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

Fang Fang* and Cornelis W. Oosterlee†

May 4, 2010

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 [9, 10], and high-order quadrature rules in the other dimension. Error analysis and experiments confirm a fast error convergence.

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 [13], 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.

Pricing European options starts from the risk-neutral valuation formula, which appears as an integration of the product of a probability density and a payoff function. The integration is either performed by numerical integration rules, which involve the Fast Fourier Transform (FFT) algorithm to speed up the computation (a particular example is the Carr-Madan method [7]), or by advanced alternatives, like methods based on series expansions (such as the Fourier cosine expansion based COS method [9]).

All these 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 [13], 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

*Delft University of Technology, Delft Institute of Applied Mathematics, Delft, the Netherlands, E-mail: f.fang@ewi.tudelft.nl

†CWI - Centrum Wiskunde & Informatica, Amsterdam, the Netherlands, E-mail: c.w.oosterlee@cwi.nl, and Delft University of Technology, Delft Institute of Applied Mathematics, Delft, the Netherlands

model were obtained mainly for Monte Carlo simulation methods [6, 4]. However, it is well-known that the development of efficient simulation methods for pricing problems with early exercise features, as they are encountered for example when pricing Bermudan or discretely-monitored barrier options, is not a trivial task.

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 [9, 10] 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 non-central chi-square distribution. 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, highest computational speed is often 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 parameters sets for which the near-singular behavior matters, and tackle the problem by a transformation from the variance domain to the *log-variance domain*. Secondly, to solve the two-dimensional problem in a robust and efficient manner, we combine the Fourier cosine expansion from [10] with quadrature rules.

The paper is organized as follows. 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 Details

In this section we give some insight in 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 Heston Model 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) [13]:

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

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

where the three non-negative parameters, $\lambda, \bar{\nu}$ 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 [12, 8], with

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

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

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

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 [4] that one often finds $2\lambda\bar{v} << \eta^2$ 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.

A lot of recent research effort has been put in the development of efficient Monte Carlo methods, based on *exact path simulation and moment matching*, for the Heston dynamics. This has brought important insights, in particular in the underlying distributions that we will briefly review here.

The exact simulation method, developed in Broadie and Kaya [6], provides, next to an exact formula to sample the log-stock price, insight in the distribution for stochastic volatility models. Integration of (1) and (2) yields [6]:

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

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

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

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

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 non-central 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:

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

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 $\gamma(\upsilon)$ is defined by

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

In [6] the cumulative distribution function of the time-integrated variance is recovered numerically from the expression,

$$Pr\left(\int_s^t \nu_\tau d\tau \le x\right) = \frac{2}{\pi}\int_0^\infty \frac{\sin(ux)}{u}\mathrm{Re}\left\{\Phi(u)\right\}du, \quad (9)$$

($\mathrm{Re}\{\cdot\}$ denoting the real part of the expression in brackets) by means of the composite Trapezoidal rule. This un-biased simulation method requires a significant amount of CPU time [4], mainly because of the numerical inversion step.

Remark 2.1 (Fast inverse Fourier transform by a Fourier cosine expansion)**.** *Application of the composite Trapezoidal rule for (9) is time-consuming, because*

the ChF is highly oscillatory, which implies that large values for N are required for accuracy.

The Fourier inversion step can be accelerated by the use of a Fourier cosine series expansion [9], see also Remark 3.1. The Fourier expansion method belongs to the spectral methods and recovers a distribution function for all $x \in \mathbb{R}$, from a ChF. The error convergence is exponential for smooth functions, i.e. of $O\left(e^{\alpha N}\right)$ with some $\alpha < 0$.

2.2 The Left-Side Tail

As a first step to understand the near-singular behavior in the variance direction, we set up a series of numerical experiments to determine the behavior of the left-side tail. 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 non-negativeness 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.

As shown 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 $q >> 0$ the density values tend 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.

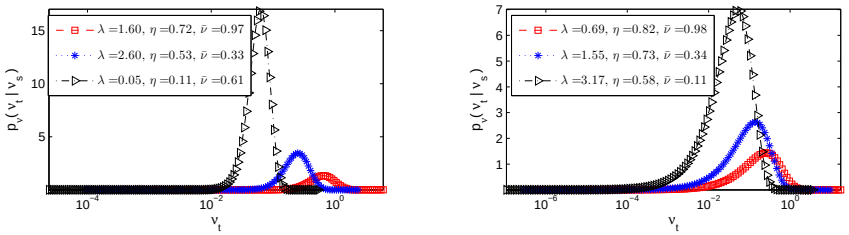
In a recent paper, [4], several challenging test cases, based on different values of λ,η and $\bar{\nu}$, were illustrated. For all those test cases we find $q \approx -0.96$, which indeed is an indication of difficult tests, see Figure 1.

The fact that q determines the decay rate of the densities' left tail can be understood if we take a closer look at Equation (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., from monotonically increasing they become monotonically decreasing.

2.3 Transformation to Log-Variance Process

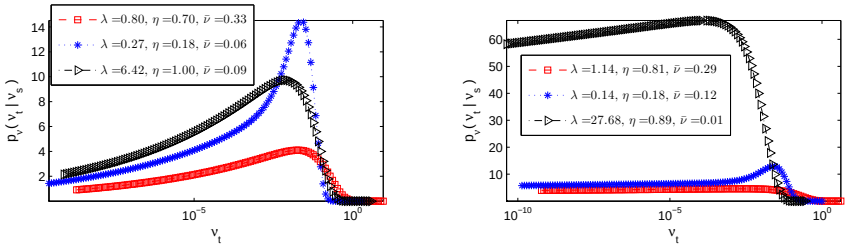
Based on the insights in the previous subsections, we propose here a solution strategy for the problem of the left-side tail: We transform the problem from the variance domain to the *log-variance domain*.

