A General Framework for Pricing Asian Options under Stochastic Volatility on Parallel Architectures

Stefania Corsaro[a], Ioannis Kyriakou[b], Daniele Marazzina[c,*], Zelda Marino[a]

[a] *Dipartimento di Studi Aziendali e Quantitativi, Università di Napoli Parthenope, via Generale Parisi 13, I-80132, Italy*
[b] *Cass Business School, City, University of London, 106 Bunhill Row, London EC1Y 8TZ, UK*
[c] *Dipartimento di Matematica, Politecnico di Milano, via Bonardi 9, I-20133, Milano*

Abstract

In this paper, we present a transform-based algorithm for pricing discretely monitored arithmetic Asian options with remarkable accuracy in a general stochastic volatility framework, including affine models and time-changed Lévy processes. The accuracy is justified both theoretically and experimentally. In addition, to speed up the valuation process, we employ high-performance computing technologies. More specifically, we develop a parallel option pricing system that can be easily reproduced on parallel computers, also realized as a cluster of personal computers. Numerical results showing the accuracy, speed and efficiency of the procedure are reported in the paper.

Keywords: finance, parallel computing, option pricing, Asian option, stochastic volatility
2010 MSC: 91G20, 91G60, 65Y05

1. Introduction

Transform methods have driven much innovation in financial engineering (see examples of application for different purposes in Carr and Madan, 1999, Albanese et al., 2004, Feng and Lin, 2013, Cai et al., 2014, Sesana et al., 2014, Fusai et al., 2016, Cui et al., 2017) and enjoyed wide use in operations research, particularly in queueing theory (see, e.g., Abate and Whitt, 1992, 1995, Abate et al., 2013, Drekic and Stanford, 2001) with applications as diverse as health care and computer systems' performance evaluation, but also in probability, insurance and radio engineering. It is the scope of the current research to advance the applicability of this indispensable tool by making use of computer technology.

In this paper, we revisit the long-standing problem of valuing non-linear derivatives contingent on the arithmetic average of the underlying asset prices with general monitoring frequency over a prespecified time period. Asian options are very popular among derivatives traders and risk managers, mainly due to the averaging's smoothing of possible market manipulations near the expiry date. Averaging also provides volatility reduction and better cash-flow matching to firms facing streams of cash flows. For this, they appear in currency, energy, metal, agricultural

*Corresponding author
Email addresses: stefania.corsaro@uniparthenope.it (Stefania Corsaro), ioannis.kyriakou@city.ac.uk (Ioannis Kyriakou), daniele.marazzina@polimi.it (Daniele Marazzina), zelda.marino@uniparthenope.it (Zelda Marino)

and freight markets and, unsurprisingly, represent a large fraction of the options traded in these markets. Nevertheless, arithmetic averages see wider application in many fields of finance, such as in project valuation (see Zahra and Reza, 2012), optimal capacity planning under average demand uncertainty (see Driouchi et al., 2006), stock-swap merger proposals (see Officer, 2006), technical analysis and algorithmic trading (see Zhu and Zhou, 2009, Kim, 2007).

There is a large body of literature on Asian options; examples include Sesana et al. (2014), Cai et al. (2015), Cui et al. (2018), Černý and Kyriakou (2011) under general Lévy and/or local volatility models. Chang and McAleer (2015) study Garch-like models, which are closely linked with econometric analysis, and their relation to option pricing; examples of option applications include Majewski et al. (2015), but also Mercuri (2011) on Asian options. For a more detailed literature survey, we refer to Fusai and Kyriakou (2016).

In this paper, we consider a general stochastic volatility model framework with or without asset price jumps. Stochastic volatility is a salient and well-documented feature of financial assets. Empirical results of Bakshi et al. (1997) suggest that stochastic volatility is a primary factor driving option prices. Diffusion-based volatility models account for dependence in increments and long-term smiles and skews, but cannot give rise to realistic short-term implied volatility patterns; this shortcoming can be overcome by introducing jumps in the returns (see Cont and Tankov, 2004). Under such models, analytical price solutions for Asian options are approximate and scarce, including Yamazaki (2014), Zeng and Kwok (2016), and Fusai and Kyriakou (2016). Instead, we resort to a recursive transform approach, which is free from restricted closeness to the true option price but rather enjoys superior efficiency that has been evidenced, for example, by Carverhill and Clewlow (1990) and Černý and Kyriakou (2011) under simpler model constructions.

We present a parallel system aiming to enhance the computational tractability of the valuation problem, an important aspect of concern of transform techniques in operations research (e.g., see Drenovak et al., 2017). By applying the parallel system to the numerical pricing procedure, we show that it leads to fast computation which further allows the users to exploit its smooth convergence, thereby producing price results at any desired accuracy level. We also conduct a theoretical analysis of the errors due to truncation and discretization involved in the transforms.

Our parallel system is inexpensive as it comprises a Non-Uniform Memory Access machine equipped by open-source software at all levels, including the operating system, the compiler, the message-passing system and the routines for main computational kernels arising in the pricing procedure. The hardware architecture is representative of many recent high-performance computing (HPC) ones. We recognize in them a mixed-memory (distributed and shared) architecture. The processors are typically grouped by their physical location on a node, a multi-core Central Processing Unit (CPU) package, and the nodes are coupled via high-speed interconnects. We point out that modern personal computers (PCs) have multi-core processors equipped with local cache memory; thus, mixed-memory models adapt to them, too. The option pricing system can then be easily ported on parallel computers, also realized as a cluster of PCs.

The proposed parallel procedure enjoys flexibility as it is built on sophisticated asset price model dynamics reflecting the market realism, while remaining computationally tractable, hence

advancing the universality of transform methods in this direction. In particular, the method relies on knowledge of the characteristic function of the (log) price conditional on the current and future variance states, which we derive for different affine stochastic volatility models including, among others, time changed Lévy processes (Carr et al., 2003) as well as the classical Heston (1993) and Bates (1996) models. We bypass increases in computational complexity and memory allocation due to increasing dimensionality by employing HPC technologies. HPC is recognized as mandatory for the effective solution of many financial problems when complex simulations must be performed in a suitable turnaround time; see, for example, Östermark (2017) where a recursive portfolio decision system is described. In this paper, we employ *domain decomposition*, according to which the computational domain is split among the involved processes. Then, each process works on a local sub-domain, communicating with the others when required. Clearly, a good strategy should require limited communication, so as to minimize the impact of communication overhead on performance. We show that our strategy leads to an inherently parallel procedure: the grid is split in such a way that processes work concurrently on a local sub-grid most of the time. In particular, communication is not required in the transform computation, which is crucial for the sake of parallel efficiency. We show that efficiency values close to the ideal ones are feasible.

The remainder of the paper is organized as follows. Section 2 presents our financial modelling framework and main preliminary results. Section 3 develops the backward-recursive integral pricing scheme, whereas Section 4 discusses its numerical implementation and introduces the parallel system. Section 5 provides a theoretical error analysis of the scheme. Section 6 presents some numerical experiments, and Section 7 concludes the paper. Proofs and additional results are collected in appendices.

2. Stochastic volatility framework

Let $(\Omega, \mathcal{F}, \hat{P})$ be a complete probability space upon which all stochastic processes are defined. We denote by $\hat{P}$ the risk neutral probability measure.

When the risk neutral dynamics of the log-price is given by a process with independent increments, the implied volatility surface follows a deterministic evolution (see Cont and Tankov, 2004); introducing stochastic volatility can tackle this difficulty. Such models are popular among practitioners and academics as they are able to account for volatility clustering and dependence in increments, and give rise to realistic implied volatility patterns (short-term and/or long-term skews). In this work, we study models from the affine class, such as Heston (1993), Bates (1996) and Carr et al. (2003), although non-affine models (e.g., see Hagan et al., 2002) represent an interesting direction for future research.

2.1. Diffusion-based stochastic volatility

In the Heston model, the stochastic variance V follows the CIR (named after Cox, Ingersoll and Ross, Cox et al., 1985; see also Feller, 1951) mean reverting, square root diffusion model

$$dV_t = \alpha\left(\beta - V_t\right)dt + \gamma\sqrt{V_t}dW_t^V,\ V_0 = v_0, \tag{1}$$

where $(W_t^V)_{t\geq 0}$ is a standard Brownian motion. The parameters α, β, γ, v_0 are positive constants. Process (1) is continuous and non-negative. Its exact behavior near zero depends on the parameters' values: if $\gamma^2 \leq 2\alpha\beta$, the process never touches zero. The log-price under the risk neutral measure $\hat{P}$ is given by

$$dX_t = \left(r - \frac{1}{2}V_t\right)dt + \sqrt{V_t}\left(\rho dW_t^V + \sqrt{1-\rho^2}dW_t^X\right), \tag{2}$$

where r is the continuously compounded risk free interest rate and $(W_t^X)_{t\geq 0}$ a standard Brownian motion. W^V and W^X are independent. Finally, ρ is the correlation coefficient between processes V and X: $\rho < 0$ leads to the so-called leverage effect of large downward price moves being associated with upward volatility moves, and is offered as an explanation for implied volatility skews (e.g., see Cont and Tankov, 2004).

2.2. Stochastic volatility model with price jumps

Diffusion-based stochastic volatility models cannot generate sufficient asymmetry in the short-term returns, hence are unable to match the empirical short-term skews. The jump diffusion stochastic volatility model of Bates (1996) addresses this shortcoming by superimposing price jumps (see Cont and Tankov, 2004). In addition, the leverage effect and the long-term skews are achieved using correlated Brownian motions in the price and variance processes. More specifically, in this model, the risk neutral log-price is given by

$$dX_t = \left(r - \lambda(e^{k(1)}-1) - \frac{1}{2}V_t\right)dt + \sqrt{V_t}\left(\rho dW_t^V + \sqrt{1-\rho^2}dW_t^X\right) + dL_t, \tag{3}$$

where $(L_t)_{t\geq 0}$ is an independent compound Poisson process with intensity λ and Gaussian distribution of jump sizes with mean μ_L and standard deviation σ_L, hence with characteristic exponent $k(iu) = iu\mu_L - \sigma_L^2 u^2/2$. Jump sizes can be represented by any other convenient distribution with known characteristic function, such as the double exponential (Kou, 2002).

2.3. Time changed Lévy models

Carr et al. (2003) extend Lévy process models by incorporating stochastic and mean reverting volatilities, by taking, for example, a homogeneous Lévy process and subordinating it by the integrated variance. The randomness of the CIR process induces stochastic volatility, and its mean reversion induces volatility clustering. The governing equations in this case are

$$\begin{aligned} X_t &= X_0 + rt + L_{Y_t}, \\ dY_t &= V_t dt, \end{aligned} \tag{4}$$

where $(L_t)_{t\geq 0}$ denotes an independent Lévy process. The normal inverse Gaussian (NIG) and the tempered stable (CGMY, named after Carr, Geman, Madan and Yor, Carr et al., 2002) processes are among the popular Lévy choices with known characteristic exponents: for NIG, $k(iu) = iu\mu + (1 - \sqrt{1+\kappa\sigma^2u^2 - 2i\theta\kappa u})/\kappa$, where $\mu = (\sqrt{1-\kappa\sigma^2-2\theta\kappa}-1)/\kappa$; for CGMY, $k(iu) = iu\mu + C\Gamma(-Y)((M-iu)^Y - M^Y + (G+iu)^Y - G^Y)$, where $\mu = C\Gamma(-Y)(M^Y - (M-1)^Y + G^Y - (G+1)^Y)$. The double exponential jump diffusion of Kou (2002) is another plausible

model as it allows controlling the relative sizes of positive and negative jumps.[1]

In all the above described models, V is given by (1). The advantage of the CIR process is in its analytical tractability; the results we derive in the following proposition form the building block for our option price recursion in Section 3.

