



CPB Netherlands Bureau for Economic  
Policy Analysis

CPB Discussion Paper | 225

# Two-dimensional Fourier cosine series expansion method for pricing financial options

Marjon Ruijter  
Kees Oosterlee



# Two-dimensional Fourier cosine series expansion method for pricing financial options

M. J. Ruijter\*      C. W. Oosterlee†

October 26, 2012

## Abstract

The COS method for pricing European and Bermudan options with one underlying asset was developed in [F. Fang, C. W. Oosterlee, 2008] and [F. Fang, C. W. Oosterlee, 2009]. In this paper, we extend the method to higher dimensions, with a multi-dimensional asset price process. The algorithm can be applied to, for example, pricing two-color rainbow options, but also to pricing under the popular Heston stochastic volatility model. For smooth density functions, the resulting method converges exponentially in the number of terms in the Fourier cosine series summations, otherwise we achieve algebraic convergence. The use of an FFT algorithm, for asset prices modelled by Lévy processes, makes the algorithm highly efficient. We perform extensive numerical experiments.

**Keywords:** Fourier cosine expansion method, European and Bermudan options, two-color rainbow options, basket options, Lévy process, Heston dynamics.

AMS Classification 65T40, 65T50, 91G60, 60E10, 62P05.

JEL Classification C02, C63, G12.

This article is published in SIAM Journal on Scientific Computing, [25], and should be cited with that publication as reference.

## 1 Introduction

In financial markets traders deal in assets and options, like the well-known call and put options. Besides these, many ‘exotic’ options have been defined, that have more complex contract details and are not traded at regulated exchanges.

One class of exotic option contracts is called the class of multi-color *rainbow options*, whose payoff may depend on multiple assets, like on the average or the maximum of asset prices. The value of the option depends on the contract details and on the underlying asset prices.

Computational finance deals with numerical and computational questions regarding efficient option pricing and calibration. Usually, an asset price model is calibrated to liquidly available plain vanilla options (calls and puts), from a regulated exchange. For the valuation of the exotic options

---

\*Centrum Wiskunde & Informatica, Amsterdam, the Netherlands, email: [m.j.ruijter@cwi.nl](mailto:m.j.ruijter@cwi.nl), and CPB Netherlands Bureau for Economic Policy Analysis, Den Haag, the Netherlands.

†Centrum Wiskunde & Informatica, Amsterdam, the Netherlands, email: [c.w.oosterlee@cwi.nl](mailto:c.w.oosterlee@cwi.nl), and Delft University of Technology, Delft, the Netherlands.

other computational methods are typically used. Option pricing techniques can be divided into the categories of Monte Carlo simulation, partial differential equation (PDE) methods and Fourier-based methods. Often Monte Carlo methods are used to price high-dimensional option contracts. The method presented here can be seen as an alternative (deterministic) pricing technique, which can deal with multi-asset option problems of *medium-sized* dimensionality, meaning 2D to approximately 5D integrals. The method we propose for pricing higher-dimensional options is based on the Fourier transform of the transitional density function and is especially suitable for asset price models in the class of Lévy processes.

In [11], an option pricing method for European options with one underlying asset has been developed, which is called the COS method. The method is based on the Fourier cosine series expansions of the discounted expected payoff. The corresponding characteristic function was used to approximate the Fourier coefficients. For smooth density functions, the error converges exponentially in  $N$ , the number of terms in the series expansions. The one-dimensional COS method has been extended in [12] to pricing Bermudan, barrier, and American options. The use of an FFT algorithm, for asset prices modelled by Lévy processes, makes the algorithm highly efficient, with a computational complexity of  $O(N \log_2 N)$ .

The previous strain of literature on the COS method was based on the one-dimensional characteristic function of a single stochastic process. In this paper, we extend the Fourier method to higher dimensions and price in particular two-color *rainbow options*, which are contracts written on two underlying assets. Well-known examples include the valuation of basket and call-on-maximum options. Methods for both European and Bermudan-style rainbow options are developed here. The resulting algorithm can be applied to models such as correlated geometric Brownian motions or multi-dimensional processes with jumps. The method is highly efficient for asset prices in the class of Lévy processes. In the literature, mainly Monte Carlo based methods are being used to solve higher dimensional pricing problems, see [2], [3], [5], and [6]. Leentvaar and Oosterleer worked on a parallel Fourier-based method ([21]) and parallel sparse grid methods (citeLeentvaar2008gridstretch) for pricing multi-asset options. The authors in [8] demonstrate an implicit PDE discretisation method for rainbow options under jump-diffusion processes. We will compare our results to reference values in the literature mentioned above.

The methodology presented here can also be applied to pricing options with one underlying asset, for which the dynamics are governed by two or more correlated stochastic processes. For example, the popular *Heston model* describes an asset price process with a stochastic volatility ([14]). The instantaneous variance process follows a mean-reverting square root (CIR) process. This model is able to capture smile and skew patterns in the implied volatility surface. Besides, the closed-form univariate characteristic function of the log-asset price process makes it easy to implement the Heston model in Fourier-based methods, see [1] and [23] for European calls. For the valuation of Bermudan and discrete barrier options, reference [13] combined the COS formula for the log-asset dimension and a quadrature rule in the log-variance dimension. Since the bivariate characteristic function of the log-asset price and variance is available, we can also apply the 2D-COS formula to this problem. We investigate the two-dimensional COS method particularly for Bermudan put options under the Heston dynamics.

The outline of this paper is as follows. We start with the presentation of the two-dimensional COS formula for pricing European rainbow options (Section 2) and the two-dimensional COS method for solving Bermudan pricing problems (Section 3). Section 4 discusses option pricing under the Heston model, which is an affine diffusion process, but not in the Lévy class. The error analysis in Section 5 indicates an exponentially converging error for smooth density functions. A non-smooth density function results in algebraic convergence. Then, in Sections 6 and 7, numerical tests are performed. The two-dimensional COS method can easily be extended to higher dimensions and we give some insights into the possibilities and difficulties in Section 8. Section 9 concludes.

## 2 European rainbow options

In this section, we explain the *two-dimensional COS formula* to approximate discounted expected payoffs. The method is based on the Fourier cosine series of the payoff function and the density. The density function of a stochastic process is usually not known, but often its characteristic function is known (see [9], [11]). This enables us to approximate the Fourier coefficients efficiently.

Let  $(\Omega, \mathcal{F}, P)$  be a probability space,  $T > 0$  a finite terminal time, and  $\mathbb{F} = (\mathcal{F}_s)_{0 \leq s \leq T}$  a filtration satisfying the usual conditions. The process  $\mathbf{X}_t = (X_t^1, X_t^2)$  denotes a two-dimensional stochastic process on the filtered probability space  $(\Omega, \mathcal{F}, \mathbb{F}, P)$ , representing the log-asset prices. We assume that the bivariate characteristic function of the stochastic process is known, which is the case, for example, for affine jump-diffusions ([9]). The value of a European rainbow option, with payoff function  $g(\cdot)$ , is given by the risk-neutral option valuation formula

$$v(t_0, \mathbf{x}) = e^{-r\Delta t} \mathbb{E}^{t_0, \mathbf{x}} [g(\mathbf{X}_T)] = e^{-r\Delta t} \iint_{\mathbb{R}^2} g(\mathbf{y}) f(\mathbf{y}|\mathbf{x}) d\mathbf{y}. \quad (1)$$

Here,  $\mathbf{x} = (x_1, x_2)$  is the current state,  $f(y_1, y_2|x_1, x_2)$  is the conditional density function,  $r$  is the risk-free rate, and time to expiration is denoted by  $\Delta t := T - t_0$ . In the derivation of the COS formula, we distinguish three different approximation steps. The errors introduced in each step are discussed in Section 5.1.

