A Novel Fourier Transform B-spline Method for Option Pricing

Gareth G. Haslip[a], Vladimir K. Kaishev[a,*]

[a]*Cass Business School, City University London.*

Abstract

We present a new efficient and robust framework for European option pricing under continuous-time asset models from the family of exponential semimartingale processes. We introduce B-spline interpolation theory to derivative pricing to provide an accurate closed-form representation of the option price under an inverse Fourier transform.

We compare our method with some state-of-the-art option pricing methods, and demonstrate that it is extremely fast and accurate. This suggests a wide range of applications, including the use of more realistic asset models in high frequency trading. Examples considered in the paper include option pricing under asset models, including stochastic volatility and jumps, computation of the Greeks, and the inverse problem of cross-sectional calibration.

Keywords: Continuous-time semimartingale models, option pricing, stochastic volatility, Fourier transform, closed-form solutions, B-splines, Peano formula, divided differences
AMS classification: 65C20, 41A15, 65R10, 65R32

*Corresponding author contact information: Cass Business School, City University London, 106 Bunhill Row, London EC1Y 8TZ, United Kingdom. Tel.: +44 (0)20 7040 8453 Fax.: +44 (0)20 7040 8572

Email addresses: gareth@haslip.co.uk (Gareth G. Haslip), v.kaishev@city.ac.uk (Vladimir K. Kaishev)

August 7, 2013

1. Introduction

Application of the Fourier transform for option pricing was pioneered by, among others, Stein and Stein (1991), and Heston (1993), who viewed the integral pricing formula as a "closed-form solution". Bakshi and Madan (2000) formalized the economic foundations for Fourier transform based pricing and the interpretation of the characteristic function spanning the pay-off universe of all derivative instruments. Carr and Madan (1999) found that, by performing the Fourier transform of the European option with respect to the strike price, the FFT can be used to perform the inversion. Following Carr and Madan (1999), the Fractional Fast Fourier Transform (FRFT) method was developed by Chourdakis (2005). The FRFT addressed a short-coming of the FFT by providing greater resolution of option prices within the range of strike prices being considered. Chourdakis (2005) demonstrated that the method could deliver prices up to 45 times faster than the FFT without substantial loss of precision. Subsequently, three new numerical methods for pricing options have emerged: (i) Integration-Along-Cut (IAC) method, described in Boyarchenko and Levendorskiĭ (2002) and refined in Levendorskiĭ and Xie (2012), (ii) the Cosine (COS) method of Fang and Oosterlee (2008) that utilizes a cosine expansion and Fourier transform inversion, providing fast and accurate prices across many strike prices simultaneously, and (iii) the Convolution (CONV) method of Lord et al. (2009) that builds upon the Quadrature (QUAD) method of Andricopoulos et al. (2003) and the enhanced QUAD-FFT method of O'Sullivan (2005), and provides efficient pricing across multiple strike prices. A few variants of the FFT have also been developed, for example, the generalized FFT of Boyarchenko and Levendorskiĭ (2002) and the inverse FFT (iFFT) of Boyarchenko and Levendorskiĭ (2008).

Alongside these developments, progress has also been made in approximation techniques for option pricing. Recently, Kristensen and Mele (2011) have provided a general approximation framework based on Taylor series expansions of the difference between the actual option price under the selected asset model and that under an auxiliary pricing model for which a closed-form solution is available. Carr and Madan (2010) provide a closed-form saddle-point approximation for a range of asset models and demonstrate its effectiveness for pricing deep out-of-the-money options for which the FFT method can be less effective. Indeed, as noted by Boyarchenko and Levendorskiĭ (2011), it should be stressed that in the context of calibration, for each maturity date, the number of strike prices is limited to about two to three dozens, and therefore it is often quicker to invert the Fourier transform directly by numerical integration than to apply the FFT method.

This paper continues the pursuit of real-time option pricing and develops a practical and efficient approximation framework to address the valuation of European options under a general continuous-time asset model from the class of exponential semimartingale processes. This class is rich and encompasses the majority of models utilized in finance for pricing derivatives. Examples include (i) the general families of (jump) diffusion processes such as the Black and Scholes (1973) model, the Merton (1976) model and the mixed exponential model of Cai and Kou (2011), (ii) pure jump Lévy processes such as the general class of linear combinations of Gamma (LG) processes, recently introduced by Kaishev (2013), which includes as special cases, the variance gamma (VG) process introduced by Madan and Seneta (1990) and the bilateral gamma process considered by Küchler and Tappe (2008), the KoBoL model of Boyarchenko and Levendorskiĭ (2000) (a special case of which is known as the CGMY processes of Carr et al. (2002)), and the generalized hyperbolic models (see Eberlein (2001)), and (iii) affine processes, characterized by Duffie et al. (2003), which include the Heston (1993) model and the Bates (1996) model. For properties of some of the particular examples of semimartin-

gale models see Eberlein et al. (2008).

We will refer to our option pricing framework as the Fourier transform B-spline method (FTBS). There are two key parts of the method. First, use of a Fourier transform based pricing integral. Second, use of B-splines, which are very flexible piecewise polynomial functions, to approximate integrand functions in the Fourier transform pricing integral. For the latter we use the Lewis-Lipton representation, provided in Lewis (2001) (see also Lewis (2000)) and Lipton (2002). This method utilizes the Fourier transform to provide an option valuation formula as a contour integral in the complex plane and can be considered as a generalization of the approach of Carr and Madan (1999). Formulas for option pricing in Lévy models, in the form of inverse Fourier transform integrals appeared already in Eq. (30) and (31) of Boyarchenko and Levendorskiĭ (1998) (see also Boyarchenko and Levendorskiĭ (2000)). Several authors have utilized the Fourier transform for option pricing. For example, Boyarchenko and Levendorskiĭ (2002) develop a numerical pricing method using a generalized FFT, and Quittard-Pinon and Randrianarivony (2010) apply this to price European options. In their more recent work, Boyarchenko and Levendorskiĭ (2011) develop new efficient implementations of the Fourier transform method: (i) based on the truncation of the infinite trapezoid rule and using a novel application of summation by parts, (ii) a new family of pricing methods entitled the payoff modification with Fourier transform methods (PMwFT), and (iii) conformal parabolic and hyperbolic iFT methods, which utilize a conformal map, and yield an integral with a much better rate of convergence. Option pricing under particular models, namely the Heston model and VG process, are considered in Levendorskiĭ (2012) and Innocentis (2011) respectively. The above methods are promising additions to the option pricing literature. We note that method (iii) above is particularly effective for accurately pricing deep out-of-the-money options, which we are not considering in our paper.

Bouziane (2008) utilizes the Fourier transform approach to the problem of pricing interest rate derivatives. In common with Carr and Madan (1999), and Chourdakis (2005), these methods introduce truncation error and sampling error as described by Lee (2004).

Generalizations of the Fourier pricing framework has recently been considered by Dufresne et al. (2009) and by Eberlein et al. (2010). The latter authors have shown that it is valid for the more general class of exponential semimartingale models of the underlying asset price dynamics.

In what follows, we utilize the extension of the Fourier transform pricing method provided by Eberlein et al. (2010), and assume the exponential semimartingale model for the asset price evolution. Our method is based on three key ideas. First, we represent the integrand of the Fourier transform pricing integral as a product of (i) a trigonometric function dependent on the option strike price, and (ii) the semimartingale process characteristic function multiplied by a Fourier transform of the option payoff with unit strike price, which is independent of the actual strike price. Second, we interpolate the strike price independent function of part (ii) by a linear combination of B-splines of a fixed (low) order. In this way, we express the Fourier transform pricing integral as a sum of integrals of a B-spline multiplied by the trigonometric function of part (i). Third, we interpret these integrals as Peano representations of divided differences of appropriate trigonometric functions. As a result, we obtain an explicit, closed-form expression for the option price in the form of a linear combination of low order divided differences of trigonometric functions, as shown in Theorem 2 of Section 3.3.3. The coefficients in this linear combination are obtained from the spline interpolation of the strike price independent function of part (ii). Consequently, the only strike price dependent part of our option pricing expression are the divided differences. Therefore, the FTBS method can be used for efficient evaluation of

option prices not just for one fixed strike price but simultaneously for a whole set of different strike prices in a given range. Although the divided differences calculation is dependent on the strike price, it does not depend on the choice of asset model and hence they can be pre-computed.

The option price evaluation under our method is very fast and accurate since the semimartingale process characteristic function of part (ii) above is interpolated and, hence, computed at a relatively small number of interpolation sites, and the knots of the interpolating spline are optimally located, as described in Section 3. To our best knowledge, optimal spline interpolation theory has not been used in the option pricing literature before. In summary, for any choice of an exponential semimartingale model for the underlying asset price process, the option price calculation using our method becomes a simple procedure of (i) fitting a spline function expressed as a linear combination of B-splines to a simple variant of the characteristic function, and (ii) computing a linear combination of the B-spline coefficients and the pre-computed divided difference factors for the choice of strike price and interest rate. As demonstrated in Section 4, this process is extremely quick and allows us to compute the option price with any required accuracy.

For example, under the VG process, we can calculate European option prices across 31 different strike prices accurate to five significant figures in under 20 microseconds, which implies a computation time of under one microsecond per option. These computation times are achieved using a modest computational environment, as described in Section 4, and do not make use of parallel processing techniques. Our method therefore has a wide range of applications in finance, ranging from pricing, marking, and hedging to calibration and high-frequency trading.

There are several key advantages of our method over existing approximation methods for option pricing. First, truncation error is eliminated by transforming the Fourier transform pricing integral to the unit interval. Second, our method is able to compute accurate option prices with lower sampling error than other methods when using only a small number of interpolation sites. This is evidenced by our numerical examples, see Figure 1 of Section 4. Finally, the divided difference calculation is model independent and can therefore be pre-computed and stored in a data file for future use. This enables the FTBS method to compute asset prices extremely quickly at a higher level of precision than is possible using alternative approximation methods, for most of the asset price models that we have investigated, see Figure 1 of Section 4, and Tables F.6, F.7, and F.8 of Appendix F. We note that we are unaware of any similar existing approximation methods in the option pricing literature. This is confirmed by the results of the numerical comparison of our FTBS method with five other state-of-the-art option pricing methods, namely the FFT, FRFT, IAC, COS, and CONV methods, which as we show are very competitive. In order to make this comparison computationally fair, we have diligently implemented all the six option pricing methods to the best of our ability in C++ using identical hardware.

In this paper, we illustrate the FTBS method for computing the price and sensitivities of European options but note that the method is more general and can be applied without adjustment to any contingent claim without early exercise or path dependent features. Additionally, the method can be adapted for pricing exotic options, as demonstrated in Haslip and Kaishev (2013) for the problem of pricing discrete lookback options.

This paper is organized as follows. The next section reviews the Fourier transform pricing framework that underlies our FTBS method. In Section 3, we introduce the application of B-spline interpolation theory to option pricing and show how the Peano representation of a

divided difference with respect to a B-spline kernel is applied to provide a closed-form formula for pricing and computing the sensitivities of European options across all strike prices. Section 4 assesses the numerical performance of the FTBS method across a wide range of different asset price models and provides detailed comparisons to the existing methods listed above. In Section 5, we consider the cross-sectional calibration of the Heston model and the stochastic volatility jump diffusion model of Bates (1996) to the implied volatility surface, and demonstrate the effectiveness of the FTBS method relative to the FFT and FRFT methods. Section 6 concludes the paper, and the appendices provide tables and some detailed proofs that are omitted from the main text.

2. Pricing options using the Fourier transform

In this section, we introduce the notation and assumptions of the continuous-time asset price model and provide a short recap of the Fourier transform pricing framework for the valuation of European options.

2.1. Notation and model assumptions

Let the stochastic process $X = (X_t)_{t\geq 0}$ with $X_0 = 0$ be defined on a continuous-time probability space $(\Omega, \mathcal{F}, \mathbf{Q})$ with a standard complete filtration $\{\mathcal{F}_t, t \geq 0\}$. We assume that X is a semimartingale with respect to the filtration $\{\mathcal{F}_t, t \geq 0\}$, so that it satisfies $X = M + A$ where $M = (M_t)_{t\geq 0}$ is a local martingale with $M_0 = 0$ and $A = (A_t)_{t\geq 0}$ is a bounded variation process with $A_0 = 0$. Let $S_t \geq 0$ denote the price at time $t > 0$ of an asset whose price dynamics follows an exponential semimartingale process given by

$$S_t = S_0 e^{(r-q)t+X_t}, \tag{1}$$

where r is the risk-free interest rate, and q is the continuous income yield provided by the asset. In Eq. (1), the probability measure $\mathbf{Q}$ is assumed to be the risk-neutral measure under the assumption of no arbitrage, where $e^{-rt}S_t$ is a martingale. For further properties of semimartingales we refer to Jacod and Shiryaev (2003).

2.2. The Fourier transform

The FTBS method is built on the foundations of the Fourier transform. We now introduce this Fourier pricing method and develop a pricing formula for European options. In Section 3, the pricing formula is approximated using an optimal spline interpolation, which provides fast convergence to the true price.

The Fourier transform method provides a direct integration formula for option prices under a range of payoffs, independent of the choice of an asset model.

Proposition 1 (European call option pricing formula). *If $M_{X_T}(v)$ exists for all $v \in (1, \alpha)$ with $\alpha > 1$, then the European call option price, $C(T, K)$, is given by*

$$C(T,K) = -\frac{Ke^{-rT}}{2\pi} \int_{iv-\infty}^{iv+\infty} e^{-izk}\phi_{X_T}(-z)\frac{dz}{z^2 - iz}, \tag{2}$$

where $M_{X_T}(v) = \int_{-\infty}^{\infty} e^{vx} f_{X_T}(x)dx$, and

$$k = \log(\frac{S_0}{K}) + (r - q)T. \tag{3}$$

PROOF. The proof is straightforward and can be found e.g. in Eberlein et al. (2010) (see Example 5.1 therein). □

The pricing formula in Eq. (2) is an interesting direct integration formula for the European call option, but in this form it exhibits a number of difficulties. First, the choice of contour integration parameter, v, is important and has a significant impact on the behavior of the integrand and, hence, the convergence properties of the formula. Lord and Kahl (2007) investigate the choice of v in depth and provide optimal choice for particular asset models when pricing European options. For a more recent investigation of the optimal choice of parameter of contour integration see Section 2.7 of Boyarchenko and Levendorskiĭ (2011). Second, the integration must be performed over an infinite interval, and care must be taken to avoid truncation error.

In our presentation of the FTBS method, we prefer to utilize an alternative form of Eq. (2) provided by Lewis (2001) and Lipton (2002).

Proposition 2 (Alternative European call option pricing formula). *If $M_{X_T}(v)$ exists for all $v \in (\alpha, \beta)$ with $\alpha < \frac{1}{2}$ and $\beta > 1$, then the European call option price, $C(T, K)$, is given by*

$$C(T,K) = S_0 e^{-qT} - \frac{\sqrt{S_0 K}\, e^{\frac{-(r+q)T}{2}}}{\pi} \int_0^\infty \mathrm{Re}\left[e^{iuk} \phi_{X_T}\left(u - \frac{i}{2} \right) \right] \frac{du}{u^2 + \frac{1}{4}}, \tag{4}$$

where k is given by Eq. (3).

Remark 1. *The pricing formula in Eq. (4) can be considered a compromise (for the purpose of ease of use) to Eq. (2), where instead of identifying the optimal choice of the contour parameter for a particular asset price model, it is fixed at a level that generally performs well across a range of models. In developing the FTBS method we have considered different contour parameters, and we note that the method is applicable for choices different than $v = 0.5$. However, for simplicity of the presentation, and ease of use of the method in deriving Theorem 2 of Section 3.3.3, we have chosen to use the above alternative form with $v = 0.5$, which works reasonably well for a wide choice of models and parameters.*

We refer the reader to Theorem 4 in Appendix C, which is a version of Theorem 2 that is applicable in the general case $v > 1$, and note that this can easily be extended to the case $v < 0$. We note that the simplified pricing formula Eq. (4) still exhibits the difficulty with the truncation error of the upper limit of integration, which we address in the next section.

3. The Fourier Transform B-spline pricing method

In this section, we provide a full derivation of the FTBS method in the context of pricing European call options on an underlying asset. In addition, we consider the problem of calculating the Greeks and apply the FTBS method to compute the sensitivities of European call options. However, as noted in the introduction, the method is more general and can be applied to a wider range of options by modifying the payoff function in the pricing formula.

3.1. Simplification of the pricing formula

We now present an alternative representation of the pricing formula in Eq. (4) in which the integrand is decomposed into the product of (i) a trigonometric function dependent on the option's strike price, and (ii) the real and imaginary parts of the semimartingale process characteristic function multiplied by the Fourier transform of the option payoff with unit strike, which is independent of the actual strike price.

Theorem 1 (Strike-separable pricing formula). *If $M_{X_T}(v)$ exists for all $v \in (\alpha, \beta)$ with $\alpha < \frac{1}{2}$ and $\beta > 1$, then the European call option price $C(T,K)$ is given by*

$$C(T,K) \;=\; S_0 e^{-qT} - \frac{1}{\pi}\sqrt{S_0 K}\, e^{\frac{-(r+q)T}{2}} I(k), \tag{5}$$

where

$$I(k) \;=\; \int_0^1 \cos\left(\tfrac{1-t}{t}k\right) s_1(t)dt + \int_0^1 \sin\left(\tfrac{1-t}{t}k\right) s_2(t)dt, \tag{6}$$

$$s_1(t) \;=\; \frac{\text{Re}\left[\phi_{X_T}(\frac{1-t}{t} - \frac{i}{2})\right]}{1 - 2t + \frac{5}{4}t^2}, \quad s_2(t) = -\frac{\text{Im}\left[\phi_{X_T}(\frac{1-t}{t} - \frac{i}{2})\right]}{1 - 2t + \frac{5}{4}t^2}, \tag{7}$$

and k is defined in Eq. (3).

