

A Fourier-Based Valuation Method for Bermudan and Barrier Options under Heston's Model*

Fang Fang[†] and Cornelis W. Oosterlee[‡]

Abstract. We develop an efficient Fourier-based numerical method for pricing Bermudan and discretely monitored barrier options under the Heston stochastic volatility model. The two-dimensional pricing problem is dealt with by a combination of a Fourier cosine series expansion, as in [F. Fang and C. W. Oosterlee, *SIAM J. Sci. Comput.*, 31 (2008), pp. 826–848, F. Fang and C. W. Oosterlee, *Numer. Math.*, 114 (2009), pp. 27–62], and high-order quadrature rules in the other dimension. Error analysis and experiments confirm a fast error convergence.

Key words. Heston model, Bermudan options, Fourier cosine expansions, numerical quadrature

AMS subject classifications. 65T40, 60E10, 91G60

DOI. 10.1137/100794158

1. Introduction. In mathematical finance, stochastic volatility models have been developed to capture the volatility smiles and skews present in market quotes. Within this class, the Heston stochastic volatility model [10], in which the variance of (the logarithm of) the stock price is modeled by a square-root process, has become popular in industrial practice. The pricing of European options is particularly efficient.

Fourier-based integration methods require the availability of the characteristic function (ChF), i.e., the Fourier transform of the probability density function of the underlying stock price. Since the ChF of Heston's model has already been given in the original paper [10], fast and accurate valuation tools for European options under Heston's model are available.

Many exotic financial products include some form of path dependency. Monte Carlo simulation methods are often used for the valuation of such products in practice. As a result, the recent numerical advances in the context of Heston's model were obtained mainly for Monte Carlo simulation methods [5, 3].

In this paper we aim to develop a stable and efficient Fourier-based valuation method that can price both Bermudan and discrete-barrier options under the Heston stochastic volatility dynamics. It is in essence a generalization of the COS [7, 8] method, which is an efficient option pricing method for (one-dimensional) Lévy processes, to the (two-dimensional) Heston model. The following three issues, however, make this topic challenging:

- *Near-singular behavior of the probability density of the variance:*

The variance in the Heston model is governed by a noncentral chi-square distribution.

*Received by the editors May 4, 2010; accepted for publication (in revised form) April 18, 2011; published electronically July 19, 2011.

<http://www.siam.org/journals/sifin/2/79415.html>

[†]Delft University of Technology, Delft Institute of Applied Mathematics, 2628 CD Delft, The Netherlands (f.fang@ewi.tudelft.nl).

[‡]CWI - Centrum Wiskunde & Informatica, NL-1090 GB Amsterdam, The Netherlands (c.w.oosterlee@cwi.nl), and Delft University of Technology, Delft Institute of Applied Mathematics, 2628 CD Delft, The Netherlands.

For some combinations of the relevant parameters, the density of the variance grows drastically in the left-side tail; i.e., the density values tend to infinitely large numbers as the variance approaches zero. Truncation of the integration range for the variance may then easily introduce significant truncation errors.

- *The integration kernel is not known explicitly:*

For path-dependent options, the pricing formula requires a two-dimensional integration over the log-stock price *and* the variance. The probability density function of the joint distribution is, however, not known in closed form and has to be recovered from the ChF.

- *Quadratic computational complexity:*

In numerical analysis, the highest computational speed is related to linear computational complexity, which means that the computational time grows only linearly w.r.t. an increasing number of unknowns and/or exponential error convergence; i.e., the error decreases exponentially with a growing number of unknowns.

A direct application of basic numerical integration rules for options with early-exercise features under Heston's model would result in quadratic computational complexity in both dimensions and would therefore cost a significant amount of CPU time.

The contributions of the present paper are the following. We determine parameter sets which give rise to near-singular behavior and tackle the problem by a transformation from the variance domain to the *log-variance domain*. Second, to solve the two-dimensional problem in a robust and efficient manner, we combine the Fourier cosine expansion from [8] with quadrature rules. In section 2, we describe the Heston asset dynamics. We focus on the issue of the left-side tail of the variance density. In section 3, the discrete pricing formula for Bermudan options is derived and an efficient recursive algorithm is developed. Minor differences when pricing discrete-barrier options are highlighted in section 4. In section 5, the error convergence and the error propagation are analyzed. Various numerical experiments are presented in section 6, and conclusions are drawn in section 7.

2. Heston model. In this section we give some insight into the Heston model. After some known results from the literature, we focus, in particular, on the *near-singular behavior* of the variance process near the origin. By means of several numerical experiments, we find the relevant parameter sets giving rise to this phenomenon and propose a transformation to deal with it when pricing options.

2.1. Basics. The Heston stochastic volatility model defines the dynamics of the logarithm of the stock price (*log-stock*), x_t , and the variance, ν_t , by the following stochastic differential equations (SDEs) [10]:

$$(1) \quad dx_t = \left(\mu - \frac{1}{2}\nu_t \right) dt + \rho\sqrt{\nu_t}dW_{1,t} + \sqrt{1-\rho^2}\sqrt{\nu_t}dW_{2,t},$$

$$(2) \quad d\nu_t = \lambda(\bar{\nu} - \nu_t) dt + \eta\sqrt{\nu_t}dW_{1,t},$$

where the three nonnegative parameters, λ , $\bar{\nu}$, and η , represent the speed of mean reversion, the mean level of variance, and the volatility of the volatility process, respectively. The Brownian motions, $W_{1,t}$ and $W_{2,t}$, are independent, and ρ is the correlation between the log-stock and the variance processes.

The square-root process defined in (2) precludes negative values for ν_t , and if ν_t reaches zero, it can subsequently become positive. The Feller condition, $2\lambda\bar{\nu} \geq \eta^2$, guarantees that ν_t stays positive; otherwise, it may reach zero. As indicated in [9, 6], with

$$q := 2\lambda\bar{\nu}/\eta^2 - 1 \quad \text{and} \quad \zeta := 2\lambda / \left((1 - e^{-\lambda(t-s)})\eta^2 \right),$$

the process $2\zeta\nu_t \sim \chi^2(q, 2\zeta\nu_s e^{-\lambda(t-s)})$, for $0 < s < t$, is governed by the noncentral chi-square distribution with degree q and noncentrality parameter $2\zeta\nu_s e^{-\lambda(t-s)}$. Therefore, the probability density function of ν_t given ν_s reads as

$$(3) \quad p_\nu(\nu_t|\nu_s) = \zeta e^{-\zeta(\nu_s e^{-\lambda(t-s)} + \nu_t)} \left(\frac{\nu_t}{\nu_s e^{-\lambda(t-s)}} \right)^{\frac{q}{2}} I_q \left(2\zeta e^{-\frac{1}{2}\lambda(t-s)} \sqrt{\nu_s \nu_t} \right),$$

where $I_q(\cdot)$ is the modified Bessel function of the first kind with order q .

The Feller condition is thus equivalent to “ $q \geq 0$.” This is difficult to satisfy in practice. It has, for example, been reported [3] that one often finds $2\lambda\bar{\nu} \ll \eta^2$ from market data, in which case the cumulative distribution of the variance shows a *near-singular behavior* near the origin, or, in other words, the left tail of the variance density grows extremely fast in value.

Such a behavior in the left tail may easily give rise to significant errors, especially for integration-based option pricing methods, for which the integration range needs to be truncated.

The exact simulation method, developed in Broadie and Kaya [5], provides, next to an exact formula to sample the log-stock price, insight into the distribution for stochastic volatility models. Integration of (1) and (2) yields [5]

$$(4) \quad x_t - x_s = \mu(t-s) - \frac{1}{2} \int_s^t \nu_\tau d\tau + \rho \int_s^t \sqrt{\nu_\tau} dW_{1,\tau} + \sqrt{1-\rho^2} \int_s^t \sqrt{\nu_\tau} dW_{2,\tau},$$

$$(5) \quad \nu_t - \nu_s = \lambda\bar{\nu}(t-s) - \lambda \int_s^t \nu_\tau d\tau + \eta \int_s^t \sqrt{\nu_\tau} dW_{1,\tau}.$$

Equation (5) can be rewritten as an equation for $\int_s^t \sqrt{\nu_\tau} dW_{1,\tau}$, which, substituted into (4), gives the following exact formula for x_t :

$$(6) \quad x_t - x_s = \mu(t-s) + \frac{\rho}{\eta} (\nu_t - \nu_s - \lambda\bar{\nu}(t-s)) + \left(\frac{\lambda\rho}{\eta} - \frac{1}{2} \right) \int_s^t \nu_\tau d\tau + \sqrt{1-\rho^2} \int_s^t \sqrt{\nu_\tau} dW_{2,\tau}.$$

Equation (6) can be used to sample x_t once the values of the variance, ν_t , and the time-integrated variance, $\int_s^t \nu_\tau d\tau$, are available. The variance is then sampled from (an approximation of) the noncentral chi-square distribution, and the time-integrated variance is sampled from a distribution which is recovered from the ChF, $\Phi(u; \nu_t, \nu_s)$, for which a closed-form

expression is available:

$$\begin{aligned}
 \Phi(v; \nu_t, \nu_s) &:= \mathbb{E} \left[\exp \left(iv \int_s^t \nu_\tau d\tau \right) \middle| \nu_t, \nu_s \right] \\
 &= \frac{I_q \left[\frac{\sqrt{\nu_t \nu_s}}{\eta^2} \frac{4\gamma(v) e^{-\frac{1}{2}\gamma(v)(t-s)}}{1 - e^{-\gamma(v)(t-s)}} \right]}{I_q \left[\frac{\sqrt{\nu_t \nu_s}}{\eta^2} \frac{4\lambda e^{-\frac{1}{2}\lambda(t-s)}}{1 - e^{-\lambda(t-s)}} \right]} \cdot \frac{\gamma(v) e^{-\frac{1}{2}(\gamma(v)-\lambda)(t-s)} (1 - e^{-\lambda(t-s)})}{\lambda(1 - e^{-\gamma(v)(t-s)})} \\
 &\quad \cdot \exp \left(\frac{\nu_s + \nu_t}{\eta^2} \left[\frac{\lambda(1 + e^{-\lambda(t-s)})}{1 - e^{-\lambda(t-s)}} - \frac{\gamma(v)(1 + e^{-\gamma(v)(t-s)})}{1 - e^{-\gamma(v)(t-s)}} \right] \right),
 \end{aligned}
 \tag{7}$$

where, again, $q = 2\lambda\bar{\nu}/\eta^2 - 1$ and $I_q(x)$ is the modified Bessel function of the first kind with order q . Variable $\gamma(v)$ is defined by

$$\gamma(v) := \sqrt{\lambda^2 - 2i\eta^2 v}.
 \tag{8}$$

2.2. The left-side tail. As a first step to understanding the near-singular behavior in the variance direction, we set up a series of numerical experiments to examine when the near-singular behavior occurs. The following results can be used as a *rule of thumb* to determine the values for which the variance density is governed by extremely large values at the left tail.