Step 1: We assume that the integrand is integrable, which is common for the problems we deal with. Because of that, we can, for given  $\mathbf{x}$ , truncate the infinite integration ranges to some domain  $[a_1, b_1] \times [a_2, b_2] \subset \mathbb{R}^2$  without losing significant accuracy. This gives the multi-D Fourier cosine expansion formulation

$$\begin{aligned} v_1(t_0, \mathbf{x}) &= e^{-r\Delta t} \int_{a_2}^{b_2} \int_{a_1}^{b_1} g(\mathbf{y}) f(\mathbf{y}|\mathbf{x}) dy_1 dy_2 \\ &= e^{-r\Delta t} \int_{a_2}^{b_2} \int_{a_1}^{b_1} g(\mathbf{y}) \sum_{k_1=0}^{+\infty} \sum_{k_2=0}^{+\infty} A_{k_1, k_2}(\mathbf{x}) \cos\left(k_1 \pi \frac{y_1 - a_1}{b_1 - a_1}\right) \cos\left(k_2 \pi \frac{y_2 - a_2}{b_2 - a_2}\right) dy_1 dy_2. \end{aligned} \quad (2)$$

The notation  $v_i$  is used for the different approximations of  $v$  and keeps track of the numerical errors that set in from each step. For final approximations we also use the “hat”-notation, like  $\hat{v}$ ,  $\hat{c}$ , etc. In the second line in (2), the conditional density is replaced by its Fourier cosine expansion in  $\mathbf{y}$  on  $[a_1, b_1] \times [a_2, b_2]$ , with series coefficients  $A_{k_1, k_2}$  defined by

$$A_{k_1, k_2}(\mathbf{x}) := \frac{2}{b_1 - a_1} \frac{2}{b_2 - a_2} \int_{a_2}^{b_2} \int_{a_1}^{b_1} f(\mathbf{y}|\mathbf{x}) \cos\left(k_1 \pi \frac{y_1 - a_1}{b_1 - a_1}\right) \cos\left(k_2 \pi \frac{y_2 - a_2}{b_2 - a_2}\right) dy_1 dy_2. \quad (3)$$

$\sum'$  in (2) means that the first term of the summation has half weight. We interchange summation and integration and define

$$V_{k_1, k_2}(T) := \frac{2}{b_1 - a_1} \frac{2}{b_2 - a_2} \int_{a_2}^{b_2} \int_{a_1}^{b_1} g(\mathbf{y}) \cos\left(k_1 \pi \frac{y_1 - a_1}{b_1 - a_1}\right) \cos\left(k_2 \pi \frac{y_2 - a_2}{b_2 - a_2}\right) dy_1 dy_2, \quad (4)$$

which are the Fourier cosine series coefficients of  $v(T, \mathbf{y}) = g(\mathbf{y})$  on  $[a_1, b_1] \times [a_2, b_2]$ .

Step 2: Truncation of the series summations gives

$$v_2(t_0, \mathbf{x}) = \frac{b_1 - a_1}{2} \frac{b_2 - a_2}{2} e^{-r\Delta t} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} A_{k_1, k_2}(\mathbf{x}) V_{k_1, k_2}(T). \quad (5)$$

Step 3: Next, the coefficients  $A_{k_1, k_2}(\mathbf{x})$  are approximated by

$$F_{k_1, k_2}(\mathbf{x}) := \frac{2}{b_1 - a_1} \frac{2}{b_2 - a_2} \iint_{\mathbb{R}^2} f(\mathbf{y}|\mathbf{x}) \cos\left(k_1 \pi \frac{y_1 - a_1}{b_1 - a_1}\right) \cos\left(k_2 \pi \frac{y_2 - a_2}{b_2 - a_2}\right) dy_1 dy_2. \quad (6)$$

The two-dimensional COS formula is based on the following goniometric relation ([29]):

$$2 \cos(\alpha) \cos(\beta) = \cos(\alpha + \beta) + \cos(\alpha - \beta). \quad (7)$$

With this we obtain

$$2F_{k_1, k_2}(\mathbf{x}) = F_{k_1, k_2}^+(\mathbf{x}) + F_{k_1, k_2}^-(\mathbf{x}), \quad (8)$$

where

$$F_{k_1, k_2}^\pm(\mathbf{x}) := \frac{2}{b_1 - a_1} \frac{2}{b_2 - a_2} \iint_{\mathbb{R}^2} f(\mathbf{y}|\mathbf{x}) \cos\left(k_1 \pi \frac{y_1 - a_1}{b_1 - a_1} \pm k_2 \pi \frac{y_2 - a_2}{b_2 - a_2}\right) dy_1 dy_2. \quad (9)$$

Now, the coefficients  $F_{k_1, k_2}^\pm(\mathbf{x})$  can be calculated by

$$\begin{aligned} F_{k_1, k_2}^\pm(\mathbf{x}) &= \frac{2}{b_1 - a_1} \frac{2}{b_2 - a_2} \operatorname{Re} \left( \iint_{\mathbb{R}^2} f(\mathbf{y}|\mathbf{x}) \exp\left(ik_1 \pi \frac{y_1}{b_1 - a_1} \pm ik_2 \pi \frac{y_2}{b_2 - a_2}\right) d\mathbf{y} \exp\left(-ik_1 \pi \frac{a_1}{b_1 - a_1} \mp ik_2 \pi \frac{a_2}{b_2 - a_2}\right) \right) \\ &= \frac{2}{b_1 - a_1} \frac{2}{b_2 - a_2} \operatorname{Re} \left( \varphi\left(\frac{k_1 \pi}{b_1 - a_1}, \pm \frac{k_2 \pi}{b_2 - a_2} \middle| \mathbf{x}\right) \exp\left(-ik_1 \pi \frac{a_1}{b_1 - a_1} \mp ik_2 \pi \frac{a_2}{b_2 - a_2}\right) \right) \\ &= \frac{2}{b_1 - a_1} \frac{2}{b_2 - a_2} \operatorname{Re} \left( \varphi_{\text{levy}}\left(\frac{k_1 \pi}{b_1 - a_1}, \pm \frac{k_2 \pi}{b_2 - a_2}\right) \exp\left(ik_1 \pi \frac{x_1 - a_1}{b_1 - a_1} \pm ik_2 \pi \frac{x_2 - a_2}{b_2 - a_2}\right) \right). \end{aligned} \quad (10)$$

$\operatorname{Re}(\cdot)$  denotes taking the real part of the input argument.  $\varphi(\cdot, \cdot | \mathbf{x})$  is the *bivariate conditional characteristic function* of  $\mathbf{X}_T$ , given  $\mathbf{X}_{t_0} = \mathbf{x}$ , ([26])

$$\varphi(\mathbf{u}|\mathbf{x}) = \mathbb{E}\left[e^{i\mathbf{u} \cdot \mathbf{X}_T} \middle| \mathcal{F}_{t_0}\right] = \iint_{\mathbb{R}^2} e^{i\mathbf{u} \cdot \mathbf{y}} f(\mathbf{y}|\mathbf{x}) d\mathbf{y}. \quad (11)$$

Examples of these characteristic functions can be found in Section 6. The last equality in (10) holds particularly for Lévy processes, for which  $\varphi_{\text{levy}}(u_1, u_2) := \varphi(u_1, u_2 | 0, 0)$ . Inserting (10) in (5) gives us the *2D-COS formula* for approximation of  $v(t_0, \mathbf{x})$ :

$$\begin{aligned} \hat{v}(t_0, \mathbf{x}) &:= \frac{b_1 - a_1}{2} \frac{b_2 - a_2}{2} e^{-r\Delta t} \sum'_{k_1=0}^{N_1-1} \sum'_{k_2=0}^{N_2-1} \frac{1}{2} \left[ F_{k_1, k_2}^+(\mathbf{x}) + F_{k_1, k_2}^-(\mathbf{x}) \right] V_{k_1, k_2}(T) \\ &= e^{-r\Delta t} \sum'_{k_1=0}^{N_1-1} \sum'_{k_2=0}^{N_2-1} \frac{1}{2} \left[ \operatorname{Re} \left( \varphi_{\text{levy}}\left(\frac{k_1 \pi}{b_1 - a_1}, + \frac{k_2 \pi}{b_2 - a_2}\right) \exp\left(ik_1 \pi \frac{x_1 - a_1}{b_1 - a_1} + ik_2 \pi \frac{x_2 - a_2}{b_2 - a_2}\right) \right) \right. \\ &\quad \left. + \operatorname{Re} \left( \varphi_{\text{levy}}\left(\frac{k_1 \pi}{b_1 - a_1}, - \frac{k_2 \pi}{b_2 - a_2}\right) \exp\left(ik_1 \pi \frac{x_1 - a_1}{b_1 - a_1} - ik_2 \pi \frac{x_2 - a_2}{b_2 - a_2}\right) \right) \right] V_{k_1, k_2}(T). \end{aligned} \quad (12)$$

With the multi-D-COS formula, calculation of the option's Greeks is straightforward, as explained for the one-dimensional case in [11].