Proposition 1. *Assume that V evolves according to (1). Define $Z_t = X_t - X_0$. Under the measure $\hat{P}$, the conditional characteristic functions on the state of the variance process at time t, $\zeta(t,u|\upsilon_0,\upsilon) = \hat{E}(\exp(iu\int_0^t V_s ds)|V_0=\upsilon_0, V_t=\upsilon)$ and $\hat{\phi}(t,u|\upsilon_0,\upsilon) = \hat{E}\left(\exp(iuZ_t)|V_0=\upsilon_0, V_t=\upsilon\right)$, are given by*

$$\begin{aligned}\zeta(t,u|\upsilon_0,\upsilon) &= \frac{\psi(u)\sqrt{e^{-(\psi(u)-\alpha)t}}\left(1-e^{-\alpha t}\right)}{\alpha\left(1-e^{-\psi(u)t}\right)} \\ &\quad\times\exp\left\{\frac{\upsilon_0+\upsilon}{\gamma^2}\left(\frac{\alpha\left(1+e^{-\alpha t}\right)}{1-e^{-\alpha t}}-\frac{\psi(u)\left(1+e^{-\psi(u)t}\right)}{1-e^{-\psi(u)t}}\right)\right\} \\ &\quad\times\frac{I_d\left(\frac{4\psi(u)\sqrt{\upsilon_0\upsilon e^{-\psi(u)t}}}{\gamma^2\left(1-e^{-\psi(u)t}\right)}\right)}{I_d\left(\frac{4\alpha\sqrt{\upsilon_0\upsilon e^{-\alpha t}}}{\gamma^2(1-e^{-\alpha t})}\right)},\end{aligned} \tag{5}$$

where $\psi(u) := \sqrt{\alpha^2-2iu\gamma^2}$ (see Broadie and Kaya, 2006), $I_d(\cdot)$ is the modified Bessel function of the first kind of order $d = 2\alpha\beta/\gamma^2 - 1$, and

i) for X as in (2),

$$\begin{aligned}\hat{\phi}(t,u|\upsilon_0,\upsilon) &= \exp\left\{iu\left(\left(r-\frac{\rho\alpha\beta}{\gamma}\right)t+\frac{\rho}{\gamma}(\upsilon-\upsilon_0)\right)\right\} \\ &\quad\times\zeta\left(t,u\left(\frac{\rho\alpha}{\gamma}-\frac{1}{2}+\frac{iu\left(1-\rho^2\right)}{2}\right)\middle|\,\upsilon_0,\upsilon\right);\end{aligned} \tag{6}$$

ii) for X as in (3),

$$\begin{aligned}\hat{\phi}(t,u|\upsilon_0,\upsilon) &= \exp\left\{iu\left(\left(r-\lambda e^{k(1)}-\frac{\rho\alpha\beta}{\gamma}\right)t+\frac{\rho}{\gamma}(\upsilon-\upsilon_0)\right)+\lambda e^{k(iu)}t\right\} \\ &\quad\times\zeta\left(t,u\left(\frac{\rho\alpha}{\gamma}-\frac{1}{2}+\frac{iu\left(1-\rho^2\right)}{2}\right)\middle|\,\upsilon_0,\upsilon\right);\end{aligned} \tag{7}$$

iii) for X as in (4),

$$\hat{\phi}(t,u|\upsilon_0,\upsilon) = e^{iurt}\zeta\left(t,-ik(iu)|\,\upsilon_0,\upsilon\right). \tag{8}$$

Proof. See Appendix A.1. ■

[1]Leverage effect can be accommodated by allowing (negative) correlation between increments in the Lévy process and increments in the activity rate process. If the Lévy process is pure jump, nonzero correlation can be induced by a jump component in the activity rate process. Carr and Wu (2004) consider examples of introducing leverage via compound Poisson jumps or infinite-activity jumps. We do not consider the latter cases in this paper since, as explained in Kallsen (2006, Section 4.6), are not of a simple structure and nontrivial ordinary differential equations have to be solved numerically in order to obtain the required characteristic function $\hat{\phi}$ as in Proposition 1 or equation (B.1), raising substantially the computational cost. This leaves us with an open numerical challenge, which might not be insurmountable and we expect by continuing research to shed further light on.

Model	$\hat{E}\left(\exp(iuZ_t)\middle\vert V_0=\upsilon_0, V_t=\upsilon\right)$
Heston (1993)	$\exp\left\{iu\left(\left(r-\frac{\rho\alpha\beta}{\gamma}\right)t+\frac{\rho}{\gamma}\left(\upsilon-\upsilon_0\right)\right)\right\}$
	$\times\zeta\left(t, u\left(\frac{\rho\alpha}{\gamma}-\frac{1}{2}+\frac{iu\left(1-\rho^2\right)}{2}\right)\middle\vert \upsilon_0, \upsilon\right)$
Bates (1996)	$\exp\left\{iu\left(\left(r-\lambda e^{k(1)}-\frac{\rho\alpha\beta}{\gamma}\right)t+\frac{\rho}{\gamma}\left(\upsilon-\upsilon_0\right)\right)+\lambda e^{k(iu)}t\right\}$
	$\times\zeta\left(t, u\left(\frac{\rho\alpha}{\gamma}-\frac{1}{2}+\frac{iu\left(1-\rho^2\right)}{2}\right)\middle\vert \upsilon_0, \upsilon\right)$
Carr et al. (2003)	$e^{iurt}\zeta\left(t, -ik(iu)\middle\vert \upsilon_0, \upsilon\right)$

Table 1: Summarized characteristic functions (see Proposition 1). Notes: Heston (1993), Bates (1996), Carr et al. (2003): $\zeta(t, u|\upsilon_0, \upsilon)$ given by (5).

Characteristic functions (6)–(8) are summarized in Table 1.

Remark 1. The analytical results presented in Proposition 1 rely on the assumption of the variance evolving according to model (1). In Appendix B, we discuss the extension to general affine stochastic volatility models lacking the analytical tractability of model (1).

3. Valuation of Asian options under stochastic volatility: a recursive-integral approach

Consider an Asian option with underlying S observed over the period $[0, T]$ at the equidistant monitoring dates $t_0 = 0, t_1 = \Delta, t_2 = 2\Delta, \ldots, t_N = N\Delta = T$. Let $\{Z_j\}_{j=1}^N$ be a collection of random variables on the probability space $(\Omega, \mathcal{F}, \hat{P})$ and $\mathbb{F} = \{\mathcal{F}_j\}_{j=1}^N$ the information filtration generated by $\{Z_j\}$, with $\mathcal{F}_0$ trivial. $\{Z_j\}$ represents the set of log-returns on S so that

$$S_k = S_0 \exp\left(\sum\nolimits_{j=1}^{k} Z_j\right),\ k = 1, \ldots, N,\ S_0 > 0.$$

In what follows, we focus on the case of a floating strike put option with payoff

$$\left(\frac{1}{N+1}\sum\nolimits_{j=0}^{N} S_j - KS_N\right)^+,$$

where $K > 0$ and $(\cdot)^+$ denotes the positive part function. The floating strike option is an important contract with a helpful structure in volatile or hardly predictable markets which justifies its high demand. Fixed exchange rate linked quanto floating strike options are, among others, particularly popular and actively traded over-the-counter as they can hedge both the foreign stock price and exchange risks for domestic investors (see Chang and Tsao, 2011). In Proposition 3 we consider the case of the call-type option, whereas in Remark 2 the fixed strike option; alternatively, one may consider applying the duality result of Eberlein et al. (2008) under stochastic volatility Lévy models.

Define

$$\xi_k = \frac{1}{N+1}\left(\frac{\sum_{j=0}^{k} S_j}{S_k} - 1\right) \geq 0, \quad 0 \leq k \leq N, \tag{9}$$

from which

$$\xi_k = \left(\frac{1}{N+1}\left(\frac{\sum_{j=0}^{k-1} S_j}{S_{k-1}} - 1\right) + \frac{1}{N+1}\right) e^{-Z_k} = \left(\xi_{k-1} + \frac{1}{N+1}\right) e^{-Z_k} \tag{10}$$

where $\xi_0 = 0$. Furthermore, in virtue of (9) and for $Y := \ln \xi$, we get

$$Y_1 = \ln \frac{1}{N+1} - Z_1, \tag{11}$$

$$Y_k = \ln\left(e^{Y_{k-1}} + \frac{1}{N+1}\right) - Z_k, \quad 1 < k \leq N. \tag{12}$$

From (9)–(12), the price of the floating strike Asian put option is given under the risk neutral measure $\hat{P}$ by

$$\hat{E}\left(e^{-rT} S_N \left(\xi_N + \frac{1}{N+1} - K\right)^+\right) = S_0 E\left(\left(e^{Y_N} + \frac{1}{N+1} - K\right)^+\right), \tag{13}$$

where the equality follows from a change to measure P with the underlying asset representing the numéraire and corresponding Radon–Nikodym derivative

$$\left.\frac{dP}{d\hat{P}}\right|_t = \frac{S_t}{\hat{E}(S_t)} = e^{-rt+Z_t}. \tag{14}$$

The measure change is valid if the discounted asset price is a true martingale. Andersen and Piterbarg (2007, Proposition 2.5) show this for the Heston model, while Bernard et al. (2017) study the special case when the Feller condition is satisfied. Cont and Tankov (2004, Lemma 15.2) also show that, in the case of an independent time change, the martingale version of the time changed Lévy process is constructed simply by applying the time change to a martingale version of the Lévy process.

This change of measure reduces effectively the pricing problem to a two-dimensional one. More specifically, the process Y is not Markov as its evolution is determined also by the level of volatility. To regain a Markov process one must consider the two-dimensional process (Y, V), then (13) can be computed recursively under P.

Theorem 2. *Let*

$$c(\Delta, y_\upsilon | x_\upsilon) = \frac{2\alpha}{\gamma^2\left(1 - e^{-\alpha\Delta}\right)} e^{-\frac{2\alpha\left(x_\upsilon e^{-\alpha\Delta} + y_\upsilon\right)}{\gamma^2\left(1-e^{-\alpha\Delta}\right)}} \left(\frac{y_\upsilon}{x_\upsilon e^{-\alpha\Delta}}\right)^{\frac{d}{2}} I_d\left(\frac{4\alpha\sqrt{x_\upsilon y_\upsilon e^{-\alpha\Delta}}}{\gamma^2\left(1 - e^{-\alpha\Delta}\right)}\right), \tag{15}$$

be the CIR variance transition density (see Cox et al., 1985), and f the log-return density conditional on the variance levels x_υ and y_υ at the endpoints of the time interval $[t_{k-1}, t_k]$. Define

$$p_N(y) = \left(e^y + \frac{1}{N+1} - K\right)^+, \tag{16}$$

$$h_k(y) = \ln\left(e^y + \frac{1}{N+1}\right), \ 0 < k < N, \tag{17}$$

$$\tilde{q}_{k-1}(x, x_\upsilon, y_\upsilon) = \int_{\mathbb{R}} p_k(x - z, y_\upsilon) f(\Delta, z | x_\upsilon, y_\upsilon) dz, \ 0 < k \leq N \tag{18}$$

$$q_{k-1}(x, x_\upsilon) = \int_{\mathbb{R}^+} \int_{\mathbb{R}} p_k(x - z, y_\upsilon) g(\Delta, z, y_\upsilon | x_\upsilon) dz dy_\upsilon, \ 0 < k \leq N, \tag{19}$$

$$= \int_{\mathbb{R}^+} \tilde{q}_{k-1}(x, x_\upsilon, y_\upsilon) c(\Delta, y_\upsilon | x_\upsilon) dy_\upsilon, \tag{20}$$

$$p_{k-1}(y, y_\upsilon) = q_{k-1}(h_{k-1}(y), y_\upsilon), \ 1 < k \leq N, \tag{21}$$

where q_{k-1} is the option value function at time t_{k-1}. Moreover,

$$g(\Delta, z, y_\upsilon|x_\upsilon) = c(\Delta, y_\upsilon|x_\upsilon)f(\Delta, z|x_\upsilon, y_\upsilon)$$

is the joint density of the log-return and variance at t_k given the information at t_{k-1} from which (18)–(20) follow. Here (18) is the convolution of the function p_k with the log-return density f.

Then, the expected value (13) is given by

$$E\left(\left(e^{Y_N} + \frac{1}{N+1} - K\right)^+\right) = q_0\left(\ln\frac{1}{N+1}, \upsilon_0\right). \tag{22}$$

Proof. See Appendix A.2. ■

A few comments are in order. First, the inner integral in (19) is isolated and defined separately in (18), and (19) is re-expressed accordingly in (20) aiming to facilitate the implementation of Algorithm 1. Second, the conditional log-return density f in (18) is not known explicitly, however through Proposition 1 and the measure change (14) we gain access to the associated characteristic function

$$\phi(\Delta, u|x_\upsilon, y_\upsilon) = e^{-r\Delta}\hat{\phi}(\Delta, u - i|x_\upsilon, y_\upsilon) \tag{23}$$

under the measure P, which serves efficiently when solving the problem numerically by Fourier transform (see Algorithm 1).