Result 2.1 (the left-side tail). *Although each of the three parameters, λ , $\bar{\nu}$, and η , in (2) plays a unique role in the tuning of the shape and the magnitude of the variance density, the decay rate at the left tail can be well characterized by values of q , whose definition interval is $[-1, \infty)$. Based on the nonnegativeness of λ , $\bar{\nu}$, and η , the near-singular problem occurs when $q \in [-1, 0]$, which is directly related to the Feller condition.*

The experiments that support this insight are set up as follows: The values of $\bar{\nu}$ and η are drawn randomly from $[0, 1]$ (we consider interval $[0, 1]$ reasonable for both $\bar{\nu}$ and η), and λ is given by $(1 + q)\eta^2/(2\bar{\nu})$. The experimental results are displayed in Figure 1.

The value of q determines the decay rate in the left tail of the variance density function, whereas the right-side tail always decays to zero rapidly. For $q \gg 0$, the density values decrease towards zero in both tails. For q smaller and approaching 0, the decay of the left-side tail slows down. Near $q = 0$, the left tail stays almost constant. For $q \in [-1, 0]$, the left tail increases drastically in value.

In a recent paper [3], several challenging test cases, based on different values of λ , η , and $\bar{\nu}$, were illustrated. For all those test cases we find $q \approx -0.96$, which indeed is an indication of difficult tests; see Figure 1.

The fact that q determines the decay rate of the densities' left tail can be understood if we take a closer look at (3) for the variance density function. When q changes sign, both functions, $(\cdot)^{q/2}$ and $I_q(\cdot)$, change shape around the origin; i.e., they go from monotonically increasing to monotonically decreasing.

2.3. Transformation to log-variance process. Based on the insights in the previous subsections, we propose here a solution strategy for the problem of the left-side tail: We transform the problem from the variance domain to the *log-variance domain*.

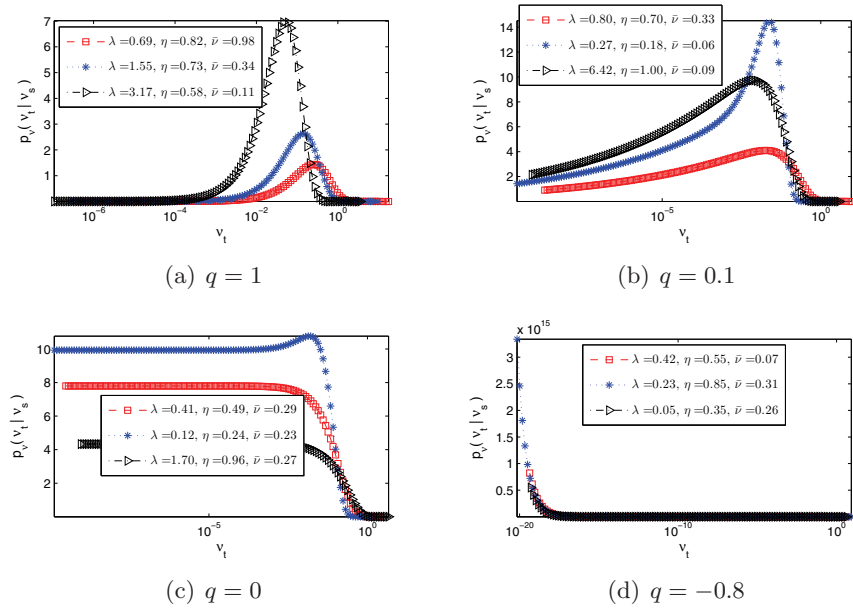


Figure 1. Decay rate in the left-side tail of the variance density as q approaches -1 from above.

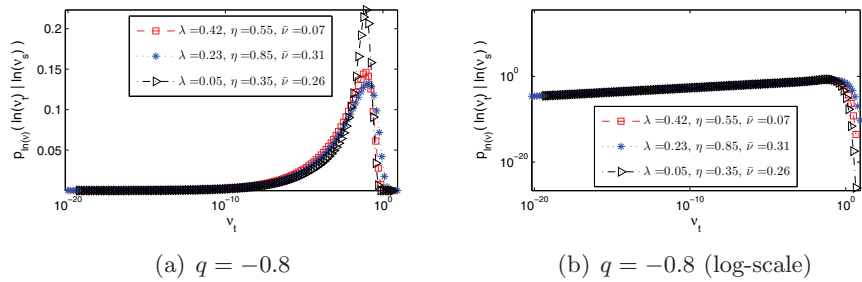


Figure 2. Decay rate of the left tail of the log-variance density in original scale and log-scale as q approaches -1 .

By the change of variables, the density of the log-variance process, based on (3), reads as

$$(9) \quad p_{\ln(\nu)}(\sigma_t | \sigma_s) = \zeta e^{-\zeta(e^{\sigma_s} e^{-\lambda(t-s)} + e^{\sigma_t})} \left(\frac{e^{\sigma_t}}{e^{\sigma_s} e^{-\lambda(t-s)}} \right)^{\frac{q}{2}} e^{\sigma_t} I_q \left(2\zeta e^{-\frac{1}{2}\lambda(t-s)} \sqrt{e^{\sigma_s} e^{\sigma_t}} \right),$$

where $\sigma_s := \ln(\nu_s)$ and $p_{\ln(\nu)}(\sigma_t | \sigma_s)$ denotes the probability density of the log-variance at a future time, given the information at the current time.

With the change of variables, a term e^{σ_t} appears, which, for $q \in [-1, 0]$, compensates the $(\cdot)^{\frac{q}{2}}$ -term, so that it converges towards zero as $\sigma_t \rightarrow -\infty$. Densities of the log-variance process for different parameter sets are *more symmetric* than those from Figure 1. It is also illustrated in log-scale, in Figure 2, that the left tails of the densities no longer increase significantly in value. Instead, these tails decay to zero rapidly as $\sigma_t \rightarrow -\infty$, although the decay rate decreases as q approaches -1 .

Before applying any numerical method, we need to define a proper truncation range for the log-variance density. For this, information about the *center* of the density as well as the decay of the left and right tails is required.

Instead of giving a rule of thumb for this truncation range, as in [7, 12], we propose to use Newton's method to determine the interval boundaries, according to a predefined error tolerance, TOL. In accordance with this tolerance, the stopping criterion of the Newton method reads as $p_{\ln(\nu)}(x|\sigma_0; T) < \text{TOL}$ for $x \in \mathbb{R} \setminus [a_\nu, b_\nu]$.

We also need the derivative of $p_{\ln(\nu)}(\sigma_t|\sigma_s)$ w.r.t. σ_t . It can be derived with the help of Maple:

$$(10) \quad \frac{dp_{\ln(\nu)}(\sigma_t|\sigma_s)}{d\sigma_t} = - \left[(-\zeta e^{\sigma_t} - q - 1) I_q \left(2\sqrt{\zeta e^{\sigma_t} u} \right) - I_{q+1} \left(2\sqrt{\zeta e^{\sigma_t} u} \right) \right] \\ \cdot \zeta e^{-u - \zeta e^{\sigma_t} + \sigma_t} \cdot \left(\frac{\zeta e^{\sigma_t}}{u} \right)^{q/2},$$

with $u := \zeta e^{\sigma_s - \lambda(t-s)}$.

A proper *initial guess* for interval boundaries is also required. We estimate the center of the truncation range by the logarithm of the mean value of the variance (see, e.g., [3]):

$$\ln(\mathbb{E}(\nu_t)) = \ln \left(\nu_0 e^{-\lambda T} + \bar{\nu} \left(1 - e^{-\lambda T} \right) \right).$$

As the left tail usually decays much slower than the right tail and because the *speed of decay* seems closely related to the value of q , we use the following values as the initial guesses for the boundaries of the truncation range $[a_\nu, b_\nu]$:

$$(11) \quad [a_\nu^0, b_\nu^0] = \left[\ln(\mathbb{E}(\nu_t)) - \frac{5}{1+q}, \ln(\mathbb{E}(\nu_t)) + \frac{2}{1+q} \right].$$

2.4. Joint distribution of log-stock and log-variance. When valuing path-dependent options, we need to know the joint distribution of the log-stock and log-variance processes at a future time, given the information at the current time, i.e., $p_{x, \ln(\nu)}(x_t, \sigma_t | x_s, \sigma_s)$, with $0 < s < t$. An analytic formula for this distribution does not exist, but we can deduce the relevant information from the Fourier domain.

The SDEs in (1), (2) indicate that the variance at a future time is independent of the log-stock value at the current time, i.e., $p_\nu(\nu_t | \nu_s, x_s) = p_\nu(\nu_t | \nu_s)$. As a result, we have

$$(12) \quad p_{x, \nu}(x_t, \nu_t | x_s, \nu_s) = p_{x|\nu}(x_t | \nu_t, x_s, \nu_s) \cdot p_\nu(\nu_t | \nu_s),$$

where we use $p_{x, \nu}$ to denote the joint probability density of the log-stock and the variance processes at a future time point, given that the information is known at the current time; $p_{x|\nu}$ denotes the probability density of the log-stock process at a future time point, given the variance value (and also given the information known at the current time). Equivalently, we have

$$(13) \quad p_{x, \ln(\nu)}(x_t, \sigma_t | x_s, \sigma_s) = p_{x|\ln(\nu)}(x_t | \sigma_t, x_s, \sigma_s) \cdot p_{\ln(\nu)}(\sigma_t | \sigma_s),$$

where $p_{x|\ln(\nu)}$ denotes the probability density of the log-stock at a future time spot, given the log-variance value as well as the information known at the current time.

The probability density of the log-variance, $p_{\ln(\nu)}(\sigma_t|\sigma_s)$, is already given in (9), and therefore we need $p_{x|\ln(\nu)}(x_t|\sigma_t, x_s, \sigma_s)$. Although there is no closed-form expression for $p_{x|\ln(\nu)}$, one can easily derive its conditional ChF, $\varphi(\omega; x_s, \sigma_t, \sigma_s)$, based on (6):

$$\begin{aligned} \varphi(\omega; x_s, \sigma_t, \sigma_s) &:= \mathbb{E}_s [\exp(i\omega x_t | \sigma_t)] \\ &= \exp\left(i\omega \left[x_s + \mu(t-s) + \frac{\rho}{\eta} (e^{\sigma_t} - e^{\sigma_s} - \lambda \bar{\nu}(t-s)) \right]\right) \\ &\quad \cdot \Phi\left(\omega \left(\frac{\lambda\rho}{\eta} - \frac{1}{2}\right) + \frac{1}{2}i\omega^2(1-\rho^2); e^{\sigma_t}, e^{\sigma_s}\right), \end{aligned} \tag{14}$$

where $\Phi(u; \nu_t, \nu_s)$ is the ChF of the time-integrated variance as given in (7).