**Remark 1** *Cosine terms facilitate the usage of the characteristic function. Fourier sine expansions may also be used, however their coefficients decrease at a lower rate for the payoff functions discussed and because of that the cosine series are to be preferred. Alternative basis functions, like certain wavelet basis functions may represent another interesting research direction for option pricing, but this is not yet known and part of future research.*

If the characteristic function is not available directly or not known analytically, it may be approximated. Local volatility models, for example, typically do not yield analytic functions  $\varphi$ , but recent research in ([24]) proposes a second-order approximation formula, so that an approximate characteristic function may be derived.

### 3 Bermudan rainbow options

We generalize the multi-D-COS method to pricing Bermudan rainbow options with a two-dimensional underlying log-asset price process,  $\mathbf{X}_t = (X_t^1, X_t^2)$ , that is in the class of Lévy processes. A Bermudan option can be exercised at a fixed set of  $\mathcal{M}$  early-exercise times  $t_0 < t_1 < \dots < t_m < \dots < t_{\mathcal{M}} = T$ , with  $\Delta t := t_{m+1} - t_m$ . The payoff function is denoted by  $g(\cdot)$ . The problem is solved backwards in time, with

$$\begin{cases} v(t_{\mathcal{M}}, \mathbf{x}) &= g(\mathbf{x}), \\ c(t_{m-1}, \mathbf{x}) &= e^{-r\Delta t} \mathbb{E} [v(t_m, \mathbf{X}_{t_m}) | \mathbf{X}_{t_{m-1}} = \mathbf{x}], \\ v(t_{m-1}, \mathbf{x}) &= \max[g(\mathbf{x}), c(t_{m-1}, \mathbf{x})], \quad 2 \leq m \leq \mathcal{M}, \\ v(t_0, \mathbf{x}_0) &= c(t_0, \mathbf{x}_0). \end{cases}$$

Function  $c(t_{m-1}, \mathbf{x})$  is called the continuation value and is approximated by the two-dimensional COS formula,

$$\hat{c}(t_{m-1}, \mathbf{x}) := \frac{b_1 - a_1}{2} \frac{b_2 - a_2}{2} e^{-r\Delta t} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} \frac{1}{2} [F_{k_1, k_2}^+(\mathbf{x}) + F_{k_1, k_2}^-(\mathbf{x})] V_{k_1, k_2}(t_m). \quad (13)$$

The Fourier coefficients of the value function in (13) are given by

$$V_{k_1, k_2}(t_m) := \frac{2}{b_1 - a_1} \frac{2}{b_2 - a_2} \int_{a_2}^{b_2} \int_{a_1}^{b_1} v(t_m, \mathbf{y}) \cos\left(k_1 \pi \frac{y_1 - a_1}{b_1 - a_1}\right) \cos\left(k_2 \pi \frac{y_2 - a_2}{b_2 - a_2}\right) dy_1 dy_2. \quad (14)$$

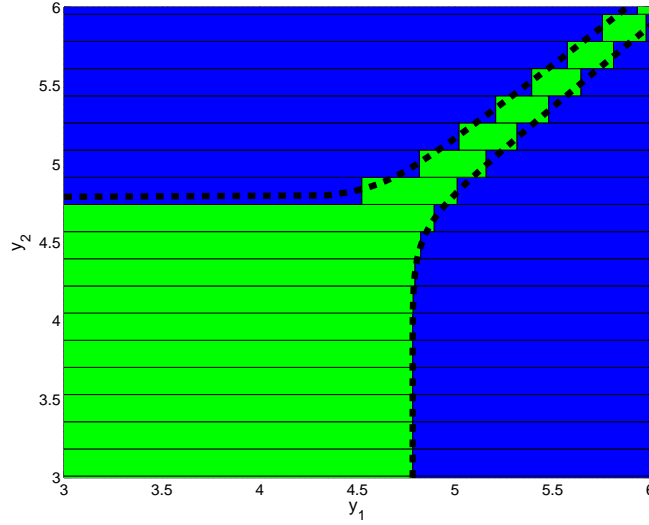
The option function is now approximated by  $\hat{v}(t_{m-1}, \mathbf{x}) := \max[g(\mathbf{x}), \hat{c}(t_{m-1}, \mathbf{x})]$ .

#### 3.1 Recursion formula for coefficients $V_{k_1, k_2}(t_m)$

In this section, a recursive algorithm to recover the coefficients  $V_{k_1, k_2}(t_m)$ , backwards in time, is derived.

In the coefficients  $V_{k_1, k_2}(t_{\mathcal{M}})$ , the terminal condition  $v(t_{\mathcal{M}}, \mathbf{y}) = g(\mathbf{y})$  appears. Some payoff functions provide analytic solutions to these coefficients in (14), otherwise they can be approximated, as explained in Section 3.2.1.

For the coefficients that are used to approximate the continuation values at times  $t_0, \dots, t_{M-2}$ , the value function,  $v(t_m, \mathbf{y}) = \max[g(\mathbf{y}), c(t_m, \mathbf{y})]$ , appears in the terms  $V_{k_1, k_2}(t_m)$  and we need to find an optimal policy for all state values  $\mathbf{y} \in [a_1, b_1] \times [a_2, b_2]$ . We divide the domain  $[a_1, b_1] \times [a_2, b_2]$  into rectangular subdomains  $\mathcal{C}^q$  and  $\mathcal{G}^p$ , so that approximately for all states  $\mathbf{y} \in \mathcal{C}^q$  it is optimal to continue and for all  $\mathbf{y} \in \mathcal{G}^p$  it is optimal to exercise the option. The concept is demonstrated in Figure 1 for a call-on-maximum option. The blue rectangles represent the early-exercise regions  $\mathcal{G}^p$ , the green ones are the continuation regions  $\mathcal{C}^q$ , and the dashed line shows an accurate boundary.

Figure 1: Rectangular regions  $\mathcal{C}^q$  (green) and  $\mathcal{G}^p$  (blue).

We can split the integral in the definition of  $V_{k_1, k_2}$  into different parts:

$$\begin{aligned}
V_{k_1, k_2}(t_m) &= \frac{2}{b_1 - a_1} \frac{2}{b_2 - a_2} \sum_p \iint_{\mathcal{G}^p} g(\mathbf{y}) \cos\left(k_1 \pi \frac{y_1 - a_1}{b_1 - a_1}\right) \cos\left(k_2 \pi \frac{y_2 - a_2}{b_2 - a_2}\right) d\mathbf{y} \\
&\quad + \frac{2}{b_1 - a_1} \frac{2}{b_2 - a_2} \sum_q \iint_{\mathcal{C}^q} c(t_m, \mathbf{y}) \cos\left(k_1 \pi \frac{y_1 - a_1}{b_1 - a_1}\right) \cos\left(k_2 \pi \frac{y_2 - a_2}{b_2 - a_2}\right) d\mathbf{y} \\
&:= \sum_p G_{k_1, k_2}(\mathcal{G}^p) + \sum_q C_{k_1, k_2}(t_m, \mathcal{C}^q), \quad (m \neq 0, \mathcal{M}).
\end{aligned} \tag{15}$$