By the change of variables, the density of the log-variance process, based on



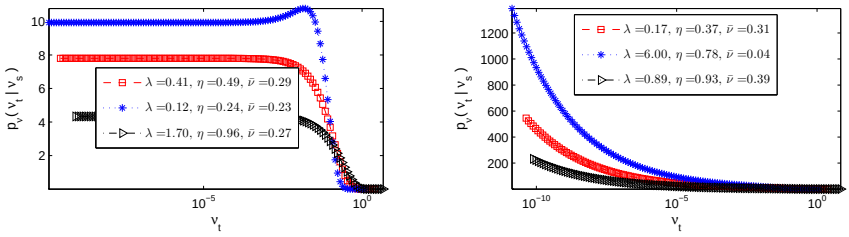
(a) $q = 5$

(b) $q = 1$



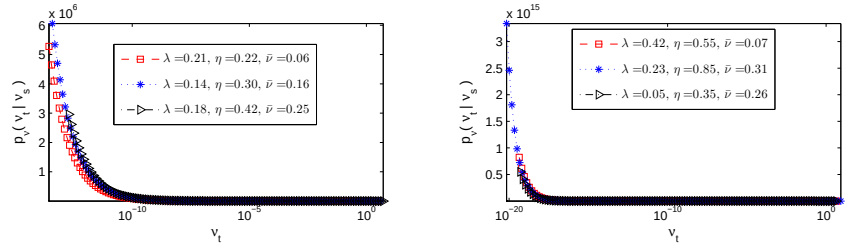
(c) $q = 0.1$

(d) $q = 0.01$



(e) $q = 0$

(f) $q = -0.2$



(g) $q = -0.5$

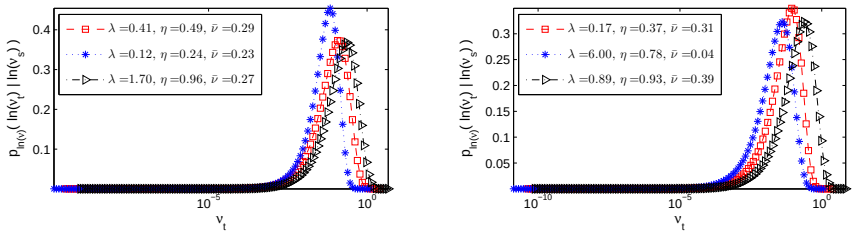
(h) $q = -0.8$

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

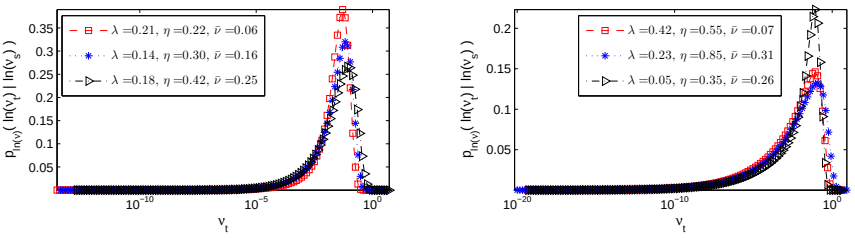
(3), reads:

$$p_{\ln(\nu)}\left(\sigma_t|\sigma_s\right) = \zeta e^{-\zeta(e^{\sigma_s}e^{-\lambda(t-s)}+e^{\sigma_t})}\left(\frac{e^{\sigma_t}}{e^{\sigma_s}e^{-\lambda(t-s)}}\right)^{\frac{q}{2}} e^{\sigma_t} I_q\left(2\zeta e^{-\frac{1}{2}\lambda(t-s)}\sqrt{e^{\sigma_s}e^{\sigma_t}}\right), \quad (10)$$

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 current time.

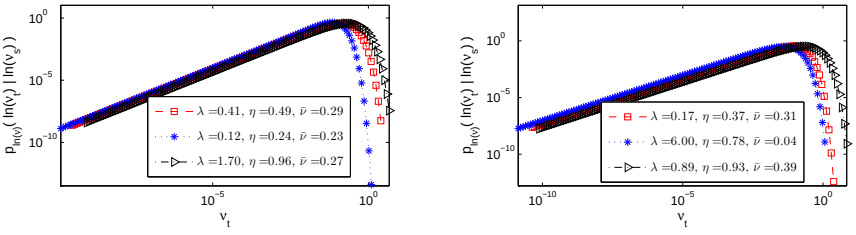


(a) $q = 0$ (b) $q = -0.2$

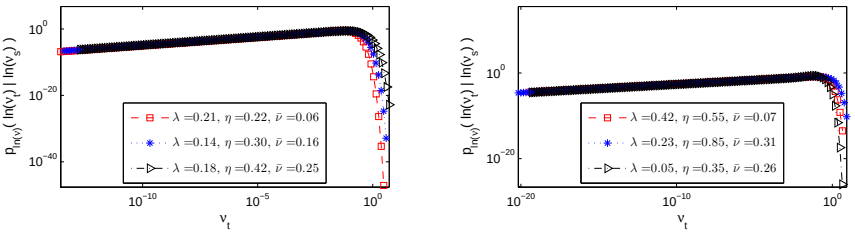


(c) $q = -0.5$ (d) $q = -0.8$

Figure 2: Decay rate of the left tail of the *log-variance density* as q approaches -1 from above



(a) $q = 0$ (b) $q = -0.2$



(c) $q = -0.5$ (d) $q = -0.8$

Figure 3: Decay rate *in log-scale* of the left tail of the *log-variance density* as q approaches -1 from above

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 $\sigma_t \to -\infty$. It is shown in Figure 2 that the 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 3, that for $q \in [-1, 0]$ the left tails of the densities do

not increase significantly in value anymore. Instead, these tails decay to zero rapidly as $\sigma_t \to -\infty$, although the decay rate decreases as q approaches -1. In Figures 2 and 3 we have only shown the problematic cases from Figure 1; the left-side tails of the cases with $q > 0$ also decay very well for the log-variance process, of course.

Remark 2.2 (Truncation range)**.** *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 [9, 16], we propose to use Newton's method to determine the interval boundaries, according to a pre-defined error tolerance, TOL. In accordance with this tolerance, the stopping criteria of the Newton method reads $p_{\ln(\nu)}(x|\sigma_0;T) < TOL$ *for* $x \in \mathbb{R}\backslash[a_\nu, b_\nu]$.

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:*

$$\frac{dp_{\ln(\nu)}(\sigma_t|\sigma_s)}{d\sigma_t} = -\left[(-\zeta e^{\sigma_t} - q - 1)\, I_q\left(2\sqrt{\zeta e^{\sigma_t} u}\right) - I_{q+1}\left(2\sqrt{\zeta e^{\sigma_t} u}\right)\right] \cdot \zeta e^{-u-\zeta e^{\sigma_t}+\sigma_t} \cdot \left(\frac{\zeta e^{\sigma_t}}{u}\right)^{q/2}, \tag{11}$$

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. [4],*

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

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* $[a_\nu, b_\nu]$*:*

$$[a_\nu^0, b_\nu^0] = \left[\ln(\mathbb{E}(\nu_t)) - \frac{5}{1+q},\ \ln(\mathbb{E}(\nu_t)) + \frac{2}{1+q}\right]. \tag{12}$$

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 from 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

$$p_{x,\nu}(x_t, \nu_t|x_s, \nu_s) = p_{x|\nu}(x_t|\nu_t, x_s, \nu_s) \cdot p_\nu(\nu_t|\nu_s), \tag{13}$$

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

$$p_{x,\ln(\nu)}(x_t, \sigma_t | x_s, \sigma_s) = p_{x|\ln(\nu)}(x_t | \sigma_t, x_s, \sigma_s) \cdot p_{\ln(\nu)}(\sigma_t | \sigma_s), \tag{14}$$

where $p_{x|\ln(\nu)}$ denotes the probability density of 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 (10) 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 characteristic function, $\varphi(\omega; x_s, \sigma_t, \sigma_s)$, based on (6):

$$\begin{aligned}
\varphi(\omega; x_s, \sigma_t, \sigma_s) &:= \mathbb{E}_s \left[\exp\left(i\omega x_t | \sigma_t \right) \right] \\
&= \exp\left(i\omega \left[x_s + \mu(t-s) + \frac{\rho}{\eta} \left(e^{\sigma_t} - e^{\sigma_s} - \lambda\bar{\nu}(t-s) \right) \right] \right) \cdot \\
&\quad \Phi\left(\omega \left(\frac{\lambda\rho}{\eta} - \frac{1}{2} \right) + \frac{1}{2} i\omega^2 (1-\rho^2); e^{\sigma_t}, e^{\sigma_s} \right),
\end{aligned} \tag{15}$$

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

$$v(x_s, \sigma_s, s) = e^{-r(t-s)} \mathbb{E}_s^{\mathbb{Q}} \left[v(x_t, \sigma_t, t) \right]. \tag{16}$$

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 (16). 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 $\mathcal{T} := \{t_m, t_m < t_{m+1} | m = 0, 1, \cdots, M\}$, with $t_M \equiv T$ and $\Delta t := t_{m+1} - t_m$, the Bermudan option pricing formula reads

$$v(x_{t_m}, \sigma_{t_m}, t_m) = \begin{cases} g(x_{t_m}, t_m) & \text{for } m = M; \\ \max\left[c(x_{t_m}, \sigma_{t_m}, t_m), g(x_{t_m}, t_m)\right] & \text{for } m = 1, 2, \cdots, M-1; \\ c(x_{t_m}, \sigma_{t_m}, t_m) & \text{for } m = 0, \end{cases} \tag{17}$$