Proposition 3 (Asian call option)**.** *Given the price of the floating strike put option, the call option price is given by*

$$S_0\left(q_0\left(\ln\frac{1}{N+1}, v_0\right) + K - \frac{e^{r\Delta} - e^{-rT}}{(N+1)(e^{r\Delta} - 1)}\right). \tag{24}$$

Proof. (24) follows from the basic parity relationship

$$\left(KS_N - \frac{1}{N+1}\sum_{j=0}^{N} S_j\right)^+ = \left(\frac{1}{N+1}\sum_{j=0}^{N} S_j - KS_N\right)^+ + KS_N - \frac{1}{N+1}\sum_{j=0}^{N} S_j.$$

■

Remark 2 (Fixed strike Asian option). Consider the Asian call option with payoff

$$\left(\frac{1}{N+1}\sum_{j=0}^{N} S_j - K\right)^+,$$

where K is the strike price. Define

$$\xi_k = \frac{1}{N+1}\left(\frac{\sum_{j=0}^{k} S_j - K(N+1)}{S_k}\right), \quad 0 \le k \le N, \tag{25}$$

from which

$$\xi_k = \frac{1}{N+1}\left(\frac{\sum_{j=0}^{k-1} S_j - K(N+1)}{S_{k-1}}\right)e^{-Z_k} + \frac{1}{N+1} = \xi_{k-1}e^{-Z_k} + \frac{1}{N+1},$$

where $\xi_0 = 1/(N+1) - K/S_0$. From (25), the price of the Asian call option is given by

$$\hat{E}\left(e^{-rT}S_N\xi_N^+\right) = S_0 E\left(\xi_N^+\right),$$

Algorithm 1: Backward pricing algorithm

Input: number of time steps N, grid parameters n, n_υ
Output: option price
% Preliminary computations
1: set grids;
2: compute CIR variance transition density c (15);
3: compute p_N (16);
4: compute characteristic function ϕ (23);
% Backward loop in time
for $k = N-1, 1$ **step** -1
  5: compute h_k (17);
  6: compute $\tilde{q}_k$ (18);
  7: compute q_k (20);
  8: compute p_k (21);
end for
9: compute q_0 on grid of n_υ initial variance values υ_0;
10: compute *option price* for given initial variance value.

where the second expectation is computed under the measure P. (The price of the put-type option is given via the relevant put-call parity.)

4. A parallel system for the recursive-integral implementation

In this section, we present the basis of the implementation of the pricing scheme. This serves as the building block for our parallel processing system with efficient memory allocation and execution time.

4.1. The basic numerical procedure

Equations (16)–(21) provide the outline of the pricing procedure based on backward time recursion. At each time step, evaluating q in (19) requires a convolution computation by discrete Fourier transform (DFT) combined with standard trapezoidal quadrature rule. Algorithm 1 provides a sketch of the numerical implementation. More in details:

1:
- *Set the grids in the state space.* We require grids for quantities x_υ, y_υ, x and y appearing in (16)–(21). We use the same grid $\boldsymbol{\upsilon} = \{\upsilon_{m_\upsilon}\}_{m_\upsilon=0}^{n_\upsilon-1}$ of size n_υ and spacing δ_υ for x_υ and y_υ, and $\boldsymbol{x} = \{x_m\}_{m=0}^{n-1}$ of size n and spacing δ for x and y, on the log scale[2] (for an example of grid construction, we refer to Fang and Oosterlee, 2011).
- *Set the grid in the Fourier space.* Define the grid $\boldsymbol{u} = \{(m_u - n/2)\delta_u\}_{m_u=0}^{n-1}$ of size n and spacing δ_u. The range of values of the grid $\boldsymbol{u}$ is chosen so that the tail of the absolute value of the characteristic function ϕ defined in (23) is sufficiently captured at the tails, i.e., $|\phi| \leq 10^{-\varrho}$ where $\varrho \in \mathbb{N}$ is guided by the desired precision. For an

[2]Numerical experiments in Fang and Oosterlee (2011) show that the order $d = 2\alpha\beta\gamma^{-2} - 1$ of the Bessel function I_d affects the behavior of the CIR density with respect to the decay rate in its left tail: decreasing d towards zero slows down the decay of the left tail; this is almost constant for d approaching zero from above; finally, when $d \in [-1, 0]$, it drastically increases in value. This issue can be avoided by transforming to the log-variance domain.

efficient DFT implementation we require that the Nyquist relation $\delta_u \delta = 2\pi/n$ holds, otherwise the original grids must be accordingly adjusted.

2: *Evaluate the CIR variance transition density c (15).* This is pre-computed once on the two-dimensional variance grid $(\boldsymbol{v}, \boldsymbol{v})$ and stored for use in the loop.

4: *Evaluate the characteristic function ϕ (23).* This is pre-computed once on the three-dimensional grid $(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{v})$ and stored for use in the loop.

6: *Compute $\tilde{q}_{k-1}$ on the grid $\boldsymbol{x}$ given the variance levels at the endpoints of the time interval $[t_{k-1}, t_k]$ represented by grid $(\boldsymbol{v}, \boldsymbol{v})$.* We compute the inverse DFT (IDFT)

$$\tilde{\boldsymbol{q}}_{k-1,\cdot,j_v,m_v} = \frac{\delta_u}{2\pi} e^{i\frac{n}{2}\delta_u(\boldsymbol{x}^{\mathrm{T}} - x_0)} \circ \sum_{m_u=0}^{n-1} e^{-i\frac{2\pi}{n} m_u m} \left(e^{-i\boldsymbol{u}^{\mathrm{T}} x_0} \circ \boldsymbol{P}_{k,\cdot,j_v} \circ \boldsymbol{\phi}_{\cdot,j_v,m_v} \right)_{m_u} \tag{26}$$

for each $j_v, m_v = 0, \ldots, n_v - 1$, where $\circ$ denotes the Hadamard element-wise product and

$$\boldsymbol{P}_{k,\cdot,j_v} = \delta e^{i(\boldsymbol{u} + \frac{n}{2}\delta_u) y_0} \circ \sum_{j=0}^{n-1} e^{i\frac{2\pi}{n} m_u j} \left(e^{-i\frac{n}{2}\delta_u \boldsymbol{y}} \circ \boldsymbol{p}_{k,\cdot,j_v}{}^{\mathrm{T}} \circ \boldsymbol{w} \right)_j \tag{27}$$

is the DFT of $\boldsymbol{p}_k$ on grid $\boldsymbol{u}$. $\boldsymbol{w}$ is the vector of trapezoidal weights.

7: *Compute q_{k-1} on the grid $\boldsymbol{x}$ given the variance level at time t_{k-1} represented by grid $\boldsymbol{v}$.* We compute this by standard trapezoidal quadrature rule:

$$\boldsymbol{q}_{k-1,\cdot,m_v} = \tilde{\boldsymbol{q}}_{k-1,\cdot,\cdot,m_v} \left(\boldsymbol{c}_{\cdot,m_v} \circ \boldsymbol{w}^{\mathrm{T}} \right)$$

for each $m_v = 0, \ldots, n_v - 1$.

8: *Compute p_{k-1} on $(\boldsymbol{x}, \boldsymbol{v})$.* We have that $p_{k-1}(\boldsymbol{x}, \boldsymbol{v}) = q_{k-1}(h_{k-1}(\boldsymbol{x}), \boldsymbol{v})$, hence we need to fit a cubic interpolating spline to $(\boldsymbol{x}, \boldsymbol{q}_{k-1,\cdot,m_v})$, for $m_v = 0, \ldots, n_v - 1$, with not-a-knot endpoint conditions to evaluate q_{k-1} at $h_{k-1}(\boldsymbol{x}) \subseteq \boldsymbol{x}$; for $h_{k-1}(\boldsymbol{x}) \nsubseteq \boldsymbol{x}$, we use linear extrapolation in e^x.

4.2. The parallel algorithm

To ensure efficient memory allocation and execution time, we introduce our parallel system. We start by describing the parallel architecture. This includes a Non-Uniform Memory Access (NUMA) machine equipped with Linux operating system. This hardware architecture is representative of many modern high-performance computers (see the TOP 500 list[3] of supercomputers). The processors are grouped by their physical location on a multi-core CPU package (the *node*), and the nodes are coupled via high-speed interconnects. At the machine level, we recognize a distributed-memory architecture; at the node level, the architecture is either distributed or shared-memory, or both. In a NUMA machine, each processor has its own local memory module that can be accessed directly at a highest speed. At the same time, it can also access any memory module belonging to another processor using a certain type of interconnection. Access to remote memory can be achieved by message-passing parallel programming. We choose this, hence we adopt a distributed-memory programming model. We employ the

[3] `https://www.top500.org/`

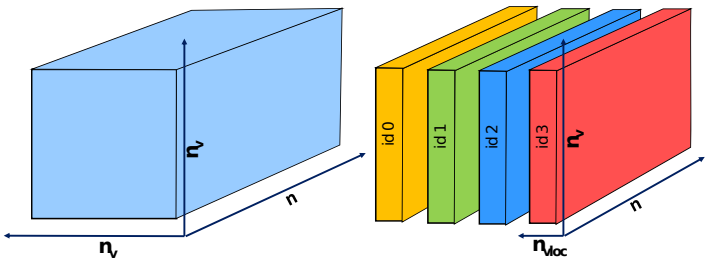

Figure 1: Parallel decomposition of the three-dimensional computational grid. Example with four processes.

Message Passing Interface (MPI) standard[4] for message passing; experimental studies on MPI latest releases show that our choice is competitive with hybrid models combining MPI and OpenMP, the native system for shared memory architectures. It is worth noting that modern PCs have multi-core processors equipped with local cache memory, and thus mixed-memory models adapt to them, too. As we explain in more detail in Section 6.2, our parallel system is based on open-source software, therefore it can be easily reproduced also on a cluster of PCs.

In what follows, we present our parallel pricing algorithm. We call *process* each processing element involved in the computation. Let *nproc* be the number of processes. For convenience, the grid sizes and *nproc* are given by powers of 2. Each process is identified by a number, the process *id*, with $id = 0, \ldots, nproc - 1$; we define the *Root* process the one identified by $id = 0$.

In Algorithm 1, the CIR transition density (see step 2) is computed on a two-dimensional grid of size $n_\upsilon \times n_\upsilon$, and the characteristic function (see step 4) on a three-dimensional grid of size $n \times n_\upsilon \times n_\upsilon$. We adopt a *domain decomposition* parallelization strategy, i.e., we split the computational domain among processes rather than computations to be performed. In particular, we split the 3-d grid $(\boldsymbol{u}, \boldsymbol{\upsilon}, \boldsymbol{\upsilon})$ of size $n \times n_\upsilon \times n_\upsilon$ among processes in such a way that each process manages a local sub-grid of size $n \times n_\upsilon^{loc} \times n_\upsilon$, where $n_\upsilon^{loc} := n_\upsilon / nproc$. As both n_υ and *nproc* are powers of 2, n_υ^{loc} is an integer. The parallelization strategy is represented in Figure 1. The grid points are independent in the sense that all the involved function values at each point depend on that point only, therefore the processes work concurrently on local sub-grids most of the time, computing their own local variables. The parallelization strategy is designed, in particular, to avoid communication for the computation of the DFT. Indeed, all the computations on the $\boldsymbol{x}$ grid are performed concurrently by processes without communication.

Functions evaluated on the 3-d grid are locally stored according to the grid decomposition represented in Figure 1; functions $\tilde{q}_k$, q_k, p_k, and c are stored in 2-d arrays, for which we use

[4] `http://www.mcs.anl.gov/research/projects/mpi/`

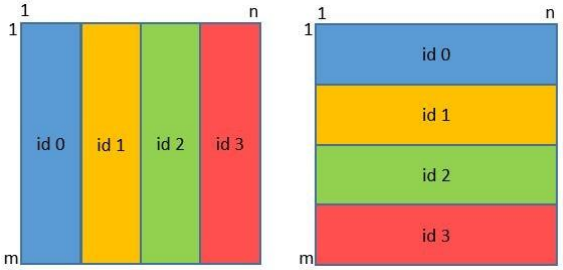

Figure 2: Left: block-column distribution. Right: block-row distribution. Example with four processes..

one-dimensional block column distribution according to the overall domain decomposition. This distribution strategy is exhibited in Figure 2, applied to a matrix of generic size $m \times n$ and $nproc = 4$: the block-column distribution is shown on the left, the block-row on the right. The block-column distribution assigns a block of $n^{loc} = \lceil n/nproc \rceil$ contiguous columns to successive processes, conceptually arranged in a $1 \times nproc$ process grid. Each process receives at most one block of columns of the matrix. Column k is stored on process $\lfloor k/n^{loc} \rfloor$. Note that if the value of $nproc$ evenly divides the value of n, as in our case, then each process owns a block of equal size. Consider step 6 of Algorithm 1. Note that (27) requires n_υ DFTs of length n. As p_k is distributed in block-column fashion, the processes compute concurrently the Fourier transforms; each process computes n_υ^{loc} DFTs of length n. Then, in (26), for each j_υ, n_υ IDFTs – one for each value m_υ – of length n are computed. Therefore, after the first stage in which the n_υ DFTs are computed, an all-to-all communication step is required to collect the DFTs. The processes exchange blocks of size $n \times n_\upsilon^{loc}$; this is the only communication required in the parallel algorithm and is performed for each monitoring date. After that, the processes compute concurrently the IDFT.

Once the time loop is completed, only process *Root* computes the solution. This step requires communication if the element of position $(-\ln(N+1), \upsilon_0)$ in q_0 (see equation 22) is not stored by process *Root*.