We approximate the terms  $C_{k_1, k_2}(t_{\mathcal{M}-1}, [z_q, z_{q+1}] \times [w_q, w_{q+1}])$  in (15), where the variables  $z_q$ ,  $z_{q+1}$ ,  $w_q$ , and  $w_{q+1}$  denote the corner points of the rectangular continuation region  $\mathcal{C}^q$ . For the integrand of the terms  $C_{k_1, k_2}$  we apply again the two-dimensional Fourier cosine expansion by inserting the COS formula for  $c(t_{\mathcal{M}-1}, \mathbf{y})$ , i.e. equation (13). The approximation reads

$$\begin{aligned}
&\hat{C}_{k_1, k_2}(t_{\mathcal{M}-1}, [z_q, z_{q+1}] \times [w_q, w_{q+1}]) \\
&:= \frac{2}{b_1 - a_1} \frac{2}{b_2 - a_2} \int_{w_q}^{w_{q+1}} \int_{z_q}^{z_{q+1}} \hat{c}(t_{\mathcal{M}-1}, \mathbf{y}) \cos\left(k_1 \pi \frac{y_1 - a_1}{b_1 - a_1}\right) \cos\left(k_2 \pi \frac{y_2 - a_2}{b_2 - a_2}\right) dy_1 dy_2 \\
&= \int_{w_q}^{w_{q+1}} \int_{z_q}^{z_{q+1}} \sum_{j_1=0}^{N_1-1} \sum_{j_2=0}^{N_2-1} e^{-r\Delta t} \frac{1}{2} \left[ F_{j_1, j_2}^+(\mathbf{y}) + F_{j_1, j_2}^-(\mathbf{y}) \right] V_{j_1, j_2}(t_{\mathcal{M}}) \cos\left(k_1 \pi \frac{y_1 - a_1}{b_1 - a_1}\right) \cos\left(k_2 \pi \frac{y_2 - a_2}{b_2 - a_2}\right) dy_1 dy_2 \\
&= \operatorname{Re} \left( \sum_{j_1=0}^{N_1-1} \sum_{j_2=0}^{N_2-1} \frac{1}{2} e^{-r\Delta t} \varphi_{levy} \left( \frac{j_1 \pi}{b_1 - a_1}, + \frac{j_2 \pi}{b_2 - a_2} \right) V_{j_1, j_2}(t_{\mathcal{M}}) M_{k_1, j_1}^+(z_q, z_{q+1}, a_1, b_1) M_{k_2, j_2}^+(w_q, w_{q+1}, a_2, b_2) \right) \\
&\quad + \operatorname{Re} \left( \sum_{j_1=0}^{N_1-1} \sum_{j_2=0}^{N_2-1} \frac{1}{2} e^{-r\Delta t} \varphi_{levy} \left( \frac{j_1 \pi}{b_1 - a_1}, - \frac{j_2 \pi}{b_2 - a_2} \right) V_{j_1, j_2}(t_{\mathcal{M}}) M_{k_1, j_1}^+(z_q, z_{q+1}, a_1, b_1) M_{k_2, j_2}^-(w_q, w_{q+1}, a_2, b_2) \right),
\end{aligned} \tag{16}$$



where the elements of square-matrices  $M^+$  and  $M^-$  are given by:

$$M_{m,n}^+(u_1, u_2, a, b) := \frac{2}{b-a} \int_{u_1}^{u_2} e^{in\pi \frac{y-a}{b-a}} \cos\left(m\pi \frac{y-a}{b-a}\right) dy, \quad (17)$$

$$M_{m,n}^-(u_1, u_2, a, b) := \frac{2}{b-a} \int_{u_1}^{u_2} e^{-in\pi \frac{y-a}{b-a}} \cos\left(m\pi \frac{y-a}{b-a}\right) dy. \quad (18)$$

We thus find

$$\hat{C}_{k_1, k_2}(t_{\mathcal{M}-1}, [z_q, z_{q+1}] \times [w_q, w_{q+1}]) = \text{Re} \left( \sum_{j_1=0}^{N_1-1} M_{k_1, j_1}^+(z_q, z_{q+1}, a_1, b_1) \mathcal{A}_{j_1, k_2}^q \right), \quad (19)$$

where

$$\begin{aligned} \mathcal{A}_{j_1, k_2}^q &:= \sum_{j_2=0}^{N_2-1} \frac{1}{2} e^{-r\Delta t} \varphi_{levy} \left( \frac{j_1\pi}{b_1-a_1}, +\frac{j_2\pi}{b_2-a_2} \right) V_{j_1, j_2}(t_{\mathcal{M}}) M_{k_2, j_2}^+(w_q, w_{q+1}, a_2, b_2) \\ &+ \sum_{j_2=0}^{N_2-1} \frac{1}{2} e^{-r\Delta t} \varphi_{levy} \left( \frac{j_1\pi}{b_1-a_1}, -\frac{j_2\pi}{b_2-a_2} \right) V_{j_1, j_2}(t_{\mathcal{M}}) M_{k_2, j_2}^-(w_q, w_{q+1}, a_2, b_2). \end{aligned} \quad (20)$$

The elements of  $N_1 \times N_2$ -matrix  $\mathcal{A}^q$  are calculated in a row-wise fashion. The row-vector  $\mathcal{A}_{j_1, \cdot}^q = \{\mathcal{A}_{j_1, k_2}^q\}_{k_2=0}^{N_2-1}$  can be written as two matrix-vector multiplications

$$\mathcal{A}_{j_1, \cdot}^q = M^+(w_q, w_{q+1}, a_2, b_2) \mathbf{w}_{j_1, \cdot}^{q+} + M^-(w_q, w_{q+1}, a_2, b_2) \mathbf{w}_{j_1, \cdot}^{q-}, \quad (21)$$

where

$$\mathbf{w}_{j_1, \cdot}^{q\pm} := \{w_{j_1, j_2}^{q\pm}\}_{j_2=0}^{N_2-1} \quad \text{with} \quad w_{j_1, j_2}^{q\pm} := \frac{1}{2} e^{-r\Delta t} \varphi_{levy} \left( \frac{j_1\pi}{b_1-a_1}, \pm \frac{j_2\pi}{b_2-a_2} \right) V_{j_1, j_2}(t_{\mathcal{M}}). \quad (22)$$

Then, the matrix  $\hat{C}_{k_1, k_2}$  is computed in a column-wise fashion. The column-vector  $\hat{C}_{\cdot, k_2} = \{\hat{C}_{k_1, k_2}\}_{k_1=0}^{N_1-1}$  is calculated by one matrix-vector product

$$\hat{C}_{\cdot, k_2}(t_{\mathcal{M}-1}, [z_q, z_{q+1}] \times [w_q, w_{q+1}]) = \text{Re} \left( M^+(z_q, z_{q+1}, a_1, b_1) \mathcal{A}_{\cdot, k_2}^q \right), \quad (23)$$

with column-vector  $\mathcal{A}_{\cdot, k_2}^q = \{\mathcal{A}_{j_1, k_2}^q\}_{j_1=0}^{N_1-1}$ .

The coefficients  $G_{k_1, k_2}([z_p, z_{p+1}] \times [w_p, w_{p+1}])$  are defined by

$$G_{k_1, k_2}([z_p, z_{p+1}] \times [w_p, w_{p+1}]) = \frac{2}{b_1-a_1} \frac{2}{b_2-a_2} \int_{w_p}^{w_{p+1}} \int_{z_p}^{z_{p+1}} g(\mathbf{y}) \cos\left(k_1\pi \frac{y_1-a_1}{b_1-a_1}\right) \cos\left(k_2\pi \frac{y_2-a_2}{b_2-a_2}\right) dy_1 dy_2. \quad (24)$$

These terms may admit an analytic solution, however, in some practical applications an analytic solution is not present. Methods to approximate these terms are proposed in Section 3.2.2.