3. The pricing method for Bermudan options. In this section, we derive the pricing formula for Bermudan options under Heston's model. This gives rise to a two-dimensional integral with a kernel which is only partly available in closed form. To evaluate this two-dimensional integral, we develop a discrete formula based on *Fourier cosine series expansions* for the integration of the part of the kernel which is not known in closed form and a *quadrature rule* for the integral of the known part of the kernel. An efficient algorithm to compute the discrete formula with the help of the FFT algorithm is introduced.

3.1. The pricing equations. For a European option, which is defined at time s and matures at time t , with $0 < s < t$, the risk-neutral valuation formula reads as

$$v(x_s, \sigma_s, s) = e^{-r(t-s)} \mathbb{E}_s^{\mathbb{Q}} [v(x_t, \sigma_t, t)]. \tag{15}$$

Here $v(x_s, \sigma_s, s)$ denotes the option price at time s , r is the risk-free interest rate, and $\mathbb{E}_s^{\mathbb{Q}}$ is the expectation operator under the risk-neutral measure, \mathbb{Q} , given the information at s .

The Markov property enables us to price a Bermudan option between two consecutive early-exercise dates by the risk-neutral valuation formula (15). This value is then called the *continuation value*. The arbitrage-free price of the Bermudan option on any early-exercise date is the maximum of the continuation value and the exercise payoff.

For M early-exercise dates and $\mathcal{T} := \{t_m, t_m < t_{m+1} | m = 0, 1, \dots, M\}$, with $t_M \equiv T$ and $\Delta t := t_{m+1} - t_m$, the Bermudan option pricing formula reads as

$$v(x_{t_m}, \sigma_{t_m}, t_m) = \begin{cases} g(x_{t_m}, t_m) & \text{for } m = M, \\ \max [c(x_{t_m}, \sigma_{t_m}, t_m), g(x_{t_m}, t_m)] & \text{for } m = 1, 2, \dots, M-1, \\ c(x_{t_m}, \sigma_{t_m}, t_m) & \text{for } m = 0, \end{cases} \tag{16}$$

with $g(x_\tau, \tau)$ being the payoff function at time τ and $c(x_\tau, \sigma_\tau, \tau)$ the continuation value at time τ .

We simplify the notation and use x_m and σ_m for x_{t_m} and σ_{t_m} , respectively. The continuation value is given by

$$c(x_m, \sigma_m, t_m) = e^{-r\Delta t} \mathbb{E}_{t_m}^{\mathbb{Q}} [v(x_{m+1}, \sigma_{m+1}, t_{m+1})], \tag{17}$$

which can be written as

$$(18) \quad c(x_m, \sigma_m, t_m) = e^{-r\Delta t} \cdot \int_{\mathbb{R}} \int_{\mathbb{R}} v(x_{m+1}, \sigma_{m+1}, t_{m+1}) p_{x, \ln(\nu)}(x_{m+1}, \sigma_{m+1} | x_m, \sigma_m) d\sigma_{m+1} dx_{m+1}.$$

With (13) we get

$$(19) \quad c(x_m, \sigma_m, t_m) = e^{-r\Delta t} \cdot \int_{\mathbb{R}} \left[\int_{\mathbb{R}} v(x_{m+1}, \sigma_{m+1}, t_{m+1}) p_{x | \ln(\nu)}(x_{m+1} | \sigma_{m+1}, x_m, \sigma_m) dx_{m+1} \right] \cdot p_{\ln(\nu)}(\sigma_{m+1} | \sigma_m) d\sigma_{m+1}.$$

Equations (16) and (19) define the problem we would like to solve numerically. The inner integral in (19) equals the pricing formula for European options defined between t_m and t_{m+1} , provided the variance value at the future time point is known.

A *scaled* log-asset price will be used from now on in this work, defined by

$$x_m = \ln(S_m/K).$$

3.2. Density recovery by Fourier cosine expansions. The COS method, based on Fourier cosine expansions, is a very efficient method for the recovery of probability density functions from the corresponding ChFs. It can therefore be efficiently used for the risk-neutral valuation formula in cases where the density is not known in closed form. We will apply the COS method to approximate the unknown conditional probability density, $p_{x | \ln(\nu)}$, in (19).

The key idea of the COS method [7] is to approximate the underlying probability density function, which is typically a smooth, real-valued function, by its Fourier cosine series expansion, taking into account that the Fourier series coefficients have a direct connection to the ChF.

First we define a truncated integration range, $[a, b] \subset \mathbb{R}$, such that

$$(20) \quad \int_a^b p_{x | \ln(\nu)}(x_{m+1} | \sigma_{m+1}, x_m, \sigma_m) dy \leq \text{TOL}_x$$

for some predefined error tolerance TOL_x . In [7] this interval is defined as

$$(21) \quad [a, b] := [\xi_1 - 12\sqrt{|\xi_2|}, \xi_1 + 12\sqrt{|\xi_2|}],$$

where ξ_n denotes the n th cumulant of the log-stock process. With an integration interval $[a, b]$ satisfying (21), we recover the probability density by its Fourier cosine series expansion:

$$(22) \quad p_{x | \ln(\nu)}(x_{m+1} | \sigma_{m+1}, x_m, \sigma_m) = \sum_{n=0}^{\infty}{}' P_n(\sigma_{m+1}, x_m, \sigma_m) \cos\left(n\pi \frac{x_{m+1} - a}{b - a}\right).$$

\sum' indicates that the first element in the summation is multiplied by one half. The coefficients P_n are the Fourier cosine coefficients, defined by

$$P_n(\sigma_{m+1}, x_m, \sigma_m) := \frac{2}{b - a} \int_a^b p_{x | \ln(\nu)}(x_{m+1} | \sigma_{m+1}, x_m, \sigma_m) \cos\left(n\pi \frac{x_{m+1} - a}{b - a}\right) dx_{m+1}.$$

By the expansion in (22), one *separates* x_{m+1} from x_m . This type of variable separation is not restricted to Fourier cosine series expansions, but in this case the Fourier expansion is advantageous, as the series coefficients have a direct relation to the ChF and are therefore known, i.e.,

$$(23) \quad P_n(\sigma_{m+1}, x_m, \sigma_m) \approx \frac{2}{b-a} \operatorname{Re} \left\{ \varphi \left(\frac{n\pi}{b-a}; x_m, \sigma_{m+1}, \sigma_m \right) e^{-in\pi \frac{a}{b-a}} \right\},$$

with $\varphi(\theta; x, \sigma_{m+1}, \sigma_m)$ given by (14).

The error in this approximation is related to TOL_x , as analyzed in [7], and (23) approximates the P_n with machine accuracy if $[a, b]$ is sufficiently wide. Subsequently, we truncate the series summation in (22).

From Fourier theory, we know that cosine series of functions belonging to $C^\infty([a, b] \subset \mathbb{R})$, with nonzero derivatives, converge exponentially w.r.t. the number of terms in the series, so that the series can be truncated without losing accuracy. By replacing P_n in (22) by (23) and truncating the series by N terms, one obtains a *semianalytic formula* which accurately approximates the probability density:

$$(24) \quad \begin{aligned} & p_{x|\ln(\nu)}(x_{m+1} | \sigma_{m+1}, x_m, \sigma_m) \\ &= \sum_{n=0}^{N-1} \frac{2}{b-a} \operatorname{Re} \left\{ \varphi \left(\frac{n\pi}{b-a}; 0, \sigma_{m+1}, \sigma_m \right) e^{in\pi \frac{x_m - a}{b-a}} \right\} \cos \left(n\pi \frac{x_{m+1} - a}{b-a} \right) + \epsilon_{\cos}. \end{aligned}$$

Here we used the fact that $\varphi(\omega; x_m, \sigma_{m+1}, \sigma_m) = e^{i\omega x_m} \varphi(\omega; 0, \sigma_{m+1}, \sigma_m)$; i.e., x_m can be separated from the σ -terms and appears as a simple exponential term. This is important for the efficient computation in the Bermudan case.

The error of this approximation, ϵ_{\cos} , decreases exponentially w.r.t. N , provided that the truncation range is set sufficiently wide (proof is given in [7]).

3.3. Discrete Fourier-based pricing formula. Equation (16) shows that the option price at time t_0 is a continuation value, which, as indicated by (19), depends on the continuation values at times t_1, t_2, \dots, t_M . The option price at time t_0 can be recovered by recursion backwards in time. This is the same approach as in [8], but here the integration is more involved because of the two-dimensional kernel.

3.3.1. Quadrature rule in log-variance dimension. Using the initial values defined in (11) and (21), we obtain the truncation range $[a_\nu, b_\nu]$ by Newton's method.

After truncating the integration region by $[a_\nu, b_\nu] \times [a, b]$, we need to compute

$$(25) \quad \begin{aligned} c_1(x_m, \sigma_m, t_m) &:= e^{-r\Delta t} \\ &\cdot \int_{a_\nu}^{b_\nu} \left[\int_a^b v(x_{m+1}, \sigma_{m+1}, t_{m+1}) p_{x|\ln(\nu)}(x_{m+1} | \sigma_{m+1}, x_m, \sigma_m) dx_{m+1} \right] \\ &\cdot p_{\ln(\nu)}(\sigma_{m+1} | \sigma_m) d\sigma_{m+1}. \end{aligned}$$

We use the notation c_i , $i = 1, \dots, 3$, to denote different approximations of continuation value, c , to keep track of the numerical errors that enter with each approximation.

There are two ways to discretize the outer integral w.r.t. σ_{m+1} , i.e., by interpolation-based quadrature rules or by a spectral series reconstruction of the interpolant (as in the COS method). Since $p_{\ln(\nu)}$ itself is known analytically, we apply a J -point quadrature integration rule (like the Gauss–Legendre quadrature rule, the composite trapezoidal rule, etc.) to the outer integral, which gives

$$(26) \quad c_2(x_m, \sigma_m, t_m) := e^{-r\Delta t} \sum_{j=0}^{J-1} w_j \cdot p_{\ln(\nu)}(\varsigma_j | \sigma_m) \cdot \left[\int_a^b v(x_{m+1}, \varsigma_j, t_{m+1}) p_{x|\ln(\nu)}(x_{m+1} | \varsigma_j, x_m, \sigma_m) dx_{m+1} \right].$$

Here the w_j are the weights of the quadrature nodes ς_j , $j = 1, 2, \dots, J-1$.