We sketch the parallel procedure in Algorithm 2, where local variables indicated by the superscript "*loc*" are computed on local grids, thus are different on each process, whereas variables without the "*loc*" mark are replicated by the processes, i.e., all processes independently compute and store them.

5. Error analysis

In this section, we study the numerical errors of our pricing procedure.

Proposition 4. *Let us define*

$$a_N = 1,\ a_{k-1} = a_k \widehat{\mu},\ b_N = \left(\frac{1}{N+1} - K\right)^+,\ b_{k-1} = \frac{a_{k-1}}{N+1} + b_k$$

for $k = N, \ldots, 1$, *with*

$$\widehat{\mu} = \max_{(x_\upsilon, y_\upsilon) \in \mathbb{R}_+^2} \int_{\mathbb{R}} e^{-z} f(\Delta, z | x_\upsilon, y_\upsilon) dz < +\infty.$$

Algorithm 2: Parallel backward pricing algorithm

Input: number of time steps N, grid parameters n, n_υ, number of processes $nproc$
Output: option price
% Parallel preliminary computations
`each process`:
1: set local grids;
2: compute CIR variance transition density c^{loc};
3: compute p_N;
% Parallel backward loop in time
`each process`:
for $k = N-1, 1$ **step** -1
4: compute h_k;
5: compute local DFT;
6: collect DFT;
7: compute ϕ^{loc};
8: compute $\tilde{q}_k^{loc}$;
9: compute q_k^{loc};
10: compute p_k^{loc};
end for
% Compute the id of the process which stores q_0
`each process`:
11: compute id_{q_0};
12: if $id_{q_0} \neq Root$, process id_{q_0} sends q_0 to $Root$;
% Process Root computes option price
`only process Root:`
13: either computes or receives q_0;
14: computes *option price*.

Then, the following bound holds:

$$p_k\left(y, y_\upsilon\right) \leq a_k e^y + b_k. \tag{28}$$

Proof. See Appendix A.3. ■

Let us define functions $\tilde{Q}_k$, Q_k, P_k as the truncated counterparts of functions $\tilde{q}_k, q_k, p_k$:

$$\begin{aligned}
P_N(y, y_\upsilon) &= p_N(y, y_\upsilon) 1_{[D_N, U_N]}(y) 1_{[d,u]}(y_\upsilon), \\
\widetilde{Q}_{k-1}(x, x_\upsilon, y_\upsilon) &= \left(\int_{\mathbb{R}} P_k(x-z, y_\upsilon) f(\Delta, z | x_\upsilon, y_\upsilon) dz \right) 1_{[\bar{D}_{k-1}, \bar{U}_{k-1}]}(x) 1_{[d,u]\times[d,u]}(x_\upsilon, y_\upsilon), \qquad (29) \\
Q_{k-1}(x, x_\upsilon) &= \int_{\mathbb{R}^+} \tilde{Q}_{k-1}(x, x_\upsilon, y_\upsilon) c(\Delta, y_\upsilon | x_\upsilon) dy_\upsilon, \qquad (30) \\
P_{k-1}(y, y_\upsilon) &= Q_{k-1}(h_{k-1}(y), y_\upsilon) 1_{[D_{k-1}, U_{k-1}]}(y),
\end{aligned}$$

with $\bar{D}_0 = \bar{U}_0 = -\ln(N+1)$,

$$D_k = \bar{D}_{k-1} - u_k, \; U_k = \bar{U}_{k-1} - d_k, \tag{31}$$

and

$$\bar{D}_k = \ln\left(e^{D_k} + \frac{1}{N+1} \right), \; \bar{U}_k = \ln\left(e^{U_k} + \frac{1}{N+1} \right). \tag{32}$$

The following theorem provides a bound for the error due to truncation to the compact intervals (d_k, u_k) and (d, u).

Theorem 5. *Let us define*

$$\begin{aligned}\tilde{a}_N &= 0,\ \widetilde{a}_{k-1} = \widetilde{a}_k\widehat{\mu} + a_k\left[F(d_k;-1) + G(u_k;-1)\right] + a_{k-1}\left[\widehat{F}(d;0) + \widehat{G}(u;0)\right],\\ \tilde{b}_N &= 0,\ \widetilde{b}_{k-1} = \widetilde{b}_k + b_k\left[F(d_k;0) + G(u_k;0) + \widehat{F}(d;0) + \widehat{G}(u;0)\right] + \frac{\widetilde{a}_{k-1}}{N+1},\end{aligned}$$

for $k = N, \ldots, 1$, *where*

$$\begin{aligned}F(z;\beta) &= \max_{(x_\upsilon,y_\upsilon)\in\mathbb{R}^2_+}\int_{-\infty}^{z} e^{\beta x} f(\Delta, x|x_\upsilon, y_\upsilon)dx,\\ \widehat{F}(z;\beta) &= \max_{x_\upsilon\in\mathbb{R}_+}\int_{0}^{z} e^{\beta y_\upsilon} c(\Delta, y_\upsilon|x_\upsilon)dy_\upsilon,\\ G(z;\beta) &= \max_{(x_\upsilon,y_\upsilon)\in\mathbb{R}^2_+}\int_{z}^{\infty} e^{\beta x} f(\Delta, x|x_\upsilon, y_\upsilon)dx,\\ \widehat{G}(z;\beta) &= \max_{x_\upsilon\in\mathbb{R}_+}\int_{z}^{\infty} e^{\beta y_\upsilon} c(\Delta, y_\upsilon|x_\upsilon)dy_\upsilon.\end{aligned}$$

The following bound then holds:

$$q_{k-1}(x, x_\upsilon) - Q_{k-1}(x, x_\upsilon) \le \widetilde{a}_{k-1}e^x + \widetilde{b}_{k-1} - \frac{\widetilde{a}_{k-1}}{N+1}.$$

Proof. See Appendix A.4. ■

The results of Theorem 5, coupled with

$$\lim_{d_k\to-\infty} F(d_k, \beta) = \lim_{u_k\to+\infty} G(u_k, \beta) = 0, \quad \beta = 0, -1, \quad \text{for any } k,$$

and

$$\lim_{d\to 0}\widehat{F}(d, 0) = \lim_{u\to+\infty}\widehat{G}(u, 0) = 0,$$

allow us to control the error due to domains' truncation

$$q_0\left(\ln\frac{1}{N+1}, v_0\right) - Q_0\left(\ln\frac{1}{N+1}, v_0\right)$$

by decreasing d_k and d and increasing u_k and u.

Next, we study the integration error when approximating the integrals by the trapezoidal quadrature rule.

Theorem 6. *Let us assume that a constant* C *exists such that*

$$\begin{aligned}&\textstyle\int_{\mathbb{R}} e^{-\beta z}\partial^{(i)}_{x_\upsilon} f(\Delta, z|x_\upsilon, y_\upsilon)dz < C, \qquad \int_{\mathbb{R}} e^{-\beta z}\partial^{(i)}_{y_\upsilon} f(\Delta, z|x_\upsilon, y_\upsilon)dz < C,\\ &\textstyle\int_{\mathbb{R}}\partial^{(i)}_{x_\upsilon} c(\Delta, y_\upsilon|x_\upsilon)dz < C, \qquad \int_{\mathbb{R}} e^{-z} f(\Delta, z|x_\upsilon, y_\upsilon)dz < C,\end{aligned}$$

where $\beta \in \{0, -1\}$, $i \in \{1, 2\}$, *and* $\partial^{(i)}_x$ *denotes the* i*-th order derivative with respect to* x. *Moreover, assume that*

$$|\partial^{(i)}_z f(\Delta, z|x_\upsilon, y_\upsilon)| < \infty, \quad |\partial^{(i)}_{y_\upsilon} c(\Delta, y_\upsilon|x_\upsilon)| < \infty,\ i = 1, 2. \tag{33}$$

Model parameter sets	
NIG+CIR	$\alpha = 3.99, \beta = 0.014, \gamma = 0.27, v_0 = 0.008836,$ $k = 0.00294, \theta = -11.00604, \sigma = 0.84059$
CGMY+CIR	$\alpha = 3.99, \beta = 0.014, \gamma = 0.27, v_0 = 0.008836,$ $C = 15.6840, G = 10.2115, M = 43.1510, Y = 0.8$
Heston	$\alpha = 1, \beta = 0.09, \gamma = 1, \rho = -0.3, v_0 = 0.09$
Bates	$\alpha = 3.99, \beta = 0.014, \gamma = 0.27, \rho = -0.79, v_0 = 0.008836,$ $\mu_L = -0.12, \sigma_L = 0.15, \lambda = 0.11$

Table 2: Model parameter sets.

n_v \ N	4	8	16	32
2^8	1.592924	1.648851	1.680491	1.695505
2^9	1.592924	1.648851	1.680466	1.689633
2^{10}	1.592924	1.648851	1.680466	1.689633

Table 3: Asian option prices under the NIG+CIR model computed using Algorithm 2 for increasing number of grid points $n_v = 2^8, 2^9, 2^{10}$ in the variance dimension, with $n = 2^{11}$ held fixed, for given number of monitoring dates $N = 4, 8, 16, 32$. Model parameters: see Table 2.

Then, the error from approximating integrals (29)–(30) by the trapezoidal rule with n and n_v points is of the order $O(n^{-2})$ and $O(n_v^{-2})$, respectively.

Proof. See Appendix A.5. ■

6. Numerical results

In this section, we study the performance of our pricing algorithm based on numerical results from tests on accuracy and parallel efficiency.

6.1. Accuracy of pricing methodology

For the sake of illustration, we consider a floating strike Asian option with time to maturity $T = 1$ under the NIG+CIR model, i.e., the NIG model with integrated CIR time change (see Section 2.3). We assume an initial price of the underlying $S_0 = 100$ and a risk free interest rate $r = 0.04$. Numerical results are presented in Tables 3–4, whereas Figure 3 and Table 5 exhibit summarized results for additional model specifications. For the purposes of our experiments, we use the stochastic volatility parameter values reported in Table 2 taken from Broadie and Kaya (2006), Andersen (2008) and Glasserman and Kim (2011), which correspond to different market calibrations and are described as challenging and intended to reflect parameter values commonly used in practice.

We investigate the impact on precision of successive grid refinements, i.e., increasing number of grid points n and n_v, for different number of monitoring dates $N = 4, 8, 16, 32$. We find that increasing n_v from 2^9 to 2^{10}, or higher, has negligible effect on precision as shown in Table 3. Moreover, Table 4 shows that increasing $n = 2^9, \ldots, 2^{14}$, with $n_v = 2^9$ fixed, results in notable precision improvement of the computed prices using our algorithm that fall eventually within the 95% confidence intervals estimated using Monte Carlo simulation.

Our numerical analysis suggests a quadratic convergence, as expected in view of Theorem 6. In Figure 3, we present on a log-log scale the convergence patterns of the computed option

n \ N	4	8	16	32
2^9	1.585	1.689	2.069	1.016
2^{10}	1.591	1.648	1.707	1.766
2^{11}	1.5929	1.6488	1.6804	1.6896
2^{12}	1.59326	1.64925	1.68066	1.69742
2^{13}	1.59335	1.64935	1.68078	1.69754
2^{14}	1.59337	1.64938	1.68081	1.69757
Conf. interval	(1.5932, 1.5934)	(1.6492, 1.6494)	(1.6807, 1.6808)	(1.6974, 1.6976)

Table 4: Asian option prices under the NIG+CIR model computed using Algorithm 2 for increasing number of grid points $n = 2^9, \ldots, 2^{14}$, with $n_v = 2^9$ held fixed, for given number of monitoring dates $N = 4, 8, 16, 32$. Also reported 95% confidence intervals for Monte Carlo price estimates based on 10^5 simulations and variance reduction using the lower bound of Fusai and Kyriakou (2016) as control variate. Model parameters: see Table 2.