We end up with the approximated coefficients

$$\hat{V}_{k_1, k_2}(t_{\mathcal{M}-1}) := \sum_p G_{k_1, k_2}(\mathcal{G}^p) + \sum_q \hat{C}_{k_1, k_2}(t_{\mathcal{M}-1}, \mathcal{C}^q). \quad (25)$$

For the other coefficients  $V_{k_1, k_2}(t_m)$ , the approximations  $\hat{c}(t_m, \mathbf{y})$  and  $\hat{V}_{j_1, j_2}(t_{m+1})$  will be used to approximate the terms  $C_{k_1, k_2}(t_m, [z_q, z_{q+1}] \times [w_q, w_{q+1}])$ , and the elements of the corresponding

matrix  $\mathcal{A}^q$  are

$$\begin{aligned} \mathcal{A}_{j_1, k_2}^q &= \sum_{j_2=0}^{N_2-1} \frac{1}{2} e^{-r\Delta t} \varphi_{levy} \left( \frac{j_1 \pi}{b_1 - a_1}, + \frac{j_2 \pi}{b_2 - a_2} \right) \hat{V}_{j_1, j_2}(t_{m+1}) M_{k_2, j_2}^+(w_q, w_{q+1}, a_2, b_2) \\ &+ \sum_{j_2=0}^{N_2-1} \frac{1}{2} e^{-r\Delta t} \varphi_{levy} \left( \frac{j_1 \pi}{b_1 - a_1}, - \frac{j_2 \pi}{b_2 - a_2} \right) \hat{V}_{j_1, j_2}(t_{m+1}) M_{k_2, j_2}^-(w_q, w_{q+1}, a_2, b_2). \end{aligned} \quad (26)$$

### Fast Fourier Transform

The matrix-vector products  $M^+ \mathbf{v}$  and  $M^- \mathbf{v}$  in the computation of matrices  $\mathcal{A}^q$  and  $\hat{C}$  can be computed efficiently by a Fourier-based algorithm, as stated in Theorem A.1 (Appendix A). The computation time achieved is  $O(N \log_2 N)$ , with  $N$  the length of the vector.

### Algorithm

We can recover the terms  $\hat{V}_{k_1, k_2}(t_m)$  recursively, starting with  $V_{k_1, k_2}(t_{\mathcal{M}})$ . The algorithm for solving the pricing problem backwards in time reads:

**Algorithm 1** (2D-COS method for pricing Bermudan rainbow options)

**Initialisation:** Calculate coefficients  $V_{k_1, k_2}(t_{\mathcal{M}})$ .

**Main loop to recover  $\hat{V}(t_m)$ :**

For  $m = \mathcal{M} - 1$  to 1:

- Determine the optimal continuation regions  $\mathcal{C}^q$  and early-exercise regions  $\mathcal{G}^p$ , as in Figure 1.
- Compute  $\hat{V}(t_m)$  from (15) with the help of the FFT algorithm.

**Final step:** Compute  $\hat{v}(t_0, \mathbf{x}_0)$  by inserting  $\hat{V}_{k_1, k_2}(t_1)$  into equation (13).

### Computational complexity

The initialisation is of order  $O(N_1 N_2)$ . In the main loop there are  $\mathcal{M} - 1$  iterations in which the following computations are performed. The construction of one matrix  $\mathcal{A}^q$  costs  $O(2N_1 N_2 \log_2 N_2)$  operations. Computation of  $\hat{C}_{k_1, k_2}(t_m, [z_q, z_{q+1}] \times [w_q, w_{q+1}])$  takes  $O(N_2 N_1 \log_2 N_1)$  operations.  $G_{k_1, k_2}([z_p, z_{p+1}] \times [w_p, w_{p+1}])$  is of order  $O(N_1 N_2)$ . The computation time is linear in the number of continuation and early-exercise regions. The final step takes  $O(N_1 N_2)$  operations.

## 3.2 Approximation methods for the coefficients $V(T)$ and $G(\mathcal{G}^p)$

In this section, we propose methods to approximate the terminal coefficients  $V_{k_1, k_2, \dots, k_n}(T)$  and the terms  $G_{k_1, k_2, \dots, k_n}(\mathcal{G}^p)$ , that are specific for the multi-D-COS method.

In the one-dimensional pricing problem, the terminal coefficients  $V_{k_1}(T)$  admit analytic solutions for several options, like put and call-based options, digital, and power options. Besides, in the 1D-COS method for pricing Bermudan options, also the terms  $G_{k_1}(\mathcal{G}^p)$  are usually known analytically.

In two dimensions, the payoff functions of, for instance, a geometric basket or a call-on-maximum option provide analytic solutions to the 2D coefficients  $V_{k_1, k_2}(T)$ , but this is generally an exception. If no exact representation is available, then they can be approximated by using *discrete Fourier cosine transforms* (DCTs) or also the *Clenshaw-Curtis quadrature rule*. The usage of DCTs is explained in Section 3.2.1. Also analytic forms for the terms  $G_{k_1, k_2, \dots, k_n}(\mathcal{G}^p)$  are in general not

available in the multi-D version. An approximation method, based on Fourier cosine expansion of the payoff function, is discussed in Section 3.2.2.

### 3.2.1 Discrete Fourier cosine transforms

In this section, we explain this idea of using DCTs to approximate the terminal coefficients  $V_{k_1, k_2}(T)$ . For this, we take  $Q \geq \max[N_1, N_2]$  grid-points for each spatial dimension and define

$$y_i^{n_i} := a_i + (n_i + \frac{1}{2}) \frac{b_i - a_i}{Q} \quad \text{and} \quad \Delta y_i := \frac{b_i - a_i}{Q}, \quad i = 1, 2. \quad (27)$$

The midpoint-rule integration gives us

$$\begin{aligned} V_{k_1, k_2}(T) &\approx \sum_{n_1=0}^{Q-1} \sum_{n_2=0}^{Q-1} \frac{2}{b_1 - a_1} \frac{2}{b_2 - a_2} g(y_1^{n_1}, y_2^{n_2}) \cos\left(k_1 \pi \frac{y_1^{n_1} - a_1}{b_1 - a_1}\right) \cos\left(k_2 \pi \frac{y_2^{n_2} - a_2}{b_2 - a_2}\right) \Delta y_1 \Delta y_2 \\ &= \sum_{n_1=0}^{Q-1} \sum_{n_2=0}^{Q-1} g(y_1^{n_1}, y_2^{n_2}) \cos\left(k_1 \pi \frac{2n_1 + 1}{2Q}\right) \cos\left(k_2 \pi \frac{2n_2 + 1}{2Q}\right) \frac{2}{Q} \frac{2}{Q}. \end{aligned} \quad (28)$$

The above two-dimensional DCT (Type II) can be calculated efficiently by, for example, MATLAB's function `dct2`. The approximated coefficients are denoted by  $V_{k_1, k_2}^{DCT}(T)$ , with the corresponding computed European option value  $\hat{v}^{DCT}(t_0, \mathbf{x})$ . Now, an extra error is introduced,

$$\begin{aligned} \epsilon_{DCT}(t_0, \mathbf{x}) &:= \hat{v}(t_0, \mathbf{x}) - \hat{v}^{DCT}(t_0, \mathbf{x}) \\ &= \frac{b_1 - a_1}{2} \frac{b_2 - a_2}{2} e^{-r\Delta t} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} F_{k_1, k_2}(\mathbf{x}) [V_{k_1, k_2}(T) - V_{k_1, k_2}^{DCT}(T)]. \end{aligned} \quad (29)$$

This error  $\epsilon_{DCT}$  converges algebraically in  $Q$ , with order two. We will confirm this by an example in Section 6.1.1. The DCT-method can be extended to higher dimensions, for the approximation of coefficients  $V_{k_1, k_2, \dots, k_n}(T)$ .

**Remark 2** *The above approximation, (28), is based on the midpoint-rule integration. Higher-order methods or adaptive quadrature rules may improve the efficiency.*

### 3.2.2 Approximation methods for $G(\mathcal{G}^p)$

The terms  $G_{k_1, k_2}$  are defined by

$$G_{k_1, k_2}([z_p, z_{p+1}] \times [w_p, w_{p+1}]) = \frac{2}{b_1 - a_1} \frac{2}{b_2 - a_2} \int_{w_p}^{w_{p+1}} \int_{z_p}^{z_{p+1}} g(\mathbf{y}) \cos\left(k_1 \pi \frac{y_1 - a_1}{b_1 - a_1}\right) \cos\left(k_2 \pi \frac{y_2 - a_2}{b_2 - a_2}\right) dy_1 dy_2. \quad (30)$$

These terms may admit an analytic solution, however, in many practical applications the calculation of coefficients  $G_{k_1, k_2}(\mathcal{G}^p)$  is time-consuming, or an analytic solution is not present. Then, we can use *discrete Fourier transforms* to approximate them, similar as in Section 3.2.1. Another way is the usage of the Fourier cosine expansion of the payoff function. First, we explain this idea in 1D.