There are merits and demerits to using high-order quadrature rules, like the Gauss–Legendre quadrature rule, and to low-order equidistant rules, like the composite trapezoidal rule. The advantage of the former is an exponential error convergence rate for integration of smooth functions, as is the case for $p_{\ln(\nu)}$, whereas the latter has only polynomial error convergence. The computational complexity of the method can be reduced by the trapezoidal rule, due to a special matrix structure which results after discretization on an equidistant grid. In numerical experiments we found that the accuracy of the trapezoidal rule was insufficient when the Feller condition was not satisfied (unless an extremely large number of integration points was used). Therefore we do not show results for this approach here.

3.3.2. COS reconstruction in log-stock dimension. In the next step, we replace $p_{x|\ln(\nu)}$, which is not known, by the COS approximation (24) and interchange the summation over n with the integration over x_{m+1} to obtain

$$(27) \quad c_3(x_m, \sigma_m, t_m) := e^{-r\Delta t} \sum_{j=0}^{J-1} w_j \sum_{n=0}^{N-1} V_{n,j}(t_{m+1}) \operatorname{Re} \left\{ \tilde{\varphi} \left(\frac{n\pi}{b-a}, \varsigma_j, \sigma_m \right) e^{in\pi \frac{x_m - a}{b-a}} \right\},$$

with

$$(28) \quad V_{n,j}(t_{m+1}) := \frac{2}{b-a} \int_a^b v(x_{m+1}, \varsigma_j, t_{m+1}) \cos \left(n\pi \frac{x_{m+1} - a}{b-a} \right) dx_{m+1},$$

and

$$(29) \quad \tilde{\varphi}(\omega, \sigma_{m+1}, \sigma_m) := p_{\ln(\nu)}(\sigma_{m+1} | \sigma_m) \cdot \varphi(\omega; 0, e^{\sigma_{m+1}}, e^{\sigma_m}).$$

The kernel function $\tilde{\varphi}$ will be the only input which characterizes the Heston model. By combining the lengthy formulas of (9) and (14), the Bessel function present in $p_{\ln(\nu)}$ cancels with the Bessel function in the denominator of φ , leaving one Bessel term, $I_q \left(e^{\frac{1}{2}(\sigma_{m+1} + \sigma_m)} \cdot 2\kappa(v) e^{-\frac{1}{2}\gamma(v)\Delta t} \right)$, with $\gamma(v)$ given by (8):

$$v = \omega \left(\frac{\lambda\rho}{\eta} - \frac{1}{2} \right) + \frac{1}{2} i\omega^2(1 - \rho^2) \quad \text{and} \quad \kappa(v) = \frac{2\gamma(v)}{\eta^2(1 - e^{-\gamma(v)\Delta t})}.$$

Coefficients $V_{n,j}(t_{m+1})$ defined in (28) can be interpreted as the Fourier cosine series coefficients of the option value at time t_{m+1} . Expression $c_3(x_m, \sigma_m, t_m)$ in (27) thus becomes a scaled inner product of the Fourier cosine series coefficients of the option price and of the underlying density.

Finally, we interchange the summations in (27), which yields the discrete formula for the continuation value:

$$(30) \quad c_3(x_m, \sigma_m, t_m) = e^{-r\Delta t} \operatorname{Re} \left\{ \sum_{n=0}^{N-1} \beta_n(\sigma_m, t_m) e^{in\pi \frac{x_m - a}{b-a}} \right\},$$

where

$$(31) \quad \beta_n(\sigma_m, t_m) := \sum_{j=0}^{J-1} w_j V_{n,j}(t_{m+1}) \tilde{\varphi} \left(\frac{n\pi}{b-a}, \varsigma_j, \sigma_m \right).$$

Equation (30) expresses the continuation value at time t_m as a series expansion. The series coefficients, which depend only on the value of the variance (and not on the log-stock value) at time t_{m+1} , are (scaled) inner products of the cosine series coefficients of the option price at time t_{m+1} and the variance-dependent ChF $\tilde{\varphi}$.

Due to the use of a quadrature rule in the log-variance dimension, we compute on a log-variance grid. The same log-variance grid is employed for all time points, which gives

$$(32) \quad c_3(x_m, \varsigma_p, t_m) = e^{-r\Delta t} \operatorname{Re} \left\{ \sum_{n=0}^{N-1} \beta_n(\varsigma_p, t_m) \exp \left(in\pi \frac{x_m - a}{b-a} \right) \right\},$$

with

$$(33) \quad \beta_n(\varsigma_p, t_m) := \sum_{j=0}^{J-1} w_j V_{n,j}(t_{m+1}) \tilde{\varphi} \left(\frac{n\pi}{b-a}, \varsigma_j, \varsigma_p \right).$$

For x_m , however, no computational grid is needed since the price is constructed from a linear combination of cosine basis functions, in which the series coefficients do not depend on x_m itself. As such, x_m can be separated from the other variables; it is present only in the cosine functions. This enables us to derive an *analytic formula* for the series coefficients, as shown in the next subsection.

One of the advantages of this *spectral dimension* is that expression (30) is known for any value of $x_m \in \mathbb{R}$, not just for discrete values. So, one can determine the *early-exercise points* rapidly by solving

$$c_3(x_m, \varsigma_j, t_m) - g(x_m) = 0, \quad j = 0, 1, \dots, J-1,$$

with an efficient root-finding procedure, like Newton's method.

When the early-exercise points, $x^*(\sigma_m, t_m)$, have been determined, procedure (16) can be used to compute the Bermudan option price. More specifically, the following hold:

- At t_M : $v(x_M, \sigma_M, t_M) = g(x_M)$.
- At t_m , with $m = 1, 2, \dots, M - 1$:

$$(34) \quad \hat{v}(x_m, \sigma_m, t_m) = \begin{cases} g(x_m) & \text{for } x \in [a, x^*(\sigma_m, t_m)], \\ c_3(x_m, \sigma_m, m) & \text{for } x \in (x^*(\sigma_m, t_m), b] \end{cases}$$

for a put option, and

$$(35) \quad \hat{v}(x_m, \sigma_m, t_m) = \begin{cases} c_3(x_m, \sigma_m, m) & \text{for } x \in [a, x^*(\sigma_m, t_m)], \\ g(x_m) & \text{for } x \in (x^*(\sigma_m, t_m), b] \end{cases}$$

for a call option.

- At t_0 : $\hat{v}(x_0, \sigma_0, t_0) = c_3(x_0, \sigma_0, t_0)$.

\hat{v} denotes that we deal with *approximate option values*, due to the various approximations involved.

With the procedure above and expression (30), we can compute recursively $\hat{v}(x_0, \sigma_0, t_0)$ from $\hat{v}(x_M, \sigma_M, t_M)$ backwards in time.

However, a more efficient technique exists. Instead of reconstructing \hat{v} for each time point, we can recover the cosine series coefficients *using backward recursion*, and only at time t_0 do we apply (30) to reconstruct \hat{v} .

3.4. Backward recursion. We show that the cosine coefficients of $\hat{v}(x_1, \sigma_1, t_1)$ can be recovered recursively, with the FFT, from those of $\hat{v}(x_M, \sigma_M, t_M)$ in $O((M-1)JN \ell)$ operations, with $\ell = \max[\log_2(N), J]$.

We first discuss the final time point, t_M . Since the option price at the maturity date equals the payoff (which does not depend on time), one can derive an analytic expression for $V_{n,j}(t_M)$ using (28):

$$(36) \quad V_{n,j}(t_M) = \begin{cases} G_n(0, b) & \text{for call options,} \\ G_n(a, 0) & \text{for put options,} \end{cases}$$

where the G_n -functions are the cosine coefficients of the payoff function $g(y)$, i.e.,

$$(37) \quad G_n(l, u) := \frac{2}{b-a} \int_l^u g(y) \cos\left(n\pi \frac{y-a}{b-a}\right) dy.$$

Given that $g(x) = [\alpha \cdot K(e^x - 1)]^+$, we have

$$(38) \quad G_n(l, u) = \frac{2}{b-a} \alpha K [\chi_k(l^*, u^*) - \psi_k(l^*, u^*)], \quad \alpha = \begin{cases} 1 & \text{for a call,} \\ -1 & \text{for a put,} \end{cases}$$

with

$$(39) \quad l^* = \begin{cases} \max(l, 0), & \alpha = 1, \\ \min(l, 0), & \alpha = -1, \end{cases} \quad u^* = \begin{cases} \max(u, 0), & \alpha = 1, \\ \min(u, 0), & \alpha = -1, \end{cases}$$

and

$$(40) \quad \chi_k(l^*, u^*) := \int_{l^*}^{u^*} e^x \cos\left(n\pi \frac{x-a}{b-a}\right) dx,$$

$$(41) \quad \psi_k(l^*, u^*) := \int_{l^*}^{u^*} \cos\left(n\pi \frac{x-a}{b-a}\right) dx.$$

χ_k and ψ_k admit the following analytic solutions:

$$(42) \quad \begin{aligned} \chi_k(l, u) &= \frac{1}{1 + \left(\frac{n\pi}{b-a}\right)^2} \left[\cos\left(n\pi \frac{u-a}{b-a}\right) e^u - \cos\left(n\pi \frac{l-a}{b-a}\right) e^l \right. \\ &\quad \left. + \frac{n\pi}{b-a} \sin\left(n\pi \frac{u-a}{b-a}\right) e^u - \frac{n\pi}{b-a} \sin\left(n\pi \frac{l-a}{b-a}\right) e^l \right], \\ \psi_k(l, u) &= \begin{cases} \left[\sin\left(n\pi \frac{u-a}{b-a}\right) - \sin\left(n\pi \frac{l-a}{b-a}\right) \right] \frac{b-a}{n\pi}, & n \neq 0, \\ u-l, & n = 0. \end{cases} \end{aligned}$$

Note that the applicability of the COS method is not limited to plain vanilla options. Analytic solutions for the Fourier cosine coefficients of binary options have also been obtained in [7], so that discontinuous payoffs can also be dealt with highly efficiently.

Subsequently, we continue with time point t_{M-1} . By inserting $V_{n,j}(t_M)$ into (33), we obtain $\beta_n(\varsigma_p, t_{M-1})$ for $p = 0, 1, \dots, J-1$. With (32) one finds an analytic formula, $c_3(x_{M-1}, \varsigma_p, t_{M-1})$, for the continuation value at time t_{M-1} . By Newton’s method, we then solve $c_3(y, \varsigma_p, t_{M-1}) - g(y) = 0$ to determine the location of the early-exercise point, $y \equiv x^*(\varsigma_p, t_{M-1})$.