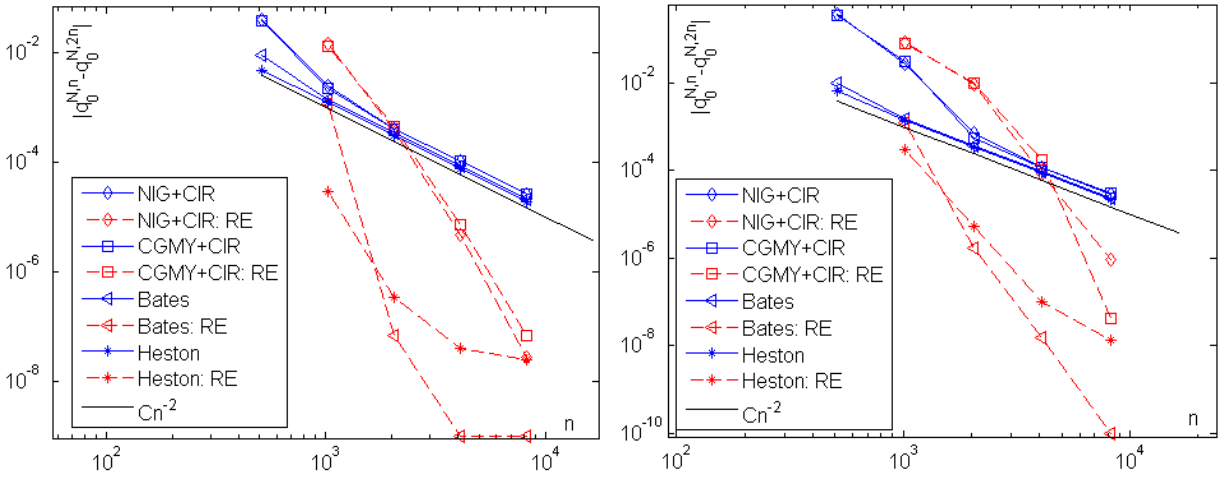

Figure 3: Convergence patterns of computed option prices under different model specifications using Algorithm 2 for increasing number of grid points $n = 2^9, \ldots, 2^{14}$, with $n_v = 2^9$ held fixed, and number of monitoring dates $N = 8$ (left plot), 16 (right plot). Price patterns resulting from Richardson extrapolation (RE) also presented. Model parameters: see Table 2.

prices under the Heston, Bates, NIG+CIR and CGMY+CIR models. We observe that smoothly diminishing error patterns are preserved under different model specifications, confirming the quadratic convergence under all models. Therefore, the following result can be applied: let $p_0^{N,n}$ be the numerical price of an option with N monitoring dates corresponding to a grid of size n; then, for sufficiently large n, we observe that $p_0^{N,n} - p_0^{N,2n} = 4(p_0^{N,2n} - p_0^{N,4n})$. Given this, we are able to accelerate the convergence rate of the price sequence $\{p_0^{N,n}\}_n$ using standard Richardson extrapolation $(0.5^2 p_0^{N,n} - p_0^{N,2n})/(0.5^2 - 1)$ (e.g., see Quarteroni et al., 2010), leading to high accuracies, e.g., at least 6 decimal digits, as shown in Table 5. The effect of Richardson extrapolation on the improvement of the accuracy is also evident in Figure 3.

In addition, we compare our option prices with relevant methods from the literature, in particular, the approximations of Zeng and Kwok (2016) and Fusai and Kyriakou (2016), and the very effective Monte Carlo method with the lower bound of the latter used as control variate. To this end, we adopt the same parameters as in Zeng and Kwok (2016) for the NIG+CIR model. From the reports in Table 6 it becomes obvious that, by increasing n, we can achieve accuracy of 5 decimal places in 2.5 to 13 seconds, depending on the number of

$(n, 2n)$	RE prices			
	NIG+CIR	CGMY+CIR	Bates	Heston
$(2^9, 2^{10})$	1.634	1.615	2.008	4.0529
$(2^{10}, 2^{11})$	1.649	1.628	2.00689	4.05299
$(2^{11}, 2^{12})$	1.64939	1.62899	2.006895	4.052992
$(2^{12}, 2^{13})$	1.649393	1.628997	2.006895	4.052992
$(2^{13}, 2^{14})$	1.649393	1.628997	2.006895	4.052992
Conf. interval	$(1.6492, 1.6494)$	$(1.6288, 1.6290)$	$(2.0068, 2.0069)$	$(4.0524, 4.0535)$

Table 5: Asian option prices under the NIG+CIR, CGMY+CIR, Bates and Heston models by Richardson extrapolation (RE) of pairs of option prices corresponding to number of grid points $(n, 2n)$ computed using Algorithm 2, with $n_\upsilon = 2^9$ held fixed, and $N = 8$ dates. Confidence intervals: refer to notes of Table 4. Model parameters: see Table 2.

	RE prices			
N / $(n, 2n)$	4	8	16	32
$(2^8, 2^9)$	2.105 (1.3)	2.158 (2.1)	2.190 (3.8)	2.207 (6.9)
$(2^9, 2^{10})$	2.10527 (2.5)	2.15886 (4.1)	2.19016 (7.1)	2.20714 (13)
$(2^{10}, 2^{11})$	2.105269 (5.1)	2.158865 (8.3)	2.190160 (15)	2.207132 (28)
$(2^{11}, 2^{12})$	2.1052692 (10)	2.1588649 (17)	2.1901597 (31)	2.2071324 (58)
Lower bound	2.105161	2.158767	2.190065	2.207046
Conf. interval	$(2.10526, 2.10527)$	$(2.15886, 2.15887)$	$(2.19015, 2.19016)$	$(2.20713, 2.20714)$

Table 6: Asian option prices for varying monitoring dates $N = 4, 8, 16, 32$ under the NIG+CIR model computed using Algorithm 2 and improved by Richardson extrapolation (RE) based on paired prices corresponding to number of grid points $(n, 2n)$ with $n_\upsilon = 2^9$ held fixed (computing times in parentheses correspond to our parallel strategy with 64 processes, see Section 6.2); lower bound of Fusai and Kyriakou (2016) (of equal accuracy to Zeng and Kwok, 2016); 95% confidence intervals based on 10^6 simulations and variance reduction using the lower bound as control variate. Model parameters: $\alpha = \beta = \gamma = \upsilon_0 = 1, k = 0.01, \theta = -0.5, \sigma = 0.1$ (Zeng and Kwok, 2016). Other parameters: $S_0 = 100, r = 0.01, T = 1$.

monitoring dates (we show later how this can be managed efficiently). Our method benefits from unlimited increase in precision (7 decimals, or more) subject to a reasonable computing time increase as shown in Table 6, but also yields lower accuracies of, say, 3 decimals, which could be relevant for practical applications, in 1.3 to 6.9 seconds. This flexibility of controlling the precision of the method as desired, renders it a suitable benchmark for other methods. The partially exact and bounded approximation by Zeng and Kwok (2016) and the lower bound of Fusai and Kyriakou (2016) strongly compete each other with a discrepancy of $\pm 10^{-4}$ between them and are accurate to 3 decimals, although the former is more computationally demanding, while, contrary to our method, they both suffer by restricted sharpness and proximity to the true option price. Although quite satisfactory, the Gram–Charlier expansion of Yamazaki (2014) is less accurate and slower requiring more characteristic function evaluations, hence we do not explicitly report numerical results in the interest of space. On the other hand, despite the use of control variates, Monte Carlo simulation still converges more slowly than our method yielding accurate results at 4–5 decimal places (with 95% confidence) with substantial excess computational cost.

We conclude this part by studying the convergence of the price of the discrete to the continuous Asian option with increasing N, which we can then exploit to reduce the impact of

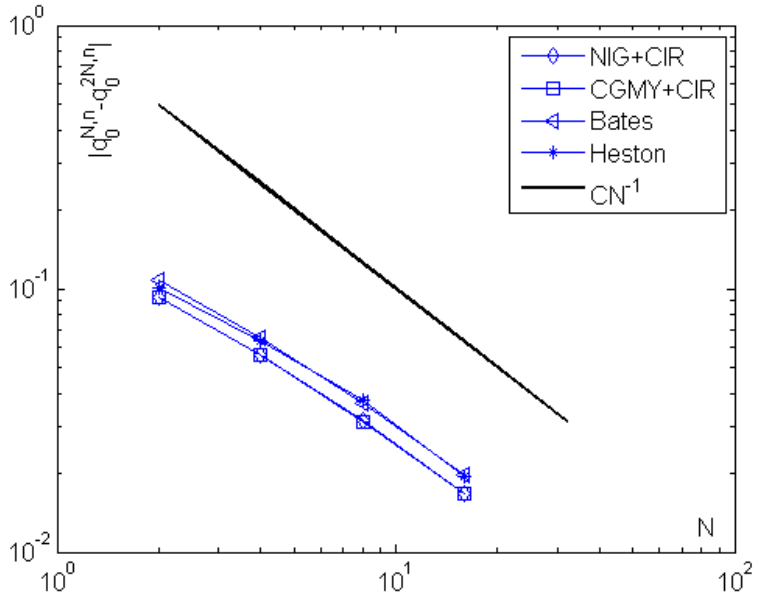

Figure 4: Convergence pattern of computed option prices using Algorithm 2 for increasing number of monitoring dates $N = 4, 8, 16, 32$, with $n = 2^{14}$ and $n_v = 2^9$ held fixed. Model parameters: see Table 2.

increasing dates on the computational speed. Figure 4 exhibits a linear convergence, coherently with the analysis of Fusai et al. (2008). This further allows the use of Richardson extrapolation as a "cost-free" way of speeding up the convergence of the discrete Asian option price $p_0^{N,n}$ to the continuous $p_0^{+\infty,n}$, i.e., $p_0^{+\infty,n} \approx (0.5p_0^{N,n} - p_0^{2N,n})/(0.5-1)$, or the discrete Asian option price for N' dates, $p_0^{N',n} \approx kp_0^{2N,n} + (1-k)p_0^{+\infty,n}$, where $k := 2N/N' < 1$. For example, under the NIG+CIR model, for $(n, 2n) = (2^{13}, 2^{14})$ (RE), $N = 16$, $N' = 50$, we compute $p_0^{+\infty,n} = 1.714351$ and $p_0^{50,n} = 1.703621$ which falls within the 95% Monte Carlo confidence interval $(1.7035, 1.7038)$.

6.2. Parallel efficiency

In this section, we analyze the performance of our parallelization strategy. First, we provide some implementation details. All the numerical tests are performed on a HP ProLiant BL460c Gen8 Server Blade, a NUMA machine consisting of 8 nodes, the *Blades*, each equipped with two quad-core Intel Xeon E5-2609. Figure 5 presents the topology of each blade.

The Linux operating system is installed on the Bladecenter. The parallel pricing algorithm is implemented in **C**; message passing is performed by means of MPI. To compute the Bessel function $I_d(z)$ (see equations 5 and 15), we use the SLATEC Common Mathematical Library, a comprehensive software library containing general purpose mathematical and statistical routines written in Fortran 77, which is freely available from NETLIB (`http://www.netlib.org/slatec/`); included routines are based on the algorithm of Amos (1986). We compute the IDFT in (26) and the DFT in (27) using `FFTW`, a **C** subroutine library for computing one- or multi-dimensional Fourier transforms of arbitrary input size and of both real and complex data (see Frigo and Johnson, 2005). It is worth noting that all software elements at different levels in our system – operating system, compiler, message-passing system, and routines for main computational kernels – are freely available, thus the parallel option pricing code that we describe in this paper can be easily used on a cluster of PCs as well as a

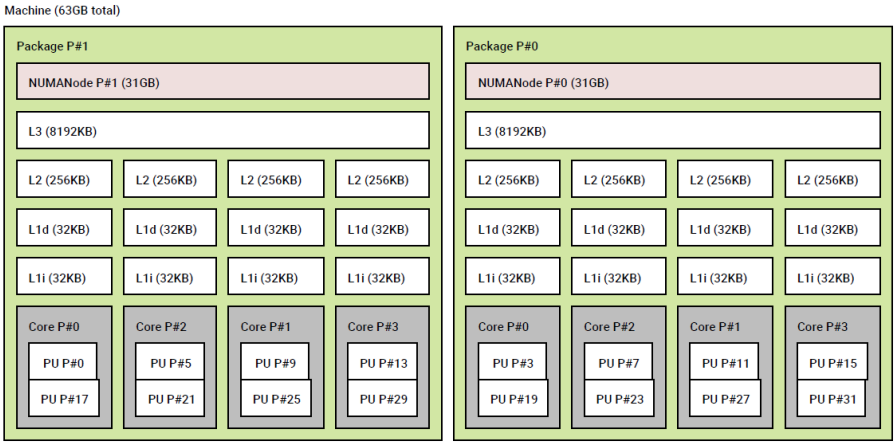

Figure 5: Architecture of a blade in the HP ProLiant BL460c Gen8 Server Blade.

multi-core PC.

We analyze the *speed-up* of the parallel code using a standard metric for evaluating a parallel software. Recalling that *nproc* is the number of processes involved in the computation, one would ideally want to solve the problem *nproc* times faster than using 1 process; the speed-up takes a value between 1 and *nproc*, providing an estimate of how effective parallelism is. This value also suggests what we should expect in terms of computing time reduction, if the number of processes were further increased. The speed-up is defined in the following way:

$$Sp_{nproc} = \frac{T_1(n, n_\upsilon)}{T_{nproc}(n, n_\upsilon)},$$

where $T_k(n, n_\upsilon)$, $k = 1, \ldots, nproc$, is the execution time with k processes on a grid of size $n \times n_\upsilon \times n_\upsilon$. The ideal value is actually never realized due to both communication time and serial parts of the algorithm that cannot be avoided. The communication overhead typically increases with the number of processes. This can be attributed to the increasing number of communications with the number of processes involved in the computations. Moreover, the single process workload decreases with *nproc*, thus the ratio between communication time and computing time increases, affecting the parallel performance.

