1/2}}{\pi} \Re\left[\int_0^\infty \exp(-ix_n\xi)\cdot\phi_X(\xi)\widehat{\widetilde{\varphi}}\Big(\frac{\xi}{a}\Big)d\xi\right], \tag{4}$$

where

$$\widehat{\widetilde{\varphi}}(\xi) = \mathcal{F}\widetilde{\varphi}(\xi) = \int_{\mathbb{R}} e^{i\xi x}\widetilde{\varphi}(x)dx.$$

When $\widehat{\widetilde{\varphi}}(\xi)$ is known, as for the linear and quadratic generators [28]

$$\widehat{\widetilde{\varphi}}^{[1]}(\xi) = \frac{12\sin^2(\xi/2)}{\xi^2(2+\cos(\xi))}, \qquad \widehat{\widetilde{\varphi}}^{[2]}(\xi) = \frac{480\sin^3(\xi/2)}{\xi^3(26\cos(\xi)+\cos(2\xi)+33)}, \tag{5}$$

the coefficients can thus be calculated efficiently using the fast Fourier transform (FFT), as described next.

When $\phi_X(\xi)$ satisfies a growth estimate of the form of equation (1), the truncation error from numerically integrating (4) will decay exponentially, and polynomially otherwise. Even so, multiplication of the chf by $\widehat{\widetilde{\varphi}}(\xi)$ in equation (4) has a damping effect which reduces aliasing caused by an otherwise insufficient choice of a (the discrete Fourier transform implies a truncation interval of $2\pi a$ in the frequency domain). This is one factor which contributes to accurate approximations at low resolutions.

2.2.2. Coefficient Approximation. To recover the orthogonal projection of the density of a random variable X, the first step is to set a resolution, for example $a = 2^P$ for $P \in \mathbb{N}$. By further specifying $\bar{P} \in \mathbb{N}$, which determines the support width of the projected density, and x_1, which determines its location in log return space, a conceptual grid $x_n = x_1 + (n-1)/a$, $n = 1, \ldots, N$, is designated where

$$N = 2^{P+\bar{P}} = a2^{\bar{P}} := a\bar{a},$$

where the choice of parameters is discussed in Section 3.6. For example, if $\mathbb{E}[X] := \mu_X$, then to center the grid over μ_X, set $x_1 = \mu_X - \left(\frac{N}{2} - 1\right)\Delta$ (where $\Delta := 1/a$), so that $\mu_X = x_{\frac{N}{2}}$. The density is then recovered on

$$[x_1, x_1 + \bar{a} - \Delta] \approx [\mu_X - \bar{a}/2, \mu_X + \bar{a}/2].$$

To discretize the integral in equation (4), by the Nyquist frequency requirement $\Delta\Delta_\xi = 2\pi/N$ the grid in frequency space is set to $\xi_j = (j-1)\Delta_\xi$, $j = 1, \ldots, N$, where

[3] Levy models, with the exception of the compound Poisson process (ie no diffusion component and finite jump activity), possess a continuous density [36].

$\Delta_\xi = 2\pi a/N = 2\pi/\bar{a}$. It is shown in [28] that the *truncated* true projection $\widetilde{f}_X(x)$ is well represented by the numerical approximation $\breve{f}_X(x)$, defined respectively by[4]

$$\widetilde{f}_X(x) := \sum_{n=1}^{N} \langle f_X, \widetilde{\varphi}_{a,n} \rangle \varphi_{a,n}(x), \qquad \breve{f}_X(x) := \sum_{n=1}^{N} \left(a^{1/2} C_{a,N} \breve{\beta}^X_{a,n} \right) \varphi_{a,n}(x),$$

where the coefficients $\langle f_X, \widetilde{\varphi}_{a,n} \rangle \approx a^{1/2} C_{a,N} \breve{\beta}^X_{a,n}$ are calculated by the discrete Fourier transform, in the absence of ChF error[5]:

$$a^{1/2} C_{a,N} \cdot \breve{\beta}^X_{a,n} = \frac{a^{-1/2}}{\pi} \Re \left\{ \sum_{j=1}^{N} \exp(-i x_n \xi_j) \cdot \phi_X(\xi_j) \widehat{\widetilde{\varphi}}\left(\frac{\xi_j}{a} \right) v_j \Delta_\xi \right\}, \tag{6}$$

where $\nu_j := 1 - (\delta_{j,1} + \delta_{j,N})/2$ and $C_{a,N}$ is a constant which depends on the selected generator φ. The full set of $\{\breve{\beta}^X_{a,n}\}_{n=1}^N$ are computed with complexity $\mathcal{O}(N \log_2(N))$ by the FFT.

As long as the numerical error is controlled, the overall convergence of the APROJ algorithm will be at least of the order of projection convergence. Define $\mathcal{H}(\mathcal{D}_d)$ to be the set of analytic functions in the strip $\mathcal{D}_d = \{z \in \mathbb{C} : \Im(z) \in (-d, d)\}$ which satisfy

$$\int_{-d}^{d} |h(x + iy)| dy \to 0, \quad \text{as } |x| \to \infty.$$

For $h \in \mathcal{H}(\mathcal{D}_d)$, we define the norm

$$\|h\|^{\mathcal{H}_d} := \lim_{\epsilon \to 0^+} \left[\int_{\mathbb{R}} |h(x + i(d - \epsilon))| dx + \int_{\mathbb{R}} |h(x - i(d - \epsilon))| dx \right].$$

We have the following result for p^{th} order B-spline generators.

Proposition 2.1. *Suppose that $\phi_X(\xi) \in \mathcal{H}(\mathcal{D}_d)$ for some $d > 0$, and let $\bar{\mu} = \bar{\mu}_X$ be an approximation to $\mathbb{E}[X]$. Fix $a = 2^P$ and $N = a \cdot \bar{a}$, where $\bar{a} = 2^{\bar{P}}$ for $\bar{P} > 1 + \log_2 |\bar{\mu}|$. Fix $x_1 = \bar{\mu} + \left(1 - \frac{N}{2}\right) \frac{1}{a}$. Then for some $0 < \gamma \leq d$*

$$\sup_{1 \leq n \leq N} \left| a^{1/2} C_{a,N} \cdot \breve{\beta}^X_{a,n} - \langle f_X, \widetilde{\varphi}_{a,n} \rangle \right| \leq \frac{a^{-1/2}}{\pi} \left(C_\gamma^{[p]}(\phi_X) \frac{e^{-(\bar{a} - 2|\bar{\mu}|)\gamma/2}}{1 - e^{-\bar{a}\gamma}} + \tau_a(X) \right),$$

where $C_\gamma^{[p]}(\phi_X)$ is a constant. If for some $c, \kappa > 0$ and $\nu \in (0, 2]$, the tail of ϕ_X satisfies

$$|\phi_X(\xi)| \leq \kappa \exp(-tc|\xi|^\nu), \quad \xi \in \mathbb{R}, \tag{7}$$

where $t > 0$ is some fixed time, then

$$\tau_a(X) = \mathcal{O}(a \exp(-tc \cdot (2\pi a)^\nu). \tag{8}$$

In this case, the largest trapezoidal error converges exponentially in $\bar{a}$, while the truncation error is exponential in a. Moreover, when $a > 2d$, $\gamma = d$.

Proof. See appendix. ∎

Note that for the linear basis we have the bound $C_\gamma^{[1]}(\phi_X) \leq 24 \|\phi_X\|^{\mathcal{H}}$ and $\tau_a(X) \leq \frac{6\kappa}{\pi} \cdot a \exp(-tc \cdot (2\pi a)^\nu)$, although the specific constants will not be required for our implementation.

[4] The term $a^{1/2}$ will be absorbed by an intermediate calculation.

[5] Error in the characteristic functions will be introduced.

2.2.3. Quadratic Basis Implementation. To implement the APROJ algorithm, we fix the quadratic basis, although the method applies more generally to p^{th} order B-splines and other generators as well. In particular,

$$f_X(x) \approx a^{1/2} C_{a,N} \sum_{n=1}^{N} \breve{\beta}^X_{a,n} \varphi^{[2]}_{a,n}(x), \quad C_{a,N} := \frac{960a^3}{N}. \tag{9}$$

The coefficients $a^{1/2} C_{a,N} \breve{\beta}^X_{a,n}$ are found using the discretization in equation (6). From the dual generator transform $\widehat{\varphi}^{[2]}(\xi)$ in equation (5), we define

$$H_1 = 1/(960a^3), \qquad H_j = \phi_X(\xi_j)\zeta_j \exp(-ix_1\xi_j), \quad 2 \le j \le N, \tag{10}$$

where

$$\zeta_j := \frac{(\sin(\xi_j/2a)/\xi_j)^3}{26\cos(\xi_j/a) + \cos(2\xi_j/a) + 33}, \quad 2 \le j \le N. \tag{11}$$

The coefficients $\breve{\beta}^X = \{\breve{\beta}^X_n\}_{n=1}^N$ are recovered by the discrete Fourier transform (DFT)

$$\breve{\beta}^X := \Re[\mathcal{D}\{H_j\}_{j=1}^N], \qquad \mathcal{D}_n\{H_j\} := \sum_{j=1}^{N} e^{-i\frac{2\pi}{N}(j-1)(n-1)} H_j, \quad n = 1, \dots, N, \tag{12}$$

For ϕ_X analytic in a strip containing $\mathcal{D}_{(-d,d)}$ with $d > 0$, trapezoidal approximations to the DFT converge exponentially with respect to $a, \bar{a}$, by Proposition 2.1.

2.3. Arithmetic Asian Options. Our main goal is to price discretely monitored arithmetic Asian options, which are contracts on the average over an observed set of prices of an underlying, with observations taken at a discrete set of $M+1$ monitoring dates, $\{0 = t_0, t_1, \dots, t_M = T\}$, with $S_0 = S(t_0)$ observed upon entering the contract. We assume a uniform spacing between observations[6], $t_m = m\Delta t = m\frac{T}{M}$, $m = 0, \dots, M$. If the density of $A_M := \frac{1}{M+1}\sum_{m=0}^{M} S_m$ is known, say f_{A_M}, then the initial value of an option paying $g(A_M)$ at time T must initially satisfy $\mathcal{V} \circ g(S_0) = e^{-rT}\int_{\mathbb{R}} g(u) f_{A_M}(u; S_0) du$.

Fixed strike vanilla Asian options (calls and puts) are priced according to the terminal payoffs with strike $W > 0$

$$g(A_M) := \begin{cases} \left(\frac{1}{M+1}\sum_{m=0}^{M} S_m - W\right)^+, & \text{for a call,} \\ \left(W - \frac{1}{M+1}\sum_{m=0}^{M} S_m\right)^+, & \text{for a put.} \end{cases} \tag{13}$$

By considering a change of numeraire, floating strike arithmetic options can be priced using an analogous formula, but only at inception [18]. On the other hand, frame projection can be used to efficiently obtain bounds on the prices of floating strike arithmetic options in terms of their geometrically averaged counterparts.

[6]This assumption is easily relaxed at a modest increase in cpu time.

3. Mean Adjusted APROJ Method. This section details the APROJ method, which combines elements of several different methods to produce a highly efficient pricing algorithm. The first step is to reduce the problem dimension by employing a technique known as the Carverhill-Clewlow-Hodges factorization [11], which has been utilized as well by [12, 22, 23, 41]. The factorization results in a recursive scheme to recover a single state variable, Y_M, defined by a sequence of intermediate variables $\{Y_m\}_{m=1}^M$. As in [41], we focus on the ChF of this process, which we extend to generalized averaging and in-progress contracts. Analyticity of the chf of Y_m at each stage is proved. To reduce the computational cost and improve accuracy, we explicitly account for the shifting mean of Y_m, by employing an alternative to the lower bound grid shift algorithm proposed in [6]. In particular, we derive upper and lower bounds on the mean of Y_m, and devise an efficient grid shift scheme.

To derive the ChF, we extend the PROJ method of [28]. By utilizing the orthogonally projected density, PROJ obtains highly accurate approximations even at low resolutions. This phenomenon is explained in [38], where for modest resolutions the least squares projection behaves like an interpolation with twice the order of accuracy. Consequently, the use of projected densities results in a substantial reduction in overall cost. Transitioning between time states m requires the calculation of a series of complex valued integrals, for which we derive accurate closed form approximations, taking advantage of the compactly supported basis elements of the PROJ method. In contrast, the globally supported basis elements of a cosine series expansion, for example, require a much more expensive procedure to evaluate the analogous integrals. The resulting algorithm achieves high accuracy at a low computational cost compared with existing methods. Moreover, Greeks are obtained at a negligible added cost.

As illustrated in [8, 15], the choice of truncation width for density expansion methods requires special care. In particular, the cumulant based approach of [19, 41] can produce inaccurate prices for heavy-tailed models. Hence, we propose an alternative method which automatically corrects the grid-width until a supplied truncation tolerance is met. Our two algorithm parameters are determined by an iterative procedure which uses the transform method of [20, 21] to estimate truncation error, as well as a proxy for the integration error incurred at each step. While the cumulant rule of [23] can be used to initialize the grid-width parameter, our method is shown to reliably detect insufficient choices, expanding the grid as needed.

3.1. Change of Variables. The Carverhill-Clewlow-Hodges factorization expresses the average in terms of a random variable Y_M, defined below, so that

$$A_M = \frac{1}{M+1}\sum_{m=0}^{M} S_m = \frac{S_0}{M+1}\left(1+\exp(Y_M)\right). \tag{14}$$

Given an approximation of the density f_{Y_M}, the value of a payoff $g(Y_M; S_0)$ satisfies

$$\mathcal{V}\circ g(S_0) = e^{-rT}\int_{\mathbb{R}} g(y;S_0) f_{Y_M}(y)dy,$$

where for vanilla options

$$g(y) := \begin{cases} \left(\dfrac{S_0(1+\exp(y))}{M+1} - W\right)^+, & \text{for a call,} \\ \left(W - \dfrac{S_0(1+\exp(y))}{M+1}\right)^+, & \text{for a put.} \end{cases} \tag{15}$$

In this way, pricing of a path-dependent Asian option is reduced to the valuation of a European option on the variable Y_M. As will be demonstrated, such a variable can also be found for generalized Asian options with fixed strikes, and for geometric Asian options with fixed and floating strikes (see [28] for the PROJ implementation for geometric Asian options).

The insight of [11] is that the arithmetic average can be expressed as

$$\begin{aligned} A_M &= \frac{S_0}{M+1}\left(1+\frac{S_1}{S_0}\left(1+\frac{S_2}{S_1}\left(\cdots\frac{S_{M-1}}{S_{M-2}}\left(1+\frac{S_M}{S_{M-1}}\right)\right)\right)\right) \\ &= \frac{S_0}{M+1}\left(1+e^{R_1}\left(1+e^{R_2}\left(\cdots e^{R_{M-1}}\left(1+e^{R_M}\right)\right)\right)\right) \\ &= \frac{S_0}{M+1}\left(1+\exp(R_1+\log\left(1+\exp(R_2+\log\left(\cdots R_{M-1}+\log\left(1+\exp(R_M)\right)\right)\right)\right)\right) \end{aligned}$$

where the log return increments are defined by[7]

$$R_m := \log(S_m/S_{m-1}), \quad m = 1,\ldots,M,$$

where we have suppressed the dependence of R_m on the time step $\Delta t = T/M$. By introducing the sequence $\{Y_m\}_{m=1}^M$, defined recursively by

$$Y_1 := R_M, \qquad Y_m := R_{M+1-m} + \log(1+\exp(Y_{m-1})), \quad m = 2,\ldots,M, \tag{16}$$

we have

$$Y_m = \log\left(\frac{1}{S_{(M-m)}}\sum_{j=1}^{m} S_{(M-m)+j}\right), \tag{17}$$

from which it follows that $\exp(Y_M) = \frac{1}{S_0}\sum_{m=1}^M S_m$, and so equation (14) holds. As in [41], we recover the ChF of ϕ_{Y_M} by computing the ChFs of the sequence $\{Y_m\}_{m=1}^M$.

3.2. The Basic Recursion. With $Z_m := \log(1+\exp(Y_m))$, the characteristic function of Y_M is found recursively from $Y_1 = R_M$ by the equation

$$Y_m = R_{M+1-m} + Z_{m-1}, \quad m = 2,\ldots,M. \tag{18}$$

Assuming exponential Levy dynamics, the log return increments R_m are independent, from which independence of R_{M+1-m} and $\log(1+\exp(Y_{m-1}))$ follow. Moreover, stationarity (and uniform monitoring) implies that $R_{M+1-m} = R$ in law for all m, where R has known ChF for many Levy processes. Hence, starting with $\phi_{Y_1}(\xi) = \phi_R(\xi)$, the ChF of Y_m is derived from that of Y_{m-1} using equation (18):

$$\phi_{Y_1}(\xi) = \phi_R(\xi), \qquad \phi_{Y_m}(\xi) = \phi_R(\xi)\phi_{Z_{m-1}}(\xi), \quad m = 2,\ldots,M.$$

Specifically,

$$\phi_{Z_{m-1}}(\xi) := \mathbb{E}\left[e^{i\xi\log(1+\exp(Y_{m-1}))}\right] = \int_{\mathbb{R}}(e^y+1)^{i\xi} f_{Y_{m-1}}(y)dy, \tag{19}$$

where $f_{Y_{m-1}}$ is approximated using $\phi_{Y_{m-1}}$.

The next result will ensure that the DFT errors, which are incurred at each density projection step, converge exponentially with respect to $a, \bar{a}$.

[7] We reserve the notation R to denote the return distribution over a time increment of size Δ_t, while R_m denotes the return random variable itself. To make the dependence on Δ_t explicit, we will at times use R_{Δ_t} to denote a generic return increment.

Proposition 3.1. *Suppose that $\phi_R(z)$ is analytic in the strip $\mathcal{D}_d := \{z \in \mathbb{C} : \Im(z) \in (-d, d)\}$, for some $d > 0$, and satisfies equation (1) for some $\kappa, c > 0$ and $\nu \in (0, 2]$. If $\{Y_m\}_{m=1}^M$ are defined by equation (16), then the ChFs satisfy*

(i) ϕ_{Y_m} is analytic in $\mathcal{D}_d$, $1 \le m \le M$, and

(ii) $|\phi_{Y_m}(\xi)| \le \kappa e^{-\Delta_t c|\xi|^\nu}$, $\xi \in \mathbb{R}$, $1 \le m \le M$.

Hence, the domain of analyticity and the decay of ϕ_{Y_m} are independent of m.

Proof. See appendix. ∎

It should also be noted that $f_{Y_m}(y) \sim e^{-d|y|}$ as $|y| \to \infty$, ie the densities have exponentially decaying tails[8], determined by the tail behavior of f_R. This follows since analyticity of ϕ_{Y_m} in $\mathcal{D}_d$ implies that $\mathbb{E}[e^{\eta Y_m}] < \infty$ for $\eta \in (-d, d)$. In particular, we are dealing with densities of rapid decrease.

3.3. APOJ Algorithm Overview. Before developing the APROJ algorithm in detail, we present the main blocks with references to their derivation in the text:

1. To account for the shifting mean of Y_m, a grid shift algorithm is derived in Section 3.4
2. The initial ChF ϕ_{Z_1} is obtained in terms of the closed form ChF ϕ_R in Section 3.5.1, where we introduce the integral matrix Ψ
3. The ChFs $\phi_{Z_{m-1}}$ are obtained recursively in Section 3.5.2
4. Given $\phi_{Z_{m-1}}$, we obtain ϕ_{Y_m} in Section 3.5.3
5. An automated method of parameter selection is detailed in Section 3.6, which is summarized by initialization Subroutine 1
6. An approximation of the integral matrix Ψ is given in Section 3.7, which is summarized by Subroutine 2
7. The final valuation step (which applies to general payoffs) is presented in Section 3.8, after recovering ϕ_{Y_M}
8. Formulas for vanilla option Greeks are provided in Section 3.10

After developing the main algorithm blocks, in Section 3.9 we summarize the routine in Algorithm 3, which calls initialization Subroutine 1 to determine parameters, and Subroutine 2 to populate the integral matrix Ψ.

3.4. Mean-adjusted Grid. We employ a grid shift to ensure that we capture to within a set tolerance the mass of $f_{Y_{m-1}}$, while the grid specific to each Y_m will belong to a single enlarged grid, for $m = 1, \dots, M-1$. The final grid corresponding to Y_M will vary slightly according to the payoff to be priced. Since the distribution of $Y_1 = R_{\Delta_t}$ is roughly centered about its mean, a natural starting grid in log return space is fixed by centering about

$$\mathbb{E}[R_{\Delta_t}] = (r - q + \omega + \mathbb{E}[L(1)])\Delta_t = c_1 \Delta_t,$$

where $c_1 = \mathbb{E}[\log(S_{t+1}/S_t)]$ is the first cumulant of log return over a unit interval, and is provided in Table 6 for common processes. For example, the Black-Scholes-Merton (BSM) model satisfies $\mathbb{E}[R_{\Delta_t}] = (r - q - \sigma^2/2)\Delta_t$, where σ is the rate of volatility.

The approach of Benhamou [6] is to approximate the mean $\mathbb{E}[Y_m] = \mathbb{E}[R_{\Delta_t}] + \mathbb{E}[\log(1 + e^{Y_{m-1}})]$ by

$$\mu_1^B := \mathbb{E}[R_{\Delta_t}], \quad \mu_m^B = \mu_1^B + \log\left(1 + e^{\mu_{m-1}^B}\right), \quad m = 2, \dots, M. \tag{20}$$

[8] The rate of decay could be faster than d, but this gives a conservative estimate.

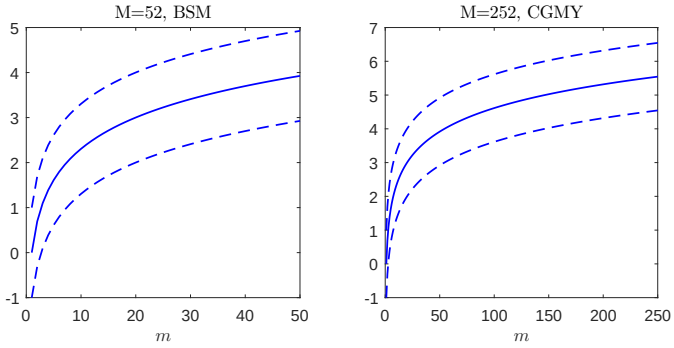

Figure 1: Plot of $\tilde{\mu}_m$, the approximated mean of Y_m, as a function of m with $r = .05$, $q = 0$ in the BSM $\sigma = 0.3$ model (Left) and the CGMY $= (0.27, 17.5, 54.8, 0.8)$ model (Right). The bounds $\bar{\mu}_m \pm \bar{a}/2$ are given by dashed lines, where $\bar{a} = 2$.