PROOF. Using the Euler identity, one can expand e^{iuk} in Eq. (4) as $\cos uk + i \sin uk$. We then express $\phi_{X_t}(u - \frac{i}{2})$ in terms of its real and imaginary parts. Multiplying together and discarding imaginary terms yields the following simplification for the integral in Eq. (4)

$$\int_0^\infty \frac{\text{Re}\left[\phi_{X_t}(u - \frac{i}{2})\right]\cos uk - \text{Im}\left[\phi_{X_t}(u - \frac{i}{2})\right]\sin uk}{u^2 + \frac{1}{4}} du,$$

which we denote by $I(k)$. We then make the change of variable $u = \frac{1-t}{t}$. Noting that the limits $u = 0$ and $u = \infty$ correspond to $t = 1$ and $t = 0$, respectively, and $du = -\frac{1}{t^2}dt$, the integral becomes

$$I(k) = \int_0^1 \frac{\text{Re}\left[\phi_{X_t}(\frac{1-t}{t} - \frac{i}{2})\right]\cos\left(\frac{1-t}{t}k\right) - \text{Im}\left[\phi_{X_t}(\frac{1-t}{t} - \frac{i}{2})\right]\sin\left(\frac{1-t}{t}k\right)}{t^2\left[\left(\frac{1-t}{t}\right)^2 + \frac{1}{4}\right]} dt.$$

Finally, by simplifying the denominator and defining $s_1(t)$ and $s_2(t)$ as above, the result follows. □

Remark 2. *We make two important observations about strike-separable pricing formula in Eq.* (5). *First, integration is now performed over the unit interval. Specifically, we have applied the change of variables $u = \frac{1-t}{t}$ which transforms the upper limit of integration from infinity to zero, and the lower limits of integration from zero to one. Thus, integration is performed over the unit interval $[0, 1]$. In this way we eliminate the truncation error which comes from the necessity to truncate the limit at infinity in the original expression of Eq.* (4). *This means that it is no longer necessary to carefully identify a truncation point for each asset price model by considering how quickly the characteristic function decays to zero. Second, we have separated the integrand into the product of $\cos\left(\frac{1-t}{t}k\right)$ or $\sin\left(\frac{1-t}{t}k\right)$, which are dependent on the strike price, and functions $s_1(t)$ and $s_2(t)$, which are independent of the strike price. This innovation is one of the foundations of the FTBS method since it allows us to price options at different strike prices in an extremely efficient manner.*

The above remark highlights a significant improvement to many of the existing numerical methods of option pricing based on the Fourier transform, most of which require careful examination of the characteristic function to minimize the truncation error in the resulting option price.

3.2. The Greeks

Pricing is one key aspect of the field of derivatives, another equally important aspect is hedging, which is crucial for effective risk management. We now consider the problem of computing the Greeks and demonstrate how Eq. (5) is easily extended to calculate the option price sensitivities. We note that here we only consider the Greek sensitivities that are model independent and therefore do not depend on the form of the characteristic function. For example, Vega is specific to the Black-Scholes model and does not exist in the same form for a general exponential semimartingale process.

Corollary 1 (The Greeks). *The sensitivities of the European call option price, $C(T,K)$, to movements in the asset price, interest rate, and the passage of time are provided by $\Delta C(T,K) = \frac{\partial C(T,K)}{\partial S_0}$ and $\Gamma C(T,K) = \frac{\partial^2 C(T,K)}{\partial S_0^2}$ with respect to the asset price, $PC(T,K) = \frac{\partial C(T,K)}{\partial r}$ with respect to the interest rate, and $\Theta C(T,K) = \frac{\partial C(T,K)}{\partial T}$ with respect to the tenor. If $M_{X_T}(v)$ exists for all $v \in (\alpha,\beta)$, with $\alpha < \frac{1}{2}$ and $\beta > 1$, then they can be computed as*

$$\Delta C(T,K) = e^{-qT} - \frac{1}{\pi}\sqrt{\tfrac{K}{S_0}}e^{\frac{-(r+q)T}{2}}\left[\tfrac{1}{2}I(k) + I'(k)\right] \tag{8}$$

$$\Gamma C(T,K) = \frac{1}{S_0\pi}\sqrt{\tfrac{K}{S_0}}e^{\frac{-(r+q)T}{2}}\left[\tfrac{1}{4}I(k) - I''(k)\right] \tag{9}$$

$$PC(T,K) = \frac{1}{\pi}\sqrt{S_0K}e^{\frac{-(r+q)T}{2}}\left[\tfrac{1}{2}I(k) - TI'(k)\right] \tag{10}$$

$$\Theta C(T,K) = -qS_0e^{-qT} + \frac{1}{\pi}\sqrt{S_0K}e^{\frac{-(r+q)T}{2}}\left[\tfrac{T}{2}I(k) - (r-q)I'(k)\right], \tag{11}$$

where $I'(k)$ and $I''(k)$ are the derivatives of the function $I(k)$, defined in Eq. (6). These are computed as

$$I'(k) = \int_0^1 \cos\left(\tfrac{1-t}{t}k\right)\Delta s_1(t)dt + \int_0^1 \sin\left(\tfrac{1-t}{t}k\right)\Delta s_2(t)dt \tag{12}$$

$$I''(k) = \int_0^1 \cos\left(\tfrac{1-t}{t}k\right)\Gamma s_1(t)dt + \int_0^1 \sin\left(\tfrac{1-t}{t}k\right)\Gamma s_2(t)dt, \tag{13}$$

where

$$\Delta s_1(t) = \frac{1-t}{t}s_2(t), \quad \text{and} \quad \Delta s_2(t) = -\frac{1-t}{t}s_1(t), \tag{14}$$

and

$$\Gamma s_1(t) = \frac{1-t}{t}\Delta s_2(t) = -\frac{(1-t)^2}{t^2}s_1(t),$$
$$\Gamma s_2(t) = -\frac{1-t}{t}\Delta s_1(t) = -\frac{(1-t)^2}{t^2}s_2(t).$$

PROOF. The proof of the formulae for the Greeks is provided by careful application of the chain rule when differentiating. We sketch the proof briefly for the case of Δ and Γ and note

that similar logic applies in the case of P or Θ. We begin with the proof for $\Delta C(T,K)$. From Eq. (5), we have

$$\begin{aligned}\Delta C(T,K) &= \frac{\partial C}{\partial S_0} = e^{-qT} - \tfrac{1}{2\pi}(S_0K)^{-\frac{1}{2}}Ke^{\frac{-(r+q)T}{2}}I(k) \\ &- \tfrac{1}{\pi}\sqrt{S_0K}e^{\frac{-(r+q)T}{2}}I'(k)\frac{\partial k}{\partial S_0}. \end{aligned} \tag{15}$$

Noting that $k = \log(\frac{S_0}{K}) + (r-q)T$, we obtain $\frac{\partial k}{\partial S_0} = \frac{1}{S_0}$. Simplifying Eq. (15) yields the required result.

To obtain $\Gamma C(T,K)$, one simply differentiates $C(T,K)$ a second time, that is, $\Gamma C(T,K) = \frac{\partial \Delta C}{\partial S_0}$, and again makes careful use of the chain rule. This yields the expression in Eq. (9). Finally, to obtain the expressions for $I'(k)$ and $I''(k)$, we differentiate Eq. (6) under the integral sign with respect to k,

$$\begin{aligned} I'(k) = \frac{d}{dk}I(k) &= \int_0^1 \frac{d}{dk}\cos\left(\tfrac{1-t}{t}k\right)s_1(t)dt + \int_0^1 \frac{d}{dk}\sin\left(\tfrac{1-t}{t}k\right)s_2(t)dt \\ &= -\int_0^1 \sin\left(\tfrac{1-t}{t}k\right)\tfrac{1-t}{t}s_1(t)dt + \int_0^1 \cos\left(\tfrac{1-t}{t}k\right)\tfrac{1-t}{t}s_2(t)dt. \end{aligned}$$

Defining $\Delta s_1(t)$ and $\Delta s_2(t)$ as in Eq. (14) provides the stated formula for $I'(k)$. The calculation of $I''(k)$ follows precisely the same logic. □

We have developed simple integration formulae for pricing options and computing their sensitivities. These form the foundations of the FTBS method and we now proceed to develop closed-form approximations to compute them. In the next section we apply spline approximation theory to interpolate functions $s_1(t)$, $s_2(t)$, $\Delta s_1(t)$, $\Delta s_2(t)$, $\Gamma s_1(t)$, and $\Gamma s_2(t)$.

3.3. Developing the approximation formula

The standard approach in option pricing for the evaluation of Eq. (5) is to apply efficient numerical integration techniques, such as the Gauss-Kronrod quadrature method, to calculate the integral to the desired level of accuracy. In this section, we introduce a general approach for evaluating Eq. (5) that utilizes spline approximation theory. We have chosen to apply spline functions in developing our option pricing method, because they allow for the accurate approximation of possibly complex and oscillatory functions. For example, see Figure 7 of Kaishev et al. (2006), where B-spline basis functions are used to approximate the Doppler function, which is wildly oscillatory, especially near the origin. There are two key ideas underlying the spline approximation component of the FTBS method. First, we interpolate the functions $s_1(t)$ and $s_2(t)$ by two linear combinations of B-splines of a fixed low order.

In this way we express each of the two integrals in the expression for $I(k)$ in Eq. (6) as a linear combination of integrals of a B-spline multiplied by $\cos(\frac{1-t}{t}k)$ and $\sin(\frac{1-t}{t}k)$, respectively. Second, we interpret these integrals as Peano representations of divided differences of appropriate trigonometric functions and obtain an explicit, closed-form expression for the option price. The latter is in the form of a linear combination of low order divided differences of trigonometric functions and is provided in Theorem 2 of Section 3.3.3. The coefficients in this linear combination are obtained from the spline interpolation of the strike price independent functions $s_1(t)$ and $s_2(t)$.

Our method is very fast and accurate for two main reasons. First, since we extract the oscillatory components, $\cos\left(\frac{1-t}{t}k\right)$ and $\sin\left(\frac{1-t}{t}k\right)$, from the integrand in Eq. (4), the remaining functions, $s_1(t)$ and $s_2(t)$, are smoother and better behaved. Hence, the approximation of these functions requires only a relatively small number of interpolation sites at which they are computed and interpolated. The second reason is that, given fixed interpolation sites, the knots of the interpolating spline are optimally located, following Gaffney and Powell (1976) and Micchelli et al. (1976). Therefore, the optimal interpolating spline provides the best possible bound for the spline interpolation error. Moreover, it can be shown (see De Boor (2001), Chapter XII) that as the number of interpolation sites and knots increases, the error bound decays at the rate $O\left(|t|^n\right)$, where $|t|:=\max_i\left(t_i-t_{i-1}\right)$ is the mesh size of the sequence of knots. In other words, the spline approximation quickly converges to the integrand function as the number of interpolation sites, and hence knots, increases. Therefore, the FTBS method provides a very accurate approximation of the option price, and requires far fewer evaluations of the approximated integrand function than using direct numerical integration. In the following sections, we provide a brief introduction to B-splines and divided differences and formalize the ideas we have described in this section to establish our main option pricing result presented in Theorem 2 of Section 3.3.3.

3.3.1. Splines, B-splines, and divided differences

In order to derive our main result given in Theorem 2, we require some background material on splines which we briefly introduce. Let n and l be positive integers and consider an interval $[a,b]\in\mathbf{R}$, partitioned by the points $a=t_n<\ldots<t_{n+l}<t_{n+l+1}=b$ called knots. We define the spline function, $s(t)$ of order n, degree $n-1$, as a piece-wise polynomial function which coincides with a polynomial of degree $n-1$ between the knots $t_n,\ldots,t_{n+l+1}$. The polynomial pieces are smoothly joined at the knots $t_{n+1},\ldots,t_{n+l}$ so that the spline is $n-2$ times continuously differentiable on the interval $[a,b]$. To introduce the set of B-spline basis functions, we add $n-1$ additional knots at each end of the interval $[a,b]$. Thus, we define the extended set of knots $\{t_i\}_{i=1}^{2n+l}$ so that $t_1=\cdots=t_n=a$, $a<t_{n+1}<\cdots<t_{n+1}<b$, and $t_{n+l+1}=\cdots=t_{2n+l}=b$.

As known by the Curry-Schoenburg theorem, polynomial splines on $\{t_i\}_{i=1}^{2n+l}$ form a linear space of functions, an element, $s(t)$, of which is represented as

$$s(t)=\sum_{i=1}^{p}c_iM_{i,n}(t), \tag{16}$$

where c_i are constant coefficients, $p=l+n$, and $M_{i,n}(t)$ are the B-spline basis functions. The latter are defined on $\{t_i\}_{i=1}^{2n+l}$ as the n-th order divided difference of the function $f(y)=n\left(\max\left\{(y-t),0\right\}\right)^{n-1}=n(y-t)_+^{n-1}$, that is,

$$M_{i,n}(t)=M_{i,n}(t;t_i,\ldots,t_{i+n})=[t_i,\ldots,t_{i+n}]f(y).$$

The n-th order $(n\geq 0)$ divided difference of a function $f(t)$ is defined recurrently as

$$[t_i,\ldots,t_{i+n}]\,f(t)=\frac{[t_{i+1},\ldots,t_{i+n}]\,f(t)-[t_i,\ldots,t_{i+n-1}]\,f(t)}{t_{i+n}-t_i}, \tag{17}$$

where $[t_i]\,f(t)=f\left(t_i\right)$ and the points $\{t_j\}_{j=i}^{i+n}$ are pairwise distinct. In the case when one or more points are repeated, a derivative based formula provided in Eq. $(B.1)$ of $Appendix\ B.1$, which appears in Ignatov and Kaishev (1989), can be used to calculate the divided difference.

The B-spline basis functions, $M_{i,n}(t)$, $i = 1, \ldots, p$, have some nice properties, see e.g. De Boor (2001), that will be very useful in deriving our pricing formula Eq. (22). In particular, we need the elegant Peano representation of the n-th order divided difference of a function $f(x)$ given as

$$[t_i, \ldots, t_{i+n}]\, f(x) = \int_{\mathbf{R}} M_{i,n}(t; t_i, \ldots, t_{i+n}) \frac{f^{(n)}(t)}{n!} dt, \tag{18}$$

where $f^{(n)}$ is the n-th derivative of f. For a proof of Eq. (18) see, for example, De Boor (2001), Chapter IX.

Note that the B-spline $M_{i,n}(t)$ appears as the kernel in the Peano representation given in Eq. (18). This is important since in the derivation of the pricing formula in Eq. (22), we need to evaluate integrals of the form $\int_{\mathbf{R}} M_{i,n}(t; t_i, \ldots, t_{i+n})\psi(t)dt$, with $\psi(t) = \cos\left(\frac{1-t}{t}k\right)$ or $\psi(t) = \sin\left(\frac{1-t}{t}k\right)$. We shall see that it is possible to express such integrals in the form of Eq. (18) and thus eliminate integration; replacing it with the evaluation of a simple low order divided difference of a function $\tilde{\psi}(t)$, whose n-th derivative coincides with $\psi(t)$. Therefore, the Peano representation plays a fundamental role in obtaining our main result, the option pricing formula given by Eq. (22) in Theorem 2.

3.3.2. Optimal B-spline interpolation

In this section, we use the spline function, $s(t)$, defined in Eq. (16), to interpolate the integrand functions, $s_1(t)$ and $s_2(t)$. from the strike-separable pricing formula, given in Theorem 1. Since these functions are defined on the interval $[0, 1]$, we assume that $[a, b] \equiv [0, 1]$ in the definition of the knot set $\{t_i\}_{i=1}^{2n+l}$. To avoid complicating the notation, we illustrate this step only for the function $s_1(t)$. The same reasoning and similar notation applies for $s_2(t)$. Denote the set of ν interpolation sites, which are alternatively referred to as data sites, as $\boldsymbol{\tau} = \{\tau_1, \ldots, \tau_\nu\}$, with $\tau_1 = 0$, $\tau_\nu = 1$, and $\tau_i < \tau_{i+1}$, $i = 1, \ldots, \nu$. The optimal spline interpolation problem can now be stated as follows. For the fixed set of appropriately located data sites $\boldsymbol{\tau}$, find the interpolating function $s(t)$, and the function $C(t)$ that satisfy the inequality

$$|s_1(t) - s(t)| \leq C(t)\|s_1^{(n)}(t)\|, \tag{19}$$

where $\|f\| = \max\{\, |f(t)| \colon a \leq t \leq b\}$ and $C(t)$ is as small as possible for all $t \in [0, 1]$. In order for the bound in Eq. (19) to be valid, we also assume that the functions $s_1(t)$ and $s_2(t)$ are continuous and have bounded n-th derivatives. We note that these assumptions hold for the implemented semimartingale asset price models, as illustrated in Section 4. Furthermore, the interpolation method developed here works even if some of these assumptions are not satisfied, in which case it may not be possible to assess the error of approximation using the bound in Eq. (19).

Gaffney and Powell (1976) and Micchelli et al. (1976) independently solved the optimal interpolation problem in Eq. (19). It turns out that the optimal interpolant, $\tilde{s}(t)$, is a spline function of order n with exactly $l = \nu - n$ optimally located internal knots, $\{\tilde{t}_i\}_{i=n+1}^{n+l}$, and linear coefficients $\tilde{c}_i$ for $i = 1, \ldots, l + n$. We refer to Gaffney (1978) for details of how to find these optimal knots and coefficients. As noted by De Boor (2001), choosing the l internal knots to be the averages of the data sites $\boldsymbol{\tau}$, that is,

$$t_{n+i} = \frac{(\tau_{i+1} + \ldots + \tau_{i+n-1})}{n-1}, i = 1, \ldots, l, \tag{20}$$

provides a very good approximation to the optimal knot set, $\{\tilde{t}_i\}_{i=n+1}^{n+l}$. We have tested the theoretically optimal knot set, implementing the subroutine SPLOPT of De Boor (2001) and

have not found any significant improvement. We therefore recommend the averaging method in Eq. (20) as the efficient knot selection procedure.

Another important parameter of the interpolation process is the degree $n-1$ of the spline. Popular choices are quadratic, $n=3$, or cubic, $n=4$. We considered both choices and found that numerically the cubic spline does not offer additional precision over the quadratic spline, while increasing the computational complexity. There is also some evidence in approximation theory literature, see, for example, Marsden (1974), suggesting that quadratic splines often provide better fits than cubic splines. We therefore have chosen to work with quadratic splines, and in the rest of the paper we therefore use $n=3$.

In summary, for given data sites $\boldsymbol{\tau}$, the choice of which is discussed in Section 3.4.1, we follow the optimal interpolation scheme described above. We obtain quadratic spline approximants, $\tilde{s}_1(t)$ and $\tilde{s}_2(t)$ to $s_1(t)$ and $s_2(t)$, that are in the form of Eq. (16). The approximants $\tilde{s}_1(t)$ and $\tilde{s}_2(t)$ are defined by the sets of knots $\{t_{1,i}\}_{i=1}^{6+l_1}$, $\{t_{2,j}\}_{j=1}^{6+l_2}$ and coefficients $c_{1,i}$, $c_{2,j}$ for $i=1,\ldots,p_1$, $j=1,\ldots,p_2$, where $p_1=l_1+3$ and $p_2=l_2+3$. We note that for the computation of the Greeks, additional splines are used to interpolate the functions $\Delta s_1(t)$, $\Delta s_2(t)$, $\Gamma s_1(t)$, and $\Gamma s_2(t)$.