The COS method for one-dimensional Bermudan options was developed in [12]. For a put option it follows that

$$V_{k_1}(t_m) = G_{k_1}(\mathcal{G}) + C_{k_1}(t_m, \mathcal{C}). \quad (31)$$

The coefficients  $C_{k_1}$  are approximated by

$$\hat{C}_{k_1}(t_m, [z_1, z_2]) = \operatorname{Re} \left( \sum_{j=0}^{N_1-1} e^{-r\Delta t} \varphi_{levy} \left( \frac{j\pi}{b-a} \right) V_j(t_{m+1}) M_{k_1, j_1}^+(z_1, z_2, a, b) \right). \quad (32)$$

In [12], the coefficients  $G_{k_1}(z_1, z_2)$  are assumed to be known analytically, and

$$\hat{V}_{k_1}(t_m) := G_{k_1}(\mathcal{G}) + \hat{C}_{k_1}(t_m, \mathcal{C}). \quad (33)$$

However, the coefficients  $G_{k_1}$  can also be approximated by a Fourier series expansion of the payoff function, i.e.

$$g(x) \approx \hat{g}(x) = \sum_{k_1=0}^{N_1-1} \cos \left( k\pi \frac{x-a}{b-a} \right) V_{k_1}(T) = \sum_{k_1=0}^{N_1-1} \operatorname{Re} \left( \exp \left( ik\pi \frac{x-a}{b-a} \right) \right) V_{k_1}(T), \quad (34)$$

with  $\{V_{k_1}(T)\}_{k_1=0}^{+\infty}$  the terminal coefficients,  $V(T, y) = g(y)$ . Inserting function  $\hat{g}$  in the terms  $G_{k_1}$  gives

$$\hat{G}_{k_1}([z_1, z_2]) := \frac{2}{b-a} \int_{z_1}^{z_2} \hat{g}(y) \cos \left( k\pi \frac{y-a}{b-a} \right) dy = \operatorname{Re} \left( \sum_{j=0}^{N_1-1} V_{j_1}(T) M_{k_1, j_1}^+(z_1, z_2, a, b) \right). \quad (35)$$

The computation of  $\hat{C}$  and  $\hat{G}$  can now be done simultaneously, as

$$\begin{aligned} \hat{V}_{k_1}(t_m) &:= \hat{G}_{k_1}(\mathcal{G}) + \hat{C}_{k_1}(t_m, \mathcal{C}) \\ &= \hat{V}_{k_1}(T) - \hat{G}_{k_1}(\mathcal{C}) + \hat{C}_{k_1}(t_m, \mathcal{C}) \\ &= \hat{V}_{k_1}(T) + \operatorname{Re} \left( \sum_{j=0}^{N_1-1} \left[ -V_{j_1}(T) + e^{-r\Delta t} \varphi_{levy} \left( \frac{j\pi}{b-a} \right) V_{j_1}(t_{m+1}) \right] M_{k_1, j_1}^+(\mathcal{C}, a, b) \right). \end{aligned} \quad (36)$$

The error of this new approach converges algebraically, as the Fourier series of the payoff function. The new approach is a little bit faster than (33) (with analytic  $G_{k_1}(\mathcal{G})$ ), however, a higher value of  $N_1$  is needed to reach the same accuracy, see Table 1 and Figure 2. However, in multiple dimensions this approach may be beneficial and time-efficient.

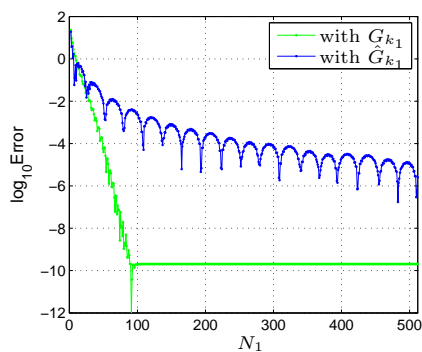


Table 1: Error, reference price is 10.479520123.

$N_1$	method [12], (33)	new method, (36)
32	-4.893e-3	-6.22e-2
64	-1.393e-6	1.14e-2
128	-2.023e-10	1.22e-3
256	-2.023e-10	-4.43e-5
512	-2.024e-10	2.52e-6
1024	-2.024e-10	-1.18e-7

Figure 2: Comparison between method (33) and (36).

Now we return to equation (30) and explain the approximation method for 2D. The Fourier cosine expansion of the payoff function can be written as

$$\begin{aligned}\hat{g}(\mathbf{y}) &:= \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} \cos\left(k_1\pi\frac{y_1-a_1}{b_1-a_1}\right) \cos\left(k_2\pi\frac{y_2-a_2}{b_2-a_2}\right) V_{k_1,k_2}(T) \\ &= \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} \frac{1}{2} \left[ \operatorname{Re}\left(\exp\left(ik_1\pi\frac{y_1-a_1}{b_1-a_1} + ik_2\pi\frac{y_2-a_2}{b_2-a_2}\right)\right) \right. \\ &\quad \left. + \operatorname{Re}\left(\exp\left(ik_1\pi\frac{y_1-a_1}{b_1-a_1} - ik_2\pi\frac{y_2-a_2}{b_2-a_2}\right)\right) \right] V_{k_1,k_2}(T).\end{aligned}\quad (37)$$

With this, the coefficients  $G_{k_1,k_2}$  can be approximated by  $\hat{G}_{k_1,k_2}$ , similar as in (16),

$$\begin{aligned}\hat{G}_{k_1,k_2}([z_p, z_{p+1}] \times [w_p, w_{p+1}]) &= \operatorname{Re}\left(\sum_{j_1=0}^{N_1-1} \sum_{j_2=0}^{N_2-1} \frac{1}{2} V_{j_1,j_2}(T) M_{k_1,j_1}^+(z_p, z_{p+1}, a_1, b_1) M_{k_2,j_2}^+(w_p, w_{p+1}, a_2, b_2)\right) \\ &\quad + \operatorname{Re}\left(\sum_{j_1=0}^{N_1-1} \sum_{j_2=0}^{N_2-1} \frac{1}{2} V_{j_1,j_2}(T) M_{k_1,j_1}^+(z_p, z_{p+1}, a_1, b_1) M_{k_2,j_2}^-(w_p, w_{p+1}, a_2, b_2)\right).\end{aligned}\quad (38)$$

We will use these approximations in Sections 6.2.1 and 6.2.2. This approximation method can be extended to higher dimensions; if the multi-D coefficients  $G_{k_1,k_2,\dots,k_n}(\mathcal{G}^p)$  are not known analytically, then we can use the terminal coefficients  $V_{k_1,k_2,\dots,k_n}(T)$  to approximate them.

## 4 Bermudan options under the Heston model

In this section, we explain the 2D-COS method for a Bermudan pricing problem, in which the asset prices follows the Heston dynamics. This two-dimensional stochastic process is one of the important processes in financial option pricing. In [13], Bermudan options under the Heston stochastic volatility model are priced using the COS formula for the log-asset dimension, combined with a quadrature rule in the log-variance dimension. For this, the closed-form density function of the transformed log-variance process is used. In our approach we employ the *bivariate characteristic function* of the log-asset price and the variance, which is also available in closed-form, and the 2D-COS formula can be applied. However, the two-dimensional stochastic process is not in the class of Lévy processes and efficient matrix-vector multiplication using the FFT algorithm (Appendix A) within the COS method will therefore be applicable in only one dimension, similar as in [28].