With the early-exercise point, $x^*(\varsigma_p, t_{M-1})$, known and $\hat{v}(x_{M-1}, \varsigma_p, t_{M-1})$ as in (34) or (35), we split the integral in (28) into two parts (for $p = 0, 1, \dots, J-1$):

$$\hat{V}_{k,p}(t_{M-1}) = \begin{cases} \hat{C}_{k,p}(x^*(\varsigma_p, t_{M-1}), b, t_{M-1}) + G_k(a, x^*(\varsigma_p, t_{M-1})) & \text{for a put,} \\ \hat{C}_{k,p}(a, x^*(\varsigma_p, t_{M-1}), t_{M-1}) + G_k(x^*(\varsigma_p, t_{M-1}), b) & \text{for a call,} \end{cases}$$

where \hat{V} , \hat{C} denote approximate values. The $\hat{C}_{k,p}$ represent the cosine coefficients of the continuation value:

$$(43) \quad \hat{C}_{k,p}(l, u, t_{M-1}) := \frac{2}{b-a} \int_l^u c_3(y, \varsigma_p, t_{M-1}) \cos\left(k\pi \frac{y-a}{b-a}\right) dy.$$

After replacing c_3 in (43) by the COS approximation, interchanging summation and integration, we obtain

$$(44) \quad \hat{C}_{k,p}(l, u, t_{M-1}) = e^{-r\Delta t} \operatorname{Re} \left\{ \sum_{n=0}^{N-1} \mathcal{M}_{k,n}(l, u) \beta_n(\varsigma_p, t_{M-1}) \right\},$$

with

$$(45) \quad \mathcal{M}_{k,n}(l, u) := \int_l^u \exp\left(in\pi \frac{y-a}{b-a}\right) \cos\left(k\pi \frac{y-a}{b-a}\right) dy.$$

Expression (45) can be obtained analytically.

The expressions above can be cast in an easy readable format in matrix/vector notation:

$$(46) \quad \hat{C}(l, u, t_{M-1}) = e^{-r\Delta t} \operatorname{Re} \{ \mathcal{M}(l, u) \mathbf{B}'(t_{M-1}) \},$$

where \mathbf{B}' indicates that the first row of matrix \mathbf{B} is multiplied by one half.

Matrix $\mathcal{M}(l, u)$ is an $N \times N$ matrix composed of elements from $\mathcal{M}_{k,n}(l, u)$, and matrix $\mathbf{B}(t_{M-1})$ is an $N \times J$ matrix, with J column vectors:

$$(47) \quad \mathbf{B}(t_{M-1}) = [\beta_0(t_{M-1}), \beta_1(t_{M-1}), \dots, \beta_{J-1}(t_{M-1})].$$

The column vectors (denoted by subscripts), $\beta_p(t_{M-1})$, are connected to the coefficients $\mathbf{V}(t_M)$, i.e., the matrix with elements $V_{n,j}(t_M)$, as follows:

$$(48) \quad \beta_p(t_{M-1}) = [\mathbf{V}(t_M) \cdot \tilde{\varphi}(\varsigma_p)] \mathbf{w},$$

where \mathbf{w} is a column vector (length J) with the quadrature weights and the (time-invariant) matrix $\tilde{\varphi}(\varsigma_p)$ is an $N \times J$ matrix with elements $\tilde{\varphi}(\frac{n\pi}{b-a}, \varsigma_j, \varsigma_p)$, as defined in (29). The operator “ \cdot ” in (48) denotes an elementwise matrix-matrix product.

From [8] we know that matrix $\mathcal{M}(l, u)$ can be written as the sum of a Hankel matrix, $\mathcal{M}_c(l, u)$, and a Toeplitz matrix, $\mathcal{M}_s(l, u)$. Because matrix-vector products with Hankel and Toeplitz matrices can be transformed into circular convolutions of two vectors, the FFT algorithm can be applied to achieve the $O(N \log_2(N))$ complexity in log-stock space. Details are given in [8].

Repeating the same computational procedure backwards in time, we can derive the equations that connect $\hat{\mathbf{V}}(t_{m-1})$ to $\hat{\mathbf{V}}(t_m)$, for $m = M - 1, M - 2, \dots, 2$:

$$(49) \quad \left\{ \begin{array}{l} \hat{\mathbf{V}}(t_m) := \begin{cases} \hat{C}(x^*(\varsigma_p, t_m), b, t_m) + G(a, x^*(\varsigma_p, t_m)) \text{ for a put,} \\ \hat{C}(a, x^*(\varsigma_p, t_m), t_m) + G(x^*(\varsigma_p, t_m), b) \text{ for a call,} \end{cases} \\ \hat{\beta}_j(t_{m-1}) := [\hat{\mathbf{V}}(t_m) \cdot \tilde{\varphi}(\varsigma_j)] \mathbf{w}, \\ \hat{\mathbf{B}}(t_{m-1}) := [\hat{\beta}_0(t_{m-1}), \hat{\beta}_1(t_{m-1}), \dots, \hat{\beta}_{J-1}(t_{m-1})], \\ \hat{C}(l, u, t_{m-1}) := e^{-r\Delta t} \operatorname{Re} \{ \mathcal{M}(l, u) \hat{\mathbf{B}}'(t_{m-1}) \}. \end{array} \right.$$

We continue the procedure until $\hat{\mathbf{V}}(t_1)$ is recovered, which is then inserted into (33) and (30) to get a *grid of option prices*, $\hat{v}(x_0, \varsigma_j, t_0)$, for $j = 0, 1, \dots, J - 1$.

Now, one can either use a *spline interpolation* to get the value of $\hat{v}(x_0, \sigma_0, t_0)$ from $\hat{v}(x_0, \varsigma_j, t_0)$ or, at the initial stage of the computation, *shift* the σ -grid, so that σ_0 lies exactly on the grid.

We summarize the backward recursion algorithm below.

ALGORITHM 1: Pricing Bermudan options under Heston's model.

Initialization:

- Find a_ν and b_ν by Newton's method (as described in section 2.3).
- Calculate $V(t_M)$ with the analytic formula (36).
- Prepare matrix $\tilde{\varphi}(\varsigma_j)$ for $j = 0, 1, \dots, J - 1$
(consisting of $\tilde{\varphi}(\frac{k\pi}{b-a}, \varsigma_j, \varsigma_p)$ as defined in (29)).

Main Loop to recover $\hat{V}(t_m)$ for $m = M - 1$ to 1:

- Determine the early-exercise point by applying Newton's method to $c_3(y, \varsigma_p, t_m) - g(y) = 0$.
- Calculate the first row and column of \mathcal{M}_s and \mathcal{M}_c as given in [8].
- For $j = 0, 1, \dots, J - 1$, calculate $\hat{\beta}_j(t_m) = [\hat{V}(t_m) \cdot \tilde{\varphi}(\varsigma_j)] \mathbf{w}$.
- Multiply the first element of $\hat{\beta}_j(t_m)$ by one half, giving $\hat{\beta}'_j(t_m)$.
- Compute, with the FFT algorithm, the column vectors of $\hat{C}(t_m)$, $e^{-r\Delta t} \operatorname{Re} \{ (\mathcal{M}_s + \mathcal{M}_c) \hat{\beta}'_j(t_{m-1}) \}$.
- Recover $\hat{V}(t_m)$ by (34) or (35).

Final step: Calculate $\hat{v}(x, \varsigma_j, t_0)$ by inserting $\hat{V}(t_1)$ into (33) and (30). Use spline interpolation to get $\hat{v}(x, \sigma_0, t_0)$.

Special attention should be given to the calculation of $\tilde{\varphi}(\omega, \sigma_{m+1}, \sigma_m)$. First, it involves a modified Bessel function of the first kind, which increases dramatically in value when $q \rightarrow -1$ and/or $\omega \rightarrow \infty$. The *scaled Bessel function* should be used instead. A robust package has been developed in [1, 2] with algorithms to compute $I_d^*(z) := \exp(-|\operatorname{Re}\{z\}|) I_d(z)$ with a complex-valued argument, z , and a real-valued order, d . As MATLAB (which we use here) incorporates this package for the MATLAB Bessel function, we replace $I_q(\cdot)$ by $e^{|\operatorname{Re}\{\cdot\}|} I_q^*(\cdot)$ during the computations.

The computation of the modified Bessel function costs significantly more (a factor of approximately 1000) CPU time than a simple multiplication, because the main part of the Bessel function algorithm is based on iterations. If the computation of the Bessel function costs \mathcal{A} times the number of operations needed for a multiplication, a matrix based on $\tilde{\varphi}(\frac{k\pi}{b-a}, \varsigma_q, \varsigma_j)$ would require $O(NJ^2\mathcal{A})$ operations to compute all matrix elements.

If one employs equidistant quadrature rules for the log-variance dimension, then for a given value of k , the input argument of the Bessel function is a function of the grid point combination, $\varsigma_q + \varsigma_j$, which gives rise to the Hankel matrix (if ς_j represents an equidistant grid). The favorable structure of a Hankel matrix enables us to determine only *one row and one column* of the $J \times J$ matrix for each value of k . The total number of operations needed is therefore reduced to $O(NJA)$. However, since the error convergence is much slower with equidistant quadrature rules, J should be set much larger than for Gaussian quadrature rules. We will discuss this trade-off effect in the section with numerical results.

With the considerations in the remarks above, the computational effort in the *initialization* step with *nonequidistant* quadrature rules is dominated by the computation of the Bessel function in matrix $\hat{\varphi}$, which is of order $O(ANJ^2)$.

The computations in the main loop of the algorithm are of order $O(MN \log_2(N)J^2)$,

dominated by the calculation of matrix $\hat{B}(t_{m-1})$. Since the computation of *vector* $\hat{\beta}_j(t_{m-1})$ costs $O(NJ)$ operations, the calculation of matrix $\hat{B}(t_{m-1})$ is of $O(NJ^2)$ complexity.

The direct computation of the matrix-matrix product in (46) would cost $O(N^2J)$ operations. The computational complexity of (46) is, however, $O(N \log_2(N)J)$, due to the special structure of matrix $\mathcal{M}(l, u)$ and the use of the FFT algorithm.

Therefore, the overall complexity is $O(\max[A, M \log_2(N)]NJ^2)$.

4. Discrete barrier options. Also, for discretely monitored barrier options, the pricing technique explained above can be used. It is even somewhat easier, as the barrier levels are known in advance, unlike the (time-dependent) early-exercise points, and need not be determined inside the recursion loop. In the following we give the pricing formula for barrier put options with *double barriers*.

For an “out” barrier put option with M monitoring dates, the pricing formula reads for $m = 0, 1, \dots, M - 1$ as

$$(50) \quad v(x_m, \sigma_m, t_m) = \begin{cases} \text{Rebate } r_b & \text{when knocked out,} \\ c(x_m, \sigma_m, t_m) & \text{otherwise,} \end{cases}$$

and

$$(51) \quad v(x_M, \sigma_M, t_M) = \begin{cases} \text{Rebate } r_b & \text{when knocked out,} \\ g(x_M) & \text{otherwise,} \end{cases}$$

where the continuation value is governed by (19), as for Bermudan options.