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

$$c(x_m, \sigma_m, t_m) = e^{-r\Delta t} \mathbb{E}^{\mathbb{Q}}_{t_m}\left[v(x_{m+1}, \sigma_{m+1}, t_{m+1})\right], \tag{18}$$

which can be written as:

$$\begin{aligned} c(x_m, \sigma_m, t_m) &= e^{-r\Delta t} \cdot \\ &\int_{\mathbb{R}}\int_{\mathbb{R}} v(x_{m+1}, \sigma_{m+1}, t_{m+1}) p_{x,\ln(\nu)}\left(x_{m+1}, \sigma_{m+1} \mid x_m, \sigma_m\right) d\sigma_{m+1} dx_{m+1}. \end{aligned} \tag{19}$$

With (14) we get:

$$\begin{aligned} c(x_m, \sigma_m, t_m) = e^{-r\Delta t} &\cdot \\ \int_{\mathbb{R}} \Big[\int_{\mathbb{R}} & v(x_{m+1}, \sigma_{m+1}, t_{m+1}) p_{x|\ln(\nu)}\left(x_{m+1} | \ \sigma_{m+1}, x_m, \sigma_m\right) dx_{m+1} \Big] \\ & p_{\ln(\nu)}\left(\sigma_{m+1} | \sigma_m\right) d\sigma_{m+1}. \end{aligned} \tag{20}$$

Equations (17) and (20) define the problem we would like to solve numerically. The inner integral in (20) 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

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

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 characteristic functions. 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 (20).

The key idea of the COS method [9] 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 characteristic function. We now describe how to recover the density function $p_{x|\ln(\nu)}$ in (20) by the COS method.

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

$$\int_a^b p_{x|\ln(\nu)}(x_{m+1}|\sigma_{m+1}, x_m, \sigma_m)dy \ \leq \mathrm{TOL}_x, \tag{21}$$

for some pre-defined error tolerance TOL_x. In [9] this interval is defined as

$$[a,b] := [\xi_1 - 12\sqrt{|\xi_2|}, \xi_1 + 12\sqrt{|\xi_2|}], \tag{22}$$

where ξ_n denotes the n-th cumulant of the log-stock process. With an integration interval $[a,b]$ satisfying (22), we recover the probability density by its Fourier cosine series expansion:

$$p_{x|\ln(\nu)}(x_{m+1}|\sigma_{m+1}, x_m, \sigma_m) = \sum_{n=0}^{\infty}{}' P_n(\sigma_{m+1}, x_m, \sigma_m) \cos\left(n\pi \frac{x_{m+1}-a}{b-a}\right). \tag{23}$$

$\sum'$ 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 (23), 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 characteristic function and are therefore known, i.e.

$$P_n(\sigma_{m+1}, x_m, \sigma_m) \approx \frac{2}{b-a}\mathrm{Re}\left\{\varphi\left(\frac{n\pi}{b-a}; x_m, \sigma_{m+1}, \sigma_m\right) \ e^{-in\pi\frac{a}{b-a}}\right\}, \tag{24}$$

with $\varphi(\theta; x, \sigma_{m+1}, \sigma_m)$ given by (15).

The error in this approximation is related to TOL_x, as analyzed in [9], and Equation (24) approximates the P_n with machine accuracy if $[a,b]$ is sufficiently wide. Subsequently, we truncate the series summation in (23).

From Fourier theory, we know that cosine series of functions belonging to $\mathbb{C}^\infty([a,b] \subset \mathbb{R})$, with non-zero derivatives, converge exponentially with respect to the number of terms in the series, so that the series can be truncated without loosing accuracy. By replacing P_n in (23) by (24) and truncating the series by N terms, one obtains a *semi-analytic formula* which accurately approximates the probability density:

$$p_{x|\ln(\nu)}(x_{m+1}|\sigma_{m+1}, x_m, \sigma_m) = \sum_{n=0}^{N-1}{}' \frac{2}{b-a}\mathrm{Re}\left\{\varphi\left(\frac{n\pi}{b-a}; 0, \sigma_{m+1}, \sigma_m\right) \ e^{in\pi\frac{x_m-a}{b-a}}\right\} \cos\left(n\pi \frac{x_{m+1}-a}{b-a}\right) + \epsilon_{cos}. \tag{25}$$

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 with respect to N, provided that the truncation range is set sufficiently wide (proof is given in [9]).

Remark 3.1 (Recover a CDF)**.** *The COS method can also be used to recover a cumulative probability distribution, $F(x)$: We simply insert the COS reconstruction of the density $f(t)$ into the definition integral of the cumulative probability after truncating the integration range, i.e.*

$$\begin{aligned} F(x) &= \int_{-\infty}^{x} f(t)dt \approx \int_{b}^{x} f(t)dt \\ &\approx \int_{b}^{x} {\sum_{n=0}^{N-1}}' \frac{2}{b-a} Re\left\{\varphi\left(\frac{n\pi}{b-a}\right)\right\} \cos\left(n\pi\frac{t-a}{b-a}\right) dt \\ &= {\sum_{n=0}^{N-1}}' \frac{2}{b-a} Re\left\{\varphi\left(\frac{n\pi}{b-a}\right)\right\} \int_{b}^{x} \cos\left(n\pi\frac{t-a}{b-a}\right) dt \\ &= {\sum_{n=0}^{N-1}}' \frac{2}{b-a} Re\left\{\varphi\left(\frac{n\pi}{b-a}\right)\right\} \psi_n(b,x), \end{aligned}$$

where $\psi_n(l,u)$ is given in (63).

3.3 Discrete Fourier-based Pricing Formula

Equation (17) shows that the option price at time t_0 is a continuation value, which, as indicated by (20), depends on the continuation values at the times $t_1, t_2, \cdots, t_M$. The option price at time t_0 can be recovered by recursion, backwards in time. This is the same approach as in [10], 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 (12) and (22), 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

$$\begin{aligned} c_1(x_m, \sigma_m, t_m) := e^{-r\Delta t} \cdot \\ \int_{a_\nu}^{b_\nu} \Big[\int_{a}^{b} v(x_{m+1}, \sigma_{m+1}, t_{m+1}) p_{x|\ln(\nu)}(x_{m+1}|\, \sigma_{m+1}, x_m, \sigma_m)\, dx_{m+1} \Big] \\ p_{\ln(\nu)}(\sigma_{m+1}|\sigma_m)\, d\sigma_{m+1}. \end{aligned} \tag{26}$$

(We use the notation $c_i, i = 1, \ldots, 3$ 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). In the latter case since the ChF of $p_{\ln(\nu)}$ is not known, one would have to use a numerical method to retrieve the series coefficients for a series reconstruction, which would add additional computational costs. However, since $p_{\ln(\nu)}$ itself is known analytically, we apply a

J-point quadrature integration rule (like Gauss-Legendre quadrature, composite Trapezoidal rule, etc.) to the outer integral, which gives

$$c_2(x_m,\sigma_m,t_m) := e^{-r\Delta t}\sum_{j=0}^{J-1} w_j \cdot p_{\ln(\nu)}(\varsigma_j|\sigma_m)\cdot \Big[\int_a^b v(x_{m+1},\varsigma_j,t_{m+1}) p_{x|\ln(\nu)}\left(x_{m+1}|\,\varsigma_j,x_m,\sigma_m\right) dx_{m+1}\Big]. \tag{27}$$

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

Remark 3.2 (Which quadrature rule to use?)**.** *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. However, the computational complexity of the method can be greatly reduced by the Trapezoidal rule, due to a special matrix structure which results after discretization on an equidistant grid. We will come back to this issue.*