In our experiments, we fix $n = 2^{11}$ and $n_\upsilon = 2^9$. Figure 6 shows the ideal speed-up as well as the observed one for $N = 4, 8, 16, 32$ monitoring dates. Note that our results remain unaffected by the underlying model specification. Also, the speed-up does not significantly decrease with increasing number of processes, i.e., the parallel algorithm exhibits good scalability. In fact, if we consider 64 processes, as for Table 6, the pricing procedure is 58–62 times faster than the serial code, depending on the number of monitoring dates.

7. Conclusions

In this work, we present a parallel pricing procedure for Asian options with discrete monitoring. Our method is suitable for general affine stochastic volatility models, including the Heston, Bates, and time changed Lévy models. We prove and demonstrate quadratic error convergence of the numerical scheme in the number of grid points. We also test its accuracy against a control

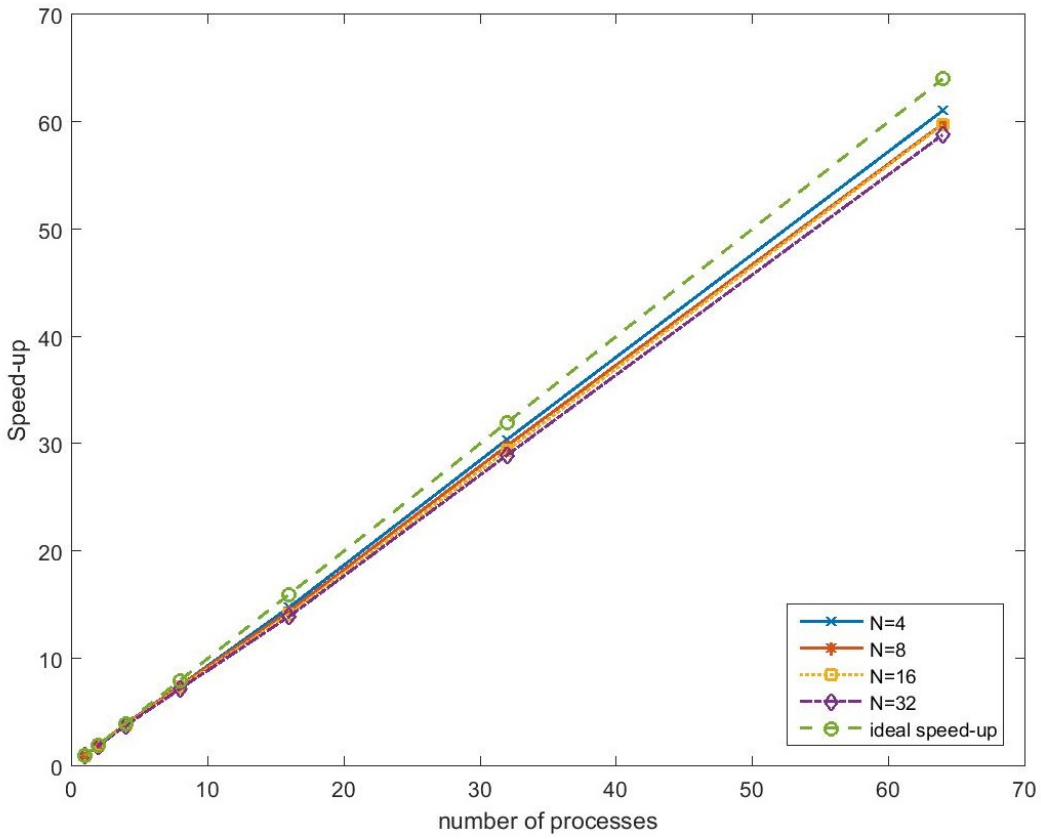

Figure 6: Speed-up for different number of monitoring dates $N = 4, 8, 16, 32$. Ideal speed-up also included.

variate Monte Carlo strategy, which is *de facto* standard for this class of models, as well as other methods from the recent literature.

To accelerate the valuation process, we develop a parallel system that is based on free software at all levels, thus can be easily ported on parallel computers, also realized as a cluster of PCs. We show that accurate results can be obtained in a suitable turnaround time. We also compute the speed-up of the parallel software in order to measure the performance of the parallel algorithm. Our results confirm the effectiveness of the adopted parallelization strategy and are unaffected by the underlying model specification.

The development of parallel strategies aiming to reduce the computing time without sacrificing for the precision of numerical procedures is an important task in finance and beyond, as discussed in the Introduction, especially when the parallel strategy is not trivial (see, for example, Corsaro et al., 2015). Future research will include, but not be limited to, the study of parallelization in multi-asset contracts.

References

Abate, J., Choudhury, G., Whitt, W., 2013. An introduction to numerical transform inversion and its application to probability models, in: Grassmann, W.K. (Ed.), Computational Probability, Chapter 8. Springer Science+Business Media, New York, pp. 257–323.

Abate, J., Whitt, W., 1992. The Fourier-series method for inverting transforms of probability distributions. Queueing Systems 10, 5–87.

Abate, J., Whitt, W., 1995. Numerical inversion of Laplace transforms of probability distributions. ORSA Journal on Computing 7, 36–43.

Albanese, C., Jackson, K., Wiberg, P., 2004. A new Fourier transform algorithm for value-at-risk. Quantitative Finance 4, 328–338.

Amos, D.E., 1986. Algorithm 644: A portable package for Bessel functions of a complex argument and nonnegative order. ACM Transactions on Mathematical Software 12, 265–273.

Andersen, L., 2008. Simple and efficient simulation of the Heston stochastic volatility model. Journal of Computational Finance 11, 1–42.

Andersen, L.B.G., Piterbarg, V.V., 2007. Moment explosions in stochastic volatility models. Finance and Stochastics 11, 29–50.

Bakshi, G., Cao, C., Chen, Z., 1997. Empirical performance of alternative option pricing models. The Journal of Finance 52, 2003–2049.

Barndorff-Nielsen, O.E., Nicolato, E., Shephard, N., 2002. Some recent developments in stochastic volatility modelling. Quantitative Finance 2, 11–23.

Barndorff-Nielsen, O.E., Shephard, N., 2001. Non-Gaussian Ornstein–Uhlenbeck-based models and some of their uses in financial economics. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63, 167–241.

Bartlett, M.S., 1938. The characteristic function of a conditional statistic. Journal of the London Mathematical Society 13, 62–67.

Bates, D.S., 1996. Jumps and stochastic volatility: exchange rate processes implicit in deutsche mark options. Review of Financial Studies 9, 69–107.

Bates, D.S., 2006. Maximum likelihood estimation of latent affine processes. Review of Financial Studies 19, 909–965.

Bernard, C., Cui, Z., McLeish, D., 2017. On the martingale property in stochastic volatility models based on time-homogeneous diffusions. Mathematical Finance 27, 194–223.

Broadie, M., Kaya, O., 2006. Exact simulation of stochastic volatility and other affine jump diffusion processes. Operations Research 54, 217–231.

Cai, N., Kou, S.G., Liu, Z., 2014. A two-sided Laplace inversion algorithm with computable error bounds and its applications in financial engineering. Advances in Applied Probability 46, 766–789.

Cai, N., Song, Y., Kou, S., 2015. A general framework for pricing Asian options under Markov processes. Operations Research 63, 540–554.

Carr, P., Geman, H., Madan, D., Yor, M., 2002. The fine structure of asset returns: An empirical investigation. The Journal of Business 75, 305–332.

Carr, P., Geman, H., Madan, D.B., Yor, M., 2003. Stochastic volatility for Lévy processes. Mathematical Finance 13, 345–382.

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

Carr, P., Wu, L., 2004. Time-changed Lévy processes and option pricing. Journal of Financial Economics 71, 113–141.

Carverhill, A., Clewlow, L., 1990. Flexible convolution. Risk 3, 25–29.

Černý, A., Kyriakou, I., 2011. An improved convolution algorithm for discretely sampled Asian options. Quantitative Finance 11, 381–389.

Chang, C.C., Tsao, C.Y., 2011. Efficient and accurate quadratic approximation methods for pricing Asian strike options. Quantitative Finance 11, 729–748.

Chang, C.L., McAleer, M., 2015. Econometric analysis of financial derivatives: An overview. Journal of Econometrics 187, 403–407.

Cont, R., Tankov, P., 2004. Financial Modelling With Jump Processes. Financial Mathematics Series, Chapman & Hall/CRC, Boca Raton.

Corsaro, S., Marazzina, D., Marino, Z., 2015. A parallel wavelet-based pricing procedure for Asian options. Quantitative Finance 15, 101–113.

Cox, J.C., Ingersoll, J. E., J., Ross, S.A., 1985. A theory of the term structure of interest rates. Econometrica 53, 385–407.

Cui, Z., Kirkby, J.L., Nguyen, D., 2017. A general framework for discretely sampled realized variance derivatives in stochastic volatility models with jumps. European Journal of Operational Research 262, 381–400.

Cui, Z., Lee, C., Liu, Y., 2018. Single-transform formulas for pricing Asian options in a general approximation framework under Markov processes. European Journal of Operational Research 266, 1134–1139.

Drekic, S., Stanford, D.A., 2001. Reducing delay in preemptive repeat priority queues. Operations Research 49, 145–156.

Drenovak, M., Ranković, V., Ivanović, M., Urošević, B., Jelic, R., 2017. Market risk management in a post-Basel II regulatory environment. European Journal of Operational Research 257, 1030–1044.

Driouchi, T., Bennett, D., Battisti, G., 2006. Capacity planning under uncertainty: an Asian option approach, in: Proceedings of 10th Annual International Conference on Real Options.

Duffie, D., Pan, J., Singleton, K., 2000. Transform analysis and asset pricing for affine jump-diffusions. Econometrica 68, 1343–1376.

Eberlein, E., Papapantoleon, A., Shiryaev, A.N., 2008. On the duality principle in option pricing: semimartingale setting. Finance and Stochastics 12, 265–292.

Fang, F., Oosterlee, C.W., 2011. A Fourier-based valuation method for Bermudan and barrier options under Heston's model. SIAM Journal on Financial Mathematics 2, 439–463.

Feller, W., 1951. Two singular diffusion problems. Annals of Mathematics 54, 173–182.

Feng, L., Lin, X., 2013. Inverting analytic characteristic functions and financial applications. SIAM Journal on Financial Mathematics 4, 372–398.

Frigo, M., Johnson, S.G., 2005. The design and implementation of FFTW3. Proceedings of the IEEE 93, 216–231.

Fusai, G., Germano, G., Marazzina, D., 2016. Spitzer identity, Wiener–Hopf factorization and pricing of discretely monitored exotic options. European Journal of Operational Research 251, 124–134.

Fusai, G., Kyriakou, I., 2016. General optimized lower and upper bounds for discrete and continuous arithmetic Asian options. Mathematics of Operations Research 41, 531–559.

Fusai, G., Marena, M., Roncoroni, A., 2008. Analytical pricing of discretely monitored Asian-style options: Theory and application to commodity markets. Journal of Banking & Finance 32, 2033–2045.

Glasserman, P., Kim, K.K., 2011. Gamma expansion of the Heston stochastic volatility model. Finance and Stochastics 15, 267–296.

Hagan, P.S., Kumar, D., Lesniewski, A.S., Woodward, D.E., 2002. Managing smile risk. Wilmott Magazine 1, 84–108.

Heston, S.L., 1993. A closed-form solution for options with stochastic volatility with applications to bond and currency options. Review of Financial Studies 6, 327–343.

Hubalek, F., Keller-Ressel, M., Sgarra, C., 2017. Geometric Asian option pricing in general affine stochastic volatility models with jumps. Quantitative Finance 17, 873–888.

Kallsen, J., 2006. A didactic note on affine stochastic volatility models, in: Kabanov, Y., Liptser, R., Stoyanov, J. (Eds.), From Stochastic Calculus to Mathematical Finance: The Shiryaev Festschrift. Springer, New York, pp. 343–368.

Kim, K., 2007. Electronic and Algorithmic Trading Technology: The Complete Guide. Complete Technology Guides for Financial Services Series, Elsevier.

Kou, S.G., 2002. A jump-diffusion model for option pricing. Management Science 48, 1086–1101.

Majewski, A.A., Bormetti, G., Corsi, F., 2015. Smile from the past: A general option pricing framework with multiple volatility and leverage components. Journal of Econometrics 187, 521–531.

Mercuri, L., 2011. Pricing Asian options in affine Garch models. International Journal of Theoretical and Applied Finance 14, 313–333.