3.3.3. The Fourier transform B-spline pricing formula

In this section, we utilize the optimal B-spline interpolation scheme presented above to develop the main result of this paper, the FTBS pricing formula. We first return to the strike-separable pricing formula in Eq. (5) of Theorem 1 and define $\tilde{C}(T,K)$ to be the approximation to Eq. (5) by replacing $s_1(t)$ and $s_2(t)$ by their spline approximants $\tilde{s_1}(t)$ and $\tilde{s_2}(t)$ respectively. That is,

$$C(T,K) \approx \tilde{C}(T,K) = S_0 e^{-qT} - \frac{\sqrt{S_0 K} e^{\frac{-(r+q)T}{2}}}{\pi} \tilde{I}(k), \tag{21}$$

where

$$\tilde{I}(k) = \int_0^1 \cos\left(\frac{1-t}{t}k\right) \tilde{s_1}(t)dt + \int_0^1 \sin\left(\frac{1-t}{t}k\right) \tilde{s_2}(t)dt.$$

We note that in what follows we assume that $M_{X_T}(v)$ exists for all $v \in (\alpha, \beta)$ with $\alpha < \frac{1}{2}$ and $\beta > 1$. To express Eq. (21) in a closed-form and eliminate integration, it suffices to (i) substitute $\tilde{s}_1(t)$ and $\tilde{s}_2(t)$ which are splines of the form given by Eq. (16), (ii) take summation in front of integration, and (iii) apply the Peano formula in Eq. (18) to each integral in the linear combination. The latter integrals are of the form $\int_{\mathbf{R}} M_{i,n}(t; t_i, \ldots, t_{i+n})\psi(t)dt$, with $\psi(t) = \cos\left(\frac{1-t}{t}k\right)$ or $\psi(t) = \sin\left(\frac{1-t}{t}k\right)$, which are in the required form to apply the Peano representation. Therefore, as described in Section 3.3.1, one needs to find the function $\tilde{\psi}(t)$ whose n-th derivative coincides with $\psi(t)$. Following this approach, we give our main result stated by the following theorem, whose proof is given in *Appendix B.2*.

Theorem 2 (The Fourier Transform B-spline pricing formula).
Let $\{t_{1,i}\}_{i=1}^{6+l_1}$, $\{t_{2,j}\}_{j=1}^{6+l_2}$ *and* $\{c_{1,i}\}_{i=1}^{p_1}, \{c_{2,j}\}_{j=1}^{p_2}$ *be the sets of knots and linear coefficients of the quadratic spline interpolants,* $\tilde{s}_1(t)$ *and* $\tilde{s}_2(t)$*, respectively, where* $p_1 = l_1 + 3$ *and* $p_2 = l_2 + 3$*. Additionally, let* $k = \log\left(\frac{S_0}{K}\right) + (r-q)T$*. The pricing formula of a European call option is given as*

$$C(T,K) \approx S_0 e^{-qT} - \frac{1}{\pi}\sqrt{S_0 K} e^{\frac{-(r+q)T}{2}} \tilde{I}(k), \tag{22}$$

where for $k \neq 0$

$$\begin{aligned}\tilde{I}(k) = 6\sum_{i=1}^{p_1} c_{1,i}[t_{1,i}, t_{1,i+1}, t_{1,i+2}, t_{1,i+3}] f_1(t,k) \\ + 6\sum_{i=1}^{p_2} c_{2,i}[t_{2,i}, t_{2,i+1}, t_{2,i+2}, t_{2,i+3}] f_2(t,k),\end{aligned} \tag{23}$$

and f_1 *and* f_2 *are defined as*

$$\begin{aligned}f_1(t,k) = \frac{1}{12}\Bigg\{ t\left(-(k^2-2t^2)\cos\frac{k(t-1)}{t} - 5kt\sin\frac{k(t-1)}{t}\right) \\ + k\,\mathrm{Ci}\left(\frac{k}{t}\right)\left(-6kt\cos k + (k^2-6t^2)\sin k\right) \\ - k\,\mathrm{Si}\left(\frac{k}{t}\right)\left(\ \ 6kt\sin k + (k^2-6t^2)\cos k\right)\Bigg\}\end{aligned}$$

$$\begin{aligned}f_2(t,k) = \frac{1}{12}\Bigg\{ t\left(-(k^2-2t^2)\cos\frac{k(t-1)}{t} - 5kt\sin\frac{k(t-1)}{t}\right) \\ + k\,\mathrm{Ci}\left(\frac{k}{t}\right)\left(\ \ 6kt\sin k + (k^2-6t^2)\cos k\right) \\ + k\,\mathrm{Si}\left(\frac{k}{t}\right)\left(-6kt\cos k + (k^2-6t^2)\sin k\right)\Bigg\}.\end{aligned}$$

For $k = 0$*,* $\tilde{I}(k)$ *simplifies to*

$$\tilde{I}(k) = \sum_{i=1}^{p_1} c_{1,i} + \sum_{i=1}^{p_2} c_{2,i}. \tag{24}$$

Note that $\mathrm{Ci}(x)$ *and* $\mathrm{Si}(x)$ *are the trigonometric special functions defined as* $\mathrm{Ci}(x) = -\int_x^\infty \frac{\cos y}{y} dy$ *and* $\mathrm{Si}(x) = \int_0^x \frac{\sin y}{y} dy$*, respectively.*

PROOF. Provided in *Appendix* $B.2$ □

While the functions f_1 and f_2 in Theorem 2 may appear complicated, they can be evaluated efficiently. Their trigonometric integrals can be computed using the Fortran library of Vandevender and Haskell (1982), which provides 17 significant figures of accuracy using series expansions with pre-computed coefficients. Calculation of the divided difference is simple using the standard recursive definition in Eq. (17) when the knots $[t_i, t_{i+1}, t_{i+2}, t_{i+3}]$ are distinct. In the case of repeated knots, the divided differences must be calculated using the derivative based definition in Eq. $(B.1)$ of *Appendix* $B.1$.

Remark 3. *We refer the reader to Theorem 4 in Appendix C, which is a version of Theorem* 2 *that is applicable in the general case* $v > 1$*, and note that this can easily be extended to the case* $v < 0$*. It is worth noting that the more general Theorem 4 does not lead to any additional computational complexity.*

The following proposition gives a bound for the absolute error of the FTBS European option price approximation. It shows that the FTBS method provides a fast convergence to the true price as the numbers, ν and l, of data sites, $\boldsymbol{\tau}$ and knots, $\{\tilde{\eta}_{j,i}\}_{i=3+1}^{3+l}$ increase, while the mesh sizes $|\tilde{\eta}_j| := \max_i (\tilde{\eta}_{j,i} - \tilde{\eta}_{j,i-1})$ go to zero.

Proposition 3 (FTBS Option Price Error Bound). *The absolute error of the FTBS European option price,* $\tilde{C}(T,K)$ *is bounded by*

$$\left|C(T,K) - \tilde{C}(T,K)\right| \leq \frac{\sqrt{S_0 K}}{\pi} e^{\frac{(r+q)T}{2}} \left(\tilde{C}_1 \left\|s_1^{(3)}(t)\right\| + \tilde{C}_2 \left\|s_2^{(3)}(t)\right\|\right), \tag{25}$$

where $\tilde{C}_1 = C_1 \int_0^1 \left|\cos\left(\frac{1-t}{t}k\right)\right| dt$, $\tilde{C}_2 = C_2 \int_0^1 \left|\sin\left(\frac{1-t}{t}k\right)\right| dt$*,* C_j*,* $j = 1, 2$ *are the constants obtained from the optimal spline interpolation,* $\tilde{s}_j(t)$ *of* $s_j(t)$*,* $j = 1, 2$*, following Gaffney and Powell (1976), and where* $\|f\| = \max\{|f(t)| : 0 \leq t \leq 1\}$*. The bound in* (25) *converges to zero as the mesh sizes* $|\tilde{\eta}_j| := \max_i (\tilde{\eta}_{j,i} - \tilde{\eta}_{j,i-1})$ *go to zero, at a rate* $O(|\tilde{\eta}|^3)$*, where* $\tilde{\eta} = \max(|\tilde{\eta}_1|, |\tilde{\eta}_2|)$*, i.e.,*

$$\left|C(T,K) - \tilde{C}(T,K)\right| = O\left(|\tilde{\eta}|^3\right) \tag{26}$$

PROOF. Provided in $Appendix\ B.3$. □

Calculation of the Greeks under the FTBS method follows the same logic as Theorem 2. It uses additional B-splines to approximate functions $\Delta s_1(t)$, $\Delta s_2(t)$, $\Gamma s_1(t)$, and $\Gamma s_2(t)$. We denote the respective approximants as $\widetilde{\Delta s}_1(t)$, $\widetilde{\Delta s}_2(t)$, $\widetilde{\Gamma s}_1(t)$, and $\widetilde{\Gamma s}_2(t)$. We can then approximate the integrals $I'(k)$ and $I''(k)$ from Eqs. (12) and (13) in closed-form using the Peano representation in Eq. (23).

Proposition 4 (FTBS formulae for the Greeks).
Let $\{t_{1,i}\}_{i=1}^{6+l_1}$*,* $\{t_{2,j}\}_{j=1}^{6+l_2}$ *and* $\{c_{1,i}\}_{i=1}^{p_1}$*,* $\{c_{2,j}\}_{j=1}^{p_2}$ *be the sets of knots and linear coefficients of the quadratic spline interpolants,* $\tilde{s}_1(t)$ *and* $\tilde{s}_2(t)$*, respectively, where* $p_1 = l_1 + 3$ *and* $p_2 = l_2 + 3$*.*

Similarly, let the knots and linear coefficients of the quadratic spline interpolants $\widetilde{\Delta s}_1(t)$*,* $\widetilde{\Delta s}_2(t)$*, and* $\widetilde{\Gamma s}_1(t)$*,* $\widetilde{\Gamma s}_2(t)$ *be denoted by* $\{t^*_{1,i}\}_{i=1}^{6+l^*_1}$*,* $\{t^*_{2,j}\}_{j=1}^{6+l^*_2}$ *and* $\{c^*_{1,i}\}_{i=1}^{p^*_1}$*,* $\{c^*_{2,j}\}_{j=1}^{p^*_2}$*, where* $p^*_1 = l^*_1 + 3$ *and* $p^*_2 = l^*_2 + 3$ *for* $* \in \{\Delta, \Gamma\}$*.*

The closed-form approximations for the Greeks are obtained by replacing functions $I(k)$*,* $I'(k)$*, and* $I''(k)$ *in Corollary* 1 *by their approximants* $\tilde{I}(k)$*,* $\tilde{I}'(k)$*, and* $\tilde{I}''(k)$*.* $\tilde{I}(k)$ *is defined in Theorem* 2*, and* $\tilde{I}'(k)$ *and* $\tilde{I}''(k)$ *are defined as*

$$\begin{aligned}\tilde{I}'(k) = 6 \sum_{i=1}^{p_1^\Delta} c_{1,i}^\Delta [t_{1,i}^\Delta, t_{1,i+1}^\Delta, t_{1,i+2}^\Delta, t_{1,i+3}^\Delta] f_1(t,k) \\ + 6 \sum_{i=1}^{p_2^\Delta} c_{2,i}^\Delta [t_{2,i}^\Delta, t_{2,i+1}^\Delta, t_{2,i+2}^\Delta, t_{2,i+3}^\Delta] f_2(t,k)\end{aligned}$$

$$\tilde{I}''(k) = 6\sum_{i=1}^{p_1^\Gamma} c_{1,i}^\Gamma[t_{1,i}^\Gamma, t_{1,i+1}^\Gamma, t_{1,i+2}^\Gamma, t_{1,i+3}^\Gamma]f_1(t,k) + 6\sum_{i=1}^{p_2^\Gamma} c_{2,i}^\Gamma[t_{2,i}^\Gamma, t_{2,i+1}^\Gamma, t_{2,i+2}^\Gamma, t_{2,i+3}^\Gamma]f_2(t,k),$$

for $k \neq 0$. If $k = 0$, this simplifies to Eq. $(C.6)$ in Theorem 2.

PROOF. The proof follows the same reasoning as for Theorem 2 and is therefore omitted. □

3.4. Implementation of the FTBS method

The FTBS method is straightforward to apply and, unlike other numerical methods for option pricing, its implementation is largely independent of the underlying continuous-time asset model. The key considerations in the implementation of the FTBS method are (i) the choice of data sites, and (ii) the pre-computation of the divided differences $[t_{l,i}, t_{l,i+1}, t_{l,i+2}, t_{l,i+3}]f_l(t,k)$ for $l = 1, 2$ in Theorem 2.

It should be noted that the spline approximants, $\tilde{s}_1(t)$ and $\tilde{s}_2(t)$, are independent of the strike price K. Since the divided differences can be pre-computed, as we will explain in Section 3.4.2, the approximants' strike price independence implies that the efficiency of the FTBS method for pricing many options simultaneously across a quantum of strike prices is similar to that of pricing a single option. This is analogous to the use of the FFT of Carr and Madan (1999) for evaluating the price of European options for a range of strike prices.

3.4.1. Optimal selection of interpolation data sites

Somewhat surprisingly, the question of selecting data sites in spline interpolation has not received significant attention in the literature on approximation theory, although some considerations can be found in De Boor (2001). The selection of data sites depends on the smoothness properties of the underlying function that is to be interpolated. In areas where the function is less smooth and exhibits higher curvature, more data sites should be allocated to capture its behavior. Intuitively, the most efficient selection of data sites should extract the most information from the function with minimum number of observations. Therefore, the allocation of data sites across the unit interval should provide a greater weight to regions where the function exhibits greater curvature. We identify the data sites to sample the semimartingale process characteristic function by analyzing the curvature of functions $s_1(t)$ and $s_2(t)$ across a range of models and parameter choices. Our selection is based on the criteria of a compromise set $\boldsymbol{\tau}$ that performs reasonably well across a large range of parameters, as opposed to being optimized for any particular model. We allocate the data sites in uniform bands: 60% to $[0, 0.2)$, 20% to $[0.2, 0.6)$ and the remainder to $[0.6, 1]$. The allocation could be improved by developing optimal data sites allocations for specific semimartingale processes and parameter ranges. We also examine the use of uniformly spaced data sites in the unit interval. Although uniform spacing of data sites produces accurate option prices, it is less efficient, since more data sites are allocated to regions of low curvature of the functions $s_1(t)$ and $s_2(t)$ than is necessary for accurate interpolation.

3.4.2. Pre-computation of divided differences

In computing the divided difference $[t_{l,i}, t_{l,i+1}, t_{l,i+2}, t_{l,i+3}]f_l(t,k)$ for $l = 1, 2$, there are two cases to consider: (i) the knots $t_{l,i}, t_{l,i+1}, t_{l,i+2}, t_{l,i+3}$ are all distinct, and (ii) some knots are repeated once or more. In the former case, we can apply the recursive definition, given in Eq.

(17), directly, which is simple to implement computationally. In the case of repeated knots, the derivative definition, given in Eq. $(B.1)$ of $Appendix\ B.1$, of a divided difference should be applied.

In our case, we have to evaluate the divided differences at the following repeated knots: $[0,0,0,t_{l,4}]f_l$, $[0,0,t_{l,4},t_{l,5}]f_l$, $[t_{l,k+3},t_{l,k+4},1,1]f_l$, and $[t_{l,k+4},1,1,1]f_l$ for $l=1,2$. Although algebraically involved, these are easily explicitly expressed with the aid of symbolic algebraic software, such as Mathematica, as shown in $Appendix\ B.4$. Note that, since the divided differences at repeated knots are evaluated just a single time at the endpoints, their complicated form does not have an impact on the time it takes to compute the option price.

Returning to the FTBS pricing formula in Theorem 2, we make an important observation. The divided differences of functions $f_1(t,k)$ and $f_2(t,k)$ across the knots $\{t_{1,i}\}_{i=1}^{6+l_1}$, $\{t_{2,i}\}_{i=1}^{6+l_2}$ depend only on the choice of knots and k. They are independent of the choice of the exponential semimartingale process. Therefore, for a fixed set of data sites and corresponding optimal knots, we can pre-compute the divided differences for different values of k and do not need to calculate this each time an option is priced for a different value of k. Furthermore, if we use the same data sites and knots for the Greeks, then the pre-computed divided differences can be applied for both pricing and computing the option price sensitivities. In all our numerical examples provided in Section 4, we pre-compute the divided differences using the allocation of data sites described in Section $3.4.1$. We find that this choice of data sites provides very accurate results across the wide range of models considered.

The pre-computation of divided differences is a significant advantage of the FTBS method, since pricing options becomes a simple procedure of fitting the functions $s_1(t)$ and $s_2(t)$ using B-spline interpolation, and then computing the sum in Eq. (23) of products of the B-spline coefficients and the pre-computed divided differences. The divided differences can be pre-computed in two ways, depending on the option pricing application. First, for the purposes of calibration, one would need to compute the divided differences for a fixed set of values of k. This set is determined by the relevant strike prices and maturity times, the interest rate, and the dividend rate, at the calibration date (see Eq. (3)). Therefore, at the beginning of the calibration process, the divided differences can be pre-computed exactly for all required values of k. This is a very fast process that takes approximately one to two milliseconds, and is negligible relative to the overall calibration time (see Table 1 of Section 5, where calibration times are measured in the order of seconds). Second, in the case of real-time applications, such as high frequency trading, it may be necessary to compute option prices for values of k that are rapidly changing. For example, as time elapses, the share price will move, and maturities will shorten. In this case, it is necessary to pre-compute the divided differences at a fine resolution of possible values of k.

We now describe an algorithm to implement the pre-computation of the divided differences in this second case. Let k_{min} and k_{max} denote the minimum and maximum moneyness of the options being considered, where we define the moneyness of an option as $k=\log\left(\frac{S_0}{K}\right)+(r-q)T$. For example, if we consider the range for $\log(\frac{S_0}{k})$ to be $[-1,2]$, $r-q$ to be $[0\%,10\%]$, and T to be $[0,10]$, then the possible range for k could be assumed to be $[k_{\text{min}},k_{\text{max}}]$, where $k_{\text{min}}=-1$ and $k_{\text{max}}=3$. Note that this covers the full market of typically traded call options. For example, if $S_0=100$, then strike prices could range from $K=15$ to 270 over any choice of $r-q\in[0\%,10\%]$ and $T\in[0,10]$. We denote the divided differences as

$$DvD_i^1(k)=[t_{1,i},t_{1,i+1},t_{1,i+2},t_{1,i+3}\,]f_1(t,k)$$
$$DvD_j^2(k)=[t_{2,i},t_{2,j+1},t_{2,j+2},t_{2,j+3}]f_2(t,k),$$

where $i = 1, \ldots, p_1$ and $j = 1, \ldots, p_2$.