By convexity of $\log(1 + e^y)$, Jensen's inequality implies $\log(1 + \exp(\mathbb{E}[Y_{m-1}])) \leq \mathbb{E}[\log(1 + \exp(Y_{m-1}))]$, so the mean shift underestimates the true mean. We employ an alternative grid-shift scheme, described next.

3.4.1. Grid Shift and Bounds. As an alternative to the grid adjustment of [6], and to bound the growth of the grid shift, we derive an upper bound on $\mathbb{E}[Y_m]$ by applying Jensen's inequality(for *concave* functions) to equation (17):

$$\mathbb{E}[Y_m] \leq \log\left(\sum_{j=1}^{m} \mathbb{E}\left[\frac{S_{(M-m)+j}}{S_{(M-m)}}\right]\right) = (r-q)\Delta_t + \log\left(\frac{\exp\left((r-q)\Delta_t m\right) - 1}{\exp\left((r-q)\Delta_t\right) - 1}\right), \tag{21}$$

since $e^{(r-q)j\Delta t} = \mathbb{E}\left[\frac{S_{(M-m)+j}}{S_{(M-m)}}\middle|\mathcal{F}_{M-m}\right] = \mathbb{E}\left[\frac{S_{(M-m)+j}}{S_{(M-m)}}\right]$, where the first equality follows from the martingale property and the second from the fact that Levy increments are independent of the current filtration, $\mathcal{F}_{M-m}$. Similarly,

$$Y_m \geq \log(m) + \frac{1}{m}\sum_{j=1}^{m} \log\left(\frac{S_{(M-m)+j}}{S_{(M-m)}}\right),$$

from which we derive $\mathbb{E}[Y_m] \geq \log(m) + \mathbb{E}[R_{\Delta_t}]\frac{m+1}{2}$. In particular, we obtain a set of upper and lower bounds on the growth of $\mathbb{E}[Y_m]$.

Proposition 3.2. *With μ_m^B defined by equation (20), and $\theta := (r-q)\Delta_t$ we have*

(22)
$$\begin{aligned}\mathbb{E}[R_{\Delta_t}]\frac{m+1}{2} + \log(m) &\leq \mathbb{E}[R_{\Delta_t}] + \log\left(1 + e^{\mu_{m-1}^B}\right)\\ &\leq \mathbb{E}[Y_m]\end{aligned}$$

(23)
$$\leq \theta + \log\left(\frac{\exp(\theta m) - 1}{\exp(\theta) - 1}\right) \leq \log(m) + \theta(m\mathbb{1}_{r\geq q} + \mathbb{1}_{r<q}).$$

With $\mu_0^B := 0$, these bounds hold for all $1 \leq m \leq M$.

Proof. Both inequalities in equation (23) follow from equation (21). To prove equation (22), define $\theta_m = \log(m) + \rho\frac{m+1}{2}$, where $\rho := \mathbb{E}[R_{\Delta_t}]$. We show that

$\mu_m^B \geq \theta_m$ by proving $\exp(\mu_m^B - \theta_m) \geq 1$ inductively, where the case of $m = 1$ holds trivially. For $m \geq 2$,

$$\begin{aligned}\exp(\mu_m^B - \theta_m) &= e^{\rho}\left(1 + e^{\mu_{m-1}^B}\right) e^{-\rho(m+1)/2}\tfrac{1}{m} \\ &\geq e^{\rho}\left(1 + e^{\rho m/2 + \log(m-1)}\right) e^{-\rho(m+1)/2}\tfrac{1}{m} \\ &= e^{\rho/2}\left(e^{-\rho m/2}\tfrac{1}{m} + \tfrac{m-1}{m}\right) \qquad (24)\end{aligned}$$

where the inequality follows by the inductive hypothesis. For $m = 2$, equation (24) becomes

$$\exp(\mu_2^B - \theta_2) \geq e^{\rho/2}\left(e^{-\rho}\tfrac{1}{2} + \tfrac{1}{2}\right) = \cosh(\rho) \geq 1.$$

It is thus sufficient to show that the lower bound in (24) is nondecreasing in m. In particular,

$$\frac{d}{dm} e^{\rho/2}\left(e^{-\rho m/2}\tfrac{1}{m} + \tfrac{m-1}{m}\right) = \frac{e^{\rho(1-m)/2}}{2m^2}\left(2\left(e^{\rho m/2} - 1\right) - \rho m\right).$$

Since $e^{\rho(1-m)/2}/2m^2 > 0$, the result follows from the fact that the second term $2\left(e^{\rho m/2} - 1\right) - \rho m := 2(e^{\lambda/2} - 1) - \lambda$ has a global minimum at $\lambda = 0$. That is, for any $\rho \neq 0$ fixed (the case of $\rho = 0$ follows immediately), the derivative is a nondecreasing function of m, and equation (22) is proved. ∎

An immediate consequence of Proposition 3.2 is that we obtain a priori a corridor in which $\mathbb{E}[Y_m]$ lies for all $1 \leq m \leq M$, in terms of the mean return and $(r - q)$:

$$\mathbb{E}[R_{\Delta_t}]\frac{m+1}{2} \leq \mathbb{E}[Y_m] - \log(m) \leq |r - q|T\frac{m}{M}, \quad \forall m \leq M. \qquad (25)$$

Hence, $\mathbb{E}[Y_m] = \log(m) + \mathcal{O}(m|r - q|\Delta_t)$ and the growth in $\mathbb{E}[Y_m]$ is no faster than $\log(m)$, independently of M (the second term is always bounded by $|r-q|T$). We also note that the upper bounds in equation (23) can be applied when $\mathbb{E}[R_{\Delta_t}]$ is unknown.

3.4.2. Grid Shift Algorithm. The APROJ grid shift is implemented by combining the innermost upper and lower bounds of Proposition 3.2. In particular, with $\mu_1^B = \mathbb{E}[R_{\Delta_t}] = c_1\Delta_t$ (see Table 6), and for $m = 2, \ldots, M$

$$\mu_m^B := \mu_1^B + \log\left(1 + e^{\mu_{m-1}^B}\right), \qquad \mu_m^U := (r - q)\Delta_t + \log\left(\frac{\exp\left((r - q)\Delta_t m\right) - 1}{\exp\left((r - q)\Delta_t\right) - 1}\right),$$

we define our grid as the lower-upper bound average

$$\tilde{\mu}_1 := c_1\Delta_t, \qquad \tilde{\mu}_m := (\mu_m^B + \mu_m^U)/2, \quad m = 2, \ldots, M, \qquad (26)$$

with maximum grid shift error $|\mathbb{E}[Y_m] - \tilde{\mu}_m| \leq (\mu_m^U - \mu_m^B)/2$.

In order to reduce the computations required below (namely in computing a matrix Ψ), we perturb each $\tilde{\mu}_m$ slightly to obtain $\bar{\mu}_m$, which belongs to an extension of the initial grid defined by $\tilde{\mu}_1$:

$$\bar{\mu}_m := \tilde{\mu}_1 + N_m\Delta, \quad N_m := \lfloor a(\tilde{\mu}_m - \tilde{\mu}_1)\rfloor, \quad m = 2, \ldots, M, \qquad (27)$$

and $\bar{\mu}_1 \equiv \tilde{\mu}_1$, $N_1 := 0$. Hence, we define the mean-adjusted grids

$$x_n^m = x_1^m + (n - 1)\Delta, \quad x_1^m := \bar{\mu}_m + (1 - N/2)\,\Delta, \quad m = 1, \ldots, M - 1, \qquad (28)$$

each corresponding to a subset of the linear basis $\{\varphi_{a,n}\}_{n=1}^{N+N_{M-1}}$, with $\varphi_{a,1}$ centered over x_1^1. In particular, the density of Y_m is recovered over $[\bar{\mu}_m - \bar{a}/2, \bar{\mu}_m + \bar{a}/2]$, $m = 1, \ldots, M$, which is illustrated in Figure 1. The choice of x_1^M will be detailed in Section 3.8. To implement the algorithm, only $\{x_1^m\}_{m=1}^M$ and $\{N_m\}_{m=1}^M$ are needed (there is no need to actually generate the grids at each stage).

3.5. Characteristic Function Recovery. We now derive the ChF recovery by successive PROJ expansions on the mean-adjusted grid. The algorithm is summarized in Section 3.9, along with a discussion of its complexity. In the algorithm description, we will denote by $\bar{\beta}_X$ the DFT approximation in the presence of ChF error, to distinguish it in the error analysis from $\breve{\beta}_X$ (which is absent ChF error).

3.5.1. Initialization. To initialize the characteristic function recursion we need

$$\phi_{Z_1}(\xi) := \mathbb{E}\left[e^{i\xi \log(1+\exp(R))}\right] = \int_{\mathbb{R}} (e^y + 1)^{i\xi} f_R(y) dy.$$

Since $\phi_R(\xi)$ is known, $\phi_{Z_1}(\xi)$ is approximated by a (quadratic) PROJ expansion of $f_R(y)$, with coefficients $\breve{\beta}_n = \breve{\beta}_{a,n}$ to yield[9]

$$\begin{aligned}
\phi_{Z_1}(\xi) &\approx \int_{\mathbb{R}} (e^y + 1)^{i\xi} \left(a^{1/2} C_{a,N} \sum_{n=1}^{N} \breve{\beta}_n \varphi_{a,n}(y) \right) dy \\
&= C_{a,N} \sum_{n=1}^{N} \breve{\beta}_n \cdot a^{1/2} \int_{I_n} (e^y + 1)^{i\xi} \varphi_{a,n}(y) dy \\
&\approx C_{a,N} \sum_{n=1}^{N} \bar{\beta}_n^1 \cdot \bar{\Psi}(\xi, n) := \bar{\phi}_{Z_1}(\xi), \qquad (29)
\end{aligned}$$

where for the quadratic basis $I_n := [x_n^1 - 3\Delta/2, x_n^1 + 3\Delta/2]$ and $C_{a,N} = 960a^3/N$.

With the initial grid implied by the choice of $x_1^1 = \mathbb{E}[R] + (1 - N/2)\Delta$, so that ϕ_{Z_1} is approximated by a projected expansion of f_R about $\mathbb{E}[R]$, the column vector $\bar{\beta}^1$ is determined by

$$\bar{\beta}^1 := \Re[\mathcal{D}\{H_j^1\}_{j=1}^N], \qquad H_j^1 := \phi_R(\xi_j) \cdot \zeta_j \exp(-i x_1^1 \xi_j), \quad j = 2, \ldots, N, \qquad (30)$$

where $H_1^1 = 1/(960a^3)$ and ζ_j is defined in equation (11). Further,

$$\Psi(\xi, n) := a^{1/2} \int_{I_n} (e^y + 1)^{i\xi} \varphi_{a,n}(y) dy, \quad n = 1, \ldots, N + N_{M-1}, \qquad (31)$$

and $\bar{\Psi}(\xi, n)$ denotes a Newton-Cotes approximation to $\Psi(\xi, n)$ (discussed in Section 3.7). From here we obtain $\bar{\phi}_{Y_2}(\xi) = \phi_R(\xi)\bar{\phi}_{Z_1}(\xi)$, which concludes the initialization.

Remark 1. *As demonstrated in Figure 2, for increasing x_n the columns in $\Psi(\xi, n)$ become progressively closer to the values of $a^{1/2}\mathcal{F}[\varphi_{a,n}]$ on $[0, 2\pi a)$. This is illustrated with the linear basis in terms of the scaled modulus*

$$\theta(\xi) := \left(\frac{\sin(\xi/2a)}{\xi/(2a)} \right)^2 = a^{1/2} \left| a^{-1/2} e^{i x_n \xi} \left(\frac{\sin(\xi/2a)}{\xi/(2a)} \right)^2 \right| = a^{1/2} |\mathcal{F}[\varphi_{a,n}]|,$$

[9] We use the notation $\bar{\beta}^1$ here to be consistent with $\bar{\beta}^m$, $m \geq 2$, although it should be noted that $\bar{\beta}^1 = \breve{\beta}^1$ in this case since ϕ_R is known.

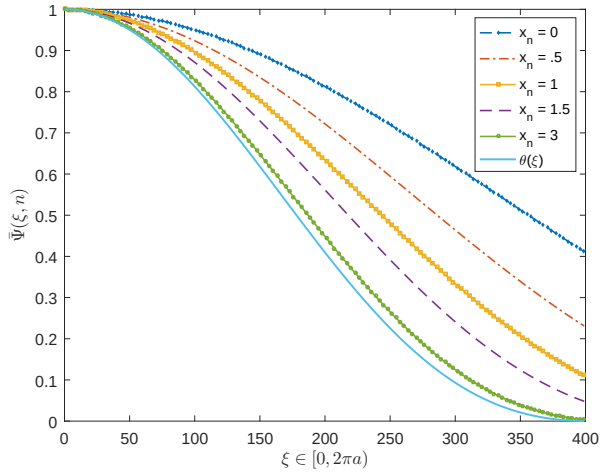

Figure 2: Convergence in x_n of $\Psi(\xi, n)$ to $a^{1/2}\mathcal{F}[\varphi_{a,n}](\xi)$, a plot of the modulus.

and reflects the fact that

$$\left| \int_{I_n} (e^y + 1)^{i\xi} \varphi_{a,n}(y) dy - \int_{I_n} e^{i\xi y} \varphi_{a,n}(y) dy \right| \to 0, \quad \text{as } x_n \to +\infty.$$

For a p^{th} order B-spline basis, we have the following characterization for large x_n.

Lemma 3.1. *With $a > 0$ fixed, the elements of $\bar{\Psi}$ behave as*

$$\Psi(\xi, n) \sim e^{ix_n\xi} \left(\frac{\sin(\xi/2a)}{\xi/(2a)} \right)^{(p+1)} + \mathcal{O}(a \cdot \exp(-x_{n-1})),$$

when x_n is large, with respect to the B-spline basis of order p.

Proof. See appendix. ∎

Especially when M is large (in which case a significant portion of $\bar{\Psi}$ will be well approximated by Lemma 3.1), the algorithm can be improved to use this result.

3.5.2. Recovery of $\phi_{Z_{m-1}}$. From the definition of Z_{m-1}, the characteristic function is approximated in terms of the PROJ expansion of $f_{Y_{m-1}}$, recovered over $[\bar{\mu}_{m-1} - \frac{\bar{a}}{2}, \bar{\mu}_{m-1} + \frac{\bar{a}}{2}]$, and corresponding to the subset of basis elements $\varphi_{a,n}$, $n = N_{m-1} + 1, \ldots, N_{m-1} + N$:

$$\begin{aligned} \phi_{Z_{m-1}}(\xi) &= \int_{\mathbb{R}} (e^y + 1)^{i\xi} f_{Y_{m-1}}(y) dy \\ &\approx \int_{\mathbb{R}} (e^y + 1)^{i\xi} \left(a^{1/2} C_{a,N} \sum_{n=1}^{N} \bar{\beta}_n^{m-1} \varphi_{a,N_{m-1}+n}(y) \right) dy \\ &\approx C_{a,N} \sum_{n=1}^{N} \bar{\beta}_n^{m-1} \cdot \bar{\Psi}(\xi, N_{m-1} + n) := \bar{\phi}_{Z_{m-1}}(\xi). \end{aligned} \tag{32}$$

As before, the grid is fixed by x_1^{m-1}, and the column vector $\bar{\beta}^{m-1} := \Re[\mathcal{D}\{H_j^{m-1}\}_{j=1}^N]$ is determined via

$$H_1^{m-1} = 1/(960a^3), \quad H_j^{m-1} := \bar{\phi}_{Y_{m-1}}(\xi_j) \cdot \zeta_j \exp(-ix_1^{m-1}\xi_j), \quad j = 2, \ldots, N. \tag{33}$$

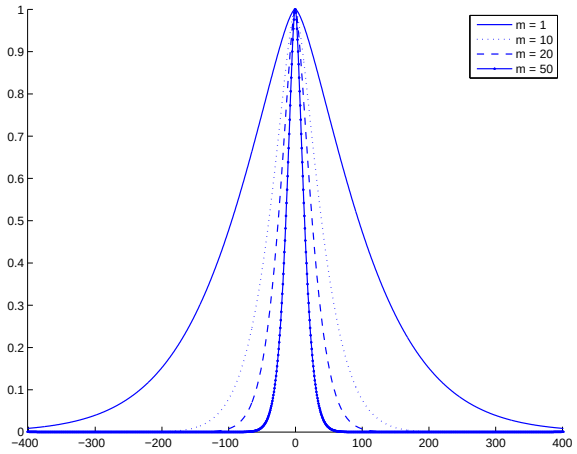

Figure 3: Modulus of ϕ_{Y_m} with $\Delta_t = 1/50$ for $(C, G, M, Y) = (.0244, .0765, 7.5515, 1.2945)$, and $r = .0367$. The x-axis: $\xi \in [-2\pi a, 2\pi a]$, $\Delta_\xi = 2\pi/\bar{a}$, where $a = 2^6$, $\bar{a} = 2^3$.

In fact, we only need the values of $\bar{\phi}_{Y_m}(\xi)$ for the discrete set of points $\xi_j = (j-1)\Delta_\xi$, $j = 1, \ldots, N$. Accordingly, if we define the $N \times (N_{M-1} + N)$ matrix $\bar{\Psi}$ by

$$\bar{\Psi}(j, n) := \bar{\Psi}(\xi_j, n), \quad j, n = 1, \ldots, N_{M-1} + N,$$

the computation at each stage can be represented as

$$\bar{\Phi}_{Z_{m-1}} = C_{a,N} \bar{\Psi}_{m-1} \bar{\beta}^{m-1} \tag{34}$$

where $\bar{\Phi}_{Z_{m-1}} = (\bar{\phi}_{Z_{m-1}}(\xi_1), \ldots, \bar{\phi}_{Z_{m-1}}(\xi_N))^\top$, and for $m = 2, \ldots, M$,

$$\bar{\Psi}_{m-1}(j, n) = \bar{\Psi}(j, N_{m-1} + n), \quad j, n = 1, \ldots, N.$$

Here, $\bar{\Psi}_{m-1}$ is defined for notational compactness and to indicate that only a subset of $\bar{\Psi}$ takes part in the matrix-vector product.

3.5.3. Recovery of ϕ_{Y_m}.

To determine $\bar{\Phi}_{Y_m}$, equation (32) yields

$$\bar{\phi}_{Y_m}(\xi) = \bar{\phi}_{Z_{m-1}}(\xi)\phi_R(\xi) = C_{a,N} \sum_{n=1}^{N} \bar{\Psi}(\xi, N_{m-1} + n) \cdot \bar{\beta}_n^{m-1} \cdot \phi_R(\xi). \tag{35}$$

In matrix form the algorithm reads

$$\Phi_R^C := C_{a,N} \Phi_R, \qquad \bar{\Phi}_{Y_m} = \left(\bar{\Psi}_{m-1} \bar{\beta}^{m-1}\right) \circ \Phi_R^C, \quad m = 2, \ldots, M, \tag{36}$$

where $\circ$ denotes the Hadamard product and $\Phi_R = (\phi_R(\xi_1), \ldots, \phi_R(\xi_N))^\top$.

An example of the modulus of recovered ChFs for the CGMY model with $M = 50$ is given in Figure 3, where the line corresponding to $m = 1$ is just $|\phi_R(\xi)|$. Notice how the ChFs collapse about the origin as m approaches M. The reflects the fact that, as m increases, the density of f_{Y_m} becomes less peaked (ie smoother), which translates into a more rapid decay of ϕ_{Y_m}.

3.6. Parameter Selection. The two parameters required to apply the APROJ method are $\bar{a}$ and N (or equivalently Δ). For several experiments in the numerical section, we fix a value of $\bar{a} = 2^{\bar{P}}$ (often excessively large to isolate the resolution error) and increase the parameter $a = 2^P$, which allows us to illustrate the convergence behavior as a function of resolution.[10] For example, Figure 7 in appendix illustrates the convergence in a for several levels of $\bar{a}$ fixed.

This section provides an automated approach to parameter selection, requiring no user input, which should facilitate implementation in practical pricing scenarios. We first fix an initial value for N and truncation multiplier L_1. For $\Delta_t \geq 1/60$ we find that $N = 2^6$ and $L_1 = 12$ provide good starting values. Similarly we initialize $N = 2^7$ and $L_1 = 16$ for $\Delta_t < 1/60$. We then initialize $\bar{a}$ given according to the cumulants of R_{Δ_t}, as proposed in [19] (without the lower bound):

$$\bar{a} \leftarrow 2 \cdot \max\left\{1,\ L_1\sqrt{c_2\Delta_t + \sqrt{c_4\Delta_t}}\right\}, \tag{37}$$

and set $\Delta \leftarrow \bar{a}/N$ (see Table 6 for c_n). Since the cumulant based approach of [19] has been shown to produce unsatisfactory estimates for certain heavy-tailed models (see for example [8, 15]), the value of $\bar{a}$ in (37) serves only as an initial guess in our algorithm, and alternative initial choices for $\bar{a}$ are also feasible. To determine if the intial choice of $\bar{a}$ and N are sufficient, we estimate the truncation error, with a tolerance ϵ_1, and a proxy for the valuation error, with a tolerance ϵ_2, increasing the values for N and $\bar{a}$ according to a set of rules. The routine is fully automated, and produces a final estimate for $\bar{a}$ and N to satisfy the user tolerance levels ϵ_1, ϵ_2.