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 (25), and interchange the summation over n with the integration over x_{m+1} to obtain:

$$c_3(x_m,\sigma_m,t_m) := e^{-r\Delta t}\sum_{j=0}^{J-1} w_j {\sum_{n=0}^{N-1}}' V_{n,j}(t_{m+1})\mathrm{Re}\left\{\tilde{\varphi}\left(\frac{n\pi}{b-a},\varsigma_j,\sigma_m\right)e^{in\pi\frac{x_m-a}{b-a}}\right\}, \tag{28}$$

with

$$V_{n,j}\left(t_{m+1}\right) := \frac{2}{b-a}\int_a^b v(x_{m+1},\varsigma_j,t_{m+1})\cos\left(n\pi\frac{x_{m+1}-a}{b-a}\right)dx_{m+1}, \tag{29}$$

and

$$\tilde{\varphi}(\omega,\sigma_{m+1},\sigma_m) := p_{\ln(\nu)}(\sigma_{m+1}|\sigma_m)\cdot\varphi\left(\omega;0,e^{\sigma_{m+1}},e^{\sigma_m}\right). \tag{30}$$

The kernel function $\tilde{\varphi}$ will be the only input which characterizes the Heston model. By combining the lengthy formulas of (10) and (15), the Bessel function present in $p_{\ln(\nu)}$ cancels with the Bessel function in the denominator of φ, leaving one Bessel-term, $I_q\left(e^{\frac{1}{2}(\sigma_{m+1}+\sigma_m)}\cdot 2\kappa(\upsilon)e^{-\frac{1}{2}\gamma(\upsilon)\Delta t}\right)$ with $\gamma(\upsilon)$ given by (8),

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

Coefficients $V_{n,j}\left(t_{m+1}\right)$ defined in (29) 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 (28) 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 (28) which yields the discrete formula for the continuation value:

$$c_3(x_m, \sigma_m, t_m) = e^{-r\Delta t}\mathrm{Re}\left\{ \sum_{n=0}^{N-1}{}' \beta_n(\sigma_m, t_m) e^{in\pi \frac{x_m - a}{b-a}} \right\}, \tag{31}$$

where

$$\beta_n(\sigma_m, t_m) := \sum_{j=0}^{J-1} w_j V_{n,j}(t_{m+1}) \tilde{\varphi}\left(\frac{n\pi}{b-a}, \varsigma_j, \sigma_m \right). \tag{32}$$

Equation (31) 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 characteristic function $\tilde{\varphi}$.

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:

$$c_3(x_m, \varsigma_p, t_m) = e^{-r\Delta t}\mathrm{Re}\left\{ \sum_{n=0}^{N-1}{}' \beta_n(\varsigma_p, t_m) \exp\left(in\pi \frac{x_m - a}{b-a} \right) \right\}, \tag{33}$$

with

$$\beta_n(\varsigma_p, t_m) := \sum_{j=0}^{J-1} w_j V_{n,j}(t_{m+1}) \tilde{\varphi}\left(\frac{n\pi}{b-a}, \varsigma_j, \varsigma_p \right). \tag{34}$$

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 only present 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 (31) 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

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

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

When the early-exercise points, $x^*(\sigma_m, t_m)$, have been determined, Procedure (17) can be used to compute the Bermudan option price. More specifically:

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

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

  for a put option, and

$$\hat{v}(x_m, \sigma_m, t_m) = \begin{cases} c_3(x_m, \sigma_m, m) & \text{for } x \in [a, x^*(\sigma_m, t_m)] \\ g(x_m) & \text{for } x \in (x^*(\sigma_m, t_m), b] \end{cases} \tag{36}$$

  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 (31), 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 we apply (31) to reconstruct $\hat{v}$.

3.4 Backward Recursion

In this subsection 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 $O\left((M-1)JN\ \ell\right)$ operations, with $\ell = \max\left[\log_2(N), J\right]$.

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 (29):

$$V_{n,j}(t_M) = \begin{cases} G_n(0, b), \text{ for call options} \\ G_n(a, 0), \text{ for put options,} \end{cases} \tag{37}$$

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

$$G_n(l, u) := \frac{2}{b-a} \int_l^u g(y) \cos\left(n\pi \frac{y-a}{b-a}\right) dy, \tag{38}$$

with

$$g(y) = \left[\alpha K \left(e^y - 1\right)\right]^+, \quad \alpha = \begin{cases} 1, & \text{for a call option} \\ -1, & \text{for a put option.} \end{cases} \tag{39}$$

The analytic solution for $G_n(l, u)$ is presented in Appendix A.

Subsequently, we continue with time point t_{M-1}. By inserting $V_{n,j}(t_M)$ into (34), we obtain $\beta_n(\varsigma_p, t_{M-1})$ for $p = 0, 1, \cdots, J-1$. With (33) 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 early-exercise point, $x^*(\varsigma_p, t_{M-1})$, known and $\hat{v}(x_{M-1}, \varsigma_p, t_{M-1})$ as in (35) or (36), we split the integral in (29) in two parts (for $p = 0, 1, \cdots, J-1.$):

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

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

$$\hat{C}_{k,p}(l, u, t_{M-1}) := \frac{2}{b-a} \int_l^u c_3(y, \varsigma_p, t_{M-1}) \cos\left(k\pi \frac{y-a}{b-a}\right) dy. \tag{40}$$

For the exact cosine coefficient of the continuation value, $C_{k,p}$, we should have used c from (40), instead of the COS approximation c_3 from (33).

After replacing c_3 in (40) by the COS approximation, interchanging summation and integration, we obtain

$$\hat{C}_{k,p}(l,u,t_{M-1}) = e^{-r\Delta t}\,\mathrm{Re}\left\{\sum_{n=0}^{N-1}{}'\,\mathcal{M}_{k,n}(l,u)\beta_n(\varsigma_p,t_{M-1})\right\}, \tag{41}$$

with

$$\mathcal{M}_{k,n}(l,u) := \int_l^u \exp\left(in\pi\frac{y-a}{b-a}\right)\cos\left(k\pi\frac{y-a}{b-a}\right)dy. \tag{42}$$

Expression (42) can be obtained analytically.

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

$$\hat{\mathrm{C}}(l,u,t_{M-1}) = e^{-r\Delta t}\,\mathrm{Re}\left\{\mathcal{M}(l,u)\mathrm{B}'(t_{M-1})\right\}, \tag{43}$$

where B' indicates that the first row of matrix 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 $\mathrm{B}(t_{M-1})$ is an $N\times J$ matrix, with J column vectors:

$$\mathrm{B}(t_{M-1}) = \left[\beta_0(t_{M-1}),\beta_1(t_{M-1}),\cdots,\beta_{J-1}(t_{M-1})\right]. \tag{44}$$

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

$$\beta_p(t_{M-1}) = \left[\mathrm{V}(t_M)\cdot\tilde{\varphi}(\varsigma_p)\right]\mathbf{w}, \tag{45}$$

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 as elements $\tilde{\varphi}\left(\frac{n\pi}{b-a},\varsigma_j,\varsigma_p\right)$, as defined in (30). The operator "$\cdot$" in (45) denotes an element-wise matrix-matrix product.

From [10] 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 Appendix A.