The option price at the maturity date, t_M , equals the payoff if the option is not knocked out (or knocked in); otherwise the option price equals the rebate. Following (28), the Fourier cosine coefficients of $v(x_m, \sigma_m, t_m)$, i.e., $V_{n,j}(t_M)$, satisfy

$$(52) \quad \begin{aligned} V_{n,j}(t_M) &= \frac{2}{b-a} \int_{[a,l] \cup [u,b]} r_b \cos\left(n\pi \frac{y-a}{b-a}\right) dy + \frac{2}{b-a} \int_l^u g(y) \cos\left(n\pi \frac{y-a}{b-a}\right) dy \\ &= \frac{2r_b}{b-a} (\psi_n(a, l) + \psi_n(u, b)) + G_n(l, u), \end{aligned}$$

with $g(y)$ defined earlier, l and u denote lower and upper barrier levels, respectively,¹ and the G_n -terms are the analytically known cosine coefficients of the payoff function $g(y)$, as given earlier.

At t_{M-1} the barrier levels split the integral in (34) or (35) into several parts:

$$(53) \quad \begin{aligned} \hat{V}_{k,p}(t_{M-1}) &= \frac{2r_b}{b-a} (\psi_k(a, l) + \psi_k(u, b)) + \frac{2}{b-a} \int_l^u c_3(y, \varsigma_p, t_{M-1}) \cos\left(k\pi \frac{y-a}{b-a}\right) dy \\ &= \frac{2r_b}{b-a} (\psi_k(a, l) + \psi_k(u, b)) + \hat{C}_{k,p}(l, u, t_{M-1}), \end{aligned}$$

where $\hat{C}_{k,p}$ are the cosine coefficients of the continuation value as given in (43).

¹For single-sided barrier options, one can simply apply the same method by setting $l = a$ or $u = b$.

We can repeat the derivation from before: We replace c_3 in (43) by the COS approximation and interchange the summation and the integration, which gives

$$(54) \quad \hat{C}(l, u, t_{M-1}) = e^{-r\Delta t} \operatorname{Re} \{ \mathcal{M}(l, u) \mathbf{B}'(t_{M-1}) \},$$

where, as before, the first row of matrix \mathbf{B} is multiplied by one half, and $\mathbf{B}(t_{M-1})$ is obtained as in (47) and (48). Matrix $\mathcal{M}(l, u)$ is an $N \times N$ matrix, which is *time invariant*, as l and u are a priori known barrier levels. As a result, this matrix $\mathcal{M}(l, u)$ (only two columns and two rows needed for the circular convolution) can be *precomputed*. Compared to Algorithm 1, the main difference is that the computation of this matrix is not in the main recursion loop.

Following the same procedure, we move backwards in time and find the equations that connect $\hat{V}(t_{m-1})$ with $\hat{V}(t_m)$ for $m = M-1, M-2, \dots, 2$. Having $\hat{V}(t_1)$ approximated, we insert it into (33) and (30) to obtain the option price $\hat{v}(x_0, \sigma_0, t_0)$.

5. Error analysis. As in [7, 8] we study here the convergence of the local error at each time lattice, as well as the propagation of the error from one time lattice to the next.

5.1. Local error. We first analyze the convergence of the *local* error

$$\epsilon(x_m, \sigma_m, t_m) := |c(x_m, \sigma_m, t_m) - c_3(x_m, \sigma_m, t_m)|.$$

We depart from (19) and denote the inner integral as $\vartheta(x_m, \sigma_{m+1}, \sigma_m)$, which actually satisfies a risk-neutral valuation formula and thus defines the continuation value at time t_m given σ_{m+1} and σ_m . For analysis purposes, we introduce an intermediate approximation after the truncation of the integration range of the outer integral by $[a_\nu, b_\nu]$:

$$(55) \quad c_0(x_m, \sigma_m, t_m) := e^{-r\Delta t} \int_{a_\nu}^{b_\nu} p_{\ln(\nu)}(\sigma_{m+1} | \sigma_m) \vartheta(x_m, \sigma_{m+1}, \sigma_m) d\sigma_{m+1}.$$

Since the option price is bounded on a bounded interval, we can assume that a positive number, δ_0 , exists with

$$\delta_0 = \sup [\vartheta(x_m, \sigma_{m+1}, \sigma_m)] \quad \forall \sigma_{m+1}, \sigma_m \in [a_\nu, b_\nu], \forall x_m \in [a, b].$$

It then follows that

$$|c - c_0| \leq \delta_0 e^{-r\Delta t} \int_{\mathbb{R} \setminus [a_\nu, b_\nu]} p_{\ln(\nu)}(\sigma_{m+1} | \sigma_m) d\sigma_{m+1},$$

which suggests that this truncation error depends purely on the decay to zero of the log-variance density function, far in the tails. One can expect larger truncation errors for the difficult parameter sets, like for $q \in (-1, 0]$ compared to $q \in (0, +\infty)$. We assume a positive number, $\delta_1(q)$, depending on q , to exist such that

$$(56) \quad e^{-r\Delta t} \int_{\mathbb{R} \setminus [a_\nu, b_\nu]} p_{\ln(\nu)}(\sigma_{m+1} | \sigma_m) d\sigma_{m+1} \leq \operatorname{TOL} \cdot \delta_1(q).$$

TOL in (56) appears because the size $[a_\nu, b_\nu]$ ensures that $p_{\ln(\nu)}(\sigma_{m+1} | \sigma_m) < \operatorname{TOL}$ for $\sigma_{m+1} \in \mathbb{R} \setminus [a_\nu, b_\nu]$. Collecting the information gives

$$|c - c_0| \leq \operatorname{TOL} \cdot \delta_1(q) \cdot \delta_0.$$

Another intermediate quantity is obtained by replacing $p_{x|\ln(\nu)}$ in (55) with the approximation by the Fourier cosine series expansion, i.e.,

$$(57) \quad \bar{c}(x_m, \sigma_m, t_m) = e^{-r\Delta t} \int_{a_\nu}^{b_\nu} p_{\ln(\nu)}(\sigma_{m+1}|\sigma_m) \tilde{\vartheta}(x_m, \sigma_{m+1}, \sigma_m) d\sigma_{m+1},$$

where $\tilde{\vartheta}$ is the *COS approximation* of ϑ :

$$\begin{aligned} \tilde{\vartheta}(x_m, \sigma_{m+1}, \sigma_m) := & \frac{2}{b-a} \int_a^b v(x_{m+1}, \sigma_{m+1}, t_{m+1}) \left[\sum_{n=0}^{N-1} \cos\left(n\pi \frac{x_{m+1}-a}{b-a}\right) \right. \\ & \left. \cdot \operatorname{Re} \left\{ \varphi\left(\frac{n\pi}{b-a}; 0, \sigma_{m+1}, \sigma_m\right) e^{in\pi \frac{x_m-a}{b-a}} \right\} \right] dx_{m+1}. \end{aligned}$$

The error analysis in [7] shows that the error due to the COS approximation,

$$\epsilon_{\cos}(N, a, b) := \sup \left[\left| \vartheta(x_m, \sigma_{m+1}, \sigma_m) - \tilde{\vartheta}(x_m, \sigma_{m+1}, \sigma_m) \right| \right] \quad \forall x_m \in [a, b], \forall \sigma_{m+1}, \sigma_m \in \mathbb{R},$$

converges *exponentially* in N for very smooth densities when the integration range $[a + x_m, b + x_m]$ is sufficiently wide. As such, we have

$$|c_0 - \bar{c}| = \epsilon_{\cos}(N, a, b) \left(e^{-r\Delta t} \int_{a_\nu}^{b_\nu} p_{\ln(\nu)}(\sigma_{m+1}|\sigma_m) d\sigma_{m+1} \right) \leq \epsilon_{\cos}(N, a, b).$$

The approximation c_3 defined in (30) can now be obtained by applying a quadrature rule to the integral of (57). Suppose that the (absolute) error from the quadrature rule is $\epsilon_Q(J)$. With the triangle inequality, it then follows that for all $x_m \in [a, b]$ and for all $\sigma_m, \sigma_{m+1} \in [a_\nu, b_\nu]$,

$$(58) \quad \begin{aligned} \epsilon(x_m, \sigma_m, t_m) = |c - c_3| & \leq |c - c_0| + |c_0 - \bar{c}| + |\bar{c} - c_3| \\ & \leq \text{TOL} \cdot \delta_0 \cdot \delta_1(q) + \epsilon_{\cos}(N, a, b) + \epsilon_Q(J) := \epsilon_{loc}. \end{aligned}$$

The local error thus consists of three parts:

1. truncation error from the log-variance domain, which depends on the decay rate to zero of the log-variance density, outside the truncation range;
2. quadrature error, which converges exponentially in J when a Gauss–Legendre quadrature rule is used (as the log-variance density belongs to C^∞);
3. COS approximation error, which converges exponentially in N when interval $[a, b]$ is set sufficiently wide.

5.2. Error propagation during recursion. In the backward recursion, we recovered the approximate Fourier cosine series coefficients $\hat{V}_{k,p}(t_m)$ instead of $V_{k,p}(t_m)$. Here, we will study the error $\epsilon_{k,p}(t_m) := |\hat{V}_{k,p}(t_m) - V_{k,p}(t_m)|$ and its evolution through time. We focus on a Bermudan put here.

Starting at t_M , $V(t_M)$ is exact since the option price at t_M is known analytically. At time t_{M-1} , an error, $\varepsilon_{k,p}(t_{M-1})$, exists because we replaced c by c_3 to determine $V_{k,p}(t_{M-1})$. Based on (43), we get

$$\varepsilon_{k,p}(t_{M-1}) = \frac{2}{b-a} \left| \int_{x^*(\varsigma_p, t_{M-1})}^b (c_3(y, \varsigma_p, t_{M-1}) - c(y, \varsigma_p, t_{M-1})) \cos \left(k\pi \frac{y-a}{b-a} \right) dy \right|.$$

The above integral can be seen as an inner product of function $(c_3 - c)$ and the cosine function, so that we can bound this error by the Cauchy–Schwarz inequality:

$$(\varepsilon_{k,p}(t_{M-1}))^2 \leq \frac{4}{(b-a)^2} \left[\int_{x^*(\varsigma_p, t_{M-1})}^b \epsilon^2(y, \sigma_{M-1}, t_{M-1}) dy \cdot \int_{x^*(\varsigma_p, t_{M-1})}^b \cos^2 \left(k\pi \frac{y-a}{b-a} \right) dy \right].$$

The early-exercise point always lies in $[a, b]$ so that $b - x^* < b - a$. With $\cos^2(x) \leq 1$, we find, for all k, p , that

$$(\varepsilon_{k,p}(t_{M-1}))^2 \leq \frac{4}{(b-a)^2} \int_{x^*(\varsigma_p, t_{M-1})}^b \epsilon^2(y, \sigma_m, t_m) dy \leq \frac{4}{b-a} \int_a^b \epsilon^2(y, \sigma_m, t_m) dy.$$

With (58) for all σ_m and y , we obtain

$$\varepsilon_{k,p}(t_{M-1}) \leq 2\epsilon_{loc}.$$

In the matrix max-norm, this reads as

$$\left\| \hat{V}(t_{M-1}) - V(t_{M-1}) \right\|_{\max} \leq 2\epsilon_{loc}.$$

In the following, we will prove, by induction, that if

$$(59) \quad \left\| \hat{V}(t_{m+1}) - V(t_{m+1}) \right\|_{\max} \sim O(\epsilon_{loc}),$$

then it will also hold for time t_m .