First we estimate the truncation error. As shown in [20] (see also [21]), the probability mass of a random variable, here R_{Δ_t}, over an interval $[l, u]$ is given by

$$\mathbb{P}[l < R_{\Delta_t} < u] = \int_{-\infty}^{\infty} e^{-i\xi(u+l)/2}\frac{\sin(\xi(u-l)/2)}{\pi\xi}\phi_{R_{\Delta_t}}(\xi)d\xi.$$

Fixing $N > 0$ and $\Delta_\xi > 0$, we have $\mathbb{P}[l < R_{\Delta_t} < u] \approx F_{\Delta_\xi,N}(l, u)$, where

$$F_{\Delta_\xi,N}(l,u) := \frac{\Delta_\xi}{\pi}\left[\gamma_1 + \sum_{1\leq|n|\leq N-1} e^{-i\gamma_2(n\Delta_\xi)}\frac{\sin(\gamma_1(n\Delta_\xi))}{n\Delta_\xi}\phi_{R_{\Delta_t}}(n\Delta_\xi)\right]$$

where $\gamma_1 = (u-l)/2$ and $\gamma_2 = (u+l)/2$. From Section 3.4.2, we know the grid shift error is bounded by $|\mathbb{E}[Y_m] - \tilde{\mu}_m| \leq (\mu_m^U - \mu_m^B)/2 := \tau_m$, and in practice we find that τ_M is the largest such error. Hence, given a grid estimate (l, u), we estimate the mass of f_R on $(\tilde{l}, \tilde{u}) = (l + \tau_M, u - \tau_M)$. If $|1 - F_{\Delta_\xi,N}(\tilde{l}, \tilde{u})| > \epsilon_1$, we double the grid size N, set $\bar{a} \leftarrow \sqrt{2}\bar{a}$, and reestimate.

As a second verification, by the martingale property of $e^{-(r-q)t}S_t$, we can utilize the following estimate to obtain a proxy for integration error incurred at each step:

[10] One could use the value of $\bar{a} = 2^{\bar{P}}$ prescribed in Corollary 5.1 which ensures $\bar{a} - 2|\bar{\mu}_M| > 0$, and hence the exponential convergence in $\bar{a}$; it is usually around $\bar{P} = 3$ for $M \leq 50$, or $\bar{P} = 4$ when $M = 250$. Since this controls the *largest* coefficient error, it tends to be conservative although robust for heavy tailed returns (for BSM, $\bar{P} = 2$ is more than sufficient for $M \leq 250$ and $\sigma \leq .5$, and practical accuracy of greater than e-04 is achieved with $\bar{P} = 0 \sim 1$). In practice, a conservative rule of thumb is to choose $\bar{P} = 4$ for heavy tailed distributions, and $\bar{P} = 1$ for diffusion models.

$$E_N := C_{a,N} \cdot \vartheta_*^{[2]} \cdot \sum_{n=1}^{N} \bar{\beta}_n^1 \exp(x_1^1 + (n-1)\Delta) = \int \breve{f}_{R_{\Delta_t}}(x) e^x dx \tag{38}$$

where $\vartheta_*^{[2]}$ is defined in Table 1. E_N approximates $\mathbb{E}[\exp(R_{\Delta_t})] = \exp((r-q)\Delta_t)$ using the projected density. Hence, once the truncation criterion is satisfied, we will further double the grid size as long as $|E_N - \exp((r-q)\Delta_t)| \cdot M > \epsilon_2$. The multiplier M is to account for the number of such approximations made during the algorithm. The resulting initialization routine is summarized in Subroutine 1. After the main algorithm, a final check will be made (see Remark 3).

Note that the parameter $\epsilon_1 =$ 5e-04, along with $\epsilon_2 =$ 5e-04 are set in Subroutine 1 to satisfy an overall valuation error tolerance of TOL:= 5e-04 or better, uniformly across models, and tends to be conservative. This is illustrated in Table 9 of the numerical section. The contract maturity is incorporated in our algorithm by the ChF $\phi_{R_{\Delta_t}}$ used to estimate the truncation and integral errors, which is a function of $\Delta_t = T/M$. Table 10 illustrates the performance for maturities of two and four years.

Subroutine 1 Initialization by automated parameter selection

For $\Delta_t \geq 1/80$, Set: $L_1 = 12, N = 2^6$; For $\Delta_t < 1/80$, Set: $L_1 = 16, N = 2^7$
Set error tolerances $\epsilon_1 =$ 5e-04; $\epsilon_2 =$ 5e-04
Calculate cumulants c_1, c_2, c_4 of R_1 (see Table 6)
$\tilde{\mu}_1 \leftarrow c_1\Delta_t$; $\theta \leftarrow (r-q)\Delta_t$; $\mu_1^B \leftarrow c_1\Delta_t$
Initialize $\bar{a} \leftarrow 2 \cdot \max\left\{1,\ L_1\sqrt{c_2\Delta_t + \sqrt{c_4\Delta_t}}\right\}$
Set(Δ, a, Δ_ξ): $\Delta \leftarrow \bar{a}/N$; $a \leftarrow 1/\Delta$; $\Delta_\xi \leftarrow 2\pi/2\bar{a}$
for $m = 2 \ldots M$ **do**
  $\mu_m^B \leftarrow \mu_1^B + \log\left(1 + e^{\mu_{m-1}^B}\right)$; $\tilde{\mu}_m \leftarrow \frac{1}{2}\left(\mu_m^B + \theta + \log\left(\frac{\exp(\theta m)-1}{\exp(\theta)-1}\right)\right)$
end for
Max grid shift error: $\tau_M := \frac{1}{2}\left(\theta + \log\left(\frac{\exp(\theta M)-1}{\exp(\theta)-1}\right) - \mu_M^B\right)$
$x_1^1 \leftarrow \tilde{\mu}_1 + (1 - N/2)\Delta$
$l \leftarrow x_1^1 + \tau_M$; $u \leftarrow (x_1^1 + \bar{a}) - \tau_M$; $\gamma_1 \leftarrow \frac{u-l}{2}$; $\gamma_2 \leftarrow \frac{u+l}{2}$
while $|1 - F_{\Delta_\xi,N}(l,u)| > \epsilon_1$ **do**
  $N \leftarrow 2N$; $\bar{a} \leftarrow \sqrt{2}\bar{a}$; Set$(\Delta, a, \Delta_\xi)$
  $x_1^1 \leftarrow \tilde{\mu}_1 + (1 - N/2)\Delta$; Update: l, u, γ_1, γ_2
end while
$\{\xi_j\}_{j=1}^N = (j-1)\Delta_\xi$, $\Phi \leftarrow \{\phi_R(\xi_j)\}_{j=1}^N$; Calculate $\{\zeta_j\}_{j=2}^N$ from (11)
Calculate $\{H_j\}_{j=1}^N$ from (30); $\{\bar{\beta}_n\}_{n=1}^N \leftarrow \Re\{\text{FFT}\{H_j\}_{j=1}^N\}$
Calculate E_N from (38)
while $|E_N - \exp(\theta)| \cdot M > \epsilon_2$ **do**
  $N \leftarrow 2N$; $\bar{a} \leftarrow \sqrt{2}\bar{a}$; Set$(\Delta, a, \Delta_\xi)$
  Recalculate: $\{\xi_j\}_{j=1}^N$, Φ, $\{\zeta_j\}_{j=2}^N$, $\{H_j\}_{j=1}^N$, $\{\bar{\beta}_n\}_{n=1}^N$ and E_N
end while
$N_m \leftarrow \lfloor a(\tilde{\mu}_m - \tilde{\mu}_1)\rfloor$; $x_1^m \leftarrow (\tilde{\mu}_1 + N_m\Delta) + (1 - N/2)\Delta$, $m = 1, \ldots, M$

3.7. Approximation of Ψ. We now discuss the numerical integration of the matrix Ψ. From equation (31), for $j = 1, \dots, N$, From equation (31), for $j = 1, \dots, N$,

$$\Psi(j,n) := a^{1/2} \int_{I_n} (e^y + 1)^{i\xi_j} \varphi_{a,n}(y) dy, \quad n = 1, \dots, N + N_{M-1},$$

which we approximate by $\bar{\Psi}$ using Newton-Cotes quadrature. By fixing the grids with $\{x_1^m\}_{m=1}^{M-1}$ defined by equation (28), each can be considered as a subset of $x_1^1 + (n-1)\Delta$, $n = 1, \dots, N + N_{M-1}$, so quadrature points (and function evaluations) can be reused in subsequent approximations. Moreover, the induced grid overlap reduces the computation[11] of $\bar{\Psi}$ from $N \times ((M-1)N)$ elements to $N \times (N + N_{M-1}) \leq N \times (\log(M-1)N)$ (see Section 3.9).

To obtain the matrix $\bar{\Psi}$ we evaluate the integrals by applying a seven point Newton-Cotes rule to each subinterval I_n^l, $l = 1, 2, 3$, where

$$I_n := [x_n - 3\Delta/2, x_n - \Delta/2] \cup [x_n - \Delta/2, x_n + \Delta/2] \cup [x_n + \Delta/2, x_n + 3\Delta/2] := I_n^1 \cup I_n^2 \cup I_n^3.$$

Combined with the known values of $\varphi^{[2]}(y)$ at each quadrature point, this results in the (composite) seven-point rule on I_n

$$\begin{aligned} Q(\nu) = \frac{1}{840} \Big\{ & 3\left[\nu_1 + \nu_{17} + 25(\nu_5 + \nu_{13}) + 46(\nu_7 + \nu_{11})\right] \\ & + \frac{27}{18}\left[\nu_2 + \nu_{16} + 4(\nu_4 + \nu_{14}) + 13(\nu_8 + \nu_{10})\right] \\ & + 34\left[\nu_3 + \nu_{15} + 6\nu_9\right] + 41\left[\nu_6 + \nu_{12}\right] \Big\}, \end{aligned}$$

where ν is defined in Subroutine 2, and represents generic values of the integrand for some (j, n) fixed.[12].

Subroutine 2 Calculation of $\bar{\Psi}$

$N_\eta = 17 + 6(N + N_{M-1} - 1)$
$\eta_k \leftarrow x_1^1 + (k-9)\Delta/6, \quad k = 1, \dots, N_\eta$
$\theta_k \leftarrow \exp\left(i\Delta_\xi \log\left(1 + \exp(\eta_k)\right)\right), \quad k = 1, \dots, N_\eta$
$\eta \leftarrow \theta$
$\bar{\Psi}(1,n) \leftarrow 1, \quad n = 1, \dots, N + N_{M-1}$
for $j = 2 \dots N$ **do**
  for $n = 1, \dots, N + N_{M-1}$ **do**
    $\nu_k \leftarrow \eta_{k+6(n-1)}, \quad k = 1, \dots, 17$
    $\bar{\Psi}(j,n) \leftarrow Q(\nu)$
  end for
  $\eta \leftarrow \eta \circ \theta$
end for

To calculate all integrals in $\Psi(j, \cdot)$ for j fixed thus requires a full grid $\{\eta_k\}_{k=1}^{N_\eta}$ of size $N_\eta = 17 + 6(N + N_{M-1} - 1)$, where[13]

$$\{\eta_k\}_{k=1}^{N_\eta} = x_1^1 - 8\Delta/6, \dots, x_1^1 + (N + N_{M-1} - 1)\Delta + 8\Delta/6, \qquad \eta_k - \eta_{k-1} = \Delta/6.$$

[11] For example, when $N = 2^{11}$ and $M = 250$, the size of $\bar{\Psi}$ is reduced from 1.04×10^8 to 7.08×10^5 elements.

[12] Note that $Q(\nu)$ requires only 17 points to evaluate to populate $\Psi(j,n)$, since $x_n - \Delta/2$ and $x_n + \Delta/2$ are each shared by two subintervals, and on the boundaries $\varphi^{[2]}(y) = 0$

[13] This grid is used to initialize the algorithm, after which the value of η is updated.

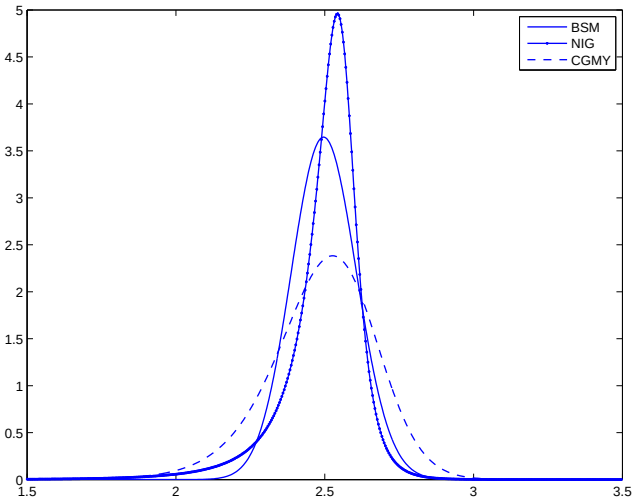

Figure 4: Plotted densities f_{Y_M}, $M = 12$, recovered by PROJ for the models: BSM(.17801), NIG(6.1882, -3.8941, .1622), CGMY(.6509, 5.853, 18.27, .8) in section 6.

Using the fact that

$$\begin{aligned}(e^y+1)^{i\xi_j} = \exp\left(i\xi_j \log\left(1+e^y\right)\right) &= \exp\left(i(\xi_{j-1}+\Delta_\xi)\log\left(1+e^y\right)\right)\\ &= \exp\left(i\Delta_\xi \log\left(1+e^y\right)\right)\cdot \exp\left(i\xi_{j-1}\log\left(1+e^y\right)\right),\end{aligned}$$

we obtain Subroutine 2 for $\bar{\Psi}$, where $\eta\circ\theta$ denotes the Hadamard product[14]. Since the quadrature rule is fixed (e.g. seven-point in our case, although alternative quadratures can be used as well), no user-supplied inputs are required. This simplifies the implementation as compared to a procedure such as ASCOS [41], which requires a specification of n_q (quadrature points for the Clenshaw-Curtis integration rule), which can vary substantially from one application to the next.

3.8. The Valuation Step. Given the approximation $\bar{\Phi}_{Y_M}$, the final step is analogous to the valuation problem for a European option. Rather than specify x_1^M as before, the valuation accuracy can be further improved by perturbing the terminal grid so that the vanilla option "kink", defined by

$$y^* := \log\left((M+1)W/S_0 - 1\right), \tag{39}$$

is a member. In this case, equation (15) can be expressed as

$$g(y) := \begin{cases}\left(\dfrac{S_0(1+\exp(y))}{M+1} - W\right)\mathbb{1}[y \geq y^*], & \text{for a call,}\\ \left(W - \dfrac{S_0(1+\exp(y))}{M+1}\right)\mathbb{1}[y \leq y^*], & \text{for a put.}\end{cases} \tag{40}$$

Initially defining $\tilde{x}_1^M = \tilde{\mu}_1 + N_M\Delta + (1-N/2)\Delta$ and $n^* = \lfloor (y^* - \tilde{x}_1^M)a + 1\rfloor$, we set

$$x_1^M := y^* - (n^*-1)\Delta, \qquad x_n^M = x_1^M + (n-1)\Delta, \quad n = 1,..,N. \tag{41}$$

[14]To evaluate the complexity, η requires on the order of $\mathcal{O}(N_\eta)$ operations to initialize (as θ), followed by $N-1$ Hadamard products for a total cost of $\mathcal{O}((N-1)N_\eta)$ operations. Each quadrature application across a row $\bar{\Psi}(j,\cdot)$ of $\bar{\Psi}$, of which there are $N-1$, requires $\mathcal{O}(N+N_{M-1})$ operations. Hence, $\bar{\Psi}$ is populated at a cost of $\mathcal{O}((N-1)(N+N_{M-1}))$ operations.

from which $y^* = x^M_{n^*}$. If we then define the terminal basis $\{\varphi^M_{a,n}(y)\}_{n=1}^N$ where $\varphi^M_{a,n}(y)$ is centered over x^M_n, the density is approximated by

$$f_{Y_M}(y) \approx \frac{1}{2\pi}\sum_{n=1}^{N}\langle \bar{\phi}_{Y_M}, \widehat{\varphi}^M_{a,n}\rangle \cdot \varphi^M_{a,n}(y) \approx a^{1/2}C_{a,N}\sum_{n=1}^{N}\bar{\beta}^M_n \varphi^M_{a,n}(y)$$

where $\varphi^M_{a,N/2}(y)$ is roughly centered over the mean of Y_M, and $\bar{\beta}^M := \Re[\mathcal{D}\{H^M_j\}_{j=1}^N]$ is determined using

$$H^M_1 = 1/(960a^3), \qquad H^M_j := \bar{\phi}_{Y_M}(\xi_j)\zeta_j \exp(-ix^M_1\xi_j), \quad j = 2,\dots,N. \tag{42}$$

The final step is to approximate the initial value by integrating the terminal payoff against the PROJ expansion of f_{Y_M} (see Figure 4):

$$\mathcal{V}\circ g(S_0) = e^{-rT}\int_{\mathbb{R}} g(y;S_0)f_{Y_M}(y)dy \approx e^{-rt_M}C_{a,N}\sum_{n=1}^{n^*+1}\bar{\beta}^M_n g_n, \tag{43}$$

where the terminal payoff coefficients are defined for $n = 1,\dots,N$ by

$$g_n := a^{1/2}\int_{x^M_n-3\Delta/2}^{x^M_n+3\Delta/2}\varphi^M_{a,n}(y)g(y)dy = \int_{-3/2}^{3/2}\varphi(y)g\left(x^M_n + \frac{y}{a}\right)dy. \tag{44}$$

Remark 2. *For a general payoff $g(y)$, equation (44) can be numerically integrated, by taking into account the piecewise definition of φ and any payoff discontinuities. In general, even when analytical formulas for g_n are known, closed form quadrature rules (such as those in Table 1 for put options) provide more numerically stable coefficients as the resolution is refined (see [29] for more discussion).*

As for European options, put-call parity can be used to price Asian call options (see equation (46)). This approach is preferred numerically since the put has a bounded payoff. For vanilla options defined in equation(40), define $C := \frac{S_0}{M+1}$ and $D := W - C$, and

$$E_n := \exp(x^M_n) = \exp(x^M_1 + (n-1)\Delta), \qquad n = 1,\dots,n^*+1.$$

The payoff coefficients of a put option are given by $g^{put}_n = 0$ for $n = n^*+2,\dots,N$, and

$$g^{put}_n := \begin{cases} D\cdot\bar{\vartheta}^{[2]}_* - C\cdot\vartheta^{[2]}_*\cdot E_n & n = 1,\dots,n^*-2 \\ D\cdot\bar{\vartheta}^{[2]}_{-1} - C\cdot\vartheta^{[2]}_{-1}\cdot E_n & n = n^*-1 \\ D\cdot\bar{\vartheta}^{[2]}_0 - C\cdot\vartheta^{[2]}_0\cdot E_n & n = n^* \\ D\cdot\bar{\vartheta}^{[2]}_1 - C\cdot\vartheta^{[2]}_1\cdot E_n & n = n^*+1 \end{cases} \tag{45}$$

where $\vartheta^{[2]}_j$ and $\bar{\vartheta}^{[2]}_j$, derived in [29], are provided in Table 1 for reference. The value is then approximated by substituting g^{put}_n for g_n in equation (43). Once the put value $\mathcal{V}^{put}$ is determined, the call value $\mathcal{V}^{call}$ satisfies (see Section 4.3)

$$\mathcal{V}^{call} = \mathcal{V}^{put} + \frac{S_0e^{-rT}}{M+1}\left(\frac{e^{(r-q)\Delta_t(M+1)}-1}{e^{(r-q)\Delta_t}-1}\right) - e^{-rT}W. \tag{46}$$

Quadratic	$\bar{\vartheta}_j^{[2]}$	$\vartheta_j^{[2]}$
$j=*$	1	$\frac{1}{5}\left[\frac{1}{2}+\frac{1}{9}\left(\cosh(5\Delta/4)+7\cosh(\Delta/2)+22\cosh(\Delta/4)\right)+\cosh(3\Delta/4)+\frac{1}{6}\cosh(\Delta)\right]$
$j=-1$	$\frac{47}{48}$	$\frac{1}{10}\left[1+\frac{1}{9}\left(e^{-5\Delta/4}+7e^{-\Delta/2}+44\cosh(\Delta/4)\right)+e^{-3\Delta/4}+\frac{1}{6}e^{-\Delta}\right.$ $\left.+\frac{7}{12}e^{\Delta/2}+\frac{49}{72}e^{5\Delta/8}+\frac{25}{72}e^{7\Delta/8}+\frac{3}{16}e^{3\Delta/4}+\frac{7}{144}e^{\Delta}\right]$
$j=0$	$\frac{1}{2}$	$\frac{1}{10}\left[\frac{7}{24}+\frac{1}{9}e^{-5\Delta/4}+\frac{1}{6}e^{-\Delta}+e^{-3\Delta/4}+\frac{7}{12}e^{-\Delta/2}+\frac{13}{12}e^{-3\Delta/8}+\frac{11}{24}e^{-\Delta/4}+\frac{47}{36}e^{-\Delta/8}\right]$
$j=1$	$\frac{1}{48}$	$\frac{1}{80}\left[e^{-9\Delta/8}+\frac{1}{6}e^{-5\Delta/4}+\frac{1}{9}e^{-11\Delta/8}+\frac{7}{18}e^{-\Delta}\right]$

Table 1: Stable closed form coefficient approximations using Boole's rule for use with terminal payoffs.

Remark 3. *While the two checks in Section 3.6 are designed to prevent an insufficient choice of Δ and $\bar{a}$ at initialization, we can use the following quantity*

$$\mathbb{E}[e^{Y_M}] = \frac{M+1}{S_0}\mathbb{E}[A_M] - 1 = \frac{e^{(r-q)\Delta_t(M+1)}-1}{e^{(r-q)\Delta_t}-1} - 1$$

to estimate the final valuation error. In particular, the error in estimating $\mathbb{E}[e^{Y_M}]$,

$$\mathcal{E}_M := \mathbb{E}[e^{Y_M}] - C_{a,N}\cdot\vartheta_*^{[2]}\cdot\sum_{n=1}^{N}\bar{\beta}_n^M\exp(x_n^M) \tag{47}$$

serves as a proxy for the error in $\mathcal{V}\circ g(S_0)$. Given an value error tolerance TOL = 5e-04, we set a mean error tolerance for $\mathcal{E}_M$ of $\epsilon_3 := TOL/10 = 5e\text{-}03$. If $\mathcal{E}_M < \epsilon_3$, the algorithm terminates. Otherwise, if this threshold is exceeded, we reenter the main loop in Algorithm 3. We will then have the new value estimate, $\mathcal{V}_N$, and the previous estimate $\mathcal{V}_{N/2}$. Hence, the new stopping criteria becomes $|\mathcal{V}_N - \mathcal{V}_{N/2}| < TOL$.