Let process  $\mathbf{X}_t = (X_t^1, X_t^2) = (X_t, \nu_t)$  represent the Heston model.  $X_t$  represents the log-asset price process and  $\nu_t$  is the variance process ( $\nu_0 \geq 0$ ), with dynamics

$$dX_t = \left(r - \frac{1}{2}\nu_t\right)dt + \rho\sqrt{\nu_t}dW_t^1 + \sqrt{1-\rho^2}\sqrt{\nu_t}dW_t^2, \quad (39)$$

$$d\nu_t = \kappa(\bar{\nu} - \nu_t)dt + \eta\sqrt{\nu_t}dW_t^1. \quad (40)$$

Here,  $r$  represents the risk free rate,  $\kappa > 0$  the mean reversion rate,  $\bar{\nu} > 0$  the long run variance,  $\eta > 0$  the volatility of variance (vol-of-vol), and  $\rho$  the correlation coefficient.  $\mathbf{W}_t$  is a two-dimensional Brownian motion on the filtered probability space  $(\Omega, \mathcal{F}, \mathbb{F}, P)$ . The variance process remains strictly positive if the Feller condition,  $2\kappa\bar{\nu} \geq \eta^2$ , is satisfied, otherwise the boundary at zero is attainable and strongly reflecting ([17]).

The process  $\mathbf{X}_t$  is affine and the *bivariate characteristic function* is of the form

$$\phi(\mathbf{u}, \mathbf{X}_t, t, T) = \mathbb{E}[e^{iu_1 X_T + iu_2 \nu_T} | X_t, \nu_t] = \exp\left(B_1(T-t, \mathbf{u})X_t + B_2(T-t, \mathbf{u})\nu_t + A(T-t, \mathbf{u})\right). \quad (41)$$

We define

$$\beta := \kappa - i\rho\eta u_1, \quad (42)$$

$$D := \sqrt{\beta^2 + \eta^2 u_1(i + u_1)}, \quad (43)$$

$$h := (\beta - D - iu_2\eta^2)/(\beta + D - iu_2\eta^2). \quad (44)$$

The functions  $A$ ,  $B_1$ , and  $B_2$  are solutions to a system of ordinary differential equations (ODEs) of Riccati type ([9]):

$$\frac{\partial B_1(t, \mathbf{u})}{\partial t} = 0, \quad (45)$$

$$\frac{\partial B_2(t, \mathbf{u})}{\partial t} = \frac{1}{2}\eta^2 B_2^2(t, \mathbf{u}) - \beta B_2(t, \mathbf{u}) - \frac{1}{2}u_1(i + u_1), \quad (46)$$

$$\frac{\partial A(t, \mathbf{u})}{\partial t} = iu_1 r + \kappa \bar{\nu} B_2(t, \mathbf{u}), \quad (47)$$

with initial conditions  $B_1(0, \mathbf{u}) = iu_1$ ,  $B_2(0, \mathbf{u}) = iu_2$ ,  $A(0, \mathbf{u}) = 0$ . Solving the ODEs gives

$$B_1(t, \mathbf{u}) = iu_1, \quad (48)$$

$$B_2(t, \mathbf{u}) = \frac{1}{\eta^2} \frac{\beta - D - (\beta + D)he^{-Dt}}{1 - he^{-Dt}}, \quad (49)$$

$$A(t, \mathbf{u}) = iu_1 r t + \frac{\kappa \bar{\nu}}{\eta^2} \left[ (\beta - D)t - 2 \ln \left( \frac{he^{-Dt} - 1}{h - 1} \right) \right]. \quad (50)$$

**Remark 3** *The characteristic function involves a multi-valued complex logarithm, in  $A(t, \mathbf{u})$ . Most software packages restrict the logarithm to its principal branch. Then, the characteristic function can become discontinuous in  $\mathbf{u}$ , which results in incorrect option prices. The same problem arises in the one-dimensional characteristic function,*

$$\phi^{1D}(u_1, \mathbf{X}_t, t, T) = \mathbb{E}[e^{iu_1 X_T} | X_t, \nu_t]. \quad (51)$$

*Solutions to the problem of choosing the correct branch here are discussed extensively in [22, 23, 1, 18]. In [23], the function  $A(t, u_1)$  ( $u_2 = 0$ ) appears in the expression for the one-dimensional characteristic function. They prove that the argument of the logarithm never crosses the negative real axis, so that the principle branch is the correct one. We have not been able to complete the proof that justifies the use of the principle branch for  $A(t, \mathbf{u})$  in 2D. However, experiments showed that the logarithm's argument does not cross the real negative real axis for the parameter values in Section 7.*

The variance process can be expressed as a time-changed squared Bessel process with dimension  $\delta := \frac{4\kappa\bar{\nu}}{\eta^2}$ , which has an absolutely continuous distribution if  $\delta > 0$  (i.e., a density exists) and a probability mass at the origin if  $\delta = 0$  ([19, 10]). The assumptions on the parameters yield  $\delta > 0$ , which justifies the use of the analytic density function in equation (1) and approximated coefficients  $F_{k_1, k_2}$  (equation (10)), with characteristic function given by equation (11).

We apply the 2D-COS method to price Bermudan options under the Heston dynamics. We take again fixed time steps  $\Delta t := t_{m+1} - t_m$  and define

$$\varphi(u_1, u_2 | x, \nu) := \phi(\mathbf{u}, \mathbf{x}, t_m, t_{m+1}) := e^{iu_1 x} e^{B_2(\Delta t, \mathbf{u})\nu} \varphi_A(\mathbf{u}), \quad \text{where } \varphi_A(\mathbf{u}) := e^{A(\Delta t, \mathbf{u})}. \quad (52)$$

The approximation of the coefficients  $C_{k_1, k_2}$  now reads

$$\begin{aligned}
& \hat{C}_{k_1, k_2}(t_{\mathcal{M}-1}, [z_q, z_{q+1}] \times [w_q, w_{q+1}]) \\
&= \operatorname{Re} \left( \sum_{j_1=0}^{N_1-1} \sum_{j_2=0}^{N_2-1} \frac{1}{2} e^{-r\Delta t} \varphi_A \left( \frac{j_1\pi}{b_1-a_1}, +\frac{j_2\pi}{b_2-a_2} \right) V_{j_1, j_2}(t_{\mathcal{M}}) M_{k_1, j_1}^+(z_q, z_{q+1}, a_1, b_1) H_{k_2, j_2}^+ \left( w_q, w_{q+1}, a_2, b_2, \frac{j_1\pi}{b_1-a_1} \right) \right) \\
&+ \operatorname{Re} \left( \sum_{j_1=0}^{N_1-1} \sum_{j_2=0}^{N_2-1} \frac{1}{2} e^{-r\Delta t} \varphi_A \left( \frac{j_1\pi}{b_1-a_1}, -\frac{j_2\pi}{b_2-a_2} \right) V_{j_1, j_2}(t_{\mathcal{M}}) M_{k_1, j_1}^+(z_q, z_{q+1}, a_1, b_1) H_{k_2, j_2}^- \left( w_q, w_{q+1}, a_2, b_2, \frac{j_1\pi}{b_1-a_1} \right) \right), \\
&:= \operatorname{Re} \left( \sum_{j_1=0}^{N_1-1} M_{k_1, j_1}^+(z_q, z_{q+1}, a_1, b_1) \mathcal{A}_{j_1, k_2}^q \right), \tag{53}
\end{aligned}$$

where the elements of matrices  $H^\pm$  are given by

$$H_{k_2, j_2}^\pm \left( w_q, w_{q+1}, a_2, b_2, \frac{j_1\pi}{b_1-a_1} \right) := \frac{2}{b_2-a_2} \int_{w_q}^{w_{q+1}} e^{y_2 B_2 \left( \Delta t, \frac{j_1\pi}{b_1-a_1}, \pm \frac{j_2\pi}{b_2-a_2} \right)} e^{\pm i j_2 \pi \frac{-a_2}{b_2-a_2}} \cos \left( k_2 \pi \frac{y_2 - a_2}{b_2-a_2} \right) dy_2. \tag{54}$$