To approximate $DvD_i^1(k)$ and $DvD_j^2(k)$ for a given $k \in [k_{\min}, k_{\max}]$, we divide the interval $[k_{\min}, k_{\max}]$ uniformly using N evenly spaced points with width $\delta = \frac{k_{\max}-k_{\min}}{N-1}$. The expressions $DvD_i^1(k_m)$ and $DvD_j^2(k_m)$ containing the divided differences should therefore be pre-computed for $i = 1, \ldots, p_1$, $j = 1, \ldots, p_2$, and $m = 1, \ldots, N$, where $k_m = k_{\min} + (m-1)\delta$.

Thanks to modern computing power, the values of $DvD_i^1(k_m)$ and $DvD_j^2(k_m)$ across permissible values of i, j, and m can comfortably be held in RAM at a very fine resolution. This reduces the need for interpolation in the case where $k_m < k < k_{m+1}$ for some $m \in 1, \ldots, N$. For example, if N is chosen such that the spacing $\delta = k_m - k_{m-1}$ between neighboring k_m is $\delta =$ 1E-5, then, for $p_1 = p_2 = p = 100$ knots, $k_{\min} = -1$ and $k_{\max} = 3$, the total memory required to store all values of the pre-computed divided differences is under one gigabyte. This is well within the capability of standard computing systems. In such an implementation, it is reasonable to choose the nearest k_m to the specified k, hence, avoiding the need for interpolation.

Since the B-spline interpolation of $s_1(t)$ and $s_2(t)$ can be performed very quickly and with good precision, the FTBS method is able to compute precise option prices in a very efficient manner. Moreover, once the divided differences have been pre-computed and the B-splines have been fitted, computing the price of options for different strike prices requires only calculating $S_0 e^{-qT} - \frac{1}{\pi}\sqrt{S_0 K} e^{\frac{-(r+q)}{2}}$ and performing $p_1 + p_2$ multiplication and addition operations, as can be seen from Eqs. (22) and (23). Since typically only 50 to 100 data sites are required, this computation is extremely fast.

If a coarser partition of $[k_{\min}, k_{\max}]$ is employed to approximate k, then simple linear interpolation can be applied to compute the required vector of divided difference between the neighboring values of k_j. This adds only a slight overhead to the calculation, requiring an extra p multiplication, p division, and $4p$ addition operations, as is evident from examining the formula for linear interpolation.

Finally, we note that in the implementation, the divided differences should be stored in contiguous memory in a simple array, since the appropriate memory offset for specified indices i, j, and m can be computed directly. This ensures that the divided difference array for valuing an option can be accessed instantly.

3.4.3. Recommended configuration and algorithm implementing the FTBS method

In our implementation of the FTBS method we have used identical sets of data sites and knots for the quadratic spline interpolants, $\tilde{s}_1(t)$ and $\tilde{s}_2(t)$ to minimize the number of evaluations of the asset price process characteristic function. In Algorithm 1 and Algorithm 2, we provide a simple implementation of the FTBS method, which is applied in Section 4 for all the numerical examples. In order to provide practical recommendations for the configuration of the FTBS method, in Table $F.3$ of Appendix F we provide prescriptions for choosing the number of data sites required to achieve a desired level of precision, for all the asset price models considered in the paper.

4. Numerical evaluation of the FTBS method

As described in Section 1, a variety of different methods for pricing European options have been developed in recent years. In order to assess the effectiveness of the FTBS method, we compare it to the key state-of-the-art methods in the literature, to both verify its accuracy in pricing European options, and provide a guide to the relative speeds of the different methods.

Algorithm 1 Pre-computation of divided differences (using identical sets of knots for interpolation of functions $s_1(t)$ and $s_2(t)$)

Input:

- Number of knots $\nu + 3$.
- Array specifying knots $\{t_i\}_{i=1}^{\nu+3}$ (note that $t_1 = t_2 = t_3 = 0$, $t_{\nu+1} = t_{\nu+2} = t_{\nu+3} = 1$).
- Array $\{k_j\}_{j=1}^{N_K}$ corresponding to selected strike prices $\{K_j\}_{j=1}^{N_K}$ defined by $k_j = \log\left(\frac{S_0}{K_j}\right) + (r-q)T$.

Output:

$[t_i, t_{i+1}, t_{i+2}, t_{i+3}]f_l(t, k_j)$, for $i = 1, \ldots, p$, $j = 1, \ldots, N_K$, and $l = 1, 2$.

for $j = 1, \ldots, N_K$ **do**
  Compute $[0, 0, 0, t_4]f_l(t, k_j)$, and $[0, 0, t_4, t_5]\, f_l(t, k_j)$, for $l = 1, 2$ using (1) and (2) in Appendix B4.
  for $i = 3, \ldots, \nu - 2$ **do**
    Compute $[t_i, t_{i+1}, t_{i+2}, t_{i+3}]f_l(t, k_j)$ for $l = 1, 2$ using the recursive definition (19).
  end for
  Compute $[t_{v-1}, t_v, 1, 1]f_l(t, k_j)$, and $[t_v, 1, 1, 1]f_l(t, k_j)$, for $l = 1, 2$, using the derivative based formula provided in Eq. (B.1) of Appendix B.1.
end for

We have diligently implemented each pricing method to the best of our ability in C++, using identical hardware, and therefore the computation times are all directly comparable. The details of the implementation of each pricing method is outlined in Appendix D.

We examine the precision of the FTBS method for a variety of different exponential semi-martingale processes. The numerical examples consider the following asset models: (i) the VG process of Madan et al. (1998), (ii) the Heston stochastic volatility model of Heston (1993), (iii) the KoBoL model of Boyarchenko and Levendorskiĭ (2000) (a special case of which is known as the CGMY processes of Carr et al. (2002)), (iv) the double exponential jump diffusion (DEJD) model of Kou (2002), and (v) the mixed exponential jump diffusion (MEJD) model of Cai and Kou (2011). Comparing the effectiveness of the FTBS method with such a variety of models assists in providing evidence for the robustness of our approach.

To judge the numerical efficiency of the FTBS method, we compare it to (i) the Fast Fourier Transform (FFT) of Carr and Madan (1999), (ii) the Fractional Fast Fourier Transform (FRFT) of Chourdakis (2005), (iii) Integration-Along-Cut (IAC) method of Levendorskiĭ and Xie (2012), (iv) the Cosine (COS) method of Fang and Oosterlee (2008), and (v) the Convolution (CONV) method of Lord et al. (2009).

Our main numerical comparison considers the computation time, for a specified level of precision, of pricing European options across multiple strikes prices. This is our main case of interest, since this is necessary for the efficient and accurate risk-neutral calibration of asset models to the observed implied volatility surface. This comparison is carried out across the VG process, Heston model, and KoBoL model. As a second comparison, we consider the efficiency of the FTBS method for pricing an European option at a single strike price.

Finally, in Appendix E, we provide some additional numerical comparisons that consider the precision of the FTBS method for computing the Greeks, and also for pricing options under the mixed exponential- and double exponential jump diffusion models.

Algorithm 2 Computation of European option prices using the FTBS method (using identical sets of knots for interpolation of functions $s_1(t)$ and $s_2(t)$)

Input:

- Number of data sites ν.
- Array specifying data sites $\boldsymbol{\tau} = \{\tau_1, \ldots, \tau_\nu\}$ with $\tau_1 = 0$, $\tau_\nu = 1$, and $\tau_i \in (0, 1)$ for $j = 2, \ldots, \nu - 1$, allocated uniformly, according to the empirically justified rule: 60% in $[0, 0.2)$, 20% in $[0.2, 0.6)$ and 20% in $[0.6, 1)$.
- The constants S_0, r, q, T, $\{K_j\}_{j=1}^{N_K}$ strikes prices.

Output: $\{C(T, K_i)\}_{j=1}^{N_K}$

1. Compute knots $\{t_i\}_{i=1}^{\nu+3}$, using the approximation in Eq. (22) for the interior knots. Therefore, $t_1 = t_2 = t_3 = 0$, $t_{3+j} = \frac{\tau_{i+j} + \tau_{i+j}}{2}$ for $j = 1, \ldots, v - 3$, and $t_{\nu+1} = t_{\nu+2} = t_{\nu+3} = 1$. Note that in our computational implementation, the same knots will be utilized to interpolate functions $s_1(t)$ and $s_2(t)$, that is we take $p = p_1 = p_2 (= \nu)$ and $t_i = t_{1,i} = t_{2,i}$, for $i = 1, \ldots, \nu + 3$.
2. Define $k_j = \log\left(\frac{S_0}{K_j}\right) + (r - q)T$, $j = 1, \ldots, N_K$. Retrieve pre-computed divided differences $[t_i, t_{i+1}, t_{i+2}, t_{i+3}]f_1(t, k_j)$, and $[t_i, t_{i+1}, t_{i+2}, t_{i+3}]f_2(t, k_j)$, for $i = 1, \ldots, p$ and $j = 1, \ldots, N_K$, from in memory array.
3. Compute quadratic spline approximants, $\tilde{s}_1(t)$ and $\tilde{s}_2(t)$ to $s_1(t)$ and $s_2(t)$ using function SPLINT of De Boor (2001). This routine will return the linear coefficients $\{c_{1,i}\}_{i=1}^{p}$ and $\{c_{2,i}\}_{i=1}^{p}$ (see Section 2.3).
4. Compute option prices using Theorem 2. That is, first compute $\tilde{I}(k_j)$, using formula (25), and then $C(T, K_j)$ using formula (24), $j = 1, \ldots, N_K$ (see section 2.3.3).

In all cases, the data sites for the B-spline approximations are allocated as described in Section 3.4.1. Our calculations utilize the exact pre-computation of divided differences as described in Section 3.4.2. Calculation times are reported in milliseconds, and the number of data sites for the FTBS method is denoted by N for consistency with other methods.

All our numerical implementations are in C++ using a standard laptop computer with an Intel Core I7-3610QM processor.

4.1. Numerical comparison for pricing options at multiple strike prices

We compare the FTBS method to the COS method, the FFT, and FRFT methods. We consider pricing European options under three different asset price models, the VG process, the Heston model, and the KoBoL (CGMY) model.

Following Chourdakis (2005), we consider three sets of parameters for each asset price model: "low", "bench", and "high". See Tables F.6, F.7, and F.8 of Appendix F for the parameter values. This is an important aspect of testing the effectiveness of different numerical methods for option pricing, since for applications such as calibration (and even for calculation of prices in a given model), it is vital that the pricing method provides accurate prices across the entire parameter space of the asset price model. For example, as demonstrated in Levendorskiĭ (2012), even a numerical scheme which calculates prices with the relative error of order of 2E-5 when used for calibration may produce the parameters rather different from the true ones, and prices of barrier options calculated using these parameters may differ by 30% from the true prices.

In the case of the VG process, and Heston model, these parameter sets are taken directly from Chourdakis (2005), and in the case of the KoBoL (CGMY) model, we have utilized parameters that represent low, medium, and high levels of volatility (based on densities a, b, and e, from Figure 1 of Carr et al. (2002)) and have used the same maturities as Chourdakis (2005).

In Figure 1, we provide a graphical comparison of the FTBS method to the above pricing methods, and across the above asset price models. Specifically, we compare the calculation time to the level of precision for pricing European options across 31 strike prices in a single computation, where the axes are in logarithmic scale. The precision level is defined as the maximum absolute error across the 31 computed option prices. Note that for the multiple strike comparison, we computed the reference option price values using integration in Mathematica to compute the inverse Fourier integral with very high numerical precision, and verified the correctness of the reference prices by applying each option pricing method at its highest level of accuracy (by setting N to a large integer).

The computation times and precision levels underlying the graphs in Figure 1 are provided in Tables F.6, F.7, and F.8, of Appendix F, where we provide several measures of precision. Namely, we give (i) the absolute error, which is the maximum absolute error, (ii) mean error, which is the average absolute error, and (iii) RMSE, which is the root mean square error.

The numerical examples provide evidence that for the problem of pricing European options across multiple strike prices, the FTBS method achieves excellent efficiency, that is, a combination of accuracy and speed, across the different exponential semimartingale models and numerical pricing methods considered. In particular, for the VG process, the FTBS method dominates the other three option pricing methods at all precision levels considered from 1E-4 to 1E-7[1].

For the Heston model, the FTBS method dominates all other comparison pricing methods except for the COS method. The FTBS method is preferable to the COS method when pricing options under the Heston model for precision levels below approximately 1E-6 precision. This can be seen visually in Figure 1, where the lines for FTBS (Heston) and COS (Heston) cross-over around 1E-6 precision for the low and high parameters sets (and around 1E-8 for the bench parameter set). For higher levels of precision the COS method is preferable.

Finally, for the KoBoL (CGMY) model, the FTBS method is preferable to the comparison pricing methods, except in the case of the high parameter set, for precision levels higher than around 3E-7, where the COS method is preferable. Therefore, for the KoBoL (CGMY) model, we conclude that the FTBS method dominates all comparison pricing methods at precision levels 1E-4 to 1E-6.

Hence, based on our numerical study, we can conclude that the FTBS method is the preferable approximation method, relative to the other methods considered, for computing option prices across multiple strikes prices for the VG process, and for the Heston model and KoBoL (CGMY) model, for precision up to the level of 1E-6. The COS method has advantages when greater precision is required for the latter models.

[1] Under the VG process, the COS method could not provide results at precision level 1E-8, so this was excluded from the comparison

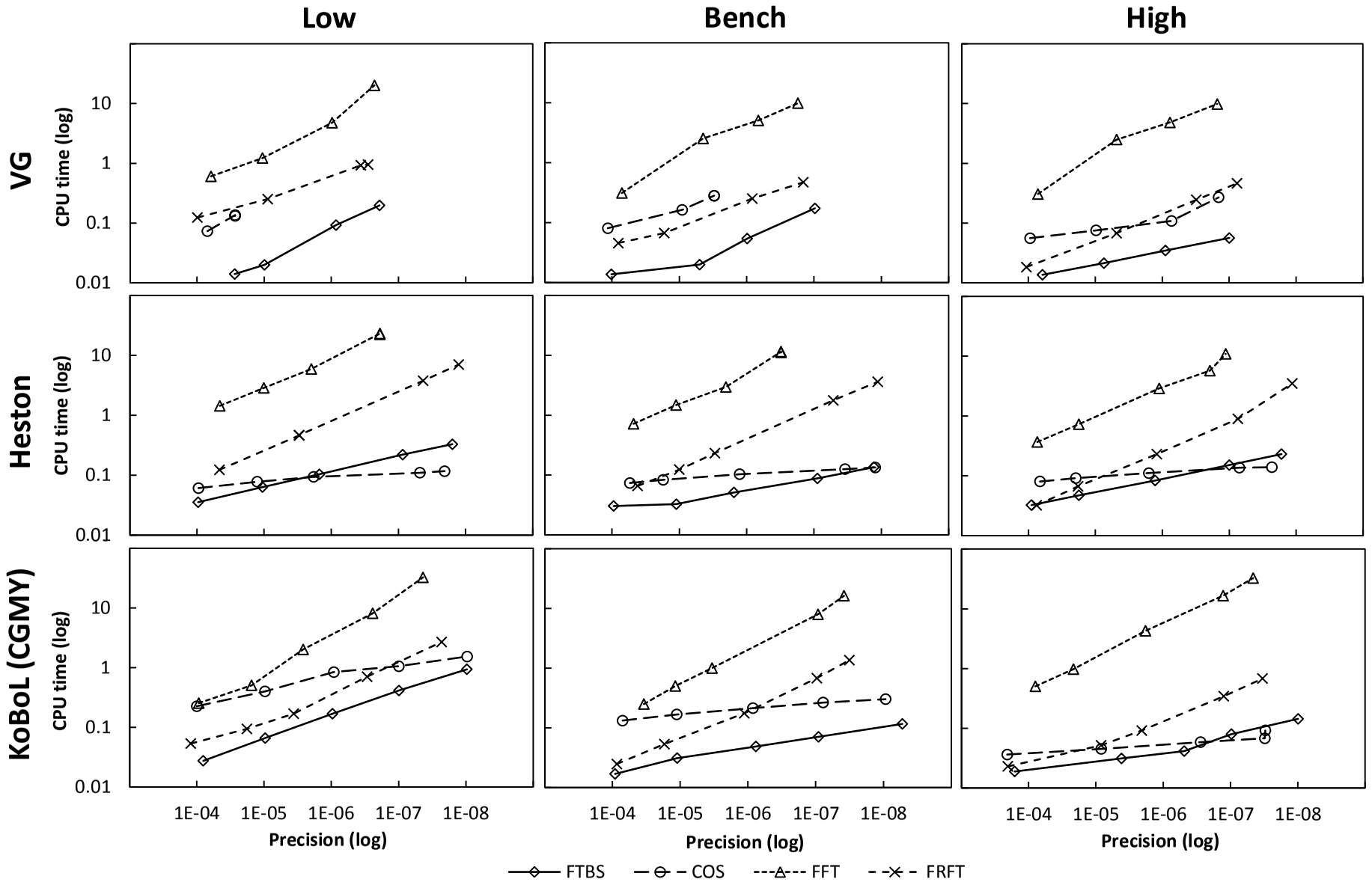

Figure 1: Comparison of precision and computation times for pricing options at multiple strike prices, for range of models and parameter sets.

4.2. Numerical comparison for pricing an option at a single strike price

While the FTBS method is best suited to the problem of pricing options across multiple strike prices, it is also an effective method for pricing an option at a single specified strike price. We now provide a numerical comparison of the FTBS method to a range of state-of-the-art methods for pricing a single European option under the VG process. Specifically, we compare to (i) the COS method, (ii) the IAC method, and (iii) the CONV method. Since the IAC method is only applicable to out-of-the-money options, and the algorithm described in Levendorskiĭ and Xie (2012) is applicable for a restricted parameter set satisfying $\nu > T$, we have only considered a single set of applicable model parameters for this comparison.

As before, each method has been diligently implemented to the best of our ability in C++, using identical hardware, and therefore the computation times are all directly comparable. The details of the implementation of each pricing method is outlined in Appendix D. We note that in the case of the COS method, convergence to the true option price is poor, and the method failed to provide accurate prices using the put option formula for the selected VG process parameter set (see the implementation notes for the COS method in Appendix D). We therefore applied the call option formula directly, instead of applying the put-call parity, as recommended by Fang and Oosterlee (2008) in Remark 5.2. Note that for the single strike comparison, we computed the reference option price values by utilizing the enhanced Simpson rule IAC method, as described in Section 5 of Levendorskiĭ and Xie (2012).