3.9. The Algorithm and its Complexity. We now summarize the proceeding steps which define the quadratic APROJ algorithm, while alternative bases can be accommodated similarly. The algorithm calls initialization Subroutine 1, although one can instead select N and Δ directly. After Subroutine 2 is called to compute $\bar{\Psi}$, the main loop begins. Note that we have designed the routine for memory efficiency by reusing the arrays H and $\bar{\beta}$.

3.9.1. Complexity. We begin with cost of initializing the matrix $\bar{\Psi}$. From Section 3.7, for a given quadrature rule the complexity associated with calculating $\bar{\Psi}$ is $\mathcal{O}((N-1)(N+N_{M-1}))$. From equation (25), we can bound the growth of N_{M-1}, and hence the dimensions of $\bar{\Psi}$. With $\tilde{\mu}_m$ defined in equation (26), it follows that

$$\begin{aligned}\tilde{\mu}_{M-1} - \tilde{\mu}_1 &\le \log(M-1) + \frac{T}{M}\left((M-2)(r-q) - (w+\mathbb{E}[L(1)])\right)\\ &\le 2\log(M-1),\end{aligned}$$

for sufficiently large M, by Proposition 3.2. For $\bar{a}\ge 2$,

$$N_{M-1} = \lfloor a(\tilde{\mu}_{M-1}-\tilde{\mu}_1)\rfloor \le \lfloor 2N\log(M-1)/\bar{a}\rfloor \le N\log(M-1). \tag{48}$$

Algorithm 3 Main Algorithm

Value error tolerance TOL:=5e-04
Call Subroutine 1 to obtain:
  Input 1: Final parameters $N, \Delta, \bar{a}, \Delta_\xi$
  Input 2: Grids $\{\xi_j\}_{j=1}^N$, $\{N_m\}_{m=1}^M$, $\{x_1^m\}_{m=1}^M$
  Input 3: Coefficient input Φ, $\{\zeta_j\}_{j=2}^N$, $\{\bar{\beta}_n\}_{n=1}^N$
Call Subroutine 2 to compute $\bar{\Psi}$
$\Phi \leftarrow C_{a,N}\Phi$; $\quad C_1 \leftarrow 1/(960a^3)$
$H_j \leftarrow \Phi_j \cdot \sum_{n=1}^N \bar{\Psi}_{j,n}\bar{\beta}_n, \quad j = 1, \ldots, N$
$\bar{\beta} \leftarrow H$
for $m = 3, \ldots, M$: **do**
  $H_1 \leftarrow C_1; \quad H_j \leftarrow \zeta_j \cdot \bar{\beta}_j \cdot \exp(-i\xi_j \cdot x_1^{m-1}), \quad j = 2, \ldots, N$
  $\bar{\beta} \leftarrow \Re[\mathrm{FFT}(H)]$
  $H_j \leftarrow \Phi_j \cdot \sum_{n=1}^N \bar{\Psi}_{j,N_{m-1}+n}\bar{\beta}_n, \quad j = 1, \ldots, N$
  $\bar{\beta} \leftarrow H$
end for
Redefine x_1^M by equation (41)
$H_1 \leftarrow C_1; \quad H_j \leftarrow \zeta_j \cdot \bar{\beta}_j \cdot \exp(-i\xi_j \cdot x_1^M), \quad j = 2, \ldots, N$
$\bar{\beta} \leftarrow \Re[\mathrm{FFT}(H)]$
Find put value $\mathcal{V}^{put}$ using equation (43) with g_n^{put} defined in (45)
For a call, use put-call parity equation (46)
Compute final error proxy $\mathcal{E}_M$ in eq. (47), and proceed as directed in Remark 3

Thus, $N + N_{M-1} \leq (N+1)\log(M-1) = \mathcal{O}(N\log(M))$, so the complexity of $\bar{\Psi}$ is $\mathcal{O}(N^2\log(M))$. Given that the computational cost of determining x_1^m and H^m, $m = 1, \ldots, M$, is on the order $\mathcal{O}(MN)$, and the final value cost is $\mathcal{O}(N)$, the remaining contribution to the algorithm's complexity resides in the cost of $\bar{\beta}^m$, $m = 1, \ldots, M$, which is on the order $\mathcal{O}(MN\log_2(N))$ when the fast Fourier transform is utilized, the matrix vector multiplications $\bar{\Psi}_{m-1}\bar{\beta}^{m-1}$, $m = 2, \ldots, M$, at a cost of $\mathcal{O}((M-1)N^2)$, and the Hadamard products $\left(\bar{\Psi}_{m-1}\bar{\beta}^{m-1}\right) \circ \Phi_R^C$, $m = 2, \ldots, M$, at a cost of $\mathcal{O}((M-1)N)$. Hence, the total cost is $\mathcal{O}(MN\log_2(N) + N^2\log(M) + MN^2) = \mathcal{O}(MN^2)$.

3.10. Greeks. We now demonstrate how price sensitivities are calculated at almost no additional cost from the valuation algorithm. Consider first the put option payoff $g(y; S_0)$ defined in equation (40), where $y^* = y^*(S_0) = \log\left((M+1)\frac{W}{S_0} - 1\right)$. First we observe that Y_M is independent of S_0. Indeed,

$$\exp(Y_M) = \frac{1}{S_0}\sum_{m=1}^M S_m = \frac{1}{S_0}\sum_{m=1}^M S_0 \exp\left(\sum_{k=1}^m R_k\right) = \sum_{m=1}^M \exp\left(\sum_{k=1}^m R_k\right).$$

From equation (43), Leibniz rule is used to determine the put option Delta, noting that $g(y^*(S_0), S_0) = 0$:

$$\boldsymbol{\Delta} := \frac{\partial \mathcal{V} \circ g}{\partial S_0} = e^{-rT}\int_{-\infty}^{y^*(S_0)} \frac{\partial g(y; S_0)}{\partial S_0} f_{Y_M}(y)dy = \frac{-e^{-rT}}{M+1}\int_{-\infty}^{y^*(S_0)} (1+e^y) f_{Y_M}(y)dy.$$

Using quantities that were already computed during the valuation stage, we find that

$$\boldsymbol{\Delta}^{put} \approx C_{a,N}\frac{-e^{-rT}}{M+1}\sum_{n=1}^{n^*+1} \bar{\beta}_n^M g_n(\boldsymbol{\Delta}^{put}). \tag{49}$$

The coefficients $g_n(\mathbf{\Delta}^{put})$ are defined similarly to equation (45), but instead of $D \cdot \bar{\vartheta}_j^{[2]} - C \cdot \vartheta_j^{[2]} \cdot E_n$ for $j \in \{*, -1, 0, 1\}$, we use $g_n(\mathbf{\Delta}^{put}) = \bar{\vartheta}_j^{[2]} + \vartheta_j^{[2]} \cdot E_n$. To determine the call Delta, equation (46) leads to the put-call parity formula

$$\mathbf{\Delta}^{call} = \mathbf{\Delta}^{put} + \frac{e^{-rT}}{M+1}\left(\frac{e^{(r-q)\Delta_t(M+1)} - 1}{e^{(r-q)\Delta_t} - 1}\right)$$

Likewise, the put (and call) option Gamma is given by

$$\begin{aligned}\mathbf{\Gamma} := \frac{\partial^2 \mathcal{V} \circ g}{\partial S_0^2} &= \frac{-e^{-rT}}{M+1}\frac{\partial}{\partial S_0}\int_{-\infty}^{y^*(S_0)} (1+e^y) f_{Y_M}(y) dy \\ &= \frac{-e^{-rT}}{M+1}\left(1+e^{y^*}\right) f_{Y_M}(y^*)\frac{\partial y^*(S_0)}{\partial S_0} = \left(\frac{W}{S_0}\right)^2 \frac{(M+1)\cdot e^{-rT} f_{Y_M}(y^*)}{W(M+1) - S_0}. \end{aligned} \tag{50}$$

For the quadratic basis we use the approximation[15]

$$f_{Y_M}(y^*) \approx a \cdot C_{a,N} \cdot \left(\varphi^{[2]}(0)\bar{\beta}_{n^*}^M + \varphi^{[2]}(1)\left(\bar{\beta}_{n^*-1}^M + \bar{\beta}_{n^*+1}^M\right)\right),$$

where $\varphi^{[2]}(0) = 3/4, \varphi^{[2]}(1) = 1/8$ and n^* is given in the previous subsection. Thus $\mathbf{\Delta}$ and $\mathbf{\Gamma}$ are computed as byproducts of the pricing algorithm.

4. Extensions. In this section we illustrate in-progress option pricing, generalized arithmetic averaging and continuously monitored option pricing.

4.1. In-Progress Options: Pricing and Greeks. Only a slight modification is required to price Asian options at arbitrary times after averaging has begun. With the arithmetic average A_M defined in equation (14), then for $\tau \in [M_s\Delta t, (M_s+1)\Delta t)$, for some $M_s < M-1$, we find a variable Y_{M-M_s} such that

$$A_M = \frac{1}{M+1}\left[\sum_{m=0}^{M_s} S_m + S(\tau)\cdot \exp(Y_{M-M_s})\right].$$

That is, M_s indexes the most recent monitoring date, and $U_{M_s} := \sum_{m=0}^{M_s} S_m$ as well as $S(\tau)$ are known at the time of pricing. Noting that for $h := (M_s+1)\Delta_t - \tau$, $S_{M_s+1} = S(\tau)e^{R(h)}$ where $R(h) \overset{d}{=} \log(S_{t+h}/S_t)$, it follows from stationarity and independence of increments that Y_{M-M_s} can be found recursively by

$$\begin{aligned}&\phi_{Y_1} = \phi_R, \qquad \bar{\phi}_{Y_m} = \phi_R \cdot \bar{\phi}_{Z_{m-1}}, \quad m = 2, \ldots, M - M_s - 1, \\ &\bar{\phi}_{Y_{M-M_s}} = \phi_{R(h)} \cdot \bar{\phi}_{Z_{M-M_s-1}}.\end{aligned}$$

When $\tau = M_s\Delta t$, $\phi_R \equiv \phi_{R(h)}$. As before, the final grid defined by $x_1^{M-M_s}$ is shifted so that the kink point

$$y^* := \log\left((M+1)W - U_{M_s}\right) - \log(S(\tau)) \tag{51}$$

is a member. For in-progress vanilla options, the payoff is expressed as

$$g(y) := \begin{cases} \left(\frac{1}{M+1}\left(U_{M_s} + S(\tau)e^y\right) - W\right)\mathbb{1}[y \geq y^*], & \text{for a call,} \\ \left(W - \frac{1}{M+1}\left(U_{M_s} + S(\tau)e^y\right)\right)\mathbb{1}[y \leq y^*], & \text{for a put,} \end{cases} \tag{52}$$

[15] For the linear basis, $f_{Y_M}(y^*) \approx a \cdot C_{a,N} \cdot \bar{\beta}_{n^*}^M$

and payoff coefficients are derived analogously. Perhaps of even more interest than the price for an in-progress option are the Greeks. For the fixed strike Asian put,

$$\mathbf{\Delta} := \frac{\partial \mathcal{V}^\tau \circ g}{\partial S(\tau)} = \frac{-e^{-r(T-\tau)}}{M+1} \int_{-\infty}^{y^*} e^y f_{Y_{M-M_s}}(y) dy,$$

where $\mathcal{V}^\tau \circ g(U_{M_s}, S(\tau)) = e^{-r(T-\tau)} \mathbb{E}[g(A_M)|U_{M_s}, S(\tau)]$. Similarly, the put (and call) option Gamma is given by

$$\mathbf{\Gamma} := \frac{\partial^2 \mathcal{V}^\tau \circ g}{\partial S^2(\tau)} = \frac{e^{-r(T-\tau)}}{(S(\tau))^2} f_{Y_{M-M_s}}(y^*) \left(W - \frac{U_{M_s}}{M+1} \right),$$

where $f_{Y_{M-M_s}}(y^*)$ is calculated as before.

4.2. Generalized Arithmetic Asian Pricing. By a slight modification of the original algorithm, the ARPOJ method is capable of pricing payoffs on generalized averages of the underlying[16]

$$A_M^\lambda := \frac{1}{M+1} \sum_{m=0}^{M} \lambda_m S_m, \tag{53}$$

where $\lambda_m > 0$, $m = 0, \ldots, M$. We have the following extension, which is proved in a similar fashion to the Carverhill-Clewlow result, noting that

$$A_M^\lambda = \frac{S_0}{M+1} \left(\lambda_0 + e^{R_1} \left(\lambda_1 + e^{R_2} \left(\cdots e^{R_{M-1}} \left(\lambda_{M-1} + \lambda_M e^{R_M} \right) \right) \right) \right).$$

In alternative representation is provided in Corollary 4.1, which prevents the matrix $\bar{\Psi}$ from becoming stage dependent, and results in an efficient algorithm.

Corollary 4.1. *Fix a set of positive weights $\lambda = \{\lambda_m\}_{m=0}^M$, and define $X_m := \frac{\lambda_m}{\lambda_{m-1}} \exp(R_m)$, $m = 1, \ldots, M$, where $R_m = \log(S_m/S_{m-1})$. Set $Y_1 = \log(X_M) = \log(\lambda_M/\lambda_{M-1}) + R_M$, and define recursively*

$$Y_m = \log \left(\frac{\lambda_{M+1-m}}{\lambda_{M+1-(m-1)}} \right) + R_{M+1-m} + Z_{m-1}, \quad m = 2, \ldots, M,$$

where $Z_m := \log(1 + \exp(Y_m))$. Then

$$A_M^\lambda \equiv \frac{\lambda_0 S_0}{M+1} \left(1 + \exp(Y_M) \right). \tag{54}$$

Proof. The proof relies on an equivalent factorization of A_M^λ,

$$A_M^\lambda = \frac{\lambda_0 S_0}{M+1} \left(1 + \frac{\lambda_1 S_1}{\lambda_0 S_0} \left(1 + \frac{\lambda_2 S_2}{\lambda_1 S_1} \left(\cdots \frac{\lambda_{M-1} S_{M-1}}{\lambda_{M-2} S_{M-2}} \left(1 + \frac{\lambda_M S_M}{\lambda_{M-1} S_{M-1}} \right) \right) \right) \right),$$

which can be verified by multiplying each of the terms. The remainder of the proof is similar to standard construction, and follows algebraically. ∎

This form of the recursion requires that $\lambda_m > 0$ for each m, in which case the structure of the APROJ algorithm is unaffected. Namely, the matrix $\bar{\Psi}$ is the same for each m, and the only real change is the grid shift, where we add $\tilde{\lambda}_m := \log(\lambda_{M+1-m}/\lambda_{M+1-(m-1)})$ to each $\tilde{\mu}_m$. The perturbed grid shifts $\bar{\mu}_m$ are defined still by equation (27).

[16] We include the term $1/(M+1)$ so that the standard average is obtained when all $\lambda_m = 1$.

4.3. Put-Call Parity. Just as for vanilla European options, put-call parity can be used to price Asian call options in terms of puts and conversely (this will be used for all numerical experiments). In the generalized setting, we have

$$e^{-rT}\mathbb{E}\left[(A_M^\lambda)^+ - (-A_M^\lambda)^+\right] = \frac{e^{-rT}}{M+1}\mathbb{E}\left[\sum_{m=0}^{M}\lambda_m S_m\right] = \frac{S_0 e^{-rT}}{M+1}\sum_{m=0}^{M}\lambda_m e^{(r-q)\Delta_t m},$$

where q is the continuous dividend yield, and $\Delta_t = T/M$ in the uniform case. Considering the fixed and floating strikes[17] together, with $\alpha = \pm 1$,

$$(\alpha(A_M - K_1 S_T - K_2))^+ = (\alpha A_M^\lambda)^+,$$

where

$$\lambda_0 = 1 - \frac{K_2}{S_0}(M+1), \quad \lambda_M = 1 - (M+1)K_1, \quad \lambda_m = 1, \quad m = 1, \ldots, M-1.$$

In this setting, with $C_M(S_0, T)$ and $P_M(S_0, T)$ denoting the call and put prices,

$$C_M(S_0,T) - P_M(S_0,T) = \frac{S_0 e^{-rT}}{M+1}\left(\frac{e^{(r-q)\frac{T(M+1)}{M}} - 1}{e^{(r-q)\frac{T}{M}} - 1}\right) - S_0 K_1 e^{-qT} - e^{-rT}K_2,$$

from which the fixed and floating strike parities are derived. Moreover, the forward contract, $g(\{S_m\}) = A_M - K_2$, is priced immediately by setting $K_1 = 0$.

It should be noted that put-call parity is a useful tool for maintaining robustness when pricing call options. Since the density of Y_M is recovered approximately over $[\bar{\mu}_M - \frac{\bar{a}}{2}, \bar{\mu}_M + \frac{\bar{a}}{2}]$, this implies a lower bound on the truncation error for pricing call options:

$$\epsilon_{trunc} \geq e^{-rT}\left(\frac{S_0(1 + \exp(\bar{\mu}_M + \bar{a}/2))}{M+1} - W\right)\int_{\bar{\mu}_M + \bar{a}/2}^{\infty} f_{Y_M}(y)dy.$$

For a heavy-tailed density, the implied truncation error can be unacceptable, in which case put-call parity can be used to price call options in terms of the *bounded* put prices.

4.4. Continuous Monitoring. As a final extension, we consider the case of continuously monitored contracts, with terminal payoffs

$$g(S) = \begin{cases} \left(\frac{1}{T}\int_0^T S(t)dt - W\right)^+ & \text{for a call,} \\ \left(W - \frac{1}{T}\int_0^T S(t)dt\right)^+ & \text{for a put.} \end{cases}$$

Let $\mathcal{V}_N(M)$ denote the discretely monitored value approximation with M monitoring dates, and with N fixed. By fixing a positive integer d, the continuously monitored option value can be approximated by a four-point Richardson extrapolation:

$$\mathcal{V}_N^\infty(d) := \frac{1}{21}\left(64\mathcal{V}_N(2^{d+3}) - 56\mathcal{V}_N(2^{d+2}) + 14\mathcal{V}_N(2^{d+1}) - \mathcal{V}_N(2^d)\right),$$

as demonstrated in [41]. We compare the extrapolation procedure[18], when applied with APROJ, to the values obtained by [41] in Table 2. For both strikes, agreement in prices is to at least three decimals.

[17] For example, a floating strike call has payoff $(A_M - K_1 S_T)^+$.

[18] For greatest efficiency, a common $\bar{\Psi}$ can be used for all four settings of M in the extrapolation procedure, by perturbing the means slightly so they align.

	$W = 90$		$W = 100$	
d	ASCOS	APROJ	ASCOS	APROJ
1	–	12.67415	–	5.11827
2	–	12.67441	–	5.11855
3	–	12.67443	–	5.11859
4	12.6748	12.67443	5.1191	5.11859
5	12.6744	12.67443	5.1186	5.11859
6	12.6743	12.67443	5.1185	5.11859

Table 2: Continuously monitored Asian option values by Richardson Extrapolation. NIG model with parameters from [23]. Values obtained by quadratic APROJ with $P = 7, \bar{P} = 4$, and seven point rule. ASCOS values given in [41].

5. Error Analysis. In this section, we provide a stability analysis of the error propagation of ChFs for $1 \le m \le M$, after which we conclude with the terminal valuation error for pricing options on the arithmetic average.

Recall that the characteristic functions for Levy processes of interest satisfy

$$|\phi_{R_{\Delta_t}}(\xi)| \le \kappa \exp(-\Delta_t c|\xi|^\nu), \quad \xi \in \mathbb{R}. \tag{55}$$

For the BSM, KOU (double exponential), and MJD (Merton's Jump Diffusion) models from Table 6, the ChF of log return satisfies $|\phi_{R_{\Delta_t}}(\xi)| \le \exp\left(-\Delta_t \frac{\sigma^2}{2}|\xi|^2\right)$, so equation (1) holds with $\nu = 2$ and $c = \frac{\sigma^2}{2}$. For the CGMY model, $\nu = Y$ and c can be taken as $c = 2C|\Gamma(-Y)\cos(\pi Y/2)| \cdot \epsilon$, for any $\epsilon \in (0,1)$. With the Normal Inverse Gaussian (NIG) model, $\nu = 1$ and $c = \delta$. For the pure jump VG, $|\phi_{R_{\Delta_t}}(\xi)| \le \kappa|\xi|^{-2\Delta_t/\nu}$, so that $\phi_{R_{\Delta_t}}$ fails to be integrable for $\Delta_t \le \nu/2$. However, by adding a Brownian motion component, $-\frac{\sigma^2}{2}\xi^2$, equation (1) is satisfied with $\nu = 2$. We have the following Corollary of Proposition 2.1.

Corollary 5.1. *Suppose that $\phi_{R_{\Delta_t}}(\xi) \in \mathcal{H}(\mathcal{D}_d)$ for some $d > 0$. Fix $a = 2^P$ and $N = a \cdot \bar{a}$, where $\bar{a} = 2^{\bar{P}}$ for $\bar{P} > 1 + \log_2|\bar{\mu}_M|$. Assume for some $c, \kappa > 0$ and $\nu \in (0,2]$, $\phi_{R_{\Delta_t}}(\xi)$ satisfies equation(55). Then for some $0 < \gamma \le d$, and a constant $C_M = \mathcal{O}(\max_{m=1,\dots,M}\|\phi_{Y_m}\|^{\mathcal{H}_d})$,*

$$\sup_{1\le n\le N}\left|a^{1/2}C_{a,N}\cdot\breve{\beta}^m_{a,n} - \langle f_{Y_m}, \widetilde{\varphi}_{a,n}\rangle\right| \le \frac{a^{-1/2}}{\pi}\left(C_M\frac{e^{-(\bar{a}-2|\bar{\mu}_M|)\gamma/2}}{1-e^{-\bar{a}\gamma}} + \tau_a(R_{\Delta_t})\right) \tag{56}$$

independently of $1 \le m \le M$ where $\tau_a(R_{\Delta_t}) = \mathcal{O}(a\exp(-\Delta t c\cdot(2\pi a)^\nu))$ is as in equation (8). For large enough $a > 0$, and $d < \infty$, γ will approach d.

5.1. Error Propagation. We can now state the core result concerning the propagation of ChF error for a given number of monitoring dates M.