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

$$\begin{cases} \hat{\mathrm{V}}(t_m) := \begin{cases} \hat{\mathrm{C}}(x^*(\varsigma_p,t_m),b,t_m) + \mathrm{G}(a,x^*(\varsigma_p,t_m)) & \text{for a put} \\ \hat{\mathrm{C}}(a,x^*(\varsigma_p,t_m),t_m) + \mathrm{G}(x^*(\varsigma_p,t_m),b) & \text{for a call} \end{cases} \\ \hat{\beta}_j(t_{m-1}) := \left[\hat{\mathrm{V}}(t_m)\cdot\tilde{\varphi}(\varsigma_j)\right]\mathbf{w} \\ \hat{\mathrm{B}}(t_{m-1}) := \left[\hat{\beta}_0(t_{m-1}),\hat{\beta}_1(t_{m-1}),\cdots,\hat{\beta}_{J-1}(t_{m-1})\right] \\ \hat{\mathrm{C}}(l,u,t_{m-1}) := e^{-r\Delta t}\,\mathrm{Re}\left\{\mathcal{M}(l,u)\hat{\mathrm{B}}'(t_{m-1})\right\} \end{cases} \tag{46}$$

We continue the procedure until $\hat{\mathrm{V}}(t_1)$ is recovered, which is then inserted into (34) and (31) to get a *grid of option prices*, $\hat{v}(x_0, \varsigma_j, t_0)$, for $j = 0, 1, \cdots, J-1$.

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;
- Calculate $\mathrm{V}(t_M)$ with the analytic formula;
- Prepare matrix $\tilde{\varphi}(\varsigma_j)$ for $j = 0, 1, \cdots, J-1$.

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

- Determine early-exercise point by Newton's method;
- Calculate the first row and column of $\mathrm{M_s}$ and $\mathrm{M_c}$.
- For $j = 0, 1, \cdots, J-1$, calculate $\hat{\beta}_j(t_m) = \left[\hat{\mathrm{V}}(t_m) \cdot \tilde{\varphi}(\varsigma_j)\right] \mathbf{w}$.
- Multiply the first element of $\hat{\beta}_j(t_m)$ by one-half.
- Compute the column vectors of $\hat{\mathrm{C}}(t_m)$, $e^{-r\Delta t}\,\mathrm{Re}\left\{\mathcal{M}\hat{\beta'}_j(t_{m-1})\right\}$, using the FFT algorithm;
- Recover $\hat{\mathrm{V}}(t_m)$ by (35) or (36).

Final step: Calculate $\hat{v}(x, \varsigma_j, t_0)$ by inserting $\hat{\mathrm{V}}(t_1)$ into (34) and (31). Use spline interpolation to get $\hat{v}(x, \sigma_0, t_0)$.

Remark 3.3 (Multiple values of S_0)**.** *Due to the use of the spectrally-oriented discretization in the log-stock dimension, the cosine-coefficients of* $\hat{\mathrm{V}}(t_m)$ *do not depend on the initial value of asset prices. Only in the final step, one needs to insert an initial value,* S_0*, into (34) and (31) to get the option price. If necessary, the method could thus price multiple options that only vary in the value of* S_0 *simultaneously, with almost no additional cost.*

Remark 3.4 (Scaled Bessel function)**.** *Special attention should be given to the calculation of* $\tilde{\varphi}(\omega, \sigma_{m+1}, \sigma_m)$*. First of all, it involves a modified Bessel function of the first kind, which increases dramatically in value when* $q \to -1$ *and/or* $\omega \to \infty$*. The* scaled Bessel function *should be used instead. A robust package has been developed in [1, 2] with algorithms to compute* $I_d^*(z) := \exp\left(-\left|Re\{z\}\right|\right) I_d(z)$ *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* $e^{|Re\{\cdot\}|} I_q^*(\cdot)$ *during the computations.*

Remark 3.5 (Computation of Bessel function)**.** *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 $\tilde{\varphi}\left(\frac{k\pi}{b-a}, \varsigma_q, \varsigma_j\right)$ 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 only determine 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 $O(NJ\mathcal{A})$. 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 *non-equidistant* quadrature rules is dominated by the computation of the Bessel function in matrix $\hat{\varphi}$, which is of order $O(\mathcal{A}NJ^2)$.

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

The direct computation of the matrix-matrix product in (43) would cost $O(N^2J)$ operations. The computational complexity of (43) 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 $O(\max[\mathcal{A}, M\log_2(N)]NJ^2)$.

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 $m = 0, 1, \cdots, M-1$:

$$v(x_m, \sigma_m, t_m) = \begin{cases} \text{Rebate } r_b, & \text{when knocked out,} \\ c(x_m, \sigma_m, t_m), & \text{otherwise,} \end{cases} \tag{47}$$

and:

$$v(x_M, \sigma_M, t_M) = \begin{cases} \text{Rebate } r_b, & \text{when knocked out,} \\ g(x_M), & \text{otherwise,} \end{cases} \tag{48}$$

where the continuation value is governed by (20), 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 (29), the Fourier cosine coefficients of $v(x_m, \sigma_m, t_M)$, i.e., $V_{n,j}(t_M)$,

satisfy

$$V_{n,j}(t_M) = \frac{2}{b-a}\int_{[a,l]\cup[u,b]} r_b \cos\left(n\pi\frac{y-a}{b-a}\right)dy + \frac{2}{b-a}\int_l^u g(y)\cos\left(n\pi\frac{y-a}{b-a}\right)dy$$
$$= \frac{2r_b}{b-a}\left(\psi_n(a,l)+\psi_n(u,b)\right) + G_n(l,u), \tag{49}$$

with $g(y)$ as defined in (39), l and u denote lower and upper barrier levels, respectively [1], and the G_n-terms are the cosine coefficients of the payoff function $g(y)$, as given in Appendix A.

At t_{M-1} the barrier levels split the integral in (35) or (36) into several parts:

$$\hat{V}_{k,p}(t_{M-1}) = \frac{2r_b}{b-a}\left(\psi_k(a,l)+\psi_k(u,b)\right) + \frac{2}{b-a}\int_l^u c_3(y,\varsigma_p,t_{M-1})\cos\left(k\pi\frac{y-a}{b-a}\right)dy$$
$$= \frac{2r_b}{b-a}\left(\psi_k(a,l)+\psi_k(u,b)\right) + \hat{C}_{k,p}(l,u,t_{M-1}). \tag{50}$$

where $\hat{C}_{k,p}$ are the cosine coefficients of the continuation value as given in (40).

We can repeat the derivation from before: We replace c_3 in (40) by the COS approximation and interchange the summation and the integration, which gives:

$$\hat{\mathrm{C}}(l,u,t_{M-1}) = e^{-r\Delta t}\,\mathrm{Re}\left\{\mathcal{M}(l,u)\mathrm{B}'(t_{M-1})\right\}, \tag{51}$$

where, as before, the first row of matrix B is multiplied by one-half, and $\mathrm{B}(t_{M-1})$ is obtained as in (44) and (45). 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 *pre-computed*. Compared to *Algorithm I* 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 $\hat{\mathrm{V}}(t_{m-1})$ with $\hat{\mathrm{V}}(t_m)$, for $m = M-1, M-2, \cdots, 2$. Having $\hat{\mathrm{V}}(t_1)$ approximated, we insert it in (34) and (31) to obtain the option price $\hat{v}(x_0,\sigma_0,t_0)$.