The final equation in (49) is equivalent to

$$\hat{C}_{k,q}(x^*(\varsigma_q), b, t_m) = \frac{2}{b-a} \int_{x^*(\varsigma_p, t_{M-1})}^b \hat{c}_3(y, \varsigma_q, t_m) \cos \left(k\pi \frac{y-a}{b-a} \right) dx_m,$$

where $\hat{c}_3(x_m, \sigma_m, t_m)$ is based on the same definition as $c_3(x_m, \sigma_m, t_m)$ in (27), except that $V_{n,j}(t_{m+1})$ is replaced by $\hat{V}_{n,j}(t_{m+1})$. As such, it holds that

$$\begin{aligned} & c_3(x_m, \sigma_m, t_m) - \hat{c}_3(x_m, \sigma_m, t_m) \\ &= e^{-r\Delta t} \sum_{j=0}^{J-1} w_j \sum_{n=0}^{N-1} \left(\hat{V}_{n,j}(t_{m+1}) - V_{n,j}(t_{m+1}) \right) \cdot \operatorname{Re} \left\{ \tilde{\varphi} \left(\frac{n\pi}{b-a}, \varsigma_j, \sigma_m \right) e^{in\pi \frac{x_m-a}{b-a}} \right\}. \end{aligned}$$

To analyze this error term, we decompose $\tilde{\varphi}$ using (29) and replace the $\text{Re}\{\cdot\}$ -term by P_n , defined in (23), which gives

$$(60) \quad c_3(x_m, \sigma_m, t_m) - \hat{c}_3(x_m, \sigma_m, t_m) = e^{-r\Delta t} \sum_{j=0}^{J-1} w_j p_{\ln(\nu)}(\varsigma_j | \sigma_m) \Theta(\varsigma_j, x_m, \sigma_m),$$

where

$$\Theta(\varsigma_j, x_m, \sigma_m) := \sum_{n=0}^{N-1} \left(\hat{V}_{n,j}(t_{m+1}) - V_{n,j}(t_{m+1}) \right) \cdot \left[P_n(\varsigma_j, x_m, \sigma_m) - \int_{\mathbb{R} \setminus [a,b]} p_{x|\ln(\nu)}(y | \varsigma_j, x_m, \sigma_m) \cos\left(n\pi \frac{y-a}{b-a}\right) dy \right].$$

From (21), we know that

$$\int_{\mathbb{R} \setminus [a,b]} p_{x|\ln(\nu)}(y | \varsigma_j, x_m, \sigma_m) \cos\left(n\pi \frac{y-a}{b-a}\right) dy \sim O(\text{TOL}_x).$$

As Θ can be viewed as an inner product of two vectors, we can apply the Cauchy–Schwarz inequality:

$$\Theta^2(\varsigma_j, x_m, \sigma_m) \leq \sum_{n=0}^{N-1} \varepsilon_{n,j}^2(t_{m+1}) \sum_{n=0}^{N-1} [P_n(\varsigma_j, x_m, \sigma_m) + O(\text{TOL}_x)]^2.$$

For smooth density functions, as we have in Heston’s model, the cosine series coefficients P_n converge exponentially in n . The sum, $\sum_{n=0}^{N-1} (P_n + O(\text{TOL}_x))^2$, is therefore a sum of a geometric series, which is thus bounded. We assume that a positive number, δ_3 , exists, which satisfies

$$\delta_3 := \sup \left[\sum_{n=0}^{N-1} [P_n(\vartheta_j, x_m, \sigma_m) + O(\text{TOL}_x)]^2 \right] \quad \forall x_m \in [a, b], \forall \sigma_m, \varsigma_j \in \mathbb{R}.$$

It then follows that

$$\Theta^2(\varsigma_j, x_m, \sigma_m) \leq \delta_3 \sum_{n=0}^{N-1} \varepsilon_{n,j}^2(t_{m+1}).$$

With (59), we can write $\varepsilon_{n,j}(t_{m+1}) \leq \sqrt{\delta_4} \epsilon_{loc}$ for some positive number δ_4 and find that

$$\Theta^2(\varsigma_j, x_m, \sigma_m) \leq \delta_3 \delta_4 N \epsilon_{loc}^2.$$

Returning to (60) and employing the Cauchy–Schwarz inequality give us

$$\begin{aligned} |c_3(x_m, \sigma_m, t_m) - \hat{c}_3(x_m, \sigma_m, t_m)| &\leq e^{-r\Delta t} \sqrt{\sum_{j=0}^{J-1} (w_j p_{\ln(\nu)}(\varsigma_j | \sigma_m))^2 \sum_{j=0}^{J-1} \Theta^2(\varsigma_j, x_m, \sigma_m)} \\ &\leq e^{-r\Delta t} \sqrt{\delta_3 \delta_4 \delta_5} \cdot \sqrt{JN} \cdot \epsilon_{loc}, \end{aligned}$$

where δ_5 is an upper bound for $\sum_{j=0}^{J-1} (w_j p_{\ln(\nu)}(\varsigma_j | \sigma_m))^2$ for all values of σ_m .

With the results above, error $\varepsilon_{k,q}(t_m)$ can be bounded as follows:

$$\begin{aligned} & \left| \hat{V}_{k,q}(x_m, \sigma_m, t_m) - V_{k,q}(x_m, \sigma_m, t_m) \right| = \left| \hat{C}_{k,q}(a, x^*(\varsigma_q, t_m), t_m) - C_{k,q}(a, x^*(\varsigma_q, t_m), t_m) \right| \\ & \leq \frac{2}{b-a} \sqrt{\int_{x^*(\varsigma_p, t_{M-1})}^b (c(y, \varsigma_q, t_m) - c_3(y, \varsigma_q, t_m))^2 dy} \sqrt{\int_{x^*(\varsigma_p, t_{M-1})}^b \cos^2 \left(k\pi \frac{y-a}{b-a} \right) dy} \\ (61) \quad & \leq 2e^{-r\Delta t} \sqrt{\delta_3 \delta_4 \delta_5} \cdot \sqrt{JN} \cdot \epsilon_{loc}. \end{aligned}$$

So, when ϵ_{loc} converges exponentially in both N and J , it holds that

$$\left\| \hat{V}(t_m) - V(t_m) \right\|_{\max} \sim O(\epsilon_{loc}).$$

The speed of convergence will, however, decrease when the number of monitoring dates increases, due to the increasing weighting term in (61). Larger values for N and J are required in that case. We will examine this via numerical experiments in the next sections.

5.3. Error analysis experiment. We check the error convergence analysis from section 5 by pricing discrete barrier options for which we set $l = a$ and $u = b$. This gives us European option prices, so we compute highly accurate reference values (accurate up to the eighth decimal place) by the European option pricing method from [7]. Since only a limited number of reference values are found in the literature, we use this special case to study the error convergence.

Three tests are extracted from [3], one relatively easy case, with $q > 0$, and two significantly more difficult cases for which $q \in [-1, 0]$:

- Test No. 1 ($q = 0.6$): $\eta = 0.5$, $\lambda = 5$, $\bar{\nu} = 0.04$, $T = 1$.
- Test No. 2 ($q = -0.84$): $\eta = 0.5$, $\lambda = 0.5$, $\bar{\nu} = 0.04$, $T = 1$.
- Test No. 3 ($q = -0.96$): $\eta = 1$, $\lambda = 0.5$, $\bar{\nu} = 0.04$, $T = 10$.

The computer used is a standard laptop with an Intel(R) 2.2GHz CPU and a 4-GB memory. The program is written in MATLAB.

Numerical methods for early-exercise or barrier options are usually either based on finite differences for PDEs [11] or on tree-based methods [13, 4]. Results with these techniques have, however, not yet been published for these parameter sets.

Other parameters to determine the values of the *put* ($\alpha = -1$) include

$$\rho = -0.9, \quad \nu_0 = 0.04, \quad S_0 = 100, \quad K = 100, \quad r = 0,$$

and we do not consider dividend payment here.

First, we analyze the error convergence in J for the Heston pricing methods with the Gauss–Legendre quadrature rule. We prescribe the predefined truncation error tolerances, TOL, in log-variance dimension as 10^{-4} , 10^{-6} , and 10^{-8} , respectively. The number of monitoring dates is set to 12, and for N we choose $N = 2^7$.

The results in Table 1 demonstrate that when N and J are sufficiently large (like $N = J = 2^7$), the truncation error, governed by “TOL,” dominates the overall error. For small values of TOL (like $\text{TOL} \leq 10^{-6}$) and N being fixed, a very fast error convergence in J is obtained (and the computational complexity is quadratic in J).

From the experiments of Test No. 1, we can conclude that for $q > 0$ highly accurate results are obtained within a fraction of a second.

Table 1

Convergence in J for Test No. 1 ($q = 0.6$) with $N = 2^7$, $M = 12$; the European option reference value is 7.5789038982.

$(J = 2^d)$ d	Fourier cosine expansion plus Gauss–Legendre rule					
	TOL = 10^{-4}		TOL = 10^{-6}		TOL = 10^{-8}	
	Time(sec)	Error	Time(sec)	Error	Time(sec)	Error
4	0.12	$-7.51 \cdot 10^{-3}$	0.12	$1.02 \cdot 10^{-2}$	0.12	1.41
5	0.43	$-3.95 \cdot 10^{-3}$	0.42	$-1.85 \cdot 10^{-5}$	0.40	$2.99 \cdot 10^{-5}$
6	1.69	$-3.95 \cdot 10^{-3}$	1.59	$-1.54 \cdot 10^{-5}$	1.54	$-6.41 \cdot 10^{-6}$
7	6.88	$-3.95 \cdot 10^{-3}$	7.07	$-1.34 \cdot 10^{-5}$	6.49	$-6.32 \cdot 10^{-7}$

We continue with the difficult test cases for which $q \rightarrow -1$. Especially for these difficult parameter sets the composite trapezoidal rule (as well as the composite Simpson rule) is not appealing, as it requires very large values of J to achieve the desired accuracy. The Gauss–Legendre rule can, however, produce satisfactory results for relatively small values of J . Therefore, we illustrate the results obtained by the Gauss–Legendre rule in log-variance dimension in Table 2.