Proposition 5.1. *Suppose that $\phi_{R_{\Delta_t}}(\xi) \in \mathcal{H}(\mathcal{D}_d)$ for some $d > 0$, and consider a p^{th} order B-spline basis generated by φ. Fix $a = 2^P$ and $N = a \cdot \bar{a}$, where $\bar{a} = 2^{\bar{P}}$ for $\bar{P} > 1 + \log_2|\bar{\mu}_M|$. Assume for some $c, \kappa > 0$ and $\nu \in (0,2]$, the tail of $\phi_{R_{\Delta_t}}(\xi)$ satisfies equation (55). The terminal ChF error satisfies $\epsilon(\bar{\phi}_{Y_M}(\xi_1)) = 0$ and*

$$|\epsilon(\bar{\phi}_{Y_M}(\xi_j))| = \mathcal{O}\left(\Delta^{(p+1)}\cdot e^{-\tilde{c}\Delta_t\left(\frac{(j-1)}{\bar{a}}\right)^\nu}\bar{a}^{1/2}\|\xi^{(p+1)}\phi_{R_{\Delta_t}}(\xi)\|_2\right), \quad 2 \le j \le N, \tag{57}$$

where $\tilde{c} := (2\pi)^\nu c$. The dependence on M is governed by the behavior of $\phi_{R_{\Delta t}}$.

Proof. Fix any $\xi \geq 0$, and let $G := \cup_{m=1,..,M} G_m$ the full truncated integration range implied by $\bar{P}$, where $G_m = [\bar{\mu}_m - \frac{\bar{a}}{2}, \bar{\mu}_m + \frac{\bar{a}}{2}]$. To manage notation, we will suppress the dependence of certain objects on m. For example, we assume by the indexing on $\bar{\beta}_n^m$ that the corresponding elements $\varphi_{a,n}$ have been shifted appropriately. We start by fixing $m \geq 3$, for which

$$\begin{aligned}
\epsilon(\bar{\phi}_{Z_{m-1}}(\xi)) &:= \phi_{Z_{m-1}}(\xi) - \bar{\phi}_{Z_{m-1}}(\xi) \\
&= \int_{\mathbb{R}} (e^y + 1)^{i\xi} f_{Y_{m-1}}(y) dy - C_{a,N} \sum_{n=1}^{N} \bar{\beta}_n^{m-1} \bar{\Psi}(\xi, n) \\
&= \int_{\mathbb{R}/G_{m-1}} (e^y + 1)^{i\xi} f_{Y_{m-1}}(y) dy \\
&+ \left(\int_{G_{m-1}} (e^y + 1)^{i\xi} f_{Y_{m-1}}(y) dy - C_{a,N} \sum_{n=1}^{N} \beta_n^{m-1} \Psi(\xi, n) \right) \\
&+ C_{a,N} \sum_{n=1}^{N} \beta_n^{m-1} (\Psi(\xi, n) - \bar{\Psi}(\xi, n)) + C_{a,N} \sum_{n=1}^{N} \bar{\Psi}(\xi, n) (\beta_n^{m-1} - \bar{\beta}_n^{m-1}) \\
&:= \big(\tau(G_{m-1}) + J_1^{m-1}(\xi) + J_2^{m-1}(\xi) \big) + J^{m-1}(\xi),
\end{aligned}$$

where the error term $J^{m-1}(\xi)$ will be further split into two components. Here we have defined β_n^{m-1} so that $a^{1/2} C_{a,N} \beta_n^{m-1} = \langle f_{Y_{m-1}}, \widetilde{\varphi}_{a,n} \rangle$, from which

$$\widetilde{f}_{Y_{m-1}}(y) := a^{1/2} C_{a,N} \sum_{n=1}^{N} \beta_n^{m-1} \varphi_{a,n}(y)$$

is the true projection truncated to the set $\{\varphi_{a,n}\}_{n=1}^N$.

Since $|(e^y + 1)^{i\xi}| = |\exp(i\xi \log(1 + e^y))| = 1$, the truncation error satisfies

$$\tau(G_{m-1}) = \int_{\mathbb{R}/G_{m-1}} (e^y + 1)^{i\xi} f_{Y_{m-1}}(y) dy \leq \int_{\mathbb{R}/G_{m-1}} f_{Y_{m-1}}(y) dy \leq \tau_M(G),$$

for $m = 1, \ldots, M$, where $\tau_M(G)$ bounds the largest such truncation error (typically, $\tau_M(G) \approx \tau(G_1)$, since f_R has the heaviest tails). The next result characterizes the convergence of J_1^{m-1}, which is governed by the projection error.

Lemma 5.2. *For $\xi \in \mathbb{R}$, $1 \leq m \leq M$, and $C_1(R_{\Delta_t}) := C_1(\varphi) \cdot \|\xi^2 \phi_{R_{\Delta_t}}(\xi)\|_2/(2\pi)$, J_1^{m-1} satisfies*

$$|J_1^{m-1}(\xi)| \leq \sqrt{\bar{a}} \cdot C_1(R_{\Delta_t}) \Delta^{(p+1)},$$

with the constant $C_1(\varphi)$ from (59), independent of $\phi_{R_{\Delta_t}}$.

Proof. In particular, by Cauchy-Schwartz

$$\begin{aligned}
J_1^{m-1}(\xi) &= \int_{G_{m-1}} (e^y + 1)^{i\xi} f_{Y_{m-1}}(y) dy - C_{a,N} \sum_{n=1}^{N} \beta_n^{m-1} \Psi(\xi, n) \\
&= \int_{G_{m-1}} (e^y + 1)^{i\xi} \left(f_{Y_{m-1}}(y) - a^{1/2} C_{a,N} \sum_{n=1}^{N} \beta_n^{m-1} \varphi_{a,n}(y) \right) dy \\
&\leq \|(e^y + 1)^{i\xi}\|_2^{G_{m-1}} \cdot \|f_{Y_{m-1}} - \widetilde{f}_{Y_{m-1}}\|_2^{G_{m-1}} \\
&\leq \|(e^y + 1)^{i\xi}\|_2^{G_{m-1}} \cdot \|f_{Y_{m-1}} - P_{\mathcal{M}_a} f_{Y_{m-1}}\|_2^{\mathbb{R}}.
\end{aligned}$$

To characterize the convergence rate of density projections onto B-spline bases, we note that φ is Riesz generator which satisfies

$$\widehat{\varphi}(0) = 1, \quad \text{and for } m \in \{0,1\}, \quad \widehat{\varphi}^{(m)}(2\pi k) = 0, \quad k \in \mathbb{Z}/\{0\}, \tag{58}$$

where $\widehat{\varphi}^{(m)}$ denotes the mth derivative of φ. In particular, the p^{th} order B-spline generator φ is a $(p+1)^{th}$ *order Riesz generator*. It then follows that for any $f_X \in L^2(\mathbb{R})$, the projection error satisfies

$$\inf_{f_a \in \mathcal{M}_a} \|f_X - f_a\|_2 \leq \|f_X - P_{\mathcal{M}_a} f_X\|_2 \leq C_1(\varphi) a^{-(p+1)} \|f_X^{(p+1)}\|_2, \tag{59}$$

where $C_1(\varphi)$ is a constant independent of f_X (see [39]). It follows that

$$\|f_{Y_m}^{(p+1)}\|_2 = \frac{1}{2\pi}\|\mathcal{F}[f_{Y_m}^{(p+1)}]\|_2 = \frac{1}{2\pi}\|(-i\xi)^{(p+1)}\phi_{Y_m}(\xi)\|_2 \leq \frac{1}{2\pi}\|\xi^{(p+1)}\phi_{R_{\Delta_t}}(\xi)\|_2 < \infty, \tag{60}$$

since for $\xi \in \mathbb{R}$, $|\phi_{Y_m}(\xi)| \leq |\phi_{R_{\Delta_t}}(\xi)|$, and the $(p+1)^{th}$ moment is finite by exponential decay of $\phi_{R_{\Delta_t}}(\xi)$. Thus if we define $C_1(R_{\Delta_t})$ as in the statement of the Lemma,

$$\|f_{Y_{m-1}} - P_{\mathcal{M}_a} f_{Y_{m-1}}\|_2^{\mathbb{R}} \leq C_1(R_{\Delta_t})\Delta^{(p+1)}, \quad \forall m \geq 2.$$

Hence, for $m \geq 2$ and $\xi \in \mathbb{R}$

$$|J_1^{m-1}(\xi)| \leq \|(e^y+1)^{i\xi}\|_2^{G_{m-1}} C_1(R_{\Delta_t})\Delta^{(p+1)} \leq \sqrt{\bar{a}} \cdot C_1(R_{\Delta_t})\Delta^{(p+1)},$$

since $|(e^y+1)^{i2\xi}| = 1$ and $|G_{m-1}| \leq \bar{a}$. ∎

Remark 4. *We should note that, while the bound in (60) is chosen to be independent of m, the behavior of this term is truly a decreasing function of m, although is difficult to quantify. This can be seen by examining the behavior of ϕ_{Y_m} from the approximations given in Figure 3 for a CGMY model.*

The next source of error materializes from the approximation of Ψ by $\bar{\Psi}$.

Lemma 5.3. *For $\xi \in \mathbb{R}$ $1 \leq m \leq M$, and $C_2(R_{\Delta_t}) := C_2(\varphi)\|\phi_{R_{\Delta_t}}\|_2/2\pi$,*

$$|J_2^{m-1}(\xi)| \leq \sqrt{\bar{a}} \cdot \epsilon(\bar{\Psi}) C_2(R_{\Delta_t}), \tag{61}$$

where the constant $C_2(\varphi)$ is the lower frame bound defined in equation (3) for the piecewise linear basis, and

$$\epsilon(\bar{\Psi}) := \sup\{|\Psi(\xi_j, n) - \bar{\Psi}(\xi_j, n)| : 1 \leq j \leq N, 1 \leq n \leq N + N_{M-1}\}.$$

Proof. By the discrete version of Cauchy-Schwartz,

$$\begin{aligned} J_2^{m-1}(\xi) &= C_{a,N} \sum_{n=1}^{N} \beta_n^{m-1}(\Psi(\xi,n) - \bar{\Psi}(\xi,n)) \\ &\leq a^{-1/2} \left(\sum_{n=1}^{N} \left(\Psi(\xi,n) - \bar{\Psi}(\xi,n)\right)^2 \right)^{1/2} \left(\sum_{n=1}^{N} \left(a^{1/2} C_{a,N} \beta_n^{m-1}\right)^2 \right)^{1/2} \\ &\leq \sqrt{\bar{a}} \cdot \epsilon(\bar{\Psi}) \cdot \Big(\sum_{n\in\mathbb{Z}} |\langle f_{Y_{m-1}}, \widetilde{\varphi}_{a,n}\rangle|^2 \Big)^{1/2} \leq \sqrt{\bar{a}} \cdot \epsilon(\bar{\Psi}) \cdot C_2(\varphi) \|f_{Y_{m-1}}\|_2. \end{aligned}$$

The term $C_2(\varphi)\|f_{Y_{m-1}}\|_2$ follows from Bessel's inequality, which is the upper *frame bound* corresponding to the piecewise linear basis. Noting that

$$\|f_{Y_{m-1}}\|_2 = \|\phi_{Y_{m-1}}\|_2/2\pi \leq \|\phi_{R_{\Delta_t}}\|_2/2\pi,$$

we have

$$|J_2^{m-1}(\xi)| \leq \sqrt{\bar{a}} \cdot \epsilon(\bar{\Psi}) \cdot C_2(\varphi)\|\phi_{R_{\Delta_t}}\|_2/2\pi. \qquad \blacksquare$$

Remark 5. *While the definition of $\epsilon(\bar{\Psi})$ is made so that we obtain an overall convergence rate in Δ when $\bar{a}$ has been fixed and a sufficiently accurate quadrature rule has been selected, the the error in $\bar{\Psi}$ tends to be much smaller for $\xi_j \in [0, 2\pi a]$ near zero than for ξ_j near $2\pi a$. If we define*

$$\epsilon_j(\bar{\Psi}_{m-1}) := \sup_{1\leq n\leq N} |\Psi(\xi_j, N_{m-1}+n) - \bar{\Psi}(\xi_j, N_{m-1}+n)|$$

then $J_2^{m-1}(\xi_j) \leq \epsilon_j(\bar{\Psi}_{m-1})\sqrt{\bar{a}} \cdot C_2(R_{\Delta_t})$. This is more than offset, however, when multiplied by $\phi_{R_{\Delta_t}}(\xi_j)$ to obtain the error in $\bar{\phi}_{Y_m}$, since $\phi_{R_{\Delta_t}}(\xi_j)$ is close to one for ξ_j near zero, and decays exponentially for larger ξ_j. In practice, the contribution of $\epsilon(\bar{\Psi})$ is dominated by the projection error when using a seven-point Newton-Cote's rule. Although Boole's rule is often sufficient (and cheaper) for $M \leq 52$, we opt for the more conservative seven-point rule in general.

For the final term, which reflects the discrete Fourier transform error inherent in $\bar{\beta}^m$, we have

$$J^{m-1}(\xi) := C_{a,N} \sum_{n=1}^{N} \bar{\Psi}(\xi, n)(\beta_n^{m-1} - \bar{\beta}_n^{m-1}) = a^{-1/2} \sum_{n=1}^{N} \bar{\Psi}(\xi, n) \cdot \epsilon(\bar{\beta}_n^{m-1}),$$

where $\epsilon(\bar{\beta}_n^{m-1}) := a^{1/2} C_{a,N}(\beta_n^{m-1} - \bar{\beta}_n^{m-1})$.

Lemma 5.4. *The error source $J^{m-1}(\xi)$ can be bounded by*

$$|J^{m-1}(\xi)| \leq \frac{\bar{a}}{\pi}\epsilon^M(a, \bar{a}) + C(J_4) \cdot \epsilon(\bar{\phi}_{Z_{m-2}})a^{-1/2}|\bar{\phi}_{Z_1}(\xi)| \tag{62}$$

where $C(J_4)$ is a constant, and

$$\epsilon^M(a, \bar{a}) := C_M \frac{e^{-(\bar{a}-2|\bar{\mu}_M|)\gamma/2}}{1 - e^{-\bar{a}\gamma}} + \tau_a(R_{\Delta_t}). \tag{63}$$

Proof. Splitting $\epsilon(\bar{\beta}_n^{m-1})$ in terms of the discrete Fourier transform and ChF errors, where $a^{1/2}C_{a,N}\breve{\beta}_n^{m-1}$ is the discrete Fourier transform approximation using the true $\phi_{Y_{m-1}}$ (see equation (6)), it follows that

$$\begin{aligned}\epsilon(\bar{\beta}_n^{m-1}) &= \left(\langle f_{Y_{m-1}}, \widetilde{\varphi}_{a,n}\rangle - a^{1/2}C_{a,N}\breve{\beta}_n^{m-1}\right) + a^{1/2}C_{a,N}\left(\breve{\beta}_n^{m-1} - \bar{\beta}_n^{m-1}\right) \\ &:= \epsilon_1(\bar{\beta}_n^{m-1}) + \epsilon_2(\bar{\beta}_n^{m-1}).\end{aligned}$$

Hence,

$$\epsilon(\bar{\phi}_{Z_{m-1}}(\xi)) = \left(\tau(G_{m-1}) + J_1^{m-1}(\xi) + J_2^{m-1}(\xi) + J_3^{m-1}(\xi)\right) + J_4^{m-1}(\xi),$$

where we have defined

$$J_3^{m-1}(\xi) := a^{-1/2}\sum_{n=1}^{N}\bar{\Psi}(\xi,n)\cdot\epsilon_1(\bar{\beta}_n^{m-1}),\quad J_4^{m-1}(\xi) := a^{-1/2}\sum_{n=1}^{N}\bar{\Psi}(\xi,n)\cdot\epsilon_2(\bar{\beta}_n^{m-1}).$$

Moreover, for the Newton-Cotes quadrature rules, $|\bar{\Psi}(\xi_j,n)|\leq 1$ for any $1\leq j,n\leq N$, and by Corollary 5.1

$$|\epsilon_1(\bar{\beta}_n^{m-1})|\leq\frac{a^{-1/2}}{\pi}\epsilon^M(a,\bar{a}),$$

where $\epsilon^M(a,\bar{a})$ is defined in equation (63).

Hence,

$$|J_3^{m-1}(\xi_j)|\leq\frac{a^{-1/2}}{\pi}\epsilon^M(a,\bar{a})\cdot a^{-1/2}\sum_{n=1}^{N}|\bar{\Psi}(\xi_j,n)|\leq\frac{\bar{a}}{\pi}\epsilon^M(a,\bar{a})$$

Note that $J_4^{m-1}(\xi)$ alone depends on $\epsilon(\bar{\phi}_{Z_{m-2}}(\xi_j))$, since

$$\begin{aligned}\epsilon_2(\bar{\beta}_n^{m-1}) &= \frac{a^{-1/2}}{\pi}\Re\left(\Delta_\xi{\sum_{j=1}^{N}}'h_{a,n}(\xi_j)\left(\phi_{Y_{m-1}}(\xi_j)-\bar{\phi}_{Y_{m-1}}(\xi_j)\right)\right)\\ &= \frac{a^{-1/2}}{\pi}\Re\left(\Delta_\xi{\sum_{j=1}^{N}}'h_{a,n}(\xi_j)\phi_{R_{\Delta_t}}(\xi_j)\epsilon(\bar{\phi}_{Z_{m-2}}(\xi_j))\right),\end{aligned}\tag{64}$$

where $h_{a,n}(\xi)$ and $h_a(\xi)$ are defined in equation (69) for the linear basis (and in general is determined by $\widehat{\widetilde{\varphi}}(\xi)$), and $\sum'$ indicates that the first and last terms in the sum are halved. If we define $\epsilon(\bar{\phi}_{Z_{m-2}}) := \max_{1\leq j\leq N}|\epsilon(\bar{\phi}_{Z_{m-2}}(\xi_j))|$, it follows that

$$\left|\Re\left(\Delta_\xi{\sum_{j=1}^{N}}'h_{a,n}(\xi_j)\phi_{R_{\Delta_t}}(\xi_j)\epsilon(\bar{\phi}_{Z_{m-2}}(\xi_j))\right)\right|\leq\epsilon(\bar{\phi}_{Z_{m-2}})\Delta_\xi{\sum_{j=1}^{N}}'h_a(\xi_j)\Re(\phi_{R_{\Delta_t}}(\xi_j)),$$

which is bounded above for all N and a, since $\Re(\phi_{R_{\Delta_t}})$ admits an upper frame bound. To derive an upper bound on $J_4^{m-1}(\xi)$, we recall the dependence of $\bar{\Psi}$ and $h_{a,n}$ on $m-1$ (through the shift x_1^{m-1}, denoted by $h_{a,n}^{m-1}$), from which equation (64) yields

$$\begin{aligned}J_4^{m-1}(\xi) &= a^{-1/2}\sum_{n=1}^{N}\bar{\Psi}_{m-1}(\xi,n)\cdot\epsilon_2(\bar{\beta}_n^{m-1})\\ &= a^{-1/2}\sum_{n=1}^{N}\bar{\Psi}_{m-1}(\xi,n)\frac{a^{-1/2}}{\pi}\Re\left(\Delta_\xi{\sum_{j=1}^{N}}'h_{a,n}^{m-1}(\xi_j)\phi_{R_{\Delta_t}}(\xi_j)\epsilon(\bar{\phi}_{Z_{m-2}}(\xi_j))\right)\\ &= \mathcal{O}\left(\frac{\epsilon(\bar{\phi}_{Z_{m-2}})}{a^{1/2}}\left|\sum_{n=1}^{N}\bar{\Psi}_{m-1}(\xi,n)\frac{a^{-1/2}}{\pi}\Re\left(\Delta_\xi{\sum_{j=1}^{N}}'h_{a,n}^{m-1}(\xi_j)\phi_{R_{\Delta_t}}(\xi_j)\right)\right|\right)\\ &= \mathcal{O}\left(\frac{\epsilon(\bar{\phi}_{Z_{m-2}})}{a^{1/2}}\left|\sum_{n=1}^{N}\bar{\Psi}_{m-1}(\xi,n)\cdot a^{1/2}C_{a,N}\bar{\beta}^1_{N_{m-1+n}}\right|\right).\end{aligned}$$

As a final simplification, we note that

$$J_4^{m-1}(\xi) = \mathcal{O}\left(\epsilon(\bar{\phi}_{Z_{m-2}})a^{-1/2}\left|\sum_{n=1}^{N}\bar{\Psi}(\xi,n)a^{1/2}C_{a,N}\bar{\beta}_n^1\right|\right) \leq \frac{C(J_4)}{a^{1/2}}\cdot\epsilon(\bar{\phi}_{Z_{m-2}})|\bar{\phi}_{Z_1}(\xi)|,$$

for some $C(J_4)$. To see that $C(J_4)$ can be chosen independently of N_{m-1}, from the decay of $\phi_{R_{\Delta_t}}(\xi)$, it follows that $f_{R_{\Delta_t}} \in C^\infty(\mathbb{R})$ has exponential decay at infinity, along with all of its derivatives [7] (see also [36]). ∎

Summarizing the obtained bounds, it follows that

$$\begin{aligned}|\epsilon(\bar{\phi}_{Z_{m-1}}(\xi))| &= \left|\left(\tau(G_{m-1}) + J_1^{m-1}(\xi) + J_2^{m-1}(\xi) + J_3^{m-1}(\xi)\right) + J_4^{m-1}(\xi)\right| \\ &\leq C^M(a,\bar{a}) + B(a,\xi)\epsilon(\bar{\phi}_{Z_{m-2}}).\end{aligned}$$

where

$$C^M(a,\bar{a}) := \tau_M(G) + \sqrt{\bar{a}}\cdot C_1(R_{\Delta_t})\Delta^{(p+1)} + \sqrt{\bar{a}}\cdot C_2(R_{\Delta_t})\epsilon(\bar{\Psi}) + \frac{\bar{a}}{\pi}\epsilon^M(a,\bar{a}), \tag{65}$$

and $B(a,\xi) := C(J_4)|\bar{\phi}_{Z_1}(\xi)|a^{-1/2}$. Iterating from $M-1$ we obtain

$$\begin{aligned}|\epsilon(\bar{\phi}_{Z_{M-1}}(\xi))| &\leq C^M(a,\bar{a})\sum_{j=0}^{M-3}B(a,\xi)^j + B(a,\xi)^{M-2}\epsilon(\bar{\phi}_{Z_1}) \\ &= C^M(a,\bar{a})\frac{1-B(a,\xi)^{M-2}}{1-B(a,\xi)} + B(a,\xi)^{M-2}\epsilon(\bar{\phi}_{Z_1}).\end{aligned}$$

Moreover, the error in $\bar{\phi}_{Z_1}$ satisfies

$$\begin{aligned}\epsilon(\bar{\phi}_{Z_1}(\xi)) &:= \phi_{Z_1}(\xi) - \bar{\phi}_{Z_1}(\xi) \\ &= \int_{\mathbb{R}}(e^y+1)^{i\xi}f_{R_{\Delta_t}}(y)dy - C_{a,N}\sum_{n=1}^{N}\bar{\beta}_n^1\bar{\Psi}(\xi,n) \\ &= \int_{\mathbb{R}/G_1}(e^y+1)^{i\xi}f_{R_{\Delta_t}}(y)dy \\ &+ \left(\int_{G_1}(e^y+1)^{i\xi}f_{R_{\Delta_t}}(y)dy - C_{a,N}\sum_{n=1}^{N}\beta_n^1\Psi(\xi,n)\right) \\ &+ C_{a,N}\sum_{n=1}^{N}\beta_n^1(\Psi(\xi,n)-\bar{\Psi}(\xi,n)) + C_{a,N}\sum_{n=1}^{N}\bar{\Psi}(\xi,n)(\beta_n^1-\bar{\beta}_n^1) \\ &:= \left(\tau(G_1) + J_1^1(\xi) + J_2^1(\xi)\right) + J_3^1(\xi),\end{aligned}$$

where we note that $\bar{\beta}_n^1 = \breve{\beta}_n^1$, since $\phi_{R_{\Delta t}}(\xi)$ is known exactly. Hence $\epsilon(\bar{\phi}_{Z_1}) \leq C^M(a,\bar{a})$, from which we derive

$$\begin{aligned}|\epsilon(\bar{\phi}_{Z_{M-1}}(\xi))| &\leq \frac{C^M(a,\bar{a})}{1-B(a,\xi)}\left(1-B(a,\xi)^{M-2} + (1-B(a,\xi))B(a,\xi)^{M-2}\right) \\ &\leq C^M(a,\bar{a})\frac{1-B(a,\xi)^{M-1}}{1-B(a,\xi)} \leq 2C^M(a,\bar{a}),\end{aligned} \tag{66}$$

for a sufficiently large.