5 Error Analysis

As in [9, 10] 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

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

We depart from (20) 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

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

an intermediate approximation, after the truncation of the integration range of the outer-integral by $[a_\nu, b_\nu]$:

$$c_0(x_m, \sigma_m, t_m) := e^{-r\Delta t} \int_{a_\nu}^{b_\nu} p_{\ln(\nu)}\left(\sigma_{m+1}|\sigma_m\right) \vartheta(x_m, \sigma_{m+1}, \sigma_m) d\sigma_{m+1}. \quad (52)$$

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

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

It then follows that

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

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, depending on q, $\delta_1(q)$, to exist such that

$$e^{-r\Delta t} \int_{\mathbb{R}\setminus[a_\nu, b_\nu]} p_{\ln(\nu)}\left(\sigma_{m+1}|\sigma_m\right) d\sigma_{m+1} \leq \text{TOL} \cdot \delta_1(q). \quad (53)$$

TOL in (53) appears because the size $[a_\nu, b_\nu]$ ensures that $p_{\ln(\nu)}(\sigma_{m+1}|\sigma_m) <$ TOL for $\sigma_{m+1} \in \mathbb{R}\setminus[a_\nu, b_\nu]$. Collecting the information gives:

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

Another intermediate quantity is obtained by replacing $p_{x|\ln(\nu)}$ in (52) with the approximation by the Fourier cosine series expansion, i.e.

$$\overline{c}(x_m, \sigma_m, t_m) = e^{-r\Delta t} \int_{a_\nu}^{b_\nu} p_{\ln(\nu)}\left(\sigma_{m+1}|\sigma_m\right) \tilde{\vartheta}(x_m, \sigma_{m+1}, \sigma_m) d\sigma_{m+1}, \quad (54)$$

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

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

The error analysis in [9] shows that the error due to the COS approximation,

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

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

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

The approximation c_3 defined in (31) can now be obtained by applying a quadrature rule to the integral of (54). Suppose that the (absolute) error from the quadrature rule is $\epsilon_Q(J)$. With the triangle inequality, it then follows that $\forall\, x_m \in [a,b]$ and $\forall\, \sigma_m, \sigma_{m+1} \in [a_\nu, b_\nu]$:

$$\begin{aligned}\epsilon(x_m,\sigma_m,t_m) = |c-c_3| &\leq |c-c_0| + |c_0-\overline{c}| + |\overline{c}-c_3| \\ &\leq \mathrm{TOL}\cdot\delta_0\cdot\delta_1(q) + \epsilon_{cos}(N,a,b) + \epsilon_Q(J) := \epsilon_{loc}. \end{aligned} \tag{55}$$

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.

One can observe the numerical convergence of the local error with respect to parameter J by setting N sufficiently large and TOL sufficiently small. This is included in Section 6.

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 $V_{k,p}(t_m)$. In this subsection, we will study the error $\varepsilon_{k,p}(t_m) := \left|\hat{V}_{k,p}(t_m) - V_{k,p}(t_m)\right|$, and its evolution through time. We focus on a Bermudan put here.

Starting at t_M, $\mathrm{V}(t_M)$ 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 (40), 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 (55) 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

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

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

$$\left|\left|\hat{\mathrm{V}}(t_{m+1}) - \mathrm{V}(t_{m+1})\right|\right|_{\max} \sim O\left(\epsilon_{loc}\right), \tag{56}$$

then it will also hold for time t_m:

The final equation in (46) 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 (28), except that $V_{n,j}(t_{m+1})$ is replaced by $\hat{V}_{n,j}(t_{m+1})$. As such, it holds that

$$\begin{aligned}
&c_3(x_m, \sigma_m, t_m) - \hat{c}_3(x_m, \sigma_m, t_m) = \\
&e^{-r\Delta t}\sum_{j=0}^{J-1} w_j {\sum_{n=0}^{N-1}}' \left(\hat{V}_{n,j}(t_{m+1}) - V_{n,j}(t_{m+1})\right) \cdot \mathrm{Re}\left\{\tilde{\varphi}\left(\frac{n\pi}{b-a}, \varsigma_j, \sigma_m\right) e^{in\pi\frac{x_m - a}{b-a}}\right\}.
\end{aligned}$$

To analyze this error term, we decompose $\tilde{\varphi}$ using (30) and replace the $\mathrm{Re}\{\cdot\}$-term by P_n, defined in (24), which gives

$$c_3(x_m, \sigma_m, t_m) - \hat{c}_3(x_m, \sigma_m, t_m) = e^{-r\Delta t}\sum_{j=0}^{J-1} w_j p_{\ln(\nu)}(\varsigma_j|\sigma_m)\Theta(\varsigma_j, x_m, \sigma_m), \tag{57}$$

where

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

From (22), we know that

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

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

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

For smooth density functions, as we have in Heston's model, the cosine series coefficients P_n converge exponentially in n. The sum, ${\sum}'(P_n + O(TOL)_x)^2$, is

therefore a sum of a geometric series, which is thus bounded. We assume that a positive number, δ_3, exists, which satisfies

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

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 (56), 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 Eq. (57) and employing the Cauchy-Schwarz inequality, gives 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:

$$\begin{aligned} &\left| \hat{V}_{k,q}(x_m, \sigma_m, t_m) - V_{k,q}(x_m, \sigma_m, t_m) \right| = \left| \hat{C}_{k,q}(a, x^*(\varsigma_q, t_m), t_m) - C_{k,q}(a, x^*(\varsigma_q, t_m), t_m) \right| \\ &\leq \frac{2}{b-a} \sqrt{\int_{x^*(\varsigma_p, t_{M-1})}^{b} (c(y, \varsigma_q, t_m) - c_3(y, \varsigma_q, t_m))^2 dy} \sqrt{\int_{x^*(\varsigma_p, t_{M-1})}^{b} \cos^2\left(k\pi \frac{y-a}{b-a} \right) dy} \\ &\leq 2e^{-r\Delta t} \sqrt{\delta_3 \delta_4 \delta_5} \cdot \sqrt{JN} \cdot \epsilon_{loc}. \end{aligned} \tag{58}$$

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

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

The speed of convergence will, however, decrease when the number of monitoring dates increases, due to the increasing weighting term in (58). Larger values for N and J are required in that case. We will examine this via numerical experiments in the next sections.

6 Numerical Results

In this section, we first confirm, by numerical experiments, in Subsection 6.1 the error convergence analysis from Section 5 by pricing *discrete barrier options* for which we set $l = a$ and $u = b$. This should give us the prices of European options with the barrier option pricing algorithm, and therefore we can generate reference values by the European version of the COS method from [9]. Since only a limited number of reference values are found in the literature, we use this special case to study the error convergence.

Subsequently, we price two Bermudan-style options with several early-exercise dates in Subsection 6.2. Their values should resemble American reference options that we use for comparison.

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.

6.1 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 8-th decimal place) by the European option pricing method from [9].

Three tests are extracted from [4], 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 ($q = -0.96$): $\eta = 1, \lambda = 0.5, \bar{\nu} = 0.04, T = 10$.

Numerical methods for early-exercise or barrier options are usually either based on finite differences for PDEs [14] or on tree-based methods [17, 18]. Results with these techniques using the parameter sets that give rise to significant pricing difficulties for early-exercise options under Heston's dynamics (i.e. the Feller condition not satisfied) have however not yet been published.

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

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

and we do not consider dividend payment here.