Officer, M.S., 2006. The market pricing of implicit options in merger collars. The Journal of Business 79, 115–136.

Östermark, R., 2017. Massively parallel processing of recursive multi-period portfolio models. European Journal of Operational Research 259, 344–366.

Quarteroni, A., Sacco, R., Saleri, F., 2010. Numerical Mathematics. Texts in Applied Mathematics, Springer-Verlag Berlin Heidelberg.

Sesana, D., Marazzina, D., Fusai, G., 2014. Pricing exotic derivatives exploiting structure. European Journal of Operational Research 236, 369–381.

Yamazaki, A., 2014. Pricing average options under time-changed Lévy processes. Review of Derivatives Research 17, 79–111.

Zahra, S., Reza, R., 2012. Asian real option: New approach to project economic valuation, in: 2012 International Conference on Information Management, Innovation Management and Industrial Engineering (ICIII), pp. 475–479.

Zeng, P., Kwok, Y.K., 2016. Pricing bounds and approximations for discrete arithmetic Asian options under time-changed Lévy processes. Quantitative Finance 16, 1375–1391.

Zhu, Y., Zhou, G., 2009. Technical analysis: An asset allocation perspective on the use of moving averages. Journal of Financial Economics 92, 519–544.

Appendix A. Proofs

Appendix A.1. Proof of Proposition 1

i) From the tower property of expectations, it holds that

$$\hat{E}\left(e^{iuZ_t}|V_0=v_0,V_t=v\right)=\hat{E}\left[\hat{E}\left(e^{iuZ_t}\middle|Y_t\right)\middle|V_0=v_0,V_t=v\right]. \tag{A.1}$$

From (1), we have that

$$\int_0^t\sqrt{V_s}dW_s^V=\frac{1}{\gamma}\left(V_t-V_0-\alpha\beta t+\alpha\int_0^t V_s ds\right),$$

hence we get from (2)

$$Z_t=rt+\frac{\rho}{\gamma}\left(V_t-V_0-\alpha\beta t\right)+\left(\frac{\rho\alpha}{\gamma}-\frac{1}{2}\right)\int_0^t V_s ds+\sqrt{1-\rho^2}\int_0^t\sqrt{V_s}dW_s^X.$$

Then, by normality of Z_t conditional on the integral of the variance process $Y_t=\int_0^t V_s ds$, (A.1) equals

$$\begin{aligned}&\exp\left\{iu\left(\left(r-\frac{\rho\alpha\beta}{\gamma}\right)t+\frac{\rho}{\gamma}\left(v-v_0\right)\right)\right\}\\&\times\hat{E}\left[\exp\left\{iu\left(\frac{\rho\alpha}{\gamma}-\frac{1}{2}+\frac{iu(1-\rho^2)}{2}\right)Y_t\right\}\middle|V_0=v_0,V_t=v\right],\end{aligned}$$

from which (6) follows by virtue of (5).

ii) (7) follows from (6) by independence of the jump component in (3).

iii) By conditioning on Y_t, we recover the characteristic function of the driving Lévy process

$$\hat{E}\left(e^{iuZ_t}\middle|\, Y_t\right) = \exp\left(iurt + k(iu)Y_t\right),$$

hence, given (5), (8) follows from

$$\hat{E}\left[\exp\left(iurt + k(iu)Y_t\right)\middle|\, V_0 = v_0, V_t = v\right].$$

Appendix A.2. Proof of Theorem 2

We prove by induction on k that $E[(e^{Y_N} + 1/(N+1) - K)^+|\mathcal{F}_k] = p_k(Y_k, V_k)$ for $k = 0, \ldots, N$. Trivially the result holds for $k = N$. Suppose that $E[(e^{Y_N} + 1/(N+1) - K)^+|\mathcal{F}_{k+1}] = p_{k+1}(Y_{k+1}, V_{k+1})$ holds for arbitrary $k < N-1$. By iterated expectations,

$$\begin{aligned} E\left(\left(e^{Y_N} + \frac{1}{N+1} - K\right)^+\middle|\, \mathcal{F}_k\right) &= E\left[E\left(\left(e^{Y_N} + \frac{1}{N+1} - K\right)^+\middle|\, \mathcal{F}_{k+1}\right)\middle|\, \mathcal{F}_k\right] \\ &= E[p_{k+1}(Y_{k+1}, V_{k+1})|\mathcal{F}_k] \\ &= E\left[p_{k+1}\left(\ln\left(e^{Y_k} + \frac{1}{N+1}\right) - Z_{k+1}, V_{k+1}\right)\middle|\, \mathcal{F}_k\right], \end{aligned}$$

where the last equality follows from (12). Then

$$\begin{aligned} & E\left[p_{k+1}\left(\ln\left(e^{Y_k} + \frac{1}{N+1}\right) - Z_{k+1}, V_{k+1}\right)\middle|\, \mathcal{F}_k\right] \\ = & \int_{\mathbb{R}^+}\int_{\mathbb{R}} p_{k+1}\left(\ln\left(e^{Y_k} + \frac{1}{N+1}\right) - z, y_v\right) g(\Delta, z, y_v|x_v)\, dz\, dy_v \\ = & \; q_k\left(\ln\left(e^{Y_k} + \frac{1}{N+1}\right), V_k\right) = p_k\left(Y_k, V_k\right), \end{aligned}$$

where the last two equalities follow from (19) and (21). Therefore, by induction, we have

$$E\left[\left(e^{Y_N} + \frac{1}{N+1} - K\right)^+\right] = E\left[\left(e^{Y_N} + \frac{1}{N+1} - K\right)^+\middle|\, \mathcal{F}_0\right] = q_0\left(\ln\frac{1}{N+1}, v_0\right).$$

Appendix A.3. Proof of Proposition 4

For $k = N$, equation (16) implies $p_N(y, y_v) \leq a_N e^y + b_N$ for any y_v. Along the arguments of the proof of Theorem 3.1 in Černý and Kyriakou (2011), we proceed with proving our thesis by induction on k. Suppose that $p_k(y, y_v) \leq a_k e^y + b_k$ holds for arbitrary $k \leq N$ and any y_v. Then from (18) we have

$$\widetilde{q}_{k-1}(x, x_v, y_v) = \int_{\mathbb{R}} p_k(x - z, y_v)\, f(\Delta, z|\, x_v, y_v)\, dz \leq a_k\widehat{\mu}e^x + b_k = a_{k-1}e^x + b_k,$$

and from (20) we obtain $q_{k-1}(x, x_v) \leq a_{k-1}e^x + b_k$. Finally, from (21) we have

$$\begin{aligned} p_{k-1}(y, y_v) &\leq a_{k-1}e^{h_{k-1}(y)} + b_k = a_{k-1}\left(e^y + \frac{1}{N+1}\right) + b_k \\ &= a_{k-1}e^y + \left(\frac{a_{k-1}}{N+1} + b_k\right) = a_{k-1}e^y + b_{k-1}. \end{aligned}$$

This completes the proof.

Appendix A.4. Proof of Theorem 5

We proceed by induction on k. More specifically, assume that the following holds for arbitrary $0 < k \leq N$

$$0 \leq p_k(y, y_\upsilon) - P_k(y, y_\upsilon) \leq \tilde{a}_k e^y + \tilde{b}_k \quad \text{for} \quad y \in [D_k, U_k], \quad y_\upsilon \in [d, u]$$

(this trivially holds for $k = N$), and for $0 \leq k < N$

$$0 \leq q_k(x, x_\upsilon) - Q_k(x, x_\upsilon) \leq \widetilde{a}_k \left(e^x - \frac{1}{N+1} \right) + \widetilde{b}_k \quad \text{for} \quad x \in [\bar{D}_k, \bar{U}_k], \quad x_\upsilon \in [d, u].$$

We have that

$$\widetilde{q}_{k-1}(x, x_\upsilon, y_\upsilon) - \widetilde{Q}_{k-1}(x, x_\upsilon, y_\upsilon) = \int_{\mathbb{R}} (p_k(x-z, y_\upsilon) - P_k(x-z, y_\upsilon)) f(\Delta, z | x_\upsilon, y_\upsilon) dz \geq 0.$$

From (31) we obtain for any $x \in [\bar{D}_{k-1}, \bar{U}_{k-1}]$, $(x_\upsilon, y_\upsilon) \in [d, u] \times [d, u]$

$$\begin{aligned}
& \widetilde{q}_{k-1}(x, x_\upsilon, y_\upsilon) - \tilde{Q}_{k-1}(x, x_\upsilon, y_\upsilon) \\
= & \int_{[d_k, u_k]} (p_k(x-z, y_\upsilon) - P_k(x-z, y_\upsilon)) f(\Delta, z | x_\upsilon, y_\upsilon) dz \\
& + \int_{\mathbb{R} \setminus [d_k, u_k]} (p_k(x-z, y_\upsilon) - P_k(x-z, y_\upsilon)) f(\Delta, z | x_\upsilon, y_\upsilon) dz \\
\leq & \int_{[d_k, u_k]} (\widetilde{a}_k e^{x-z} + \widetilde{b}_k) f(\Delta, z | x_\upsilon, y_\upsilon) dz + \int_{\mathbb{R} \setminus [d_k, u_k]} p_k(x-z, y_\upsilon) f(\Delta, z | x_\upsilon, y_\upsilon) dz \\
\leq & \int_{\mathbb{R}} (\widetilde{a}_k e^{x-z} + \widetilde{b}_k) f(\Delta, z | x_\upsilon, y_\upsilon) dz + \int_{\mathbb{R} \setminus [d_k, u_k]} (a_k e^{x-z} + b_k) f(\Delta, z | x_\upsilon, y_\upsilon) dz \\
\leq & \widetilde{a}_k \widehat{\mu} e^x + \widetilde{b}_k + a_k e^x \left[F(d_k; -1) + G(u_k; -1) \right] + b_k \left[F(d_k; 0) + G(u_k; 0) \right],
\end{aligned}$$

and

$$\begin{aligned}
& q_{k-1}(x, x_\upsilon) - Q_{k-1}(x, x_\upsilon) \\
= & \int_{\mathbb{R}_+} \left(\widetilde{q}_{k-1}(x, x_\upsilon, y_\upsilon) - \tilde{Q}_{k-1}(x, x_\upsilon, y_\upsilon) \right) c(\Delta, y_\upsilon | x_\upsilon) dy_\upsilon \\
= & \int_{[d, u]} \left(\widetilde{q}_{k-1}(x, x_\upsilon, y_\upsilon) - \tilde{Q}_{k-1}(x, x_\upsilon, y_\upsilon) \right) c(\Delta, y_\upsilon | x_\upsilon) dy_\upsilon \\
& + \int_{[0, d] \cup [u, +\infty)} \widetilde{q}_{k-1}(x, x_\upsilon, y_\upsilon) c(\Delta, y_\upsilon | x_\upsilon) dy_\upsilon \\
\leq & \widetilde{a}_k \widehat{\mu} e^x + \widetilde{b}_k + a_k e^x \left[F(d_k; -1) + G(u_k; -1) \right] + b_k \left[F(d_k; 0) + G(u_k; 0) \right] \\
& + (a_{k-1} e^x + b_k) \left[\widehat{F}(d, 0) + \widehat{G}(u, 0) \right] \\
\leq & \widetilde{a}_{k-1} e^x + \widetilde{b}_{k-1} - \frac{\widetilde{a}_{k-1}}{N+1}.
\end{aligned}$$

From (32) we have for any $y \in [D_{k-1}, U_{k-1}]$ that $h_{k-1}(y) \in [\bar{D}_{k-1}, \bar{U}_{k-1}]$ and, thus,

$$\begin{aligned} 0 &\leq p_{k-1}(y, y_\upsilon) - P_{k-1}(y, y_\upsilon) = q_{k-1}(h_{k-1}(y), y_\upsilon) - Q_{k-1}(h_{k-1}(y), y_\upsilon) \\ &\leq \tilde{a}_{k-1}\left(e^{h_{k-1}(y)} - \frac{1}{N+1}\right) + \tilde{b}_{k-1} = \tilde{a}_{k-1}e^{y} + \tilde{b}_{k-1}. \end{aligned}$$