In Figure 2, we provide a graphical comparison of the FTBS method to the above pricing methods for the VG process, where in this case the precision level is defined as the absolute error. It is seen that the FTBS method dominates the COS and CONV methods at all precision levels, but does not outperform the IAC method for pricing a single option, although we note the above parameter restriction for the IAC method.

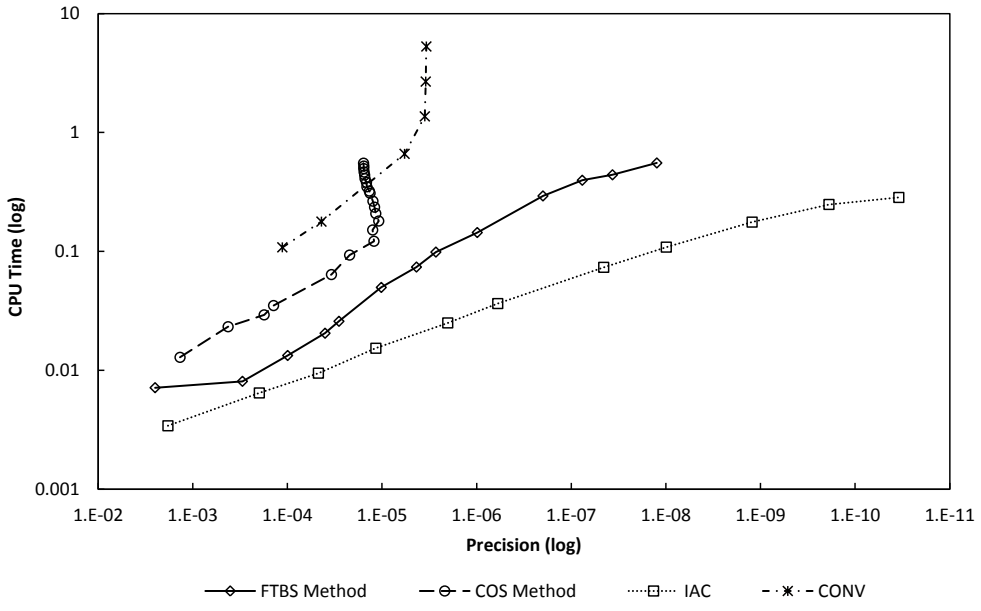

Figure 2: Comparison of precision and computation times for pricing a single option for the VG process. Parameters: $S_0 = 1, K = 1.1, \mu = \frac{1}{480}, \sigma = \frac{1}{4\sqrt{15}}, \nu = \frac{20}{3}, r = 0.1, q = 0, T = 0.5$.

5. Inverse calibration problem

As an effective test of robustness across options of all tenors and moneyness, and across the entire parameter space of the asset model, we consider the effectiveness of the FTBS method in the inverse calibration problem, as described by Cont and Tankov (2004). The inverse calibration problem seeks to identify the asset model parameters such that the discounted asset price is a martingale, and the observed option prices in the market are given by their discounted risk-neutral expectations. There are many approaches to solving the inverse calibration problem. They normally involve repeated computation of the price of options under the selected asset model for many different combinations of the parameters. For example, the popular least squares calibration method requires an exhaustive search of the parameter space across observed options of all maturities and strike prices.

Several examples in the literature consider this problem. Duffie et al. (2000) perform a cross-sectional calibration using observed option prices on the S&P 500 index for a wide range of continuous-time asset models, including stochastic volatility and stochastic volatility jump diffusion models. More recently, Schoutens et al. (2005) perform a similar analysis using a large data set of European options. This data set consists of 144 European call option prices with maturities ranging from less than one month up to 5.16 years. The prices are based on the implied volatility surface of the Eurostoxx 50 index, which had a value of 2461.44 on October 7th, 2003. In Figure 3, we show the implied volatility smile curves across the different maturities at this date.

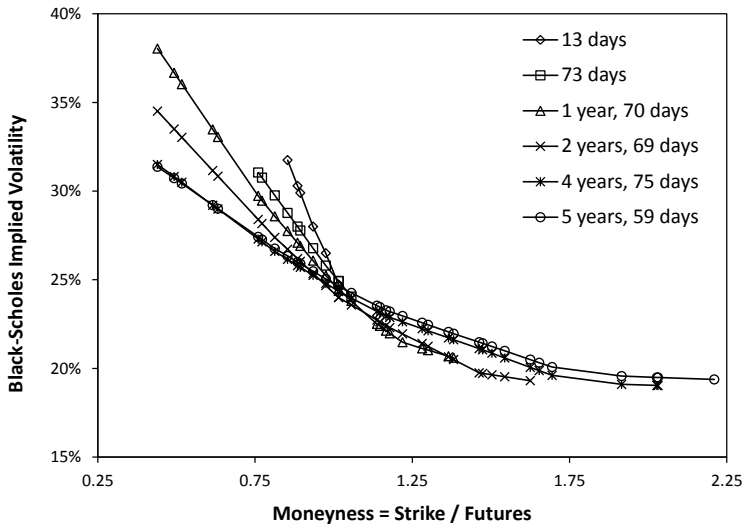

Figure 3: Implied volatility curves for Eurostoxx 50 index on October 7th, 2003

For our calibration example, we use the more recent data set of Schoutens et al. (2005), since it contains a wider range of maturities and, hence, provides a better test for the robustness of numerical pricing methods. We consider two continuous-time models for the inverse calibration problem: (i) the Heston stochastic volatility model, and (ii) an extension of the Heston model introduced by Bates (1996) that includes log-normally distributed jumps that arrive according to a Poisson process.

In our implementation of the cross-sectional calibration, we utilize a global optimization algorithm, as described in Ingber (1989), to perform an exhaustive search of the parameter space. Using this algorithm, we seek to find the model parameters that provide the closest match to

the observed implied volatility surface, subject to the Feller condition, $2\kappa_v\theta_v \leq \sigma_v^2$, being satisfied to ensure positivity of the asset price process. The inverse calibration problem therefore provides an excellent test for the robustness of the FTBS method as it requires accurate pricing at all maturities, strikes, and combinations of the model parameters. For comparison, we also consider the same problem tackled by the FFT and the FRFT.

In Table 1, we show the results of the calibration exercise using different methods and the reference parameters identified by Schoutens et al. (2005). For the Heston model, all three methods provide similar calibrations and are in line with the reference parameters. The FTBS method provides the fastest calibration, requiring just 9.2 seconds to find the optimum parameters over a total of $21,564$ repeated price evaluations of the 144 call options. The FRFT method performed the same task in 14.6 seconds, while the FFT method took a lengthy 714 seconds. We note that it was necessary to change the so called dampening parameter for the FFT and FRFT from the recommended value of Chourdakis (2005) to a value of three to avoid infinite option prices being generated for some parameter sets.

For the Bates model, both the FFT and FRFT methods exhibit numerical instability over the full range of parameters. In addition, the parameters obtained for the FFT and FRFT methods are very different from the reference parameters and substantially increase the pricing error compared to the calibrated Heston parameters. Therefore, we omit the results from these methods. Using the FTBS method, the Bates model is calibrated successfully, and we find two sets of parameters which, while being quite different, provide a close fit to the observed option data. The first is in broad agreement with Schoutens et al. (2005), in which the jump process has a positive mean, and the asset price process and volatility process are perfectly negatively correlated. The second set of parameters is structurally quite close to those identified in Duffie et al. (2000), featuring negative mean jumps and correlation of -0.79 between the asset price and the volatility processes. The latter, we believe, has better economic interpretation and suggests that the inverse calibration process requires additional constraints to ensure the resulting fit can be used reliably in important applications, such as pricing exotic options.

Table 1: Calibration results for the Heston and Bates models

	Heston				Bates		
	Ref.	FTBS	FFT	FRFT	Ref.	FTBS_1	FTBS_2
N	-	50	4096	64	-	150	150
κ_v	0.6067	0.6118	0.6011	0.6297	0.4963	0.5258	0.7423
θ_v	0.0654	0.0646	0.0646	0.0642	0.0650	0.0609	0.0357
σ_v	0.2928	0.2811	0.2788	0.2842	0.2286	0.2530	0.2302
ρ	-0.7571	-0.6689	-0.6690	-0.6670	-0.9900	-0.9900	-0.7923
λ	-	-	-	-	0.1382	0.7906	0.1368
μ_J	-	-	-	-	0.1791	0.0861	-0.1435
σ_J	-	-	-	-	0.1346	0.0670	0.3541
Err		1.90E-04	1.90E-04	2.69E-04		1.16E-04	1.76E-04
Time		9.2s	714.2s	14.6s		135.5s	135.9s

Ref. denotes the reference calibration parameters from Schoutens et al. (2005).

6. Conclusion

In this paper, we have presented an entirely new approach to approximating European-style option prices that utilizes B-spline interpolation theory to provide an efficient closed-form solution. Our framework works across the family of continuous-time exponential semimartingale processes and other models whose characteristic function satisfies certain regularity constraints. Our novel use of the Peano representation of a divided difference makes the FTBS method extremely efficient. It enables us to evaluate the pricing integral in closed-form once the integrands have been replaced by the B-spline approximants.

Through a very careful comparison to other methods in the literature, we have demonstrated that the FTBS method provides accurate prices across the most widely adopted continuous-time asset models and can provide computation times as low as one to two microseconds per option when pricing a basket of options across a range of strike prices. Based on our numerical study, we can conclude that the FTBS method is the preferable approximation method, relative to the other methods considered, for computing option prices across multiple strikes prices for the VG process, and for the Heston model and KoBoL (CGMY) model, for precision up to the level of 1E-6.

The applications of the FTBS framework are therefore wide ranging and will enable more realistic asset models to be utilized in areas of finance in which computation time is a key consideration. For example, pricing, marking, and hedging of large derivative portfolios as well as high frequency trading are all areas that can benefit from accurate real-time pricing under realistic continuous-time asset models.

We believe that our method also has useful applications in parameter estimation in econometrics, when calibration incorporates cross-sectional option price information. We have provided a simple example of this by calibrating the Heston and Bates models to cross-sectional option data. We have demonstrated the robustness of the FTBS method over the full range of maturities and moneyness of options as well as the full model parameter space. The FTBS method also has applications in maximum likelihood estimation, such as that considered by Kimmel et al. (2007), in which cross-sectional information is incorporated and which requires computation of option prices at each iteration of the likelihood search.

Acknowledgments

We would like to thank Sergei Levendorskiĭ and Jiayao Xie for kindly providing their Matlab code for the IAC method, and the two anonymous referees for their valuable comments and suggestions which helped to significantly improve the revised version of our paper.

Appendix A. Fourier transform

Table A.2: Payoff functions and their Fourier transforms for European-style contingent claims

Instrument	Payoff Function $w(x)$	Fourier Transform $\hat{w}(z)$	Strip of Regularity $\mathcal{S}_w$
Call	$(e^x - K)^+$	$\frac{-K^{iz+1}}{z^2-iz}$	$\operatorname{Im} z > 1$
Put	$(K - e^x)^+$	$\frac{-K^{iz+1}}{z^2-iz}$	$\operatorname{Im} z < 0$
Covered Call	$\min(e^x, K)$	$\frac{K^{iz+1}}{z^2-iz}$	$0 < \operatorname{Im} z < 1$
Binary Call	$e^x 1_{\{e^x>K\}}$	$-\frac{K^{iz+1}}{iz+1}$	$\operatorname{Im} z > 1$
Binary Put	$e^x 1_{\{e^x<K\}}$	$\frac{K^{iz+1}}{iz+1}$	$\operatorname{Im} z < 1$
Money Market	1	$2\pi\delta(z)$	$\operatorname{Im} z = 0$

Note that $\delta(z)$ is the Dirac-delta function.

Appendix B. Fourier transform B-spline method and related proofs

Appendix B.1. The divided difference at repeated knots

$$[\underbrace{t_0, \cdots, t_0}_{v_0}, \cdots, \underbrace{t_m, \cdots, t_m}_{v_m}] f = \sum_{i=0}^{m} \frac{D^{v_i-1} g_i(t_i)}{(v_i - 1)!}, \tag{B.1}$$

where

$$g_i(t) = \frac{f(t)}{\prod_{\substack{j=0 \\ j\neq i}}^{l} (t - t_j)^{v_j}},$$

and $D^{v_i-1} g_i(t_i)$ is the $(v_i - 1)$-th derivative of $g_i(t)$ evaluated at $t = t_i$. For a proof see Ignatov and Kaishev (1989).

Appendix B.2. Proof of Theorem 2

We begin with the first integral in Eq. (21).

$$\begin{aligned}
\int_0^1 \cos\left(\frac{1-t}{t}k\right) \tilde{s}_1(t)dt &= \int_0^1 \cos\left(\frac{1-t}{t}k\right) \sum_{i=1}^{p_1} c_{1,i} M_{i,3}(t)dt \\
&= \sum_{i=1}^{p_1} c_{1,i} \int_0^1 \cos\left(\frac{1-t}{t}k\right) M_{i,3}(t)dt \\
&= \sum_{i=1}^{p_1} c_{1,i} \left(3! \int_0^1 \frac{f_1^{(3)}\left(\frac{1-t}{t}k\right)}{3!} M_{i,3}(t)dt\right) \\
&= 6 \sum_{i=1}^{p_1} c_{1,i} [t_{1,i}, \cdots, t_{1,i+n}] f_1(t,k),
\end{aligned}$$

where f_1 is chosen to satisfy $f_1^{(3)}(\frac{1-t}{t}k) = \cos(\frac{1-t}{t}k)$ and we have applied the Peano representation of the divided difference. We identify f_1 through repeated integration of $\cos(\frac{1-t}{t}k)$ with respect to t. This is easily performed using symbolic integration in tools such as Mathematica and yields the result presented in Theorem 2. The formula in Eq. $(C.6)$ follows from Eq. (23) noting that the cosine is equal to one for $k = 0$ and the integral of a B-spline is also one. The proof for the second integral is similar and is therefore omitted. □

Appendix B.3. Proof of Proposition 3 *(FTBS Lookback Option Price Error Bound)*
From (Theorem 1, formula for the price $C(T, K)$) we have

$$
\begin{aligned}
&\left|C(T,K) - \tilde{C}(T,K)\right| = \\
&= \left|\frac{-\sqrt{S_0 K}}{\pi} e^{\frac{(r+q)T}{2}} \left(\int_0^1 \cos\left(\frac{1-t}{t}k\right)(s_1(t) - \tilde{s}_1(t))\, dt + \int_0^1 \sin\left(\frac{1-t}{t}k\right)(s_2(t) - \tilde{s}_2(t))\, dt \right)\right| \\
&\leq \frac{\sqrt{S_0 K}}{\pi} e^{\frac{(r+q)T}{2}} \left(\int_0^1 \left|\cos\left(\frac{1-t}{t}k\right)(s_1(t) - \tilde{s}_1(t))\right| dt + \int_0^1 \left|\cos\left(\frac{1-t}{t}k\right)(s_1(t) - \tilde{s}_1(t))\right| dt \right) \\
&\leq \frac{\sqrt{S_0 K}}{\pi} e^{\frac{(r+q)T}{2}} \left(\underset{t\in[0,1]}{max} |s_1(t) - \tilde{s}_1(t)| \int_0^1 \left|\cos\left(\frac{1-t}{t}k\right)\right| dt \right. \\
&\qquad\qquad \left. + \underset{t\in[0,1]}{max} |s_2(t) - \tilde{s}_2(t)| \int_0^1 \left|\sin\left(\frac{1-t}{t}k\right)\right| dt \right) \\
&\leq \frac{\sqrt{S_0 K}}{\pi} e^{\frac{(r+q)T}{2}} \left(\left\|s_1^{(3)}(t)\right\| C_1 \int_0^1 \left|\cos\left(\frac{1-t}{t}k\right)\right| dt + \left\|s_2^{(3)}(t)\right\| C_2 \int_0^1 \left|\sin\left(\frac{1-t}{t}k\right)\right| dt \right)
\end{aligned}
$$

which is the asserted bound noting that $\tilde{C}_1 = C_1 \int_0^1 \left|\cos\left(\frac{1-t}{t}k\right)\right| dt$ and $\tilde{C}_2 = C_2 \int_0^1 \left|\sin\left(\frac{1-t}{t}k\right)\right| dt$. In the last inequality we have used the optimal spline interpolation bounds $\underset{t\in[0,1]}{max} |(s_i(t) - \tilde{s}_i(t))| = \|s_i(t) - \tilde{s}_i(t)\| \leq C_i \left\|s_i^{(3)}(t)\right\|$, $i = 1, 2$ obtained by Gaffney and Powell (1976). The rate $O\left(|\tilde{\eta}|^3\right)$ in (26) now follows applying to (25) a result by De Boor (2001) (Theorem 22, page 154). This completes the proof of Proposition 3. □

Appendix B.4. Explicit formula for divided differences at repeated knots
We begin by noting that while functions $f_l(t, k)$, $l = 1, 2$, are undefined at $t = 0$, their limit as $t \to 0$ exists. Therefore, in computing the divided differences at the repeated knot zero, we set $t_{l,1} = t_{l,2} = t_{l,3} = \epsilon > 0$, $l = 1, 2$, and consider the limit of the divided differences as $\epsilon \to 0$.