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

$(J = 2^d)$	Fourier cosine expansion plus composite Trapezoidal Rule					
	TOL $= 10^{-4}$		TOL $= 10^{-6}$		TOL $= 10^{-8}$	
d	time(sec)	error	time(sec)	error	time(sec)	error
4	0.05	-4.53 10^{-3}	0.06	4.89 10^{-2}	0.06	2.09
5	0.15	-7.04 10^{-3}	0.15	-3.97 10^{-5}	0.16	7.66 10^{-4}
6	0.56	-4.93 10^{-3}	0.55	-3.37 10^{-5}	0.56	-5.28 10^{-7}
7	2.34	-4.29 10^{-3}	2.35	-1.29 10^{-5}	2.42	-4.08 10^{-7}
$(J = 2^d)$	Fourier cosine expansion plus Gauss-Legendre Rule					
	TOL $= 10^{-4}$		TOL $= 10^{-6}$		TOL $= 10^{-8}$	
d	time(sec)	error	time(sec)	error	time(sec)	error
4	0.12	-7.51 10^{-3}	0.12	1.02 10^{-2}	0.12	1.41
5	0.43	-3.95 10^{-3}	0.42	-1.85 10^{-5}	0.40	2.99 10^{-5}
6	1.69	-3.95 10^{-3}	1.59	-1.54 10^{-5}	1.54	-6.41 10^{-6}
7	6.88	-3.95 10^{-3}	7.07	-1.34 10^{-5}	6.49	-6.32 10^{-7}

First of all, we compare the error convergence in J for the Heston pricing methods with the composite Trapezoidal rule (upper part of Table 1) with results obtained by the Gauss-Legendre quadrature rule (lower part of Table 1). We prescribe the pre-defined 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 TOL$\leq 10^{-6}$) and N is fixed, a very fast error convergence in J is obtained (and the computational complexity is quadratic in J). As pointed out earlier, with the composite Trapezoidal rule, the calculation of $\tilde{\phi}$ in the initialization phase requires less CPU time than with the non-equidistant Gauss-Legendre rule. From the experiments of Test No. 1, we can conclude that for $q > 0$ both methods give highly accurate results within a fraction of a second.

We continue with the difficult test cases for which $q \to -1$. While seemingly pleasant in both CPU time and convergence for Test No.1, the composite Trapezoidal rule (as well as the composite Simpson rule) is no longer appealing when q is less than zero, as it requires very large values of J to achieve the desired accuracy. The Gauss-Legendre rule can, however, still produce satisfactory results for relatively small values of J. Therefore, we only illustrate the results obtained by the Gauss-Legendre rule in log-variance dimension in Table 2.

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

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

Compared to Test No.1, the absolute errors in the Tests No. 2 and No.3 are larger for the same N and the same J. When $q \to -1$, 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 values of J are 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. So, if we can find a proxy for the Bessel function which can be computed in a cheap way (like the moment matching based functions in [4]), the overall computation time could be significantly reduced. We leave this for further research.

Next, we examine the error convergence in N, keeping the number of points

in log-variance direction, J, fixed. The results are presented in Table 3. One can observe that the error convergence is faster than quadratic (the linear increments in CPU time are not shown).

Table 3: Convergence in N; COS + Gauss-Legendre, $M = 12$, TOL= 10^{-7}, $J = 2^7$ for Test No.1 and $J = 2^8$ for Test No.2.

	$d : (N = 2^d)$			
Test:	4	5	6	7
No. 1 ($q = 0.6$)	2.94 10^{-1}	-1.63 10^{-2}	-3.01 10^{-5}	-1.79 10^{-6}
No. 2 ($q = -0.84$)	7.32 10^{-1}	-9.75 10^{-2}	-2.30 10^{-2}	-1.72 10^{-4}

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 4). 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 of CPU time in the main loop, which is in accordance with the error analysis.

Table 4: 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.

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

6.2 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 return thus 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, PDE or Monte Carlo method.

Two parameter test sets are used here. One is chosen in the PDE-based finite differences literature, for example in [14], with $q > 0$; and the second is with $q \in [-1, 0]$, inspired by results with a tree-based method in [17]. The reference value for the first test case is available and accurate up to the 6th digit, see [14].

For the latter test Bermudan reference values are not available. So we provide our results that 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:

- 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 $\rho = 0.1$,

which gives $q > 0$. So, a very accurate and efficient pricing performance is expected from our method. Results are presented in Table 5, 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].

Table 5: 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)		
ref.val.	2.000000	1.107621	0.520030	0.213677	0.082044	total	Init.	Loop
M=10	-1.80 10^{-2}	-4.79 10^{-3}	-2.85 10^{-3}	-1.31 10^{-3}	-5.18 10^{-4}	6.9	6.34	0.57
M=20	-9.54 10^{-3}	-2.39 10^{-3}	-1.40 10^{-3}	-6.65 10^{-4}	-2.78 10^{-4}	7.5	6.36	1.13
M=40	-5.14 10^{-3}	-1.07 10^{-3}	-5.50 10^{-4}	-2.54 10^{-4}	-1.22 10^{-4}	8.9	6.57	2.32
M=80	-2.83 10^{-3}	-2.86 10^{-4}	2.75 10^{-5}	5.42 10^{-5}	-8.43 10^{-7}	14.1	7.35	6.70

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 [17], where the parameters were set as:

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

However, reference values were not available in the paper, so that we provide our results as a reference in Table 6.

Table 6: Results of Test No. 5 ($q = -0.47$); COS + Gauss-Legendre; $N = 2^8, J = 2^8$ and TOL= 10^{-7}.

	S_0			time (sec)		
M	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

[2] Although 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 [16, 10].

7 Conclusions

In this paper, we have focused on 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. An efficient 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 with respect to 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 also highly satisfactory then. 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.

For the near future research, we expect a significant speed-up when the Bessel function computations can be replaced by the computation of an accurate proxy. Replacing the MATLAB implementation by an efficient C code would further reduce CPU time.

A Derivations

In this appendix we repeat some basic derivations and results obtained in [9] and needed in the present paper.

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

$$G_n(l,u) = \frac{2}{b-a}\alpha K \left[\chi_k(l^*,u^*) - \psi_k(l^*,u^*)\right], \quad \alpha = \begin{cases} 1 & \text{for a call,} \\ -1 & \text{for a put,} \end{cases} \tag{59}$$

with

$$l^* = \begin{cases} \max(l,0) & \alpha = 1, \\ \min(l,0) & \alpha = -1. \end{cases} \qquad u^* = \begin{cases} \max(u,0) & \alpha = 1, \\ \min(u,0) & \alpha = -1. \end{cases} \tag{60}$$

and

$$\chi_k(l^*,u^*) := \int_{l^*}^{u^*} e^x \cos\left(n\pi\frac{x-a}{b-a}\right) dx, \tag{61}$$

$$\psi_k(l^*,u^*) := \int_{l^*}^{u^*} \cos\left(n\pi\frac{x-a}{b-a}\right) dx. \tag{62}$$

χ_k and ψ_k admit the following analytic solutions:

$$\begin{aligned}
\chi_k(l,u) &= \frac{1}{1+\left(\frac{n\pi}{b-a}\right)^2}\left[\cos\left(n\pi\frac{u-a}{b-a}\right)e^u - \cos\left(n\pi\frac{l-a}{b-a}\right)e^l\right. \\
&\quad \left. + \frac{n\pi}{b-a}\sin\left(n\pi\frac{u-a}{b-a}\right)e^u - \frac{n\pi}{b-a}\sin\left(n\pi\frac{l-a}{b-a}\right)e^l\right], \\
\psi_k(l,u) &= \begin{cases} \left[\sin\left(n\pi\frac{u-a}{b-a}\right) - \sin\left(n\pi\frac{l-a}{b-a}\right)\right]\frac{b-a}{n\pi} & n\neq 0, \\ (u-l) & n=0. \end{cases}
\end{aligned} \tag{63}$$

We also show that the matrix $\mathcal{M}(l,u)$ with elements $\mathcal{M}_{k,n}(l,u)$ in (42) is the sum of a Hankel matrix, $M_s(l,u)$, and a Toeplitz matrix, $M_c(l,u)$, and that the matrix-vector-product, $\mathcal{M}(l,u)\mathbf{u}$, can then be computed by means of the FFT in $O(N\log_2 N)$ operations.