Appendix A.5. Proof of Theorem 6

Consider the derivatives with respect to x and y. The proof proceeds by induction on k. Function P_N is piecewise differentiable and satisfies

$$0 \leq \partial_y P_N(y, y_\upsilon) \leq a_N e^y + c_N \quad \text{for} \quad y \in (D_N, U_N), \quad y_\upsilon \in (d, u)$$

with $c_N = 0$. In addition, assume that

$$0 \leq \partial_y P_k(y, y_\upsilon) \leq a_k e^y + c_k \quad \text{for} \quad y \in (D_k, U_k), \quad y_\upsilon \in (d, u)$$

for arbitrary $k < N$ and some constant c_k. Note that $\partial_y P_k(y, y_\upsilon) = P_k(y, y_\upsilon) = 0$ for any $y \notin (D_k, U_k)$, $y_\upsilon \notin (d, u)$. Function $0 \leq \partial_x P_k(x - z, y_\upsilon) f(\Delta, z | x_\upsilon, y_\upsilon)$ is dominated by an integrable function of x in a compact interval, thus we can interchange integration and differentiation to obtain for $x \in (\bar{D}_{k-1}, \bar{U}_{k-1})$, $x_\upsilon \in (d, u)$

$$\begin{aligned} 0 \leq \partial_x \widetilde{Q}_{k-1}(x, x_\upsilon, y_\upsilon) &= \int_{\mathbb{R}} \partial_x P_k(x - z, y_\upsilon) f(\Delta, z | x_\upsilon, y_\upsilon) dz \leq \int_{\mathbb{R}} (a_k e^{x-z} + c_k) f(\Delta, z | x_\upsilon, y_\upsilon) dz \\ &\leq a_{k-1}e^x + c_k. \end{aligned}$$

The same bound holds for Q implying that $\partial_x Q_{k-1}$ is continuous in $(\bar{D}_{k-1}, \bar{U}_{k-1})$. As $h'_{k-1}(y) \in (0, 1)$ for all y, we also obtain

$$0 \leq \partial_y P_{k-1}(y, y_\upsilon) \leq \partial_x Q_{k-1}(x, y_\upsilon)|_{x=h_{k-1}(y)} \leq a_{k-1}e^{h_{k-1}(y)} + c_k = a_{k-1}e^y + c_{k-1}, \tag{A.2}$$

where $c_{k-1} = c_k + a_{k-1}/(N+1)$. Similarly we proceed with the second order derivatives and obtain

$$0 \leq \partial_{yy} P_k(y, y_\upsilon) \leq a_k e^y + c_k. \tag{A.3}$$

We are now able to define the discretization error from evaluating integral (29) with the trapezoidal rule with n points. More specifically, in computing $\widetilde{Q}_{k-1}$ we have

$$\widetilde{Q}_{k-1}(x, x_\upsilon, y_\upsilon) = \int_{D_k}^{U_k} P_k(y, y_\upsilon) f(\Delta, x - y | x_\upsilon, y_\upsilon) dy,$$

which, when approximated by the trapezoidal rule, yields an error bounded (see Quarteroni et al., 2010) by

$$\frac{(U_k - D_k)^3 E}{12n^2},$$

with

$$\begin{aligned} E &= \max_{y\in[D_k,U_k]} |\partial_{yy}(P_k(y,y_\upsilon)f(\Delta,x-y|x_\upsilon,y_\upsilon))| \\ &\leq \max_{y\in[D_k,U_k]} |(a_k e^y+c_k)f(\Delta,x-y|x_\upsilon,y_\upsilon)+(a_k e^y+b_k)\partial_{zz}f(\Delta,z|x_\upsilon,y_\upsilon)|_{z=x-y} \\ &\quad -2(a_k e^y+c_k)\partial_z f(\Delta,z|x_\upsilon,y_\upsilon)|_{z=x-y}|\,, \end{aligned}$$

due to (28) (recall that $P_k \leq p_k$ by definition), (A.2) and (A.3). Note that E is bounded by assumption (33).

Consider now the derivatives with respect to x_υ and y_υ. We have that

$$\begin{aligned} \partial_{x_\upsilon}\widetilde{Q}_{k-1}(x,x_\upsilon,y_\upsilon) &= \int_{\mathbb{R}} P_k(x-z,y_\upsilon)\partial_{x_\upsilon}f(\Delta,z|x_\upsilon,y_\upsilon)dz\, 1_{[\bar{D}_{k-1},\bar{U}_{k-1}]}(x)1_{[d,u]\times[d,u]}(x_\upsilon,y_\upsilon) \\ &\leq a_k e^x\int_{\mathbb{R}} e^{-z}\partial_{x_\upsilon}f(\Delta,z|x_\upsilon,y_\upsilon)dz + b_k\int_{\mathbb{R}}\partial_{x_\upsilon}f(\Delta,z|x_\upsilon,y_\upsilon)dz =: A_k(x,x_\upsilon,y_\upsilon), \end{aligned}$$

and, similarly,

$$\begin{aligned} \partial_{x_\upsilon}Q_{k-1}(x,x_\upsilon) &\leq \int_{\mathbb{R}_+} A_k(x,x_\upsilon,y_\upsilon)c(\Delta,y_\upsilon|x_\upsilon)dy_\upsilon + (a_{k-1}e^x+b_k)\int_{\mathbb{R}_+}\partial_{x_\upsilon}c(\Delta,y_\upsilon|x_\upsilon)dy_\upsilon, \\ \partial_{y_\upsilon}P_{k-1}(y,y_\upsilon) &= \partial_{y_\upsilon}Q_{k-1}(h_{k-1}(y),y_\upsilon) \\ &\leq \int_{\mathbb{R}_+} A_k(h_{k-1}(y),y_\upsilon,z_\upsilon)c(\Delta,z_\upsilon|y_\upsilon)dz_\upsilon + (a_{k-1}e^y+b_{k-1})\int_{\mathbb{R}_+}\partial_{y_\upsilon}c(\Delta,z_\upsilon|y_\upsilon)dz_\upsilon \\ &=: B_{k-1}(y,y_\upsilon) \end{aligned}$$

and

$$\begin{aligned} \partial_{y_\upsilon}\widetilde{Q}_{k-1}(x,x_\upsilon,y_\upsilon) &\leq a_k e^x\int_{\mathbb{R}} e^{-z}\partial_{y_\upsilon}f(\Delta,z|x_\upsilon,y_\upsilon)dz + b_k\int_{\mathbb{R}}\partial_{y_\upsilon}f(\Delta,z|x_\upsilon,y_\upsilon)dz \\ &\quad + \int_{\mathbb{R}} B_k(x-z,y_\upsilon)f(\Delta,z|x_\upsilon,y_\upsilon)dz. \end{aligned}$$

We proceed similarly with the second order derivatives. Thus, we have that $\widetilde{Q}_{k-1}$ is differentiable and $\widetilde{Q}_{k-1}$, $\partial_{y_\upsilon}\widetilde{Q}_{k-1}$ and $\partial_{y_\upsilon y_\upsilon}\widetilde{Q}_{k-1}$ are bounded.

Therefore, the discretization error from evaluating integral (30) with the trapezoidal rule with n_υ points is bounded by

$$\frac{(u-d)^3 E_\upsilon}{12n_\upsilon^2},$$

with

$$\begin{aligned} E_\upsilon &= \max_{y_\upsilon\in[d,u]} \Big|\partial_{y_\upsilon y_\upsilon}\widetilde{Q}_{k-1}(x,x_\upsilon,y_\upsilon)c(\Delta,y_\upsilon|x_\upsilon)+2\partial_{y_\upsilon}\widetilde{Q}_{k-1}(x,x_\upsilon,y_\upsilon)\partial_{y_\upsilon}c(\Delta,y_\upsilon|x_\upsilon) \\ &\quad +\widetilde{Q}_{k-1}(x,x_\upsilon,y_\upsilon)\partial_{y_\upsilon y_\upsilon}c(\Delta,y_\upsilon|x_\upsilon)\Big|\,. \end{aligned}$$

Note that E_υ is bounded due to assumption (33).

Appendix B. Extension to general affine stochastic volatility models

When deriving the results in Proposition 1 we exploit the analytical tractability of the square root volatility. Nevertheless, we can extend to general affine stochastic volatility models in the

absence of an analytical solution for the characteristic function of the log-return conditional on the current and terminal variance states. In particular, from Bartlett (1938, p. 62–63) and Bates (2006, Proposition 1),

$$\hat{\phi}(t,u|v_0,v) = \frac{\int e^{-iv\omega}\hat{E}\left(\exp(i\omega V_t + iuZ_t)|V_0 = v_0\right)d\omega}{\int e^{-iv\omega}\hat{E}\left(\exp(i\omega V_t)|V_0 = v_0\right)d\omega}, \tag{B.1}$$

where $\hat{E}\left(\exp(i\omega V_t + iuZ_t)|V_0 = v_0\right)$ admits explicit representations under various affine volatility models (see Table B.7); for more details about the derivation of the relevant expressions, we refer, for example, to Hubalek et al. (2017). In what follows, we present concrete paradigms with jumps in the volatility.

Appendix B.1. Stochastic volatility with concurrent and correlated price and variance jumps

In addition to the Heston and Bates models, Duffie et al. (2000) include jumps in the variance process. The governing equations are

$$\begin{aligned} dX_t &= \left(r - \lambda k(1,0) - \frac{1}{2}V_t\right)dt + \sqrt{V_t}\left(\rho dW_t^V + \sqrt{1-\rho^2}dW_t^X\right) + dL_t^X, \\ dV_t &= \alpha\left(\beta - V_t\right)dt + \gamma\sqrt{V_t}dW_t^V + dL_t^V, \end{aligned}$$

where processes $(L_t^V)_{t\geq 0}$ and $(L_t^X)_{t\geq 0}$ have a common Poisson driver, hence jumps in price and volatility occur concurrently, have correlated sizes with parameters $\rho_{X,V}$, μ_X, σ_X, μ_V, and jump transform $k(u_1,u_2) := \exp(\mu_X u_1 + \sigma_X^2 u_1^2/2)/(1-\rho_{X,V}\mu_V u_1 - \mu_V u_2) - 1$. The volatility jumps are positive (the jump size is exponential).

Appendix B.2. Non-Gaussian Ornstein–Uhlenbeck stochastic volatility models

Barndorff-Nielsen and Shephard (2001) and Barndorff-Nielsen et al. (2002) consider Lévy-driven positive Ornstein–Uhlenbeck (OU) models for the variance process:

$$dV_t = -\lambda V_t dt + dL_t,$$

where $\lambda > 0$ and the background driving Lévy process $(L_t)_{t\geq 0}$ is a zero-drift subordinator with Laplace exponent $k(u)$. For example, the OU process with a gamma stationary law corresponds to a compound Poisson subordinator with exponential jump sizes (see Cont and Tankov, 2004, Example 15.1) with $k(u) = \nu u/(\alpha - u)$, $\nu, \alpha > 0$. Another concrete case is that of the variance process with an inverse Gaussian stationary law with $k(u) = \nu u/\sqrt{\alpha^2 - 2u}$, $\nu, \alpha > 0$. The risk neutral specification of the log-price is given by

$$dX_t = \left(r - \lambda k(\rho) - \frac{1}{2}V_t\right)dt + \sqrt{V_t}dW_t^X + \rho dL_t,$$

where the standard Brownian motion W^X is independent of L, $\rho \leq 0$ and the ρdL_t term accounts for the leverage effect.

Table B.7 summarizes the joint laws of (V,Z) for use in (B.1).

Model	$\hat{E}\left(\exp(i\omega V_t + iuZ_t)\middle\vert V_0 = \upsilon_0\right)$
Duffie et al. (2000)	$\exp\left\{t\left(r - \frac{\rho\alpha\beta}{\gamma} - \lambda k(1,0)\right)iu\right.$
	$+\frac{\alpha\beta}{\gamma^2}\left[\alpha t + \ln\left(\frac{\vartheta^2(u)(1+\varphi_s^2(\omega,u))}{\gamma^2(2(iu\rho\gamma-\alpha)i\omega-\gamma^2\omega^2+iu(iu-1))}\right)\right]$
	$+\lambda\int_0^t k\left(iu, \frac{\varphi_s(\omega,u)\vartheta(u)+\alpha-iu\rho\gamma}{\gamma^2}\right)ds$
	$\left.+\frac{\varphi_t(\omega,u)\vartheta(u)+\alpha-iu\rho\gamma}{\gamma^2}\upsilon_0\right\}$
Barndorff-Nielsen and Shephard (2001),	$\exp\{t(r-\lambda k(\rho))iu$
Barndorff-Nielsen et al. (2002)	$+\lambda\int_0^t k\left(e^{-\lambda s}i\omega + \frac{1-e^{-\lambda s}}{2\lambda}(iu-1)iu + \rho iu\right)ds$
	$\left.+\left[e^{-\lambda t}i\omega + \frac{1-e^{-\lambda t}}{2\lambda}(iu-1)iu\right]\upsilon_0\right\}$

Table B.7: Summarized characteristic functions (see equation B.1). Notes: Duffie et al. (2000): $\varphi_t(\omega,u) := \tan\left[\vartheta(u)t/2 + \arctan\left((iu\rho\gamma - \alpha + i\omega\gamma^2)/\vartheta(u)\right)\right]$ and $\vartheta(u) := \sqrt{(iu-1)iu\gamma^2 - (\alpha - iu\rho\gamma)^2}$.