Proposition 5. *Divided difference of* $[\epsilon,\epsilon,\epsilon,t_{1,4}]\,f_1$
This is given by

$$
\begin{aligned}
[\epsilon,\epsilon,\epsilon,t_{1,4}]\,f_1 = \frac{1}{12\epsilon^3(\epsilon-t_{1,4})^3}\Bigg(&\epsilon^3 k\,\mathrm{Ci}\left(\frac{k}{t_{1,4}}\right)\left(6kt_{1,4}\cos(k)-\left(k^2-6t_{1,4}^2\right)\sin(k)\right)+\\
&\epsilon^3 k\,\mathrm{Ci}\left(\frac{k}{\epsilon}\right)\left(\left(k^2-6t_{1,4}^2\right)\sin(k)-6kt_{1,4}\cos(k)\right)+\\
&k\left(\epsilon^3\left(Si\left(\frac{k}{t_{1,4}}\right)-Si\left(\frac{k}{\epsilon}\right)\right)\left(\left(k^2-6t_{1,4}^2\right)\cos(k)+\right.\right.\\
&6kt_{1,4}\sin(k)\Big)-k(\epsilon-t_{1,4})\,\mathrm{sinc}\left(\frac{k}{\epsilon}\right)\left(\cos(k)\left(-6\epsilon^2t_{1,4}+2\epsilon k^2-k^2t_{1,4}\right)+\right.\\
&\left.6\epsilon^2k\sin(k)\right)\Big)+\epsilon^3\cos(k)\left(\epsilon\left(2\epsilon^2-6\epsilon t_{1,4}-k^2+6t_{1,4}^2\right)\cos\left(\frac{k}{\epsilon}\right)+\right.\\
&\left.t_{1,4}\left(k^2-2t_{1,4}^2\right)\cos\left(\frac{k}{t_{1,4}}\right)\right)+\epsilon\left(\epsilon^3k(\epsilon-6t_{1,4})\sin(k)\cos\left(\frac{k}{\epsilon}\right)+\right.\\
&\epsilon^2t_{1,4}\sin\left(\frac{k}{t_{1,4}}\right)\left(\left(k^2-2t_{1,4}^2\right)\sin(k)-5kt_{1,4}\cos(k)\right)+\\
&5\epsilon^2kt_{1,4}^2\sin(k)\cos\left(\frac{k}{t_{1,4}}\right)+\sin\left(\frac{k}{\epsilon}\right)\left(k\cos(k)\left(-\epsilon^4+2\epsilon^2\left(k^2+\right.\right.\right.\\
&\left.\left.\left.3t_{1,4}^2\right)-3\epsilon k^2t_{1,4}+k^2t_{1,4}^2\right)+\epsilon^2\sin(k)\left(2\epsilon^3-6t_{1,4}\left(\epsilon^2+k^2\right)+5\epsilon k^2+6\epsilon t_{1,4}^2\right)\right)\Bigg).
\end{aligned}
\tag{B.2}
$$

PROOF. The simplest way to compute the divided difference with repeated knots is by using Mathematica or a similar symbolic algebraic software package. The result in Eq. $(B.2)$ can be generated using the following expression in Mathematica

```
f1[t_] :=
  1/12 (k CosIntegral[k/t] (-6 k t Cos[k] + (k^2 - 6 t^2) Sin[k]) +
     t (-(k^2 - 2 t^2) Cos[(k (-1 + t))/t] -
        5 k t Sin[(k (-1 + t))/t]) -
     k ((k^2 - 6 t^2) Cos[k] + 6 k t Sin[k]) SinIntegral[k/t]);
g0[t0_, t1_] := f1[t0]/(t0 - t1);
g1[t0_, t1_] := f1[t1]/(t1 - t0)^3;
FullSimplify[1/2*D[g0[\[Epsilon], t14], {\[Epsilon], 2}] +
   g1[\[Epsilon], t14]]
```

Similar logic is used to compute the other divided differences $[\epsilon,\epsilon,t_{l,4},t_{l,5}]f_l$, $[t_{l,p_{l-1}},t_{l,p_l},1,1]f_l$, and $[t_{l,p_l},1,1,1]f_l$ for $l=1,2$. □

In the case of divided differences with repeated arguments, $[\epsilon,\epsilon,\epsilon,t_{1,4}]\,f_1$, $[\epsilon,\epsilon,\epsilon,t_{2,4}]\,f_2$, $[\epsilon,\epsilon,t_{1,4},t_{1,5}]\,f_1$, and $[\epsilon,\epsilon,t_{2,4},t_{2,5}]\,f_2$, we are concerned with their limits as $\epsilon\to 0$. The latter are easily determined, noting in particular that $\lim_{x\to\pm\infty}\mathrm{Ci}(x)=0$, $\lim_{x\to\pm\infty}\mathrm{Si}(x)=\pm\frac{\pi}{2}$, and $\lim_{x\to\pm\infty}\mathrm{Sinc}(x)=0$. This leads to the following simplification of Eq. $(B.2)$ as $\epsilon\to 0$

$$
\begin{aligned}
[0,0,0,t_{1,4}]\, f_1 = \frac{1}{24t_{1,4}^3}\Bigg(&-10At_{1,4}^2 \sin\left[\frac{A(-1+t_{1,4})}{t_{1,4}}\right] + \\
&2A\,\mathrm{Ci}\left[\frac{\mathrm{Sign}[A]A}{t_{1,4}}\right]\left(\left(A^2-6t_{1,4}^2\right)\sin A - 6At_{1,4}\cos A\right) \\
&- 2t_{1,4}\left(A^2-2t_{1,4}^2\right)\cos\left[\frac{A(-1+t_{1,4})}{t_{1,4}}\right] + \\
&A\left(6At_{1,4}\sin[A] + \left(A^2-6t_{1,4}^2\right)\cos A\right)\left(\mathrm{Sign}[A]\pi - 2\,\mathrm{Si}\left[\frac{A}{t_{1,4}}\right]\right)\Bigg).
\end{aligned}
\tag{B.3}
$$

Similar analysis leads to the following expressions for the other repeated divided differences at zero.

$$
\begin{aligned}
[0,0,0,t_{2,4}]\, f_2 = \frac{1}{24t_{2,4}^3}\Bigg(&-10At_{2,4}^2\cos\left[\frac{A(-1+t_{2,4})}{t_{2,4}}\right] + 2A\,\mathrm{Ci}\left[\frac{A}{t_{2,4}}\right]\left(\left(A^2-6t_{2,4}^2\right)\cos A + \right. \\
&\left. 6At_{2,4}\sin A\right) + 2t_{2,4}\left(A^2-2t_{2,4}^2\right)\sin\left[\frac{A(-1+t_{2,4})}{t_{2,4}}\right] - A\left(-6At_{2,4}\cos A + \right. \\
&\left.\left(A^2-6t_{2,4}^2\right)\sin A\right)\left(\mathrm{Sign}[A]\pi - 2\,\mathrm{Si}\left[\frac{A}{t_{2,4}}\right]\right)\Bigg),
\end{aligned}
\tag{B.4}
$$

$$
\begin{aligned}
[0,0,t_{1,4},t_{1,5}]\, f_1 = \frac{1}{24t_{1,4}^2(t_{1,4}-t_{1,5})t_{1,5}^2}\Bigg(&\mathrm{Sign}[A]A^3\pi\left(-t_{1,4}^2+t_{1,5}^2\right)\cos A + \\
&2t_{1,4}t_{1,5}\left(-\left(A^2-2t_{1,4}^2\right)t_{1,5}\cos\left[\frac{A(-1+t_{1,4})}{t_{1,4}}\right] + \right. \\
&\left. t_{1,4}\left(A^2-2t_{1,5}^2\right)\cos\left[\frac{A(-1+t_{1,5})}{t_{1,5}}\right]\right) + 2A\left(t_{1,5}^2\,\mathrm{Ci}\left[\frac{\mathrm{Sign}[A]A}{t_{1,4}}\right]\left(-6At_{1,4}\cos A + \right.\right. \\
&\left.\left.\left(A^2-6t_{1,4}^2\right)\sin A\right) + t_{1,4}^2\,\mathrm{Ci}\left[\frac{\mathrm{Sign}[A]A}{t_{1,5}}\right]\left(6At_{1,5}\cos A - \left(A^2-6t_{1,5}^2\right)\sin A\right) + \right. \\
&\left. t_{1,5}\left(t_{1,4}\left(\mathrm{Sign}[A]3A\pi(-t_{1,4}+t_{1,5})\sin A + 5t_{1,4}t_{1,5}\left(\sin\left[A\left(-1+\right.\right.\right.\right.\right.\right. \\
&\left.\left.\left.\left.\left.\left.\frac{1}{t_{1,4}}\right)\right] + \sin\left[\frac{A(-1+t_{1,5})}{t_{1,5}}\right]\right)\right) - t_{1,5}\left(\left(A^2-6t_{1,4}^2\right)\cos A + \right. \right. \\
&\left.\left. 6At_{1,4}\sin A\right)\mathrm{Si}\left[\frac{A}{t_{1,4}}\right]\right) + t_{1,4}^2\left(\left(A^2-6t_{1,5}^2\right)\cos A + 6At_{1,5}\sin A\right)\mathrm{Si}\left[\frac{A}{t_{1,5}}\right]\Bigg)\Bigg),
\end{aligned}
\tag{B.5}
$$

$$
\begin{aligned}
[0,0,t_{2,4},t_{2,5}]\, f_2 = \frac{1}{24t_{2,4}^2(t_{2,4}-t_{2,5})t_{2,5}^2}\Bigg(&2At_{2,4}t_{2,5}\left(\text{Sign}[A]*3A\pi(-t_{2,4}+t_{2,5})\cos A+\right.\\
&\left.5t_{2,4}t_{2,5}\left(-\cos\left[\frac{A(-1+t_{2,4})}{t_{2,4}}\right]+\cos\left[\frac{A(-1+t_{2,5})}{t_{2,5}}\right]\right)\right)+\\
&\text{Sign}[A]*A^3\pi t_{2,4}^2\sin A-\text{Sign}[A]*A^3\pi t_{2,5}^2\sin A+\\
&2At_{2,5}^2\,\text{Ci}\left[\frac{A}{t_{2,4}}\right]\left(\left(A^2-6t_{2,4}^2\right)\cos A+\right.\\
&\left.6At_{2,4}\sin A\right)-2At_{2,4}^2\,\text{Ci}\left[\frac{A}{t_{2,5}}\right]\left(\left(A^2-6t_{2,5}^2\right)\cos A+6At_{2,5}\sin A\right)+\\
&2A^2t_{2,4}t_{2,5}^2\sin\left[\frac{A(-1+t_{2,4})}{t_{2,4}}\right]-4t_{2,4}^3t_{2,5}^2\sin\left[\frac{A(-1+t_{2,4})}{t_{2,4}}\right]\\
&-2t_{2,4}^2t_{2,5}\left(A^2-2t_{2,5}^2\right)\sin\left[\frac{A(-1+t_{2,5})}{t_{2,5}}\right]+\\
&2A\left(t_{2,5}^2\left(-6At_{2,4}\cos A+\left(A^2-6t_{2,4}^2\right)\sin A\right)\text{Si}\left[\frac{A}{t_{2,4}}\right]+\right.\\
&\left.t_{2,4}^2\left(6At_{2,5}\cos A-A^2\sin A+6t_{2,5}^2\sin A\right)\text{Si}\left[\frac{A}{t_{2,5}}\right]\right).
\end{aligned}
\tag{B.6}
$$

Appendix C. The FTBS method for alternative contour integrals

Theorem 3 (Strike-separable pricing formula). *An alternative form of Eq.* (5) *for computing* $C(T,K)$ *is given by*

$$
C(T,K) = -\frac{Ke^{-rT+vk}}{\pi} I(k;v) \tag{C.1}
$$

where

$$
I(k;v) = \int_0^1 \cos\left(\frac{1-t}{t}k\right) s_1(t;v)dt + \int_0^1 \sin\left(\frac{1-t}{t}k\right) s_2(t;v)dt
$$

$$
s_1(t;v) = \frac{\text{Re}\left[\phi_{X_T}(-\frac{1-t}{t}-iv)\right]\left[\left(\frac{1-t}{t}\right)^2+v-v^2\right]+\text{Im}\left[\phi_{X_T}(-\frac{1-t}{t}-iv)\right]\left(\frac{1-t}{t}\right)(2v-1)}{t^2\left(\left[\left(\frac{1-t}{t}\right)^2+v-v^2\right]^2+\left(\frac{1-t}{t}\right)^2(2v-1)^2\right)}
$$

$$
s_2(t;v) = \frac{-\text{Re}\left[\phi_{X_T}(-\frac{1-t}{t}-iv)\right]\left(\frac{1-t}{t}\right)(2v-1)+\text{Im}\left[\phi_{X_T}(-\frac{1-t}{t}-iv)\right]\left[\left(\frac{1-t}{t}\right)^2+v-v^2\right]}{t^2\left(\left[\left(\frac{1-t}{t}\right)^2+v-v^2\right]^2+\left(\frac{1-t}{t}\right)^2(2v-1)^2\right)}
$$

and $v \in (1,\alpha)$*, as in Proposition 1.*

PROOF. Note that since the option price provided by the integral in Eq. (2) is real-valued, we can rewrite it as

$$
C(T,K) = -\frac{Ke^{-rT}}{2\pi}\int_{iv-\infty}^{iv+\infty}\text{Re}\left[\phi_{X_T}(-z)\frac{e^{-izk}}{z^2-iz}\right]dz, \text{ with } v\in(1,\alpha),
$$

We now denote the real and imaginary parts of $\phi_{X_T}(-z)$ as $\phi_{Re}(z_1,v)$ and $\phi_{Im}(z_1,v)$ so that

$$
\phi_{X_T}(-z) = \phi_{X_T}(-(z_1+iv)) = \phi_{Re}(z_1,v)+i\phi_{Im}(z_1,v)
$$

where $z_1 \in (-\infty, \infty)$ along the integration path. For brevity, we write ϕ_{Re} and ϕ_{Im}, instead of $\phi_{Re}(z_1, v)$ and $\phi_{Im}(z_1, v)$, keeping in mind that these are functions of z_1 and v. We now rewrite e^{-izk},

$$e^{-izk} \;=\; e^{-i(z_1+iv)k} = e^{vk}e^{-iz_1k} = e^{vk}\left[\cos(z_1k) - i\sin(z_1k)\right].$$

Therefore (2) becomes

$$C(T,K) = -\frac{Ke^{-rT}}{2\pi}\int_{-\infty}^{\infty} \mathrm{Re}\left[(\phi_{Re}+i\phi_{Im})\,\frac{e^{vk}\left[\cos(z_1k) - i\sin(z_1k)\right]}{\underbrace{(z_1+iv)^2 - i(z_1+iv)}_{(*)}}\right] dz_1 \qquad \text{(C.2)}$$

We expand $(*)$ into its real and imaginary parts

$$(*) = z_1^2 + 2ivz_1 - v^2 - iz_1 + v = z_1^2 + v - v^2 + iz_1(2v-1)$$

Multiplying the numerator and denominator of the integrand in Eq. $(C.2)$ by the complex congruent of $(*)$ provides

$$C(T,K) = -\frac{Ke^{-rT+vk}}{2\pi}\int_{-\infty}^{\infty}\mathrm{Re}\left[\frac{\overbrace{(\phi_{Re}+i\phi_{Im})}^{a_1+a_2i}\overbrace{\left[\cos(z_1k) - i\sin(z_1k)\right]}^{b_1-b_2i}\overbrace{\left(z_1^2+v-v^2-iz_1(2v-1)\right)}^{c_1-c_2i}}{(z_1^2+v-v^2)^2 + z_1^2(2v-1)^2}\right] dz_1.$$

Now, noting that $\mathrm{Re}[(a_1+a_2i)(b_1-b_2i)(c_1-c_2i)] = a_1b_1c_1 + a_2b_2c_1 + a_2b_1c_2 - a_1b_2c_2$, we have the following expression for the pricing integral

$$\begin{aligned} C(T,K) \;=\; & -\frac{Ke^{-rT+vk}}{2\pi}\left[\int_{-\infty}^{\infty}\frac{\cos(z_1k)\left[\phi_{Re}\left(z_1^2+v-v^2\right) + \phi_{Im}z_1(2v-1)\right]}{(z_1^2+v-v^2)^2 + z_1^2(2v-1)^2}dz_1\right. \\ & \left. + \int_{-\infty}^{\infty}\frac{\sin(z_1k)\left[-\phi_{Re}z_1(2v-1) + \phi_{Im}\left(z_1^2+v-v^2\right)\right]}{(z_1^2+v-v^2)^2 + z_1^2(2v-1)^2}dz_1\right]. \end{aligned} \qquad \text{(C.3)}$$

In order to simplify Eq. $(C.3)$, we now show that the integrands in Eq. $(C.3)$ are even function with respect to z_1. Examining the characteristic function of X_T along the integration path shows that

$$\begin{aligned} \phi_{X_T}(-z) \;&=\; E_{\mathcal{Q}}\left(e^{-izX_T}\right) = E_{\mathcal{Q}}\left(e^{-i(z_1+iv)X_T}\right) \\ &=\; E_{\mathcal{Q}}\left(e^{-iz_1X_Te^{vX_n}}\right) \\ &=\; E_{\mathcal{Q}}\left(e^{vX_n}\left[\cos(-z_1) + i\sin(-z_1)\right]\right), \text{ since } X_T \in \mathbf{R} \\ &=\; E_{\mathcal{Q}}\left(e^{vX_n}\left[\cos(z_1) - i\sin(z_1)\right]\right) \\ &=\; E_{\mathcal{Q}}\left(e^{vX_n}\cos(z_1)\right) - iE_{\mathcal{Q}}\left(e^{vX_n}\sin(z_1)\right) \\ &=\; \phi_{Re}(z_1,v) + i\phi_{Im}(z_1,v), \end{aligned}$$

where ϕ_{Re} and ϕ_{Im} are as defined in Eq. $(C.3)$. Observe that $\phi_{Re}(z_1, v) = \phi_{Re}(-z_1, v)$ and $\phi_{Im}(z_1, v) = -\phi_{Im}(-z_1, v)$, so that the real part of $\phi_{X_T}(-z)$ is even and the imaginary part is odd. Now, in $(C.3)$, denote the first integrand by $f_1(z_1, v)$ and the second by $f_2(z_1, v)$ so that

$$C(T, R) = -\frac{R^{1-v}}{2\pi}\left[\int_{-\infty}^{\infty} f_1(z_1, v)dz_1 + \int_{-\infty}^{\infty} f_2(z_1, v)dz_1\right].$$

By inspection, using the even property of ϕ_{Re} and cosine, and the odd property of ϕ_{Im} and sine, we have that both $f_1(z_1, v) = f_1(-z_1, v)$ and $f_2(z_1, v) = f_2(-z_1, v)$.

Using this even function property we can modify the integration limits of $(C.3)$ to $[0, \infty)$, noting the even function property and scaling by a factor of 2. We then make the substitution $z_1 = \frac{1-t}{t}$ in Eq. $(C.3)$ which provides the required result. □

Theorem 4 (The FTBS pricing formula for alternative contour integrals).
Let $\{t_{1,i}\}_{i=1}^{6+l_1}$, $\{t_{2,j}\}_{j=1}^{6+l_2}$ and $\{c_{1,i}(v)\}_{i=1}^{p_1}, \{c_{2,j}(v)\}_{j=1}^{p_2}$ be the sets of knots and linear coefficients of $\tilde{s}_1(t; v)$ and $\tilde{s}_2(t; v)$ respectively, which are the the quadratic spline interpolants of functions $s_1(t; v)$ and $s_2(t; v)$, as specified in Theorem 3, where $p_1 = l_1 + 3$ and $p_2 = l_2 + 3$. Additionally, let $k = \log\left(\frac{S_0}{K}\right) + (r - q)T$, and $v \in (1, \alpha)$, as in Proposition 1. The pricing formula of a European call option is given as

$$C(T, K) \approx -\frac{Ke^{-rT+vk}}{\pi}\tilde{I}(k; v), \tag{C.4}$$

where for $k \neq 0$

$$\begin{aligned}\tilde{I}(k; v) = 6\sum_{i=1}^{p_1} c_{1,i}(v)[t_{1,i}, t_{1,i+1}, t_{1,i+2}, t_{1,i+3}]f_1(t, k) \\ + 6\sum_{i=1}^{p_2} c_{2,i}(v)[t_{2,i}, t_{2,i+1}, t_{2,i+2}, t_{2,i+3}]f_2(t, k),\end{aligned} \tag{C.5}$$

and f_1 and f_2 are defined as in Theorem 2. For $k = 0$, $\tilde{I}(k; v)$ simplifies to

$$\tilde{I}(k; v) = \sum_{i=1}^{p_1} c_{1,i}(v) + \sum_{i=1}^{p_2} c_{2,i}(v). \tag{C.6}$$

PROOF. The proof utilizes Eq. $(C.1)$ of Theorem 3, and is similar to that Theorem 2, and therefore is omitted. □

Appendix D. Numerical implementation of pricing methods

The FFT and FRFT methods:
Our implementation of the FFT and FRFT methods carefully follows the description of Chourdakis (2005), and utilizes the numerical library of Frigo and Johnson (1998) in C++, which is known to be one of the fastest implementations of the FFT available. Under the FFT and FRFT, it is necessary to truncate the upper limit of the Fourier integral and select the number of grid points for the numerical computation. We follow the recommendations of Chourdakis (2005) in configuring each of the methods for the respective asset models.