Table 2

Convergence in J as $q \rightarrow -1$; Fourier cosine expansion plus Gauss–Legendre rule, $N = 2^8$, $M = 12$, TOL = 10^{-7} ; the European reference values are 6.2710582179 (Test No. 2) and 13.0842710701 (Test No. 3).

$(J = 2^d)$ d	Test No. 2 ($q = -0.84$)				Test No. 3 ($q = -0.96$)			
	Time(sec)			Error	Time(sec)			Error
	Total	Init.	Loop		Total	Init.	Loop	
6	3.03	2.85	0.18	5.63	3.11	2.93	0.18	-22.7
7	13.3	12.78	0.56	$6.89 \cdot 10^{-3}$	12.1	11.55	0.53	$-8.51 \cdot 10^{-2}$
8	56.4	52.32	4.07	$-2.12 \cdot 10^{-5}$	55.7	51.74	4.00	$-1.60 \cdot 10^{-3}$

Compared to Test No. 1, the absolute errors in Tests No. 2 and No. 3 are larger for the same N and the same J . When $q \rightarrow -1$, the left-side tail of the log-variance density function tends to converge slower to zero. As a result, the truncation range in the log-variance dimension is set very wide (by Newton’s method) to reach the same tolerance level, TOL. The wider the truncation range, the larger the values of J required for the same level of accuracy. However, the error convergence in J is still reasonably fast.

The results presented in Table 2 indicate that, as q approaches -1 , the initialization step dominates the overall computational time, in particular the expensive computation of the Bessel function. The computations in the main loop of the pricing algorithm cost less than 8 percent of the total time.

We also check the propagation of the error through time. For this, we fixed N and J and measured the error convergence for increasing values of M (presented in Table 3). We employ somewhat different values for J here to indicate that it does not need to be a power of 2. The results confirm that the local error grows only very slowly for $q > 0$ and somewhat faster for $q \in [-1, 0]$. The overall error can be further reduced by setting larger values for J and/or N . Doubling parameter M corresponds to doubling CPU time in the main loop, which is in accordance with the error analysis.

Table 3

Error propagation in M ; COS + Gauss-Legendre, TOL = 10^{-7} ; $N = 2^7$, $J = 100$ for Test No. 1 and $N = 2^8$, $J = 300$ for Test No. 2.

Test	M		
	10	20	40
No. 1 ($q = 0.6$)	$-2.14 \cdot 10^{-6}$	$-3.13 \cdot 10^{-6}$	$-4.92 \cdot 10^{-6}$
No. 2 ($q = -0.84$)	$-2.56 \cdot 10^{-5}$	$-2.71 \cdot 10^{-5}$	$-7.02 \cdot 10^{-4}$

6. Numerical results for Bermudan options. We will now consider Bermudan options and use Algorithm 1 to price them. With increasing values for the number of exercise dates, M , the prices of Bermudan options converge towards the equivalent American options. The M time lattices can be viewed as a discretization in time.

Tree-based methods that are used to price American options using M time steps thus return prices of the equivalent Bermudan options with M exercise dates. The same holds for other pricing methods: If M time steps are used in a path simulation for American options, then the price of a Bermudan option with M early-exercise dates is computed.

This insight enables us to take a reference value from the American option pricing literature here, with our choice of parameter M resembling the number of time steps used in a tree-based or Monte Carlo method.

Two parameter test sets are used here. One is chosen in the PDE-based finite differences literature, for example in [11], with $q > 0$; and the second is with $q \in [-1, 0]$, inspired by results with a tree-based method in [13]. The reference value for the first test case is available and accurate up to the sixth digit; see [11]. For the latter test, Bermudan reference values are not available. So we provide our results, which may serve as a reference test for future computations by other pricing methods.

The most commonly used test parameters for American options under the Heston dynamics in the literature read as follows:

- Test No. 4 ($q = 0.98$): $S_0 = \{8, 9, 10, 11, 12\}$, $K = 10$, $T = 0.25$, $r = 0.1$, $\lambda = 5$, $\eta = 0.9$, $\bar{\nu} = 0.16$, $\nu_0 = 0.0625$, and $\rho = 0.1$.

This gives $q > 0$, so that a very accurate and efficient pricing performance is expected. Results are presented in Table 4, where CPU time is measured for five different values of S_0 computed simultaneously. The convergence of the Bermudan options to the American option reference values is clearly visible.²

A negative correlation coefficient, ρ , is often observed in market data. A test example for a Bermudan put with this parameter and $q \in [-1, 0]$ was given in [13], where the parameters were set as follows:

- Test No. 5 ($q = -0.47$): $S_0 = \{90, 100, 110\}$, $K = 100$, $T = 0.25$, $r = 0.04$, $\lambda = 1.15$, $\eta = 0.39$, $\rho = -0.64$, $\bar{\nu} = 0.0348$, and $\nu_0 = 0.0348$.

However, reference values were not available in the paper, so that we provide our results as a reference in Table 5.

²Although it is not our main concern in this paper, one can obtain American option prices much more rapidly by extrapolating prices of Bermudan options with small values of M . Details are given in [12, 8].

Table 4

Errors of Test No. 4 ($q = 0.98$); *COS* + *Gauss-Legendre*; $N = 2^7$, $J = 2^7$, and $TOL = 10^{-7}$, plus reference values.

S_0	8	9	10	11	12	Time (sec)		
						Total	Init.	Loop
Ref. val.	2.000000	1.107621	0.520030	0.213677	0.082044			
$M = 10$	$-1.80 \cdot 10^{-2}$	$-4.79 \cdot 10^{-3}$	$-2.85 \cdot 10^{-3}$	$-1.31 \cdot 10^{-3}$	$-5.18 \cdot 10^{-4}$	6.9	6.34	0.57
$M = 20$	$-9.54 \cdot 10^{-3}$	$-2.39 \cdot 10^{-3}$	$-1.40 \cdot 10^{-3}$	$-6.65 \cdot 10^{-4}$	$-2.78 \cdot 10^{-4}$	7.5	6.36	1.13
$M = 40$	$-5.14 \cdot 10^{-3}$	$-1.07 \cdot 10^{-3}$	$-5.50 \cdot 10^{-4}$	$-2.54 \cdot 10^{-4}$	$-1.22 \cdot 10^{-4}$	8.9	6.57	2.32
$M = 80$	$-2.83 \cdot 10^{-3}$	$-2.86 \cdot 10^{-4}$	$2.75 \cdot 10^{-5}$	$5.42 \cdot 10^{-5}$	$-8.43 \cdot 10^{-7}$	14.1	7.35	6.70

Table 5

Results of Test No. 5 ($q = -0.47$); *COS* + *Gauss-Legendre*; $N = 2^8$, $J = 2^8$, and $TOL = 10^{-7}$.

M	S_0			Time (sec)		
	90	100	110	Total	Init.	Loop
20	9.9783714	3.2047434	0.9273568	68.9	58.2	10.7
40	9.9916484	3.2073345	0.9281068	81.9	59.3	22.6
60	9.9957789	3.2079202	0.9280425	93.2	59.4	33.8

7. Conclusions. In this paper, we have presented a robust and efficient approach for pricing Bermudan and discretely monitored barrier options under Heston's stochastic volatility model with a Fourier-based method. The near-singular problem in the left-side tail of the Heston variance density has been dealt with by a change of variables to the log-variance domain. A discrete pricing formula is derived by applying a Fourier series expansion technique to the log-stock dimension and a quadrature rule to the log-variance dimension. By means of an error analysis we have determined the various sources for the errors, which are verified by numerical experiments.

The pricing method exhibits a fast error convergence. Furthermore, the method is robust w.r.t. parameter variations. For pricing early-exercise options for which the parameters in the Heston model satisfy the Feller condition, the new solution method gives highly accurate option prices within a fraction of a second. The challenge was, however, to price options in case the Feller condition was not satisfied. The computation of the Bessel functions in the initialization step of the algorithm dominates the overall computation time in that case. The error convergence is then also highly satisfactory. Choosing approximately 128 points in the log-stock and in the log-variance dimension is usually sufficient for an error reduction of the order 10^{-4} , even if the Feller condition is not satisfied.

The method presented here may serve as a reference technique, which can also be used for other, more elaborate, stochastic volatility models, like Heston models with stochastic interest rates, as long as the ChF or the probability density function of the joint density of the state variables involved can be determined.

REFERENCES

- [1] D. E. AMOS, *A Subroutine Package for Bessel Functions of a Complex Argument and Nonnegative Order*, Sandia National Laboratory Report SAND85-1018, Sandia National Laboratories, Albuquerque, NM, 1985.

- [2] D. E. AMOS, *Algorithm 644: A portable package for Bessel functions of a complex argument and non-negative order*, ACM Trans. Math. Software, 12 (1986), pp. 265–273.
- [3] L. ANDERSEN, *Simple and efficient simulation of the Heston stochastic volatility model*, J. Comput. Finance, 11 (2008), pp. 1–42.
- [4] N. BELIAEVA AND S. NAWALKHA, *A Simple Approach to Pricing American Options under the Heston Stochastic Volatility Model*, working paper, 2010; available online from <http://ssrn.com/abstract=1107934>.
- [5] M. BROADIE AND O. KAYA, *Exact simulation of stochastic volatility and other affine jump diffusion processes*, Oper. Res., 54 (2006), pp. 217–231.
- [6] J. C. COX, J. E. INGERSOLL, AND S. A. ROSS, *A theory of the term structure of interest rates*, Econometrica, 53 (1985), pp. 385–407.
- [7] F. FANG AND C. W. OOSTERLEE, *A novel pricing method for European options based on Fourier-cosine series expansions*, SIAM J. Sci. Comput., 31 (2008), pp. 826–848.
- [8] F. FANG AND C. W. OOSTERLEE, *Pricing early-exercise and discrete barrier options by Fourier-cosine series expansions*, Numer. Math., 114 (2009), pp. 27–62.
- [9] W. FELLER, *Two singular diffusion problems*, Ann. of Math. (2), 54 (1951), pp. 173–182.
- [10] S. HESTON, *A closed-form solution for options with stochastic volatility with applications to bond and currency options*, Rev. Financ. Stud., 6 (1993), pp. 327–343.
- [11] K. ITO AND J. TOIVANEN, *Lagrange multiplier approach with optimized finite difference stencils for pricing American options under stochastic volatility*, SIAM J. Sci. Comput., 31 (2009), pp. 2646–2664.
- [12] R. LORD, F. FANG, F. BERVOETS, AND C. W. OOSTERLEE, *A fast and accurate FFT-based method for pricing early-exercise options under Lévy processes*, SIAM J. Sci. Comput., 30 (2008), pp. 1678–1705.
- [13] M. VELLEKOOP AND H. NIEUWENHUIS, *A tree-based method to price American options in the Heston model*, J. Comput. Finance, 13 (2009), pp. 1–21.