The behavior of $C^M(a,\bar{a})$ can be characterized by noting that with $\bar{a}$ chosen sufficiently large, the truncation error $\tau_M(G)$ is dominated by the other sources. Further, as $\epsilon(\bar{\Psi})$ can be made negligible by a sufficient choice of quadrature, and $\epsilon^M(a,\bar{a})$ converges exponentially in $\bar{a}, a$, the error behaves like $\mathcal{O}(\Delta^{(p+1)})$, which is the projection convergence with respect to the B-spline basis of order p. In particular, from equation (65) we have

$$C^M(a,\bar{a}) = \mathcal{O}(\sqrt{\bar{a}} \cdot C_2(R_{\Delta_t})\Delta^{(p+1)}).$$

Recalling that $\bar{\phi}_{Y_M} = \bar{\phi}_{Z_{M-1}}\phi_{R_{\Delta t}}$,

$$|\epsilon(\bar{\phi}_{Y_M}(\xi_j))| \leq 2C^M(a,\bar{a}) \cdot |\phi_{R_{\Delta_t}}(\xi_j)| = \mathcal{O}(\sqrt{\bar{a}} \cdot \Delta^{(p+1)}|\phi_{R_{\Delta_t}}(\xi_j)|), \quad 2 \leq j \leq N,$$

where we note that $\epsilon(\bar{\phi}_{Y_M}(\xi_1)) = 0$, since $\bar{\phi}_{Y_M}(\xi_1) = 1$ is enforced by the algorithm. Equation (57) then follows from the assumed decay of $\phi_{R_{\Delta_t}}$. ∎

5.2. Valuation Error. The terminal valuation error for a contract on the arithmetic average is now characterized. We show that for bounded payoffs[19] (as for a put, with put-call parity to price a call), the error converges on the order of projection error, $\mathcal{O}(\Delta^{(p+1)})$. If we define $\mathcal{E}(\mathcal{V}_N) := e^{rT}\left(\mathcal{V} \circ g(S_0) - \mathcal{V}_N \circ g(S_0)\right)$, we obtain

$$\begin{aligned}
\mathcal{E}(\mathcal{V}_N) &= \int_{\mathbb{R}} g(y) f_{Y_M}(y)dy - C_{a,N}\sum_{n=1}^{N} \bar{\beta}_n^M g_n \\
&= \int_{\mathbb{R}/G_M} g(y) f_{Y_M}(y)dy + \left(\int_{G_M} g(y) f_{Y_M}(y)dy - C_{a,N}\sum_{n=1}^{N} \beta_n^M g_n\right) \\
&\quad + C_{a,N}\sum_{n=1}^{N}\left(\beta_n^M - \breve{\beta}_n^M + \breve{\beta}_n^M - \bar{\beta}_n^M\right) g_n := \tilde{\tau}_M(G) + \mathcal{E}_1 + \mathcal{E}_2.
\end{aligned}$$

Assuming g is bounded, we have

$$\tilde{\tau}_M(G) = \int_{\mathbb{R}/G_M} g(y) f_{Y_M}(y)dy \leq \|g\|_\infty \cdot \mathbb{P}[Y_M \in G_M^c] = \|g\|_\infty \cdot \tau_M(G),$$

which is controlled by the choice of $\bar{a}$ sufficiently large.[20] In particular, once $\bar{a}$ is fixed, the attainable accuracy of the algorithm is limited by the truncation error.

[19] This assumption is not essential, although it simplifies the analysis.

[20] For unbounded g, as long as the price is finite, the integral $\int_{\mathbb{R}} g(y) f_{Y_M}(y)dy < \infty$, hence $\int_{\mathbb{R}/G_M} g(y) f_{Y_M}(y)dy \to 0$ as $G_M \uparrow \mathbb{R}$. That is, $\tau_M(G) \to 0$ as the truncation error decreases.

Since the coefficients g_n are exact[21], the second source of error satisfies

$$\begin{aligned}\mathcal{E}_1 &= \int_{G_M} g(y) f_{Y_M}(y) dy - C_{a,N} \sum_{n=1}^{N} \beta_n^M g_n \\ &= \int_{G_M} g(y) f_{Y_M}(y) dy - \sum_{n=1}^{N} \langle f_{Y_M}, \widetilde{\varphi}_{a,n} \rangle \int_{G_M} g(y) \varphi_{a,n}(y) dy \\ &= \int_{G_M} g(y) \left(f_{Y_M}(y) - \sum_{n=1}^{N} \langle f_{Y_M}, \widetilde{\varphi}_{a,n} \rangle \varphi_{a,n}(y) \right) dy \\ &\leq \|g\|_2^{G_M} \cdot \|f_{Y_M} - P_{\mathcal{M}_a} f_{Y_M}\|_2^{\mathbb{R}} = \mathcal{O}(\Delta^{(p+1)}). \end{aligned} \tag{67}$$

The third source of error, which accounts for the trapezoidal approximation to the projection coefficients as well as the terminal ChF error, satisfies

$$\begin{aligned}\mathcal{E}_2 &= C_{a,N} \sum_{n=1}^{N} \left(\beta_n^M - \breve{\beta}_n^M + \breve{\beta}_n^M - \bar{\beta}_n^M \right) g_n \\ &= \sum_{n=1}^{N} a^{1/2} C_{a,N} \left(\beta_n^M - \breve{\beta}_n^M + \breve{\beta}_n^M - \bar{\beta}_n^M \right) \cdot \int_{G_M} g(y) \varphi_{a,n}(y) dy. \end{aligned}$$

We will need the following result.

Lemma 5.5. *For any $N_M \in \mathbb{Z}$, it holds that*

$$\sum_{n=1}^{N} C_{a,N} \breve{\beta}_{N_M+n}^1 = \mathcal{O}(1),$$

where $\breve{\beta}_{N_M+n}^1$ are the DFT coefficients of $f_{R_{\Delta_t}}$ corresponding to x_n^M, which are absent of ChF error since $\phi_{R_{\Delta_t}}$ is known (these are not calculated explicitly). For $\bar{a}$ sufficiently large, $\tilde{a} > 0$ can be chosen so that the sum is strictly less than one $\forall a \geq \tilde{a}$.

Proof. We have the following bound

$$\begin{aligned}\sum_{n=1}^{N} C_{a,N} \breve{\beta}_{N_M+n}^1 &\leq \int_{G_M} \left(\sum_{n \in \mathbb{Z}} a^{1/2} C_{a,N} |\breve{\beta}_n^1| \cdot \varphi_{a,n}(y) \right) dy \\ &= \int_{G_M} |\breve{f}_R(y)| dy \leq \int_{G_M} |f_R(y)| dy + \int_{G_M} |\breve{f}_R(y) - f_R(y)| dy, \end{aligned}$$

and the result follows from Corollary 5.1 after applying Cauchy-Swartz inequality to the second integral, and a similar argument as the proof of Lemma 5.2. ∎

We can now provide the convergence rate for the third error source.

Lemma 5.6. *The term $\mathcal{E}_2$ is characterized by*

$$\mathcal{E}_2 = O\left(C^M(a, \bar{a})\right), \tag{68}$$

where $C^M(a, \bar{a})$ is defined in equation (65).

[21] For numerical reasons we have elected instead to use a more stable approximation than the exact coefficients.

Proof. We consider each error $\epsilon_1(\bar{\beta}_n^M) := a^{1/2}C_{a,N}(\beta_n^M - \breve{\beta}_n^M)$ and $\epsilon_2(\bar{\beta}_n^M) := a^{1/2}C_{a,N}(\breve{\beta}_n^M - \bar{\beta}_n^M)$ in turn. Noting that, by Corollary 5.1

$$\sup_{1\le n\le N}|\epsilon_1(\bar{\beta}_n^M)| = a^{1/2}C_{a,N}\sup_{1\le n\le N}\left|\beta_n^M - \breve{\beta}_n^M\right| \le \frac{a^{-1/2}}{\pi}\epsilon^M(a,\bar{a}),$$

where $\epsilon^M(a,\bar{a})$ is defined in equation (63), we have

$$\begin{aligned}
&\sum_{n=1}^N a^{1/2}C_{a,N}\left(\beta_n^M - \breve{\beta}_n^M\right)\int_{G_M} g(y)\varphi_{a,n}(y)dy \\
&\le \left(\sum_{n=1}^N\left(\frac{a^{-1/2}}{\pi}\epsilon^M(a,\bar{a})\right)^2\right)^{1/2}\left(\sum_{n=1}^N\left(\int_{G_M} g(y)\varphi_{a,n}(y)dy\right)^2\right)^{1/2} \\
&= \frac{\bar{a}^{1/2}}{\pi}\epsilon^M(a,\bar{a})\cdot\left(\sum_{n=1}^N|\langle g\mathbb{1}_{G_M},\varphi_{a,n}\rangle|^2\right)^{1/2} \le \frac{\bar{a}^{1/2}}{\pi}\epsilon^M(a,\bar{a})\cdot C_3(\varphi)\cdot\|g\|_2^{G_M},
\end{aligned}$$

where $C_3(\varphi)$ is the upper frame bound of the *dual basis*, $\{\widetilde{\varphi}_{a,n}\}_{n\in\mathbb{Z}}$, and is the inverse of the lower from bound of the "primal" basis[22]. Since $\|g\|_2^{G_M} \le \|g\|_\infty \bar{a}^{1/2}$ (for bounded payoffs), the final inequality is on the order $\mathcal{O}(\bar{a}\epsilon^M(a,\bar{a}))$.
Considering $\epsilon_2(\bar{\beta}_n^M)$, we have (noting the dependence of $h_{a,n}^M(\xi_j)$ on the grid $\{x_n^M\}$)

$$\begin{aligned}
\epsilon_2(\bar{\beta}_n^M) &= \frac{a^{-1/2}}{\pi}\Re\left\{\Delta_\xi{\sum_{j=1}^N}' h_{a,n}^M(\xi_j)\phi_{R_{\Delta_t}}(\xi_j)\epsilon(\bar{\phi}_{Z_{M-1}}(\xi_j))\right\} \\
&= \mathcal{O}\Big(\frac{\epsilon(\bar{\phi}_{Z_{M-1}})}{a^{1/2}\pi}\Re\Big\{\Delta_\xi{\sum_{j=1}^N}' h_{a,n}^M(\xi_j)\phi_{R_{\Delta_t}}(\xi_j)\Big\}\Big) = \mathcal{O}\big(\epsilon(\bar{\phi}_{Z_{M-1}})\sqrt{a}C_{a,N}\breve{\beta}_{N_M+n}\big),
\end{aligned}$$

where $\epsilon(\bar{\phi}_{Z_{M-1}}) := \sup_{1\le j\le N}|\epsilon(\bar{\phi}_{Z_{M-1}}(\xi_j))|$. Hence,

$$\begin{aligned}
\sum_{n=1}^N g_n\frac{\epsilon_2(\bar{\beta}_n^M)}{a^{1/2}} &= \mathcal{O}\left(\epsilon(\bar{\phi}_{Z_{M-1}})\sum_{n=1}^N\left|\int_{G_M} g(y)\varphi_{a,n}(y)dy\right|a^{1/2}C_{a,N}\breve{\beta}_{N_M+n}\right) \\
&= \|g\|_\infty^{G_M}\epsilon(\bar{\phi}_{Z_{M-1}})\mathcal{O}\left(\sum_{n=1}^N a^{1/2}C_{a,N}\breve{\beta}_{N_M+n}\right) = \mathcal{O}\left(\|g\|_\infty^{G_M}\epsilon(\bar{\phi}_{Z_{M-1}})\right)
\end{aligned}$$

by Lemma 5.5, and $\left|\int_{G_M} g(y)\varphi_{a,n}(y)dy\right| \le \|g\|_\infty^{G_M}\int_{G_M}\varphi_{a,n}(y)dy = a^{-1/2}\|g\|_\infty^{G_M}$.

For bounded payoffs, $\|g\|_\infty^{G_M} \le \|g\|_\infty < \infty$, and $\epsilon(\bar{\phi}_{Z_{M-1}}) = \mathcal{O}(C^M(a,\bar{a}))$ by equation (66). Hence, by the definition of $C^M(a,\bar{a})$,

$$\mathcal{E}_2 = \mathcal{O}\left(C^M(a,\bar{a}) + \bar{a}\epsilon^M(a,\bar{a})\right) = O\left(C^M(a,\bar{a})\right). \qquad \blacksquare$$

[22] Duality is used here to obtain a tighter bound, by a factor of $\bar{a}^{1/2}$, than if the standard techniques were applied, which in turn allows us to dominate this source of error by the one derived from $\epsilon_2(\bar{\beta}_n^M)$ next.

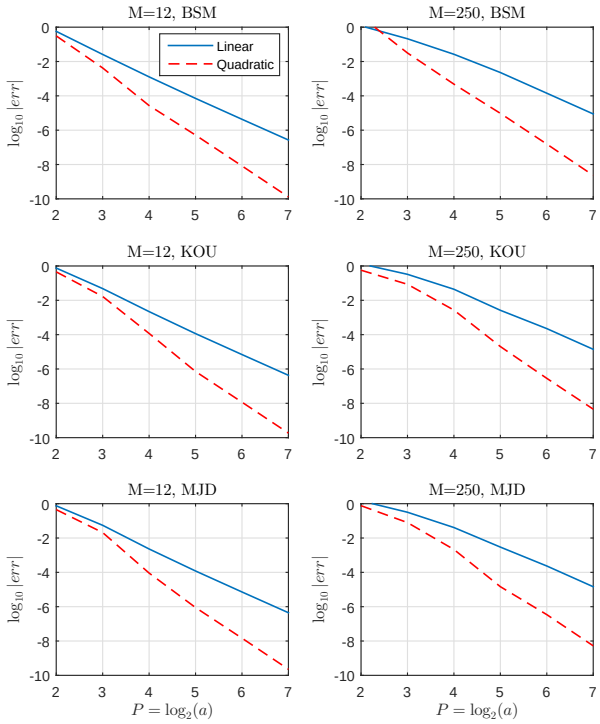

Figure 5: Convergence of linear vs. quadratic APROJ. Parameters as in [22]. Errors are max over strikes $\{90, 100, 110\}$. For MJD and BSM, $\bar{P} := \log_2(\bar{a}) = 3$; for KOU $\bar{P} = 4$. Reference values by linear APROJ with $P = 10$ given in Table 8.

Remark 6. *Combining equations* (67) *and* (68)*, and assuming that* $\bar{a}$ *is chosen to make* $\tau_M(G)$ *(and hence* $\tilde{\tau}_M(G)$*) negligible, we conclude*

$$\mathcal{V} \circ g(S_0) - \mathcal{V}_N \circ g(S_0) = \mathcal{O}\left(e^{-rT} \cdot C^M(a, \bar{a})\right) = \mathcal{O}(\Delta^{(p+1)}).$$

This of course requires that the error contributed by $\bar{\Psi}$ *has been controlled by the choice of quadrature, a choice which may vary by basis. Figure 5 illustrates the difference in convergence rates for the APROJ method with linear and quadratic B-splines.*

6. Numerical Experiments. A major improvement over the breakthrough pricing methods of Clewlow (1990), Benhamou (2002), and later Fusai and Meucci (2008), referred to as FM, was the improved convolution method of Cerny and Kyriakou (2009), referred to as CK. The method of CK represented a major improvement in speed[23], but also demonstrated that references prices reported by the other three are less precise than the four to five decimal places claimed, often correct to only two or three decimals. The ASCOS method of [41] is capable of obtaining precise estimates of prices, but it does not seem to compete with CK in terms of cpu time[24]. The primary drawbacks of ASCOS are its global basis functions, which require several

[23]The results for CK were obtained in MATLAB 7.2 on Dell Latitude 620 Intel(R) Dual Core T7200, 2GHz, 2Gb RAM.

[24]The results for ASCOS were obtained in MATLAB 7.7 with Intel(R) Core(TM)2 Duo CPU E6550, 2.33GHz and 4MB cache size.

Model (Param.)	vol	Calibrated Parameters	Strike 90	100	110
BSM (σ)	0.1	(0.1)	11.581134	3.338617	0.273759
	0.3	(0.3)	13.669816	7.698599	3.896399
	0.5	(0.5)	17.192393	12.091536	8.314413
NIG (ν, σ, θ)	0.1	(0.1222, 0.0879, -0.1364)	11.640247	3.323853	0.158354
	0.3	(0.1222, 0.2637, -0.4091)	13.700850	7.342655	3.278604
	0.5	(0.1222, 0.4395, -0.6819)	16.763062	11.235866	7.168361
CGMY (C, G, M, Y)	0.1	(0.2703, 17.56, 54.82, 0.8)	11.639881	3.324584	0.157877
	0.3	(0.6509, 5.853, 18.27, 0.8)	13.701604	7.347424	3.283082
	0.5	(0.9795, 3.512, 10.96, 0.8)	16.768352	11.244236	7.176236

Table 3: Calibrated parameters from [12]; values reported here to an additional decimal, obtained by quadratic APROJ with $P = 9, \bar{P} = \log_2(\bar{a}) = 3$. Other parameters: $M = 50$, $r = .04$, $q = 0$, $T = 1$.

hundred quadrature points per element of a matrix analogous to $\bar{\Psi}$, and the fixed truncation support (no mean-adjustment)[25]. We also compare to the recent method of Levendorskii and Xie [32], denoted LX, which takes two forms: LX(f) for the flat iFT method, and LX(p) for the parabolic iFT method.[26] Given the difference in computing power for experiments conducted with each of these methods, all comparisons must be understood within context.

Through numerical experiments[27] we demonstrate that APROJ is not only highly accurate (on the level of CK and LX), but is also faster than the state-of-the-art methods to obtain the same or superior accuracy, typically on the order of a 10- to 100-fold improvement. This is true for both linear and quadratic implementations. Given that the initial peak of f_R is quickly softened to obtain f_{Y_m}, we find that quadratic APROJ is remarkably accurate for Asian option pricing, and is presented next. Numerical results for linear APROJ (not presented), are also impressive.

In the first few sets of experiments, to isolate the rate of convergence of APROJ, we conservatively fix $\bar{P} := \log_2(\bar{a}) = 3$ for pure diffusion models, and $\bar{P} = 3 \sim 4$ for heavy-tailed models, such as CGMY and NIG. For most cases, a smaller value of $\bar{P}$ would have sufficed (especially with BSM experiments), and reduced the computation time.[28] Sensitivity of APROJ with respect to the choice of $\bar{P}$ is illustrated in Figure 7. While these experiments are chosen to illustrate the error decay as a function of resolution, once the truncation parameter $\bar{P}$ is fixed, the attainable accuracy is always limited by the truncation error. The final set of experiments investigates the automated approach to parameter selection which is often much more efficient, improving cpu times even further.

Our first set of experiments compares the convergence and cpu time of APROJ against the method of CK for $M = 50$ and strikes $\{90, 100, 110\}$. The specifications

[25]The author's indicate that a grid adjustment is possible, but to do so would require re-computation of the matrix analogous to $\bar{\Psi}$ at each step (or every several steps), and would incur a substantial cost.

[26]The results of LX were obtained in MATLAB 7.11.0, with an Intel(R) Celeron(R) Processor T1600, 1.66GHz, 667MHz FSB and 1MB cache.

[27]The results for APROJ were obtained in MATLAB 8.0 with Intel(R) Core(TM) i5-3470T CPU, 2.90GHz with 3MB cache size.

[28]Moreover, Boole's rule, which is faster than the seven point rule, obtains nearly identical results in many of the cases. However, it is safer in practice to use the more accurate method, so this is how we present the results.