We use $e^{i\alpha} = \cos\alpha + i\sin\alpha$ in the definition of $\mathcal{M}_{k,n}(l,u)$ in (42) and find that

$$\mathcal{M}_{k,n}(l,u) = -\frac{i}{\pi}\left(\mathcal{M}^c_{k,n}(l,u) + \mathcal{M}^s_{k,n}(l,u)\right), \tag{64}$$

where

$$\mathcal{M}^c_{k,n}(l,u) := \begin{cases} \dfrac{(u-l)\pi i}{(b-a)} & k=n=0, \\ \dfrac{\exp\left(i(n+k)\dfrac{(u-a)\pi}{b-a}\right) - \exp\left(i(n+k)\dfrac{(l-a)\pi}{b-a}\right)}{n+k} & \text{otherwise} \end{cases} \tag{65}$$

and

$$\mathcal{M}^s_{k,n}(l,u) := \begin{cases} \dfrac{(u-l)\pi i}{b-a} & k=n, \\ \dfrac{\exp\left(i(n-k)\dfrac{(u-a)\pi}{b-a}\right) - \exp\left(i(n-k)\dfrac{(l-a)\pi}{b-a}\right)}{n-k} & k\neq n. \end{cases} \tag{66}$$

The matrices

$$\mathcal{M}_c(l,u) := \{\mathcal{M}^c_{k,n}(l,u)\}_{k,n=0}^{N-1} \qquad \text{and} \qquad \mathcal{M}_s(l,u) := \{\mathcal{M}^s_{k,n}(l,u)\}_{k,n=0}^{N-1}$$

have special structure: The matrix $\mathcal{M}_c(l,u)$ is a *Hankel* matrix,

$$\mathcal{M}_c(l,u) = \begin{bmatrix} \mathrm{m}_0 & \mathrm{m}_1 & \mathrm{m}_2 & \cdots & \mathrm{m}_{N-1} \\ \mathrm{m}_1 & \mathrm{m}_2 & \cdots & \cdots & \mathrm{m}_N \\ \vdots & & & & \vdots \\ \mathrm{m}_{N-2} & \mathrm{m}_{N-1} & \cdots & & \mathrm{m}_{2N-3} \\ \mathrm{m}_{N-1} & \cdots & & \mathrm{m}_{2N-3} & \mathrm{m}_{2N-2} \end{bmatrix}_{N\times N} \tag{67}$$

and $\mathcal{M}_s(l,u)$ is a *Toeplitz* matrix,

$$\mathcal{M}_s(l,u) = \begin{bmatrix} \mathrm{m}_0 & \mathrm{m}_1 & \cdots & \mathrm{m}_{N-2} & \mathrm{m}_{N-1}, \\ \mathrm{m}_{-1} & \mathrm{m}_0 & \mathrm{m}_1 & \cdots & \mathrm{m}_{N-2}, \\ \vdots & & \ddots & & \vdots \\ \mathrm{m}_{2-N} & \cdots & \mathrm{m}_{-1} & \mathrm{m}_0 & \mathrm{m}_1 \\ \mathrm{m}_{1-N} & \mathrm{m}_{2-N} & \cdots & \mathrm{m}_{-1} & \mathrm{m}_0 \end{bmatrix}_{N\times N} \tag{68}$$

with

$$\mathrm{m}_n := \begin{cases} \dfrac{(u-l)\pi}{b-a} i & n = 0 \\ \frac{1}{n}\left[\exp\left(in\dfrac{(u-a)\pi}{b-a}\right) - \exp\left(in\dfrac{(l-a)\pi}{b-a}\right)\right] & n \neq 0 \end{cases} \tag{69}$$

The matrix-vector product can now be transformed into a circular convolution, see also, for example, in [10]. The product $\mathcal{M}_s(l,u)\mathbf{u}$ is equal to the first N elements of $\mathbf{m}_s \circledast \mathbf{u}_s$ with the $2N$-vectors:

$$\mathbf{m}_s = \left[\mathrm{m}_0, \mathrm{m}_{-1}, \mathrm{m}_{-2}, \cdots, \mathrm{m}_{1-N}, 0, \mathrm{m}_{N-1}, \mathrm{m}_{N-2}, \cdots, \mathrm{m}_1\right]^T,$$

and $\mathbf{u}_s = \left[u_0, u_1, \cdots, u_{N-1}, 0, \cdots, 0\right]^T$.

A Hankel matrix times a column vector, $\mathcal{M}_c(l,u)\mathbf{u}$, is equal to the first N elements of $\mathbf{m}_c \circledast \mathbf{u}_c$, in reversed order, with the $2N$-vectors defined by

$$\mathbf{m}_c = \left[\mathrm{m}_{2N-1}, \mathrm{m}_{2N-2}, \cdots, \mathrm{m}_1, \mathrm{m}_0\right]^T$$

and $\mathbf{u}_c = \left[0, \cdots, 0, u_0, u_1, \cdots, u_{N-1}\right]^T$.

References

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

[2] Amos D.E., Algorithm 644: A Portable Package for Bessel Functions of a Complex Argument and Nonnegative Order, *ACM Trans. Math. Software*, 12(3): 265–273, 1986.

[3] Andricopoulos A.D., Widdicks M., Duck P.W. and Newton D.P., Universal Option Valuation Using Quadrature, *J. Financial Economics*, 67(3): 447-471, 2003.

[4] Andersen L., Simple and efficient simulation of the Heston stochastic volatility model, *J. Comp. Finance,* 11: 1–42, 2008.

[5] Andricopoulos A.D., Widdicks M., Duck P.W., and Newton D.P., Extending quadrature methods to value multi-asset and complex path dependent options, *J. Fin. Economics*, 83: 471–499, 2007,

[6] Broadie M. and Kaya O., Exact Simulation of Stochastic Volatility and Other Affine Jump Diffusion Processes, *Operations Research,* 54(2): 217–231, 2006.

[7] Carr P.P. and Madan D.B. , Option Valuation Using the Fast Fourier Transform, *J. Comp. Finance*, 2: 61–73, 1999.

[8] Cox J.C., Ingersoll J.E., and Ross S.A., A theory of the term structure of interest rates, *Econometrica*, 53(2): 385–407, 1985.

[9] Fang F. and Oosterlee C.W., A Novel Method For European Options Based On Fourier-Cosine Series Expansions, *SIAM J. Sci. Comput.* 31(2): 826–848, 2008.

[10] Fang F. and Oosterlee C.W., Pricing Early-Exercise and Discrete Barrier Options By Fourier-Cosine Series Expansions, *Numerische Mathematik*, 114(1): 27–62, 2009.

[11] Duffie D., Pan J. and Singleton K., Transform Analysis and Asset Pricing for Affine Jump-diffusions, *Econometrica*, 68: 1343–1376, 2000.

[12] Feller W., Two Singular Diffusion Problems, *Annals of Mathematics*, 54:173-182, 1951.

[13] Heston S., A Closed-form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options, *Review of Financial Studies*, 6: 327-343, 1993.

[14] Ito K. and Toivanen J., Lagrange Multplier Approach with Optimized Finite Difference Stencils for Pricing American Options under Stochastic Volatility. *SIAM J. Sci. Comput.*, 31: 2646–2664, 2009.

[15] Kahl C., and Jackel P., Fast Strong Approximation Monte-Carlo Schemes for Stochastic Volatility Models. *Quant. Finance*, 6(6): 513–536, 2006.

[16] Lord R., Fang F., Bervoets F. and Oosterlee C.W., A Fast and Accurate FFT-based Method for Pricing Early-exercise Options under Lévy Processes. *SIAM J. Sci. Comput.* 30: 1678–1705, 2008.

[17] Vellekoop M. and Nieuwenhuis H., A Tree-based Method to price American Options in the Heston Model, *J. Comp. Finance* 13(1), 1–21, 2009.

[18] Beliaeva N. and Nawalkha S., *A Simple Approach to Pricing American Options Under the Heston Stochastic Volatility Model*, working paper, available at SSRN: http://ssrn.com/abstract=1107934, 2010.