As before, the elements of  $N_1 \times N_2$ -matrix  $\mathcal{A}^q$  are calculated in a row-wise fashion and the row-vector  $\mathcal{A}_{j_1, \cdot}^q = \{\mathcal{A}_{j_1, k_2}^q\}_{k_2=0}^{N_2-1}$  can be written as two matrix-vector multiplications

$$\mathcal{A}_{j_1, \cdot}^q = H^+ \left( w_q, w_{q+1}, a_2, b_2, \frac{j_1\pi}{b_1-a_1} \right) \mathbf{w}_{j_1, \cdot}^{q+} + H^- \left( w_q, w_{q+1}, a_2, b_2, \frac{j_1\pi}{b_1-a_1} \right) \mathbf{w}_{j_1, \cdot}^{q-}, \tag{55}$$

where

$$\mathbf{w}_{j_1, \cdot}^{q\pm} := \{w_{j_1, j_2}^{q\pm}\}_{j_2=0}^{N_2-1} \quad \text{with} \quad w_{j_1, j_2}^{q\pm} := \frac{1}{2} e^{-r\Delta t} \varphi_A \left( \frac{j_1\pi}{b_1-a_1}, \pm \frac{j_2\pi}{b_2-a_2} \right) V_{j_1, j_2}(t_{\mathcal{M}}). \tag{56}$$

The above equations are similar to equations (16) and (20) for pricing rainbow options. However, the matrices  $H^\pm$  cannot be decomposed into a Hankel and Toeplitz matrix anymore and the computation of  $\mathcal{A}_{j_1, \cdot}^q$  is therefore of order  $O(N_2^2)$ . For the numerical tests in Section 7 we will take a fixed grid  $\{w_q\}$ . Then, the matrices  $H^\pm$  need to be computed only once and this computation is part of the algorithm's initialisation.

## 5 Error convergence and computational domain

The method parameters of the 2D-COS method are the integration ranges,  $[a_1, b_1]$  and  $[a_2, b_2]$ , and the number of Fourier cosine terms,  $N_1$  and  $N_2$ . Convergence in the number of cosine terms is discussed in the error analysis in Section 5.1. Section 5.2 gives some suggestions for the choice of the computational domain.

### 5.1 Error analysis

The error analysis is similar to the analyses in [11] and [12]. We start with the local error of the COS formula. Then, a result for the propagating error is presented.

The local error of the COS formula is defined by

$$\epsilon(t_{m-1}, \mathbf{x}) := c(t_{m-1}, \mathbf{x}) - \hat{c}(t_{m-1}, \mathbf{x}). \tag{57}$$

Let's assume that the terms  $V_{k_1, k_2}(t_m)$  are known analytically. Then, errors are introduced in three steps (see Section 2), the truncation of the integration range, the substitution of the density by its cosine series expansion on the truncated range, and the substitution of the series coefficients by the characteristic function approximation. The key to bound the error lies in the decay rate of the Fourier-cosine series coefficients. We discuss the three errors one after the other:

Step 1: The integration range truncation error:

$$\epsilon_1(t_{m-1}, \mathbf{x}) := c(t_{m-1}, \mathbf{x}) - c_1(t_{m-1}, \mathbf{x}) = e^{-r\Delta t} \iint_{\mathbb{R}^2 \setminus [a_1, b_1] \times [a_2, b_2]} v(t_m, \mathbf{y}) f(\mathbf{y}|\mathbf{x}) d\mathbf{y}. \quad (58)$$

Step 2: The series truncation error:

$$\epsilon_2(t_{m-1}, \mathbf{x}) := c_1(t_{m-1}, \mathbf{x}) - c_2(t_{m-1}, \mathbf{x}) = \frac{b_1 - a_1}{2} \frac{b_2 - a_2}{2} e^{-r\Delta t} \sum_{\substack{k_1, k_2 \\ k_1 \geq N_1 \text{ or } k_2 \geq N_2}}^{+\infty} A_{k_1, k_2}(\mathbf{x}) V_{k_1, k_2}(t_m). \quad (59)$$

The convergence rate of Fourier cosine series depends on the properties of the approximated functions in the expansion interval. The coefficients  $A_{k_1, k_2}$  usually decay faster than  $V_{k_1, k_2}$ . With [4, 11], we find that the error converges exponentially in  $N_1$  and  $N_2$  for density functions in the class  $C^\infty([a_1, b_1] \times [a_2, b_2])$ . A density function with discontinuity in one of its derivatives results in an algebraic convergence of the Fourier cosine series expansion.

Step 3: The error related to approximating  $A_{k_1, k_2}(\mathbf{x})$  by  $F_{k_1, k_2}(\mathbf{x})$  (equation (6)):

$$\begin{aligned} \epsilon_3(t_{m-1}, \mathbf{x}) &:= c_2(t_{m-1}, \mathbf{x}) - \hat{c}(t_{m-1}, \mathbf{x}) \\ &= \frac{b_1 - a_1}{2} \frac{b_2 - a_2}{2} e^{-r\Delta t} \sum'_{k_1=0}^{N_1-1} \sum'_{k_2=0}^{N_2-1} (A_{k_1, k_2}(\mathbf{x}) - F_{k_1, k_2}(\mathbf{x})) V_{k_1, k_2}(t_m) \\ &= -e^{-r\Delta t} \iint_{\mathbb{R}^2 \setminus [a_1, b_1] \times [a_2, b_2]} \left[ \sum'_{k_1=0}^{N_1-1} \sum'_{k_2=0}^{N_2-1} \cos\left(k_1 \pi \frac{y_1 - a_1}{b_1 - a_1}\right) \cos\left(k_2 \pi \frac{y_2 - a_2}{b_2 - a_2}\right) V_{k_1, k_2}(t_m) \right] f(\mathbf{y}|\mathbf{x}) d\mathbf{y}. \end{aligned} \quad (60)$$

Addition of the three errors gives

$$\begin{aligned} \epsilon(t_{m-1}, \mathbf{x}) &= e^{-r\Delta t} \iint_{\mathbb{R}^2 \setminus [a_1, b_1] \times [a_2, b_2]} [v(t_m, \mathbf{y}) - \hat{v}(t_m, \mathbf{y})] f(\mathbf{y}|\mathbf{x}) d\mathbf{y} \\ &\quad + e^{-r\Delta t} \iint_{[a_1, b_1] \times [a_2, b_2]} [v(t_m, \mathbf{y}) f(\mathbf{y}|\mathbf{x}) - \hat{v}(t_m, \mathbf{y}) \hat{f}(\mathbf{y}|\mathbf{x})] d\mathbf{y}. \end{aligned} \quad (61)$$

**Result 1** *Let's assume that the terms  $V_{k_1, k_2}(t_m)$  are exact. If the integration domain  $[a_1, b_1] \times [a_2, b_2]$  is chosen sufficiently wide, then the series truncation error  $\epsilon_2$  dominates the overall local error. Then, for smooth density functions the error  $\epsilon$  converges exponentially in  $N_1$  and  $N_2$ , otherwise it goes algebraically. The extra error  $\epsilon_{DCT}$ , introduced by approximation of the terms  $V_{k_1, k_2}(T)$  with DCTs, has been discussed in Section 3.2.1.*

We now discuss the error in the terms  $V_{k_1, k_2}(t_m)$  and define

$$\epsilon_{k_1, k_2}(t_m, \mathcal{C}^q) := C_{k_1, k_2}(t_m, \mathcal{C}^q) - \hat{C}_{k_1, k_2}(t_m, \mathcal{C}^q). \quad (62)$$



The terms  $G_{k_1, k_2}(\mathcal{G}^p)$  are assumed to be exact, so that the error in the Fourier coefficients is given by

$$\varepsilon_{k_1, k_2}(t_m) := V_{k_1, k_2}(t_m) - \hat{V}_{k_1, k_2}(t_m) = \sum_q \varepsilon_{k_1, k_2}(t_m, \mathcal{C}^q). \quad (63)$$

**Result 2** *With  $[a_1, b_1] \times [a_2, b_2] \subset \mathbb{R}^2$  chosen sufficiently wide and a probability density function  $f$  in  $C^\infty([a_1, b_1] \times [a_2, b_2])$ , error  $\varepsilon_{k_1, k_2}(t_m)$  converges exponentially in  $N_1$  and  $N_2$  for  $1 \leq m \leq \mathcal{M}-1$ . If the local error converges algebraically, then so