		Quadratic APROJ, $P = \log_2(a)$							APROJ	CK
	vol	1	2	3	4	5	6	7	cpu(sec)	cpu(sec)
BSM	.1	4.8e+00	1.1e+00	1.6e-01	7.3e-03	3.1e-05	2.4e-07	3.1e-09	.008/.202	1.0
	.3	3.8e+00	9.2e-02	8.8e-04	5.6e-06	6.2e-08	8.8e-10	1.3e-11	.003/.009	.3
	.5	1.7e+00	4.8e-03	3.0e-05	2.9e-07	3.9e-09	5.8e-11	9.5e-13	.001/.008	.3
NIG	.1	4.3e+00	1.1e+00	1.5e-01	1.7e-02	2.2e-04	1.3e-06	1.1e-08	.026/.203	3.7
	.3	2.9e+00	1.8e-01	6.4e-03	4.4e-05	1.4e-07	2.6e-09	3.9e-11	.003/.028	1.8
	.5	2.6e+00	2.1e-02	2.2e-04	1.6e-06	5.9e-09	6.8e-09	4.8e-09	.003/.009	1.8
CGMY	.1	3.3e+00	1.1e+00	1.5e-01	1.6e-02	2.2e-04	1.3e-06	9.9e-09	.027/.201	8.5
	.3	2.9e+00	1.7e-01	6.2e-03	4.2e-05	1.1e-07	2.4e-09	3.6e-11	.004/.027	4.1
	.5	3.0e+00	2.1e-02	2.1e-04	1.5e-06	3.9e-08	1.4e-08	7.7e-09	.004/.009	2.1

Table 4: Parameters from CN [12]. For ARPOJ with $\bar{P} = 3$ and the seven point rule, each cpu pair $\cdot/\cdot$ reports the time to achieve an error of $\mathrm{TOL}_1/\mathrm{TOL}_2$, where TOL_1 is on the order of e-03∼e-04 and TOL_2 is on the order of e-06∼e-07. The error is taken as the maximum abs. error over the strike set $\{90, 100, 110\}$. CK prices are on the order of e-05∼e-06. Ref prices are provided in Table 3.

considered are the log-normal, ie Black-Scholes-Merton (BSM), the Normal Inverse Gaussian (NIG), and the Carr-Geman-Madan-Yor (CGMY) model. Three test cases are considered for each model, with parameters calibrated by [12] to a fixed volatility (vol) in the set $\{0.1, 0.3, 0.5\}$. Recovered values, as well as calibrated parameters are provided for each strike in Table 3. For the NIG model, we use the alternative ChF form with parameters (ν, σ, θ) to maintain consistency with [12], which has Levy symbol

$$\psi_L(\xi) = \frac{1}{\nu}\left(1 - \sqrt{1 - 2\theta\nu i\xi + \nu\sigma^2\xi^2}\right).$$

In Table 4, we see rapid convergence of the quadratic APROJ method, which is implemented with the seven point rule and $\bar{P} = 3$. By $P = 5$, accuracy on the order e-07∼e-09 is achieved for vol $\in \{0.3, 0.5\}$ and for all models. With $P = 7$, accuracy on the order e-08 is achieved for all models and levels of vol. In the far right column of Table 4 we provide the cpu times reported by [12] to achieve within four to five correct decimals, which are at least a factor of 10 more than the time required for APROJ to reach e-06∼e-07 accuracy (with only one exception), and are often *more than 100 times* that of APROJ. This is consistent across all models and specifications as well as strikes tested. Similar results hold for the linear implementation of APROJ.

For the set of experiments in Table 4 involving the CMGY (KoBoL) model, we can also compare our results to those of LX [32], using the parabolic method LX(p). When vol = 0.1, they report a max error of 6.7e-05 over strikes in $\{90, 100, 110\}$ at a cost of 1.581 seconds (compared to an APROJ accuracy of 1.3e-06 in 0.201 seconds). When vol =0.3, they achieve 3.9e-06 in 1.037 seconds (compared to 1.2e-07 in 0.027 seconds), and when vol =0.5 they achieve 4.6e-06 in 0.684 seconds (compared to 1.5e-06 in 0.009 seconds). In each of these cases, APROJ obtains greater accuracy and at the same time provides a 7.8, 38, and 76-fold time reduction respectively.

The second set of experiments compares the convergence and cpu time of APROJ against the ASCOS method for $M \in \{12, 50, 250\}$ and strikes $\{90, 100, 110\}$. For this case, we specify the NIG model with parameter set in [23],

$$(\alpha, \beta, \delta) = (6.1882, -3.8941, 0.1622), \quad r = 0.0367, \quad q = 0, \quad T = 1, \quad S_0 = 100,$$

and ChF given in Table 6 of the appendix. Table 5 reports the convergence of quadratic APROJ, along with the reference prices. Reference prices as well as re-

M	strike	Quadratic APROJ, $P = \log_2(a)$ 1	2	3	4	5	6	7	NIG reference
	90	5.3e-01	2.0e-01	6.4e-03	1.7e-04	3.4e-08	2.2e-09	3.8e-11	12.62243
12	100	1.9e+00	4.0e-01	1.9e-03	8.4e-04	8.4e-06	7.2e-08	1.1e-09	5.06060
	110	1.5e+00	3.0e-01	9.2e-03	9.1e-04	1.6e-05	6.3e-08	1.3e-09	1.01355
	90	6.6e-01	4.2e-01	4.3e-02	2.5e-03	1.9e-07	1.1e-09	6.6e-11	12.66126
50	100	2.3e+00	5.0e-01	1.4e-02	8.3e-03	2.7e-05	1.0e-07	1.9e-09	5.10370
	110	2.1e+00	6.1e-01	9.8e-02	2.3e-03	2.0e-04	7.6e-07	4.4e-09	1.03770
	90	3.3e-01	4.4e-01	5.6e-02	6.4e-03	6.6e-05	6.0e-07	2.3e-08	12.67176
250	100	1.9e+00	3.9e-01	3.6e-02	1.8e-02	3.6e-05	2.9e-06	1.7e-07	5.11556
	110	1.8e+00	6.2e-01	1.3e-01	1.1e-02	2.5e-04	3.7e-06	3.0e-07	1.04448

Table 5: NIG parameters from FM [23], $(\alpha, \beta, \delta) = (6.1882, -3.8941, 0.1622)$, and $r = 0.0367, q = 0, T = 1, S_0 = 100$. Convergence for quadratic APROJ with $\bar{P} = 4$ and seven-point rule. Reference values obtained by quadratic APROJ with $P = 9, \bar{P} = 4$, and seven point rule.

M		ASCOS $N = 128$ $n_q = 200$	$N = 256$ $n_q = 400$	$N = 384$ $n_q = 600$	Quadratic APROJ $\bar{P} = 4$ Seven-Point		
12	|err.|	2.0e-03	1.74e-04	5.16e-06	9.1e-04	1.6e-05	6.3e-08
	(sec)	(2.41)	(15.13)	(46.09)	(.017)	(.085)	(.314)
50	|err.|	2.26e-04	6.94e-05	2.17e-06	2.0e-04	7.6e-07	4.4e-09
	(sec)	(2.43)	(15.16)	(46.22)	(.190)	(.731)	(2.94)
250	|err.|	7.8e-03	9.33e-05	8.49e-06	2.5e-04	3.7e-06	2.8e-07
	(sec)	(2.42)	(15.23)	(46.68)	(.717)	(2.94)	(11.42)

Table 6: NIG parameters from FM [23], $(\alpha, \beta, \delta) = (6.1882, -3.8941, 0.1622)$, and $r = 0.0367, q = 0, T = 1, S_0 = 100$. APROJ with $\bar{P} = 4$, seven point rule. Corresponding values of P for each accuracy are given in Table 5. Absolute errors for strike $W = 110$.

ported cpu times are provided for $\bar{P} = 4$ and the seven point rule[29]. In Table 6 the performance of APROJ is compared to ASCOS, with similar findings as in the first set of experiments. For example, when $M = 12$, ASCOS requires 15.13 seconds to achieve 1.74e-04 accuracy, while APROJ reaches 1.6e-05 accuracy in 0.085 seconds, an almost 200-fold improvement. To reach 6.3e-08 accuracy takes APROJ 0.314 seconds compared to 46.09 seconds for ASCOS to reach 5.15e-06. For other cases of *comparable accuracy*, the improvement is by at least a factor of 10 or more.

We next consider a KoboL (CGMY) model from Levendorskii and Xie [32], with parameters $CGMY = (1.1136, 3, 10, 0.2)$, or in terms of the KoBoL [7, 9] parameterization $(c, \lambda_-, \lambda_+, \nu) = (1.1136, -10, 3, 0.2)$. As demonstrated in Table 7, the APROJ method converges rapidly to high accuracy. Two methods from [32] are provided for comparison, the LX(f) method and LX(p), neither of which seems to dominate the other in terms of speed or accuracy from the experiments provided in [32]. In this case, LX(f) is slower to converge (in terms of cpu), but for strikes $\{90, 100, 100\}$, both methods of [32] reach an accuracy of about (2.1e-07, 7.8e-07, 1.7e-06) respectively. The APROJ method with $P = 5$ achieves accuracy of (9.3e-09, 7.5e-08, 5.0e-06), with a cpu time reduction factor of 9.65 for the LX(p) method and a 338-fold reduction for LX(f). To fully understand the difference in efficiency between these two methods however, experiments will need to be conducted on the machine.

[29]The necessarily larger value of $\bar{P}$ is detected by recovering the value of $\bar{\beta}_1^1$ prior to the algorithm's initialization.

strike	Ref.	Quadratic APROJ, $P = \log_2(a)$ 2	3	4	5	LX(f)	LX(p)
90	14.795530855	6.349e-02	1.161e-04	1.136e-05	9.312e-09	2.1e-07	2.1e-07
100	8.281218252	2.973e-02	2.641e-04	3.467e-05	7.533e-08	7.8e-07	7.8e-07
110	3.718094231	1.523e-01	1.040e-03	1.951e-04	5.002e-06	1.7e-06	1.8e-06
	cpu (sec)	0.003	0.007	0.016	0.082	27.77	0.792

Table 7: CGMY (KoBoL) Parameters from Levendorskii and Xie [32]: $S_0 = 100$, $M = 12$, $T = 1$, $r = 0.04$, $q = 0$, $CGMY = (1.1136, 3, 10, 0.2)$; in terms of KoBoL parameterization, $(c, \lambda_-, \lambda_+, \nu) = (1.1136, -10, 3, 0.2)$. Convergence for quadratic APROJ with $\bar{P} = 4$ and seven-point rule. Reference values obtained by quadratic APROJ with $P = 11, \bar{P} = 5$, and seven point rule, and verified to seven decimals with prices of [32]. The LX(f) and LX(p) methods are respectively the flat and parabolic Fourier transform methods of [32].

Model	Parameters	Strike	Reference Values $M = 12$	$M = 250$
BSM	$\sigma = 0.17801$	90	11.9049157	11.9405632
		100	4.8819616	4.9521569
		110	1.3630380	1.4133670
KOU	$\sigma = 0.120381$	90	12.713070	12.753177
	$\lambda = 0.330966, \quad p = 0.2071$	100	5.017859	5.070220
	$\eta_1 = 9.65997, \quad \eta_2 = 3.13868$	110	1.041531	1.076568
MJD	$\sigma = 0.126349$	90	12.710669	12.749182
	$\lambda = 0.174814$	100	5.011290	5.063823
	$\mu_J = -0.390078, \quad \sigma_J = 0.338796$	110	1.051633	1.087406

Table 8: Model parameters from [22, 23], and $r = 0.0367, q = 0, T = 1, S_0 = 100$. Reference values by Linear APROJ, $P = 10$. For MJD and BSM, $\bar{P} = 3$; for KOU $\bar{P} = 4$.

Now we consider the BSM model, Merton's Jump Diffusion (MJD), and Kou's double exponential (KOU) model, which characteristic functions given in Figure 6. Parameters are as in [23] (later used in [22]), which are provided in Table 8 along with reference values. The parameter setting for BSM is also considered in [12]. Convergence is compared for the linear and quadratic implementation of APROJ in Figure 5.

The first observation is that the prices obtained for BSM agree with those of CK [12] to 7 decimals (the other two models are not reported in [12]), while the method of FM [23] is accurate to only about 2-3 decimals in most cases with cpu times in excess of 5 seconds (this is pointed out as well in [12]). Greater accuracy is obtained by APROJ in just milliseconds. When $M = 250$, $K = 100$, the price to seven decimals is given by 4.9521569, as computed by CK and APROJ. FM obtains 4.95233, while the maturity randomization methods of Fusai, Marazzina and Marena (FMM) [22] report prices of 4.95212 and 4.95242, using density recursion and price recursion respectively, with cpu times of 38.32 seconds and 95.80 seconds[30]. We find similar results for the models of KOU and MJD, where the prices of FMM agree with those computed by ARPOJ (given in Table 8) to 3 or 4 decimals with FMM cpu times in the dozens of seconds, compared to milliseconds for APROJ.

The previous experiments illustrate the convergence of APROJ as a function of the resolution when the grid width is fixed a priori. The final set of experiments

[30]The results for FMM were obtained in MATLAB 7.4 on a personal computer with Intel(R) Core 2 Quad Q6600, 2.4GHz, 4Gb RAM.

Model	W	M	Ref	|Err|	cpu(sec)	NL_1	NL_2	$\log_2(N)$
CGMY	90	12	11.999099	2.33e-06	0.006	1	1	6
(0.2703, 17.56, 54.82, 0.8)	100	250	3.643684	7.01e-07	0.083	2	1	8
CGMY	90	12	15.061188	5.63e-04	0.005	1	1	6
(1.1136, 3, 10, 0.2)	100	250	8.644264	7.80e-06	0.471	2	2	9
MJD	90	12	13.134793	6.10e-04	0.005	1	1	6
(0.13, 0.17, -0.39, 0.34)	100	250	5.480458	1.02e-04	0.103	1	2	8
MJD	90	12	12.704098	1.02e-06	0.005	1	1	6
(0.1, 3, -0.05, 0.086)	100	250	5.620436	6.79e-06	0.030	1	1	7
BSM	90	12	11.949574	4.22e-07	0.006	1	1	6
$\sigma = 0.1$	100	250	3.639486	2.02e-06	0.029	1	1	7
BSM	90	12	13.854399	1.96e-06	0.006	1	1	6
$\sigma = 0.3$	100	250	7.939288	4.92e-06	0.028	1	1	7
NIG	90	12	12.290729	5.89e-05	0.005	1	1	6
(8, -1, 0.2)	100	250	4.610758	1.70e-07	0.423	3	1	9
Kou	90	12	13.564345	9.71e-04	0.005	1	1	6
(0.15, 0.4, 0.2, 9, 3)	100	250	6.297930	2.33e-05	0.473	2	2	9

Table 9: Call price errors for quadratic APROJ with automated parameter selection. Cpu times represent full cost including parameter determination. Columns NL_1 and NL_2 are the number of loops required in initialization (Subroutine 1) and the main algorithm (Algorithm 3) before tolerance is met, where $\epsilon_1 = 5e\text{-}04, \epsilon_2 = 5e\text{-}04$, and $\epsilon_3 = 5\text{e-}03$ in Algorithm 3. N is the final grid size. In all cases, $S_0 = 100$, $r = 0.05$, $q = 0$, $T = 1$. MJD params: $(\sigma, \lambda, \mu_J, \sigma_J)$. Kou params: $(\sigma, \lambda, p, \eta_1, \eta_2)$. NIG params: (α, β, δ)

analyzes the ability of the APROJ algorithm to accurately select parameters without user input, as described in Section 3.6 (and implemented in Subroutine 1) to achieve a practical accuracy of about TOL = 5e-04 or better.[31] While our choice of tolerance reflects a level that is conservative for practical applications, given the presence of truncation error, it will not typically yield an accuracy beyond e-06 with the chosen tolerance.

Table 9 considers several models and settings for M and W, with reference prices obtained by APROJ with $N = 2^{13}$. Based on the prescription given in Section 3.6, for $M = 12$ the algorithm is initialized with $N = 2^6$ and and grid width multiplier $L_1 = 12$, while for $M = 250$ we set $N = 2^7$ and $L_1 = 16$. The column labeled $\log_2(N)$ reports the final value after satisfying all error tolerances. The column labeled NL_1 is the number of iterations required in initialization Subroutine 1 before the error tolerances ϵ_1 and ϵ_2 were satisfied (so $NL_1 = 1$ implies that the initial estimate of N and $\bar{a}$ were sufficient). Column NL_2 is the number of loops in the main Algorithm 3 before the terminal valuation criteria was satisfied.

Ideally, since the cost of Subroutine 1 is negligible, we would prefer it to identify insufficient settings of N and $\bar{a}$ prior to entering Algorithm 3. Either way we see that these three consistency checks are more than sufficient to achieve desired accuracy. Column cpu(sec) reports the time in seconds for the full procedure, which is generally fractions of a second, including the cost of demeriting initial values for N and $\bar{a}$. From the columns NL_1 and NL_2, we see that the method is reliable at detecting insufficient initial choices of the parameters N and $\bar{a}$. In particular, the method is able to correct for the often unreliable estimate provided by cumulants alone.

[31] We have selected to the parameters $\epsilon_1, \epsilon_2, \epsilon_3$ to attain an accuracy of TOL = 5e-04 or better. However, these parameters, as well as the initial value of N and L_1 can be increased if the desired accuracy is beyond what is required in practice.

Model	T	M	Ref	|Err|	cpu(sec)	NL_1	NL_2	$\log_2(N)$
CGMY	2	50	5.969720	1.79e-06	0.008	1	1	6
	2	250	5.977913	2.60e-07	0.084	2	1	8
(0.2703, 17.56, 54.82, 0.8)	4	50	9.896446	1.09e-06	0.008	1	1	6
	4	250	9.902125	7.24e-07	0.031	1	1	7
NIG	2	50	7.331371	3.56e-05	0.014	2	1	7
	2	250	7.344673	9.35e-05	0.082	2	1	8
(8, -1, 0.2)	4	50	11.592820	5.46e-04	0.007	1	1	6
	4	250	11.606504	6.31e-05	0.080	2	1	8
Kou	2	50	9.911194	9.80e-05	0.013	2	1	7
	2	250	9.930953	1.39e-06	0.465	2	2	9
(0.15, 0.4, 0.2, 9, 3)	4	50	15.244009	1.94e-05	0.014	2	1	7
	4	250	15.268818	3.93e-07	0.459	2	2	9

Table 10: Call price errors for quadratic APROJ with automated parameter selection. Columns NL_1 and NL_2 are the number of loops required in initialization (Subroutine 1) and the main algorithm (Algorithm 3) before tolerance is met, where $\epsilon_1 = 5e\text{-}04, \epsilon_2 = 5e\text{-}04$, and $\epsilon_3 = 5e\text{-}03$ in Algorithm 3. N is the final grid size. In all cases, $S_0 = W = 100$, $r = 0.05$, $q = 0$.

The algorithm's reliability extends to larger maturities T illustrated in Table 10, for which we consider durations of two and four years with heavy-tailed models. This is especially beneficial given that the standard cumulant-based approach often underestimates the required grid-width for longer maturities. We conclude that APROJ is capable of obtaining accurate prices at a very small computational cost when the algorithm, rather than the user, determines the required values of N and $\bar{a}$ needed to achieve the designated tolerance.

7. Implementation into Premia 22. We implemented the APROJ-method for the fixed strike vanilla Asian options (calls and puts) in

- the Kou model (see Example 4);
- the NIG model (see Example 3);
- the CGMY (KoBoL) model (see Example 1).

Note that in the program implemented into Premia 22 for the Asian options one can manage by three parameters of the algorithm: the scale of log-price range L, the number of discrete monitoring points M, and the x-grid parameter n that fixes the number of the x-grid points as $N = 2^n$. Parameter L controls the size of the truncated region in x-space. The typical values of the parameter for Lévy models are varying from $L = 8$ to $L = 15$.

Appendix A. Proofs.