The COS method:
Our implementation of the COS method precisely follows the description provided in Fang and Oosterlee (2008). In particular, we have utilized Eq. (30) in Fang and Oosterlee (2008), which enables option prices for different strikes to be obtained in one single numerical experiment. The COS method requires a truncation limit for integration, and we have followed the recommendation $L = 10$, specified in Eq. (49) and the calculation of the cumulants specified in Table 11, both in Fang and Oosterlee (2008). We note that in the case of the Heston model, we use only the first and second cumulant in defining the truncation range, as recommended by Fang and Oosterlee (2008). Finally, we note that following Remark 5.2 of Fang and Oosterlee (2008), we have computed put option prices, and used the put-call parity to compute the required call option prices. This is noted to remove sensitivity with respect to the choice of the truncation parameter L.

The IAC method:
We have implemented the IAC method using the enhanced Simpson rule for the VG process described in Levendorskiĭ and Xie (2012). We thank Sergei Levendorskiĭ and Jiayao Xie for kindly providing their Matlab code which we directly converted into C++ in our implementation.

The CONV method:
Our implementation of the CONV method carefully follows Eq. (27) of Lord et al. (2009), and again we utilize the numerical library of Frigo and Johnson (1998) for the computation of the FFT. We note that Eq. (27) of Lord et al. (2009) is intended for use in pricing early exercise options, and therefore provides option prices across a range of initial underlying asset prices (as opposed to strike prices). Since we are not concerned with option prices at different initial asset prices, we replace the outer discrete Fourier transform in Eq. (27) of Lord et al. (2009), with a single summation corresponding to the required initial asset price. This provides the optimal performance of the CONV method when it is applied for pricing European options at a single initial asset price.

Appendix E. Other numerical comparisons

We now provide further numerical comparisons to validate the numerical accuracy of the FTBS method for computing the Greeks, and also the accuracy of options prices under additional asset price models.

The Greeks:
We compute the Greeks of European options under the Heston model using the FTBS method and compare them to the results of Kristensen and Mele (2011) under their Taylor expansion approximation method. The results of our comparison of the Greeks are provided in (a), (b), (c), and (d) of Table F.9. Note that the reported exact option sensitivities are those provided in Kristensen and Mele (2011). Across all levels of initial share price and volatility, the FTBS method produces accurate results using only 50 data sites. Moving to 100 data sites yields results to seven significant figures. The total computation time of the option price Delta's and Gamma's in (a) and (b) of Table F.9 are just 0.02 milliseconds respectively using 50 data sites for the 11 initial share prices, or equivalently, under two microseconds per option. In (c) and (d) of Table F.9, the total calculation time increases to 0.84 milliseconds since we need to re-fit the spline approximants for each initial volatility level V_0. Overall, we have demonstrated that the FTBS method provides greater precision than the approximation method of Kristensen and Mele (2011), since we have precisely matched the reference prices, while achieving computation times as low as two microseconds per option.

The double exponential jump diffusion process:
We consider the pricing example under the double exponential jump diffusion process of Kou (2002) that uses the enhanced FFT method described in Boyarchenko and Levendorskiĭ (2002). Quittard-Pinon and Randrianarivony (2010) produce the reference prices, and we provide our numerical comparison in Table F.4. Using 500 data sites, the FTBS method produces prices that are in agreement with the reference prices and are more accurate than the FFT results by about one significant figure. Our computation time is only 0.2ms in total for the 11 options.

The mixed exponential jump diffusion model:
Cai and Kou (2011) report the prices of European call options under the Mixed Exponential Jump Diffusion model using the EI method of Petrella (2004). Our results are presented in Table F.5. Using 100 data sites, the FTBS method produces near identical prices. The FTBS computation time is 0.05ms per option.

Appendix F. Tables

Table F.3: Recommended configuration of FTBS method for desired level of precision

Accuracy	VG	Heston	KoBoL / CGMY
1E-4	25	25	30
1E-5	35	50	70
1E-6	200	80	180
1E-7	400	170	450
1E-8	-	250	1050

Above is based on numerical results from Section 4 by considering minimum number of knots required for desired precision across "low", "bench", and "high" parameter sets.

Table F.4: Comparison to the generalized FFT method results of Quittard-Pinon and Randrianarivony (2010), C_1 series, Table 2 (DEJD model)

	FFT Error	FTBS Error
90	-1.51E-08	6.56E-09
92	-1.61E-08	1.37E-10
94	-1.16E-08	-1.42E-09
96	-3.77E-09	-9.51E-10
98	2.02E-09	-6.65E-10
100	3.05E-08	-5.84E-10
102	4.62E-08	9.19E-10
104	3.28E-08	-9.62E-10
106	1.53E-08	-3.82E-10
108	7.86E-09	-6.95E-10
110	9.25E-09	-5.71E-09
Total time	16ms	0.2ms
Speed-up Factor		$80\times$

DEJD parameters: $S = 100, \sigma = 0.16, \lambda = 1, \eta_1 = 10, \eta_2 = 5, p = 0.4, r = 0.05, q = 0$.
Calculation time reported for pricing 11 options. FFT method is using $N = 4096$ and FTBS method uses $N = 500$.

Table F.5: Comparison to the EI method results of Cai and Kou (2011), Table 1 (MEJD model)

		$\sigma = 0.2$			$\sigma = 0.3$		
η_1	λ	EI Price	FTBS	Diff.	EI Price	FTBS	Diff.
20	1	10.97472	10.97472	0.00000	14.59752	14.59753	0.00001
20	3	11.94485	11.94485	0.00000	15.29993	15.29994	0.00001
20	5	12.83076	12.83077	0.00001	15.96677	15.96678	0.00001
40	1	10.57572	10.57572	0.00000	14.31636	14.31637	0.00001
40	3	10.82050	10.82050	0.00000	14.48475	14.48476	0.00001
40	5	11.05846	11.05845	0.00001	14.65079	14.65079	0.00000

Mixed Exponential parameters: $S = 100, \sigma = 0.2, \lambda = 5, \eta_1 = 20, \theta_1 = 20, \eta_2 = 50, \theta_2 = 50, p = 0.4, p_1 = 1.2, p_2 = -0.2, q_1 = 1.3, q_2 = -0.3, r = 0.05, q = 0, T = 1, K = 100$.
EI parameters $A = 18, n = 30, X = 10,000$. FTBS method using $N = 100$.

Table F.6: Heston model option pricing results

		Low						Bench						High					
Accuracy	Pricing Method	N	Mean Error	RMSE	Abs. Error	Time (ms)	Speed up to FFT	N	Mean Error	RMSE	Abs. Error	Time (ms)	Speed up to FFT	N	Mean Error	RMSE	Abs. Error	Time (ms)	Speed up to FFT
1E-04	FFT	10	9.4E-06	1.6E-05	4.6E-05	1.43	-	9	2.8E-05	3.1E-05	4.8E-05	0.72	-	8	4.7E-05	5.1E-05	7.4E-05	0.37	-
	FRFT	6	1.1E-05	1.5E-05	4.6E-05	0.12	12	5	2.1E-05	2.4E-05	4.2E-05	0.07	11	4	2.5E-05	3.4E-05	7.5E-05	0.03	11
	COS	17	4.6E-05	5.3E-05	9.5E-05	0.06	24	21	3.1E-05	3.5E-05	5.4E-05	0.07	10	23	5.2E-05	5.4E-05	6.7E-05	0.08	5
	FTBS	23	4.2E-05	5.0E-05	9.6E-05	0.04	41	20	8.3E-05	8.3E-05	9.5E-05	0.03	24	21	8.7E-05	8.7E-05	9.0E-05	0.03	11
1E-05	FFT	11	2.5E-06	3.8E-06	1.0E-05	2.86	-	10	5.8E-06	6.9E-06	1.1E-05	1.48	-	9	1.1E-05	1.2E-05	1.8E-05	0.73	-
	FRFT	8	5.7E-07	1.1E-06	3.0E-06	0.46	6	6	3.4E-06	4.5E-06	1.0E-05	0.12	12	5	1.1E-05	1.2E-05	1.8E-05	0.07	11
	COS	22	7.5E-06	8.3E-06	1.3E-05	0.08	37	24	1.0E-05	1.1E-05	1.7E-05	0.08	18	26	1.3E-05	1.4E-05	2.0E-05	0.09	8
	FTBS	44	4.7E-06	5.5E-06	1.1E-05	0.06	45	22	9.6E-06	9.7E-06	1.1E-05	0.03	45	32	1.7E-05	1.7E-05	1.8E-05	0.05	15
1E-06	FFT	12	9.0E-07	1.0E-06	2.0E-06	5.91	-	11	8.5E-07	9.9E-07	2.0E-06	2.96	-	11	5.0E-07	6.2E-07	1.1E-06	2.89	-
	FRFT	8	5.9E-07	1.1E-06	3.0E-06	0.46	13	7	1.7E-06	2.0E-06	3.0E-06	0.23	13	7	8.1E-07	9.0E-07	1.2E-06	0.23	13
	COS	26	1.1E-06	1.2E-06	1.8E-06	0.09	63	29	6.4E-07	7.3E-07	1.3E-06	0.10	29	31	1.0E-06	1.1E-06	1.6E-06	0.11	26
	FTBS	75	6.4E-07	7.7E-07	1.5E-06	0.10	57	36	1.4E-06	1.4E-06	1.5E-06	0.05	58	60	1.2E-06	1.2E-06	1.3E-06	0.08	35
1E-07	FFT	14	1.4E-07	1.6E-07	1.9E-07	23.10	-	13	1.5E-07	1.8E-07	3.1E-07	11.55	-	12	1.1E-07	1.2E-07	1.9E-07	5.72	-
	FRFT	11	9.7E-09	1.6E-08	4.3E-08	3.77	6	10	3.3E-08	3.6E-08	5.2E-08	1.77	7	9	4.3E-08	4.8E-08	7.4E-08	0.90	6
	COS	31	2.4E-08	2.8E-08	4.8E-08	0.11	210	35	2.0E-08	2.2E-08	3.5E-08	0.13	92	37	4.2E-08	4.6E-08	7.1E-08	0.14	42
	FTBS	165	3.0E-08	3.8E-08	8.6E-08	0.22	105	60	3.9E-08	4.3E-08	8.9E-08	0.09	132	110	9.7E-08	9.7E-08	1.0E-07	0.15	38
1E-08	FFT	14	1.4E-07	1.6E-07	1.9E-07	22.41	-	13	1.5E-07	1.8E-07	3.1E-07	11.18	-	13	5.5E-08	6.5E-08	1.1E-07	10.94	-
	FRFT	12	2.6E-09	4.4E-09	1.3E-08	7.03	3	11	4.8E-09	5.8E-09	1.1E-08	3.62	3	11	8.7E-09	8.9E-09	1.1E-08	3.49	3
	COS	34	1.2E-08	1.4E-08	2.1E-08	0.12	192	38	6.7E-09	7.6E-09	1.2E-08	0.13	84	39	1.3E-08	1.5E-08	2.3E-08	0.14	78
	FTBS	250	5.0E-09	6.6E-09	1.6E-08	0.33	68	100	1.1E-08	1.1E-08	1.3E-08	0.14	83	170	1.6E-08	1.6E-08	1.7E-08	0.23	47

Option pricing parameters under Heston model:

Low. $S_0 = 1, V_0 = 0.01, \kappa = 1.0, \theta = 0.09, \sigma = 0.05, \rho = -0.5, r = q = 0, T = 0.1.$

Bench. $S_0 = 1, V_0 = 0.09, \kappa = 3.0, \theta = 0.09, \sigma = 0.15, \rho = -0.5, r = q = 0, T = 0.25.$

High. $S_0 = 1, V_0 = 0.81, \kappa = 9.0, \theta = 0.09, \sigma = 0.45, \rho = -0.5, r = q = 0, T = 1.$

Upper integration limits for FFT and FRFT chosen optimally to minimize pricing error.

Calculation time reported for pricing 31 options with $K = 0.85$ to 1.15 in steps of 0.01.

Table F.7: Variance gamma process option pricing results

		Low						Bench						High					
Accuracy	Pricing Method	N	Mean Error	RMSE	Abs. Error	Time (ms)	Speed up vs. FFT	N	Mean Error	RMSE	Abs. Error	Time (ms)	Speed up vs. FFT	N	Mean Error	RMSE	Abs. Error	Time (ms)	Speed up vs. FFT
1E-04	FFT	10	8.9E-06	1.7E-05	6.2E-05	0.60	-	9	3.0E-05	3.5E-05	7.1E-05	0.31	-	9	3.9E-05	4.4E-05	7.2E-05	0.31	-
	FRFT	7	9.0E-05	9.0E-05	9.9E-05	0.12	5	5	3.5E-05	4.1E-05	8.0E-05	0.05	7	4	4.3E-05	5.2E-05	1.1E-04	0.02	17
	COS	27	2.1E-05	2.6E-05	7.0E-05	0.07	8	30	5.2E-05	6.0E-05	1.2E-04	0.08	4	20	5.2E-05	5.9E-05	9.4E-05	0.06	5
	FTBS	21	1.2E-05	1.5E-05	2.7E-05	0.01	43	20	9.2E-05	9.2E-05	1.0E-04	0.01	23	21	6.0E-05	6.0E-05	6.2E-05	0.01	22
1E-05	FFT	11	2.2E-06	3.6E-06	1.1E-05	1.23	-	12	1.6E-06	2.0E-06	4.4E-06	2.54	-	12	2.8E-06	3.2E-06	4.8E-06	2.50	-
	FRFT	8	6.3E-06	6.4E-06	8.8E-06	0.25	5	6	6.5E-06	7.8E-06	1.7E-05	0.07	38	6	2.8E-06	3.1E-06	4.8E-06	0.07	37
	COS	50	6.3E-06	9.1E-06	2.7E-05	0.14	9	60	3.4E-06	4.2E-06	9.1E-06	0.16	16	28	5.1E-06	5.7E-06	9.7E-06	0.08	33
	FTBS	32	2.3E-06	3.1E-06	9.9E-06	0.02	61	31	2.1E-06	2.5E-06	5.0E-06	0.02	128	33	7.1E-06	7.1E-06	7.4E-06	0.02	117
1E-06	FFT	13	2.3E-07	3.7E-07	9.7E-07	4.77	-	13	2.7E-07	3.1E-07	6.7E-07	5.07	-	13	4.0E-07	4.6E-07	7.7E-07	4.82	-
	FRFT	10	7.9E-08	1.2E-07	3.6E-07	0.93	5	8	4.6E-07	5.2E-07	8.2E-07	0.25	20	8	1.5E-07	1.8E-07	3.1E-07	0.25	20
	COS	50	6.3E-06	9.1E-06	2.7E-05	0.13	36	100	1.2E-06	1.5E-06	3.0E-06	0.28	18	40	3.9E-07	4.4E-07	7.3E-07	0.11	45
	FTBS	170	1.5E-07	2.7E-07	8.5E-07	0.09	52	95	2.7E-07	3.6E-07	9.8E-07	0.05	94	60	8.8E-07	8.8E-07	9.0E-07	0.03	139
1E-07	FFT	15	7.4E-08	1.0E-07	2.3E-07	20.14	-	14	7.5E-08	8.8E-08	1.7E-07	9.87	-	14	8.3E-08	9.6E-08	1.5E-07	9.80	-
	FRFT	10	5.5E-08	8.4E-08	2.8E-07	0.95	21	9	6.5E-08	7.7E-08	1.4E-07	0.47	21	9	4.2E-08	4.7E-08	7.6E-08	0.46	21
	COS	50	6.3E-06	9.1E-06	2.7E-05	0.13	153	220	1.2E-06	1.5E-06	3.0E-06	0.28	17	100	7.8E-08	8.4E-08	1.4E-07	0.27	36
	FTBS	380	6.7E-08	8.7E-08	1.9E-07	0.20	102	330	1.8E-08	3.0E-08	9.7E-08	0.17	57	100	9.7E-08	9.7E-08	1.0E-07	0.06	174

Option pricing parameters under Variance gamma process:

Low. $S_0 = 1, \mu = -0.1, \sigma = 0.15, \nu = 0.1, r = q = 0, T = 0.1.$

Bench. $S_0 = 1, \mu = -0.2, \sigma = 0.3, \ \nu = 0.2, r = q = 0, T = 0.25.$

High. $S_0 = 1, \mu = -0.3, \sigma = 0.45, \nu = 0.3, r = q = 0, T = 1.$

Upper integration limits for FFT and FRFT chosen optimally to minimize pricing error.

Calculation time reported for pricing 31 options with $K = 0.85$ to 1.15 in steps of 0.01.

Table F.8: KoBoL (CGMY) process option pricing results

Accuracy	Pricing Method	Low N	Low Mean Error	Low RMSE	Low Abs. Error	Low Time (ms)	Low Speed up vs. FFT	Bench N	Bench Mean Error	Bench RMSE	Bench Abs. Error	Bench Time (ms)	Bench Speed up vs. FFT	High N	High Mean Error	High RMSE	High Abs. Error	High Time (ms)	High Speed up vs. FFT
1E-04	FFT	8	4.9E-05	5.5E-05	9.3E-05	0.25	-	8	1.8E-05	2.1E-05	3.4E-05	0.25	-	9	3.8E-05	4.4E-05	7.9E-05	0.50	-
	FRFT	5	6.2E-05	6.9E-05	1.2E-04	0.05	5	4	3.6E-05	4.3E-05	8.6E-05	0.02	10	4	1.5E-04	1.6E-04	2.0E-04	0.02	22
	COS	73	4.2E-05	5.0E-05	1.0E-04	0.23	1	43	3.7E-05	4.2E-05	7.1E-05	0.13	2	12	1.9E-04	1.9E-04	2.1E-04	0.04	14
	FTBS	26	3.8E-05	4.3E-05	8.0E-05	0.03	9	15	8.4E-05	8.4E-05	9.1E-05	0.02	15	17	1.5E-04	1.5E-04	1.6E-04	0.02	27
1E-05	FFT	9	7.1E-06	8.0E-06	1.5E-05	0.51	-	9	6.0E-06	6.9E-06	1.2E-05	0.50	-	10	1.1E-05	1.3E-05	2.2E-05	0.98	-
	FRFT	6	7.7E-06	9.0E-06	1.8E-05	0.09	5	5	9.0E-06	1.1E-05	1.7E-05	0.05	9	5	3.9E-06	4.6E-06	8.4E-06	0.05	19
	COS	129	3.5E-06	4.4E-06	9.7E-06	0.40	1	54	6.1E-06	6.8E-06	1.1E-05	0.17	3	15	7.2E-06	7.3E-06	8.3E-06	0.05	22
	FTBS	67	3.8E-06	4.4E-06	9.6E-06	0.07	8	30	1.1E-05	1.1E-05	1.1E-05	0.03	16	30	2.2E-06	2.4E-06	4.2E-06	0.03	31
1E-06	FFT	11	1.5E-06	1.7E-06	2.6E-06	2.03	-	10	2.0E-06	2.2E-06	3.4E-06	1.00	-	12	8.5E-07	9.9E-07	1.8E-06	4.29	-
	FRFT	7	2.0E-06	2.3E-06	3.6E-06	0.17	12	7	6.1E-07	7.0E-07	1.1E-06	0.18	6	6	9.8E-07	1.1E-06	2.1E-06	0.09	47
	COS	215	2.8E-07	3.6E-07	9.2E-07	0.84	2	69	4.4E-07	5.1E-07	8.4E-07	0.21	5	19	2.2E-07	2.3E-07	2.8E-07	0.06	73
	FTBS	180	2.8E-07	3.5E-07	9.6E-07	0.17	12	50	7.0E-07	7.0E-07	7.6E-07	0.05	21	40	2.9E-07	3.4E-07	4.9E-07	0.04	103
1E-07	FFT	13	1.3E-07	1.5E-07	2.4E-07	8.17	-	13	4.7E-08	5.4E-08	9.2E-08	7.95	-	14	6.8E-08	7.7E-08	1.3E-07	16.60	-
	FRFT	9	1.3E-07	1.6E-07	2.9E-07	0.71	12	9	4.4E-08	5.1E-08	9.6E-08	0.68	12	8	5.3E-08	6.3E-08	1.3E-07	0.34	48
	COS	331	2.5E-08	3.5E-08	9.9E-08	1.06	8	85	4.1E-08	4.7E-08	7.8E-08	0.27	30	21	2.0E-08	2.2E-08	3.1E-08	0.07	245
	FTBS	439	2.3E-08	3.0E-08	9.9E-08	0.42	20	74	4.4E-08	5.1E-08	9.1E-08	0.07	113	80	8.6E-08	8.7E-08	9.7E-08	0.08	208
1E-08	FFT	15	1.9E-08	2.2E-08	4.3E-08	32.70	-	14	1.5E-08	1.9E-08	3.8E-08	16.29	-	15	2.6E-08	3.0E-08	4.6E-08	32.68	-
	FRFT	11	8.5E-09	1.0E-08	2.3E-08	2.70	12	10	2.0E-08	2.2E-08	3.2E-08	1.36	12	9	1.5E-08	1.7E-08	3.3E-08	0.68	48
	COS	500	2.9E-09	3.9E-09	9.6E-09	1.54	21	99	4.9E-09	5.5E-09	9.3E-09	0.30	54	30	1.7E-08	1.8E-08	3.0E-08	0.09	354
	FTBS	1010	1.9E-09	2.9E-09	9.7E-09	0.94	35	122	2.7E-09	3.1E-09	5.3E-09	0.12	140	150	9.1E-09	9.1E-09	9.9E-09	0.14	227

Option pricing parameters under KoBoL (CGMY) model (to 7 decimal places):

Low. $S_0 = 1, \lambda_+ = 22.9666295, \lambda_- = -6.9666295, c_+ = c_- = 5, \nu = 0.25, r = q = 0, T = 0.1.$

Bench. $S_0 = 1, \lambda_+ = 22.9666295, \lambda_- = -6.9666295, c_+ = c_- = 5, \nu = 0.5, r = q = 0, T = 0.25.$

High. $S_0 = 1, \lambda_+ = 7.6353590, \lambda_- = -4.3295739, c_+ = c_- = 5, \nu = 0.5, r = q = 0, T = 1.$

Upper integration limits for FFT and FRFT chosen optimally to minimize pricing error.

Calculation time reported for pricing 31 options with $K = 0.85$ to 1.15 in steps of 0.01.

Table F.9: Comparison to tables of Kristensen and Mele (2011). Heston parameters: $K = 1000, V_0 = 0.5172, \kappa = 0.1465, \theta = 0.5172,$ $\sigma = 0.5786, \rho = -0.0243, r = q = 0$. Calculation time reported for 11 options. KM denotes the method of Kristensen and Mele (2011).

S_0	Exact	KM		FTBS (50)		FTBS (100)	
	Delta	Delta	Error	Delta	Error	Delta	Error
950	44.2794	44.2819	0.0025	44.2792	0.0002	44.2794	0.0000
960	46.2918	46.2940	0.0022	46.2917	0.0001	46.2918	0.0000
970	48.2928	48.2945	0.0017	48.2928	0.0000	48.2928	0.0000
980	50.2776	50.2788	0.0012	50.2775	0.0001	50.2776	0.0000
990	52.2414	52.2421	0.0007	52.2414	0.0000	52.2414	0.0000
1000	54.1800	54.1801	0.0001	54.1800	0.0000	54.1800	0.0000
1010	56.0893	56.0890	-0.0003	56.0893	0.0000	56.0893	0.0000
1020	57.9657	57.9649	-0.0008	57.9658	0.0001	57.9657	0.0000
1030	59.8058	59.8046	-0.0012	59.8059	0.0001	59.8058	0.0000
1040	61.6066	61.6049	-0.0017	61.6067	0.0001	61.6066	0.0000
1050	63.3654	63.3633	-0.0021	63.3655	0.0001	63.3654	0.0000
		Total Time		0.02ms		0.03ms	

(a) Comparison to Table 1 (Delta A.1).

S_0	Exact	KM		FTBS (50)	
	Gamma	Gamma	Error	Gamma	Error
950	0.2016	0.2016	0.0000	0.2016	0.0000
960	0.2008	0.2007	-0.0001	0.2008	0.0000
970	0.1994	0.1993	-0.0001	0.1994	0.0000
980	0.1975	0.1975	0.0000	0.1975	0.0000
990	0.1952	0.1951	-0.0001	0.1952	0.0000
1000	0.1925	0.1924	-0.0001	0.1925	0.0000
1010	0.1893	0.1893	0.0000	0.1893	0.0000
1020	0.1859	0.1858	-0.0001	0.1859	0.0000
1030	0.1821	0.1820	-0.0001	0.1821	0.0000
1040	0.1780	0.1780	0.0000	0.1780	0.0000
1050	0.1737	0.1737	0.0000	0.1737	0.0000
	Total Time			0.02ms	

(b) Comparison to Table 1 (Gamma B.1).

V(0)	Exact	KM		FTBS (50)		FTBS (100)	
	Delta	Delta	Error	Delta	Error	Delta	Error
0.1	51.9512	51.9534	0.0022	51.9512	0.0000	51.9512	0.0000
0.2	52.6614	52.6621	0.0007	52.6614	0.0000	52.6614	0.0000
0.3	53.2189	53.2193	0.0004	53.2189	0.0000	53.2189	0.0000
0.4	53.6929	53.6932	0.0003	53.6929	0.0000	53.6929	0.0000
0.5	54.1121	54.1123	0.0002	54.1122	0.0001	54.1121	0.0000
0.6	54.4920	54.4921	0.0001	54.4920	0.0000	54.4920	0.0000
0.7	54.8416	54.8418	0.0002	54.8417	0.0001	54.8416	0.0000
0.8	55.1673	55.1674	0.0001	55.1673	0.0000	55.1673	0.0000
0.9	55.4732	55.4733	0.0001	55.4733	0.0001	55.4732	0.0000
1.0	55.7625	55.7626	0.0001	55.7626	0.0001	55.7625	0.0000
1.1	56.0376	56.0377	0.0001	56.0377	0.0001	56.0376	0.0000
		Total Time:		0.84ms		1.63ms	

(c) Comparison to Table 1 (Delta A.2).

V_0	Exact	KM		FTBS (50)	
	Gamma	Gamma	Error	Gamma	Error
0.1	0.4464	0.4434	-0.0030	0.4464	0.0000
0.2	0.3123	0.3118	-0.0005	0.3123	0.0000
0.3	0.2539	0.2538	-0.0001	0.2539	0.0000
0.4	0.2193	0.2193	-0.0000	0.2194	0.0000
0.5	0.1958	0.1957	-0.0001	0.1958	0.0000
0.6	0.1784	0.1784	-0.0000	0.1784	0.0000
0.7	0.1650	0.1649	-0.0001	0.1650	0.0000
0.8	0.1541	0.1541	0.0000	0.1541	0.0000
0.9	0.1451	0.1451	0.0000	0.1451	0.0000
1.0	0.1375	0.1375	0.0000	0.1375	0.0000
1.1	0.1309	0.1309	-0.0000	0.1309	0.0000
	Total Time			0.84ms	

(d) Comparison to Table 1 (Gamma B.2).

References

Andricopoulos, A. D., Widdicks, M., Duck, P. W., and Newton, D. P. Universal option valuation using quadrature methods. *Journal of Financial Economics*, 67 (2003), 447–71.

Bakshi, G., and Madan, D. Spanning and derivative-security valuation. *Journal of Financial Economics*, 55 (2000), 205–38.

Bates, D. S. Jumps and stochastic volatility: Exchange rate processes implicit in Deutsche mark options. *Review of Financial Studies*, 9 (1996), 69–107.

Black, F., and Scholes, M. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81 (1973), 637–54.

Bouziane, M. *Pricing interest-rate derivatives: a Fourier-transform based approach*. 607. Springer Verlag (2008).

Boyarchenko, M., and Levendorskiĭ, S. Refined and enhanced fast Fourier transform techniques, with an application to the pricing of barrier options. *Available at SSRN: http://ssrn.com/abstract=1142833*, (2008).

Boyarchenko, S. I., and Levendorskiĭ, S. On rational pricing of derivative securities for a family of non-Gaussian processes. *Preprint 98/7, Institut für Mathematik, Universität Potsdam*, (1998).

Boyarchenko, S. I., and Levendorskiĭ, S. Option pricing for truncated Lévy processes. *International Journal of Theoretical and Applied Finance*, 3 (2000), 549–52.

Boyarchenko, S. I., and Levendorskiĭ, S. *Non-Gaussian Merton-Black-Scholes Theory*. World Scientific Singapore (2002).

Boyarchenko, S. I., and Levendorskiĭ, S. New efficient versions of Fourier transform method in applications to option pricing. *Available at SSRN: http://ssrn.com/abstract=1846633*, (2011).

Cai, N., and Kou, S. G. Option pricing under a mixed-exponential jump diffusion model. *Management Science*, 57 (2011), 2067–80.

Carr, P., and Madan, D. Option valuation using the fast Fourier transform. *Journal of Computational Finance*, 2 (1999), 61–73.

Carr, P., and Madan, D. Saddlepoint methods for option pricing. *Journal of Computational Finance*, 13 (2010), 49–61.

Carr, P. P., Geman, H., Madan, D., and Yor, M. The fine structure of asset returns: An empirical investigation. *The Journal of Business*, 75 (2002), 305–33.

Chourdakis, K. Option pricing using the fractional FFT. *Journal of Computational Finance*, 8 (2005), 1–18.

Cont, R., and Tankov, P. *Financial modelling with jump processes* volume 2. Chapman Hall (2004).

De Boor, C. *A practical guide to splines* volume 27. Springer Verlag (2001).

Duffie, D., Filipovi, D., and Schachermayer, W. Affine processes and applications in finance. *The Annals of Applied Probability*, 13 (2003), 984–1053.

Duffie, D., Pan, J., and Singleton, K. Transform analysis and asset pricing for affine jump-diffusions. *Econometrica*, 68 (2000), 1343–76.

Dufresne, D., Garrido, J., and Morales, M. Fourier inversion formulas in option pricing and insurance. *Methodology and Computing in Applied Probability*, 11 (2009), 359–83.

Eberlein, E. Application of generalized hyperbolic Lévy motion to finance. *In O.E. Barndorff-Nielsen, T. Mikosch, and S. Resnick (Eds.), Lévy Processes Theory and Applications*, (2001), 319–37.

Eberlein, E., Glau, K., and Papapantoleon, A. Analysis of Fourier transform valuation formulas and applications. *Applied Mathematical Finance*, 17 (2010), 211–40.

Eberlein, E., Papapantoleon, A., and Shiryaev, A. N. On the duality principle in option pricing: semimartingale setting. *Finance and Stochastics*, 12 (2008), 265–92.

Fang, F., and Oosterlee, C. W. A novel pricing method for European options based on Fourier-cosine series expansions. *SIAM Journal on Scientific Computing*, 31 (2008), 826–48.

Frigo, M., and Johnson, S. G. FFTW: An adaptive software architecture for the FFT. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on* (pp. 1381–4). IEEE volume 3 (1998).

Gaffney, P., and Powell, M. Optimal interpolation. *Numerical Analysis*, 506 (1976), 90–9.

Gaffney, P. W. To compute the optimal interpolation formula. *Mathematics of Computation*, 32 (1978), 763–77.

Haslip, G. G., and Kaishev, V. K. Lookback option pricing using the Fourier transform B-spline method. *Available at SSRN: http://ssrn.com/abstract=2306729*, (2013).

Heston, S. L. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Review of Financial Studies*, 6 (1993), 327–43.

Ignatov, Z. G., and Kaishev, V. K. A probabilistic interpretation of multivariate B-splines and some applications. *Serdica*, 15 (1989), 91–9.

Ingber, L. Very fast simulated re-annealing. *Mathematical and Computer Modelling*, 12 (1989), 967–73.
Innocentis, M. D. Fast calculation of prices and sensitivities of European options under variance gamma. *Available at SSRN: http://ssrn.com/abstract=1916020*, (2011).
Jacod, J., and Shiryaev, A. N. *Limit theorems for stochastic processes* volume 288. Springer-Verlag, Berlin (2003).
Kaishev, V. K. Lévy processes induced by Dirichlet (B-) splines: modelling multivariate asset price dynamics. *Mathematical Finance*, 23 (2013), 217–47.
Kaishev, V. K., Dimitrova, D., Haberman, S., and Verrall, R. *'Geometrically Designed, Variable Knot Regression Splines: Variation Diminishing Optimality of Knots', Statistical Research Paper 29*. Cass Business School, ISBN 1-905752-03-2, available at: http://www.cass.city.ac.uk/__data/assets/pdf_file/0018/372 (2006).
Kimmel, R. et al. Maximum likelihood estimation of stochastic volatility models. *Journal of Financial Economics*, 83 (2007), 413–52.
Kou, S. G. A jump-diffusion model for option pricing. *Management Science*, 48 (2002), 1086–101.
Kristensen, D., and Mele, A. Adding and subtracting Black-Scholes: A new approach to approximating derivative prices in continuous-time models. *Journal of Financial Economics*, 102 (2011), 390–414.
Küchler, U., and Tappe, S. Bilateral gamma distributions and processes in financial mathematics. *Stochastic Processes and their Applications*, 118 (2008), 261–83.
Lee, R. W. Option pricing by transform methods: extensions, unification and error control. *Journal of Computational Finance*, 7 (2004), 51–86.
Levendorskiĭ, S. Efficient pricing and reliable calibration in the Heston model. *Available at SSRN: http://ssrn.com/abstract=1979225*, (2012).
Levendorskiĭ, S., and Xie, J. Fast pricing and calculation of sensitivities of out-of-the-money European options under Lévy processes. *Journal of Computational Finance*, 15 (2012), 71–133.
Lewis, A. Option valuation under stochastic volatility. *Option Valuation under Stochastic Volatility*, (2000).
Lewis, A. A simple option formula for general jump-diffusion and other exponential Lévy processes. *Unpublished working paper*, (2001).
Lipton, A. The vol smile problem. *Risk magazine*, 15 (2002), 61–6.
Lord, R., Fang, F., Bervoets, F., and Oosterlee, C. W. A fast and accurate FFT-based method for pricing early-exercise options under Lévy processes. *SIAM Journal on Scientific Computing*, 30 (2009), 1678–705.
Lord, R., and Kahl, C. Optimal Fourier inversion in semi-analytical option pricing. *Journal of Computational finance*, 10 (2007), 1–30.
Madan, D. B., Carr, P. P., and Chang, E. C. The variance gamma process and option pricing. *European Finance Review*, 2 (1998), 79–105.
Madan, D. B., and Seneta, E. The variance gamma (VG) model for share market returns. *Journal of business*, (1990), 511–24.
Marsden, M. Quadratic spline interpolation. *American Mathematical Society*, 80 (1974).
Merton, R. C. Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics*, 3 (1976), 125–44.
Micchelli, C., Rivlin, T., and Winograd, S. The optimal recovery of smooth functions. *Numerische Mathematik*, 26 (1976), 191–200.
O'Sullivan, C. Path dependant option pricing under Lévy processes. In *EFA 2005 Moscow Meetings Paper* (2005).
Petrella, G. An extension of the Euler Laplace transform inversion algorithm with applications in option pricing. *Operations Research Letters*, 32 (2004), 380–9.
Quittard-Pinon, F., and Randrianarivony, R. How to price efficiently European options in some geometric Lévy processes models? *International Journal of Business*, 13 (2010), 301–14.
Schoutens, W., Simons, E., and Tistaert, J. Model risk for exotic and moment derivatives. *Exotic Option Pricing and Advanced Lévy Models*, (2005), 67–98.
Stein, E. M., and Stein, J. C. Stock price distributions with stochastic volatility: an analytic approach. *Review of Financial Studies*, 4 (1991), 727–52.
Vandevender, W. H., and Haskell, K. H. The SLATEC mathematical subroutine library. *ACM SIGNUM Newsletter*, 17 (1982), 16–21.