Proof of Proposition 3.1. We proceed by induction where $m = 1$ follows from $Y_1 = R_M$. Fix $m \geq 2$ and assume (i) and (ii) hold for $m-1$. First we show finiteness of $\phi_{Z_{m-1}}(z)$ for any fixed $z = x + i\eta \in \mathcal{D}_d$. Consider the case of $\eta \in (-d, 0)$ (finiteness for $\eta \in [0, d)$ follows immediately). Since $\phi_{Y_{m-1}}(x+i\eta) = \int_{\mathbb{R}} e^{i(x+i\eta)y} f_{Y_{m-1}}(y)dy < \infty$, ie the integral exists and is finite, $e^{i(x+i\eta)y} f_{Y_{m-1}}(y)$ must be absolutely integrable in y, from which $\int_{\mathbb{R}} e^{-\eta y} f_{Y_{m-1}}(y)dy < \infty$. Note that

$$\int_{\mathbb{R}} |e^{i(x+i\eta)\log(1+e^y)} f_{Y_{m-1}}(y)|dy \leq e^{-\eta\log(2)} \int_{-\infty}^{0} f_{Y_{m-1}}(y)dy + \int_{0}^{\infty} e^{-\eta\log(1+e^y)} f_{Y_{m-1}}(y)dy.$$

Model	$\psi_L(\xi)$, $\omega = -\psi_L(-i)$, Param. Restrictions	Cumulants
BSM	$\psi_L(\xi) = -\frac{\sigma^2}{2}\xi^2$	$c_1 = \gamma$
	$\omega = -\frac{\sigma^2}{2}$	$c_2 = \sigma^2$
	$\sigma > 0, \quad \mathcal{I}_L = \mathbb{R}$	$c_4 = 0$
MJD	$\psi_L(\xi) = -\frac{\sigma^2}{2}\xi^2 + \lambda\Big(\exp(i\xi\mu_J - \frac{\sigma_J^2}{2}\xi^2) - 1\Big)$	$c_1 = \gamma + \lambda\mu_J$
	$\omega = -\frac{\sigma^2}{2} - \lambda\Big(\exp(\mu_J + \frac{\sigma_J^2}{2}) - 1\Big)$	$c_2 = \sigma^2 + \lambda(\mu_J^2 + \sigma_J^2)$
	$\lambda, \sigma_J, \sigma > 0, \quad \mathcal{I}_L = \mathbb{R}$	$c_4 = \lambda(\mu_J^4 + 6\sigma_J^2\mu_J^2 + 3\sigma_J^4\lambda)$
CGMY	$\psi_L(\xi) = C\Gamma(-Y)\left((M - i\xi)^Y - M^Y + (G + i\xi)^Y - G^Y\right)$	$c_1 = \gamma + C\Gamma(1-Y)(M^{Y-1} - G^{Y-1})$
	$\omega = -C\Gamma(-Y)((M-1)^Y - M^Y + (G+1)^Y - G^Y)$	$c_2 = C\Gamma(2-Y)(M^{Y-2} + G^{Y-2})$
	$C, G > 0, M > 1, Y \in (0,1)\cup(1,2), \quad \mathcal{I}_L = [-M, G]$	$c_4 = C\Gamma(4-Y)(M^{Y-4} + G^{Y-4})$
NIG	$\psi_L(\xi) = -\delta\left(\sqrt{\alpha^2 - (\beta + i\xi)^2} - \sqrt{\alpha^2 - \beta^2}\right)$	$c_1 = \gamma + \delta\beta/\sqrt{\alpha^2 - \beta^2}$
	$\omega = \delta\left(\sqrt{\alpha^2 - (\beta+1)^2} - \sqrt{\alpha^2 - \beta^2}\right)$	$c_2 = \delta\alpha^2(\alpha^2 - \beta^2)^{-3/2}$
	$\alpha, \delta > 0, \beta \in (-\alpha, \alpha - 1), \quad \mathcal{I}_L = [\beta \pm \alpha]$	$c_4 = 3\delta\alpha^2(\alpha^2 + 4\beta^2)(\alpha^2 - \beta^2)^{-7/2}$
KOU	$\psi_L(\xi) = -\frac{\sigma^2}{2}\xi^2 + \lambda\Big(\frac{p\eta_1}{\eta_1 - i\xi} + \frac{(1-p)\eta_2}{\eta_2 + i\xi} - 1\Big)$	$c_1 = \gamma + \frac{\lambda p}{\eta_1} - \frac{\lambda(1-p)}{\eta_2}$
	$\omega = -\frac{\sigma^2}{2} - \lambda\Big(\frac{p\eta_1}{\eta_1 - 1} + \frac{(1-p)\eta_2}{\eta_2 + 1} - 1\Big)$	$c_2 = \sigma^2 + 2\frac{\lambda p}{\eta_1^2} + 2\frac{\lambda(1-p)}{\eta_2^2}$
	$\lambda, \sigma > 0, p \in [0,1], \eta_1 > 1, \eta_2 > 0, \quad \mathcal{I}_L = (-\eta_1, \eta_2)$	$c_4 = 24\lambda(\frac{p}{\eta_1^4} + \frac{1-p}{\eta_2^4})$
VG	$\psi_L(\xi) = -\frac{\sigma^2}{2}\xi^2 - \frac{1}{\nu}\log\left(1 - i\nu\theta\xi + \nu\frac{\sigma_V^2}{2}\xi^2\right)$	$c_1 = \gamma + \theta$
	$\omega = -\frac{\sigma^2}{2} + \frac{1}{\nu}\log\left(1 - \nu\theta - \nu\frac{\sigma_V^2}{2}\right)$	$c_2 = \sigma^2 + \sigma_V^2 + \nu\theta^2$
	$\nu, \sigma_V > 0, \sigma \geq 0, \quad \mathcal{I}_L = \left[\frac{\theta}{\sigma_V^2} \pm \sqrt{\frac{\theta^2}{\sigma_V^4} + \frac{2}{\nu\sigma_V^2}}\right]$	$c_4 = 3(\sigma_V^4\nu + 2\theta^4\nu^3 + 4\sigma_V^2\theta^2\nu^2)$

Figure 6: Symbols $\psi_L(\xi)$, cumulants c_n of $\log(S_{t+1}/S_t)$, parameter restrictions and strip of analyticity $\mathcal{I}_L$ for tractable Levy processes. $\gamma := r - q - \psi_L(-i) = r - q + \omega$. Note that $\mathbb{E}[\log(S_{t+1}/S_t)] = c_1 = r - q + w + \mathbb{E}[L(1)]$, and $\mathbb{E}[R_{\Delta_t}] = c_1\Delta_t$.

To bound the second integral, note that $\exists\tilde{\eta} \in (-d, \eta)$, and $\tau > 0$ s.th $\forall y > \tau$, $-\tilde{\eta}y > -\eta\log(1 + e^y)$. Hence,

$$\int_\tau^\infty e^{-\eta\log(1+e^y)} f_{Y_{m-1}}(y)dy \leq \int_\tau^\infty e^{-\tilde{\eta}y} f_{Y_{m-1}}(y)dy \leq \int_{\mathbb{R}} e^{-\eta y} f_{Y_{m-1}}(y)dy < \infty,$$

so $\phi_{Z_{m-1}}(z)$ exists and is finite $\forall z \in \mathcal{D}_d$. To prove continuity, fix any $\{z_n\} \in \mathcal{D}_d$ with $z_n \to z \in \mathcal{D}_d$. Let $G \subset \mathcal{D}_d$ be a bounded open set containing the tail of $\{z_n\}$, so $\bar{G} \subset \mathcal{D}_d$. With $\bar{\eta} := \max\{|\eta| : z = x + i\eta \in \bar{G}\}$, note that for any $z \in \bar{G}$ it holds

$$|e^{iz\log(1+e^y)} f_{Y_{m-1}}(y)| \leq e^{\bar{\eta}\log(1+e^y)} f_{Y_{m-1}}(y) = |e^{i\bar{z}\log(1+e^y)} f_{Y_{m-1}}(y)|,$$

where $\bar{z} = x - i\bar{\eta}$ for arbitrary $x \in \mathbb{R}$, since $\log(1 + e^y) \geq 0$ for all $y \in \mathbb{R}$. Hence

$$\sup_{z\in\bar{G}} |\phi_{Z_{m-1}}(z)| \leq |\phi_{Z_{m-1}}(\bar{z})| < \infty,$$

where finiteness of $\bar{z} \in \bar{G} \subset \mathcal{D}_d$ was proved above, so by dominated convergence

$$\lim_{z_n\to z} \phi_{Z_{m-1}}(z_n) = \int_{\mathbb{R}} \lim_{z_n\to z} \exp\left(iz_n\log(1 + e^y)\right) f_{Y_{m-1}}(y)dy = \phi_{Z_{m-1}}(z).$$

Analyticity is now proved as follows. Fix any positively oriented triangle $\Gamma \in \mathcal{D}_d$. By Fubini's theorem

$$\int_\Gamma \phi_{Z_{m-1}}(z)dz = \int_{\mathbb{R}} f_{Y_{m-1}}(y) \int_\Gamma \exp\left(iz\log(1 + e^y)\right) dzdy = 0,$$

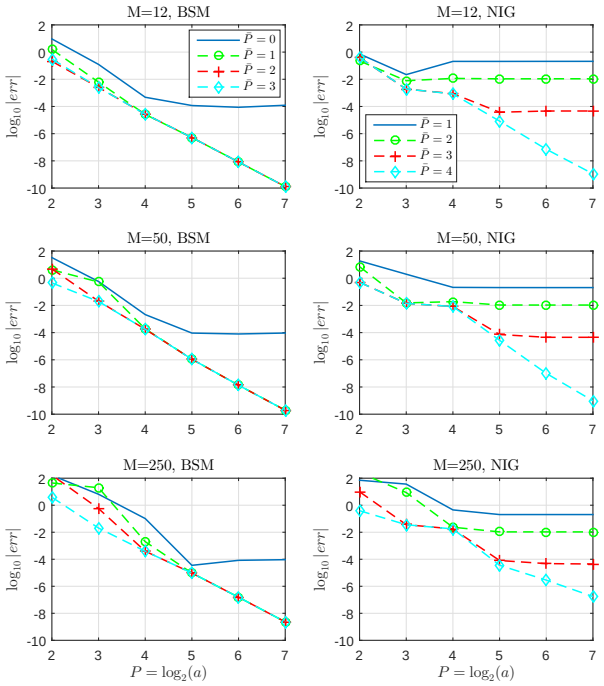

Figure 7: Convergence in $\bar{P}$ of quadratic APROJ prices for BSM and NIG models (one legend for each model). Parameters and reference values as in Table 8, strike $W = 100$.

where the final equality holds by Cauchy's theorem. Hence, by Morera's theorem, we conclude that $\phi_{Z_{m-1}}(z)$ is analytic on $\mathcal{D}_d$, and so too is $\phi_{Y_m}(z) = \phi_R(z)\phi_{Z_{m-1}}(z)$. The growth estimate (ii) follows immediately from $|\phi_{Z_{m-1}}(\xi)| \leq 1$ for $\xi \in \mathbb{R}$. ∎

Proof of Lemma 3.1. Let $[-\lambda, \lambda]$ be the support of φ. For $a > 0$, $\xi \in [0, 2\pi a)$,

$$\begin{aligned}
\left|a^{1/2}\mathcal{F}[\varphi_{a,n}](\xi) - \bar{\Psi}(\xi, n)\right| &\leq a^{1/2} \int \varphi_{a,n}(y) \left|e^{i\xi y} - e^{i\xi \log(1+\exp(y))}\right| dy \\
&= \int_{-\lambda}^{\lambda} \varphi(y) \left|e^{i\xi(x_n + \frac{y}{a})}\right| \left|1 - e^{i\xi\left(\log(1+\exp(x_n+\frac{y}{a})) - (x_n+\frac{y}{a})\right)}\right| \\
&\leq |\xi| \int_{-\lambda}^{\lambda} \varphi(y) \left|\log\left(1 + \exp\left(x_n + \frac{y}{a}\right)\right) - \left(x_n + \frac{y}{a}\right)\right| dy \\
&\leq 2\pi a \left(\log(1 + \exp(x_{n-1})) - x_{n-1}\right) \int_{-\lambda}^{\lambda} \varphi(y) dy \\
&= 2\pi a \left(\log(1 + \exp(x_{n-1})) - x_{n-1}\right),
\end{aligned}$$

where the next to last line follows since $\log(1 + \exp(x)) - x$ is strictly decreasing. An asymptotic expansion yields

$$\log(1 + \exp(x_{n-1})) - x_{n-1} \sim e^{-x_{n-1}} - e^{-2x_{n-1}}/2 + \mathcal{O}(e^{-3x_{n-1}}),$$

and the result follows from $\mathcal{F}[\varphi_{a,n}](\xi)$. ∎

Proof of Proposition 2.1. We provide a proof here for linear case, with a bound on the term $C_\gamma^{[1]}(\phi_X)$. The more general case of a p^{th} order basis is discussed in [28].

First define

$$h_{a,n}(\xi) := 12\frac{\sin^2(\xi/2a)}{(\xi/a)^2(2+\cos(\xi/a))}\exp(-ix_n\xi) := h_a(\xi)\exp(-ix_n\xi), \tag{69}$$

and $\xi_j = (j-1)\Delta_\xi$ where $\Delta_\xi = 2\pi a/N$. We have that $\epsilon(\breve{\beta}^X_{a,n}) := a^{1/2}C_{a,N}\cdot\bar{\beta}^X_{a,n} - \langle f_X, \widetilde{\varphi}_{a,k}\rangle$ satisfies

$$\begin{aligned}\epsilon(\breve{\beta}^X_{a,n}) &= \frac{a^{-1/2}}{\pi}\Re\left(\Delta_\xi\sum_{j=1}^{N}\nu_j\phi_X(\xi_j)h_{a,n}(\xi_j) - \int_0^\infty \phi_X(\xi)h_{a,n}(\xi)d\xi\right)\\ &= \frac{a^{-1/2}}{\pi}\Re\left(\Delta_\xi\sum_{j=1}^{\infty}\tilde{\nu}_j\phi_X(\xi_j)h_{a,n}(\xi_j) - \int_0^\infty \phi_X(\xi)h_{a,n}(\xi)d\xi\right.\\ &\qquad\left.+\Delta_\xi\sum_{j=N}^{\infty}\bar{\nu}_j\phi_X(\xi_j)h_{a,n}(\xi_j)\right) := \frac{a^{-1/2}}{\pi}\left(\epsilon_{trap}(a,\bar{a}) + \tau_a(X)\right),\end{aligned}$$

where $\nu_j := 1-(\delta_{j,1}+\delta_{j,N})/2$, $\tilde{\nu}_j = 1-\delta_{j,1}/2$, and $\bar{\nu}_j = 1-\delta_{j,N}/2$. To apply Theorem 3.2.1 in [37], we must show that the presence of $h_a(\xi)$ does not affect the integrand's analyticity or the finiteness of the Hardy norm, both of which will follow if we can bound $h_a(\xi)$ in a strip contained within $\mathcal{D}_d$ (note that Proposition 3.1 of [28] demonstrates the existence of a bound). Consider $\widehat{\widetilde{\varphi}}(\xi) = \frac{12\sin^2(\xi/2)}{\xi^2(2+\cos(\xi))} = h_a(a\xi)$, and let $z = x+iy$. Note first that

$$\begin{aligned}|2+\cos(x+iy)| &= \tfrac{1}{2}\left|4 + e^{-y}(\cos(x)+i\sin(x)) + e^{y}(\cos(x)-i\sin(x))\right|\\ &= \left(\sinh^2(y)\sin^2(x) + (\cosh(y)\cos(x)+2)^2\right)^{1/2}.\end{aligned}$$

For $|y|\le 1/2$, $\cosh(y)\le 3/2$, from which $(\cosh(y)\cos(x)+2)^2 \ge 1/4$, and $|2+\cos(x+iy)|\ge 1/2$, uniformly in x. Similarly, for $|y|\le 1$,

$$\left|\frac{\sin\left(\frac{x+iy}{2}\right)}{x+iy}\right|^2 = \frac{\sinh^2\left(\frac{y}{2}\right)\cos^2\left(\frac{x}{2}\right) + \cosh^2\left(\frac{y}{2}\right)\sin^2\left(\frac{x}{2}\right)}{y^2+x^2} \le 1,$$

uniformly in x. Hence, $\forall|y|\le 1/2$, $|\widehat{\widetilde{\varphi}}(x+iy)|\le 24$, so for $|y|\le a/2$, $|\widehat{\widetilde{\varphi}}((x+iy)/a)|\le 24$, $\forall x\in\mathbb{R}$. Thus, $\phi_X\cdot h_{a,n}\in\mathcal{H}(\mathcal{D}_\gamma)$ where $\gamma = \gamma(a) = d\wedge a/2$, and $C_\gamma(\phi_X) := \|\phi_X\cdot h_{a,n}\|^{\mathcal{H}_\gamma} \le 24\|\phi_X\|^{\mathcal{H}_\gamma}$. For a sufficiently large, the integrand is bounded within $\mathcal{D}_d$ (for any finite $d>0$). Moreover, since $\bar{P} > 1+\log_2|\bar{\mu}|$, it holds that $\bar{a}/2 > |\bar{\mu}|$ and so $|x_n|\le|\bar{\mu}|+\bar{a}/2 < \bar{a}$, $\forall 1\le n\le N$. Thus by Theorem 3.2.1 in [37], $\epsilon_{trap}(a,\bar{a})$ converges exponentially in $\bar{a}$, according to the bound given.

The truncation error depends on the tail behavior of ϕ_X. Since $|h_{a,n}(\xi)|\le\frac{12a^2}{\xi^2}$, and $|\phi_X(\xi)|$ satisfies equation (7), the truncation error is bounded by

$$\Delta_\xi\sum_{j=N}^{\infty}\bar{\nu}_j\phi_X(\xi_j)h_{a,n}(\xi_j) \le 12\kappa a^2\int_{2\pi a}^\infty\frac{e^{-tc|\xi|^\nu}}{\xi^2}d\xi \le 12\kappa a^2 e^{-tc|2\pi a|^\nu}\int_{2\pi a}^\infty\frac{1}{\xi^2}d\xi,$$

and the result follows after simplifying. ∎

[1] H. Albrecher, P. Mayer, and W. Schoutens. General lower bounds for arithmetic asian option prices. *App. Math. Fin.*, 15(2):123–149, 2008.

[2] H. Albrecher and M. Predota. Bounds and approximations for discrete Asian options in a variance-gamma model. *Grazer Math. Ber.*, 345:35–57, 2002.

[3] B. Alziary, J.P. Dechamps, and P.F. Koehl. Pde approach to asian options: analytical numerical evidence. *J. Banking and Fin.*, 21:613–640, 1997.

[4] J. Andreasen. The pricing of discretely sampled Asian and lookback options: a change of numeraire approach. *J. Comput. Fin.*, 1(3):15–36, 1998.

[5] O. Barndorff-Nielsen. Processes of normal inverse Gaussian type. *Finance and Stochastics*, 2(1):41–68, 1997.

[6] E. Benhamou. Fast Fourier transform for discrete Asian options. *J. Comput. Fin.*, 6(1):49–61, 2002.

[7] S. Boyarchenko and S. Levendorskii. *Non-Gaussian Merton-Black-Scholes Theory*. Adv. Ser. Stat. Sci. Appl. Probab. World Scientific Publishing Co., River Edge, NJ,, volume 9 edition, 2002.

[8] S. Boyarchenko and S. Levendorskii. Efficient versions of the Fourier transform in applications to option pricing. *J. Comp. Fin.*, 18(2), 2014.

[9] S. Boyarchenko and S. Levendorskii. Option pricing for truncated Levy processes. *International J. of Theoretical and Applied Finance*, 3(3):549–552, July 2000.

[10] P. Carr, H. Geman, D. B. Madan, and M. Yor. The fine structure of asset returns: an empirical investigation. *J. Business*, 75:305–332, 2002.

[11] A. Carverhill and L. Clewlow. Flexible convolution: Valuing average rate (Asian) options. *Risk*, 3(4):25–29, 1990.

[12] A. Cerny and I. Kyriakou. An improved convolution algorithm for discretely sampled Asian options. *Quant. Fin.*, 11:381–389, 2011.

[13] O. Christensen. *An Introduction to Frames and Riesz Bases*. Birkhauser Boston, 2003.

[14] T. Dai and Y. Lyuu. Accurate and efficient lattice algorithms for American-style Asian options with range bounds. *Appl. Math. Comput.*, 209:238–253, 2009.

[15] M. De Innocentis and S. Levendorskii. Pricing discrete barrier options and credit default swaps under Levy processes. *Quant. Fin.*, 14(8):1337–1365, 2014.

[16] P. Den Iseger and E. Oldenkamp. Pricing guaranteed return rate products and discretely sampled Asian options. *J. Comput. Fin.*, 9(3):1–39, 2006.

[17] J. Dewynne and P. Wilmott. Asian options as linear complementary problems. *Advances in Futures and Options Research*, 8:145–173, 1995.

[18] E. Eberlein and A. Papapantoleon. Equivalence of floating and fixed strike Asian and lookback options. *Stochastic Processes and their Applications*, 115(1):31–40, 2005.

[19] F. Fang and C.W. Oosterlee. A novel pricing method for European options based on Fourier cosine series expansions. *SIAM J. Sci. Comput.*, 31:826–848, 2008.

[20] L. Feng and X. Lin. Inverting analytic characteristic functions with financial applications. *SIAM J. Finan. Math.*, 4(1):372–398, 2013.

[21] L. Feng and V. Linetsky. Pricing discretely monitored barrier options and defaultable bonds in Levy process models: a fast Hilbert transform approach. *Math. Finan.*, 18(3):337–384, July 2008.

[22] G. Fusai, D. Marazzina, and M. Marena. Pricing discretely monitored Asian options by maturity randomization. *SIAM J. Finan. Math.*, 2:383–403, 2011.

[23] G. Fusai and A. Meucci. Pricing discretely monitored Asian options under Levy processes. *J. Banking Fin.*, 32(1):2076–2088, 2008.

[24] H. Geman and D.B. Madan. *Risks in returns: a pure jump perspective*. In Exotic Option Pricing and Advanced Levy Models. Wiley, 2005.

[25] C. Heil. *A Basis Theory Primer*. Birkhauser, expanded edition edition, 2011.

[26] N. Ju. Pricing asian and basket options via taylor expansion. *J. Comput. Fin.*, 5(3):79–103, 2002.

[27] A. Kemna and A. Vorst. A pricing method for options based on average asset values. *J. of Banking and Fin.*, 14:113–129, 1990.

[28] J.L. Kirkby. Efficient option pricing by frame duality with the fast Fourier transform. *SIAM J. Finan. Math.*, 6(1):713–747, 2015.

[29] J.L. Kirkby. Robust option pricing with characteristic functions and the B-spline order of density projection. J. Comput. Fin., Forthcoming, 2016.

[30] J.L. Kirkby and S. Deng. Static hedging and pricing of exotic options with payoff frames.

Available at SSRN: http://ssrn.com/abstract=2501812, 2015.

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

[32] S.Z. Levendorskii and J. Xie. Pricing discretely sampled Asian options under Levy processes. Available at SSRN: http://papers.ssrn.com/abstract=2088214, 2012.

[33] D. Madan and E. Seneta. The variance gamma (v.g.) model for share market returns. *J. Business*, 63:511–524, 1990.

[34] J. Nielsen and K. Sandmann. Pricing bounds on Asian options. *J. Fin. and Quant. Anal.*, 38(2), 2003.

[35] P. Sabino. Monte Carlo methods and path-generation techniques for pricing multi-asset path-dependent options. SSRN Working Paper, 2007.

[36] K-I. Sato. *Levy Processes and Infinitely Divisible Distributions*. Cambridge University Press, Cambridge, UK, 1999.

[37] F. Stenger. *Numerical Methods based on Sinc and Analytic functions*. Springer-Verlag, New York, 1993.

[38] M. Unser and I. Daubechies. On the approximation power of convolution-based least squares versus interpolation. *IEEE Transactions on Signal Processing*, 45(7):1697–1711, July 1997.

[39] Michael Unser. Vanishing moments and the approximation power of wavelet expansions. In *Proceedings of the 1996 IEEE international conference on image processing*, 1996.

[40] J. Vecer. A new pde approach for pricing arithmetic average Asian options. *J. Comput. Fin.*, 4(4):105–115, 2001.

[41] B. Zhang and C.W. Oosterlee. Efficient pricing of European-style Asian options under exponential Levy processes based on Fourier cosine expansions. *SIAM J. Finan. Math.*, 4:399–426, 2013.

[42] J.E. Zhang. Pricing continuously sampled Asian options with perturbation method. *J. Futures Markets*, 23(6):535–560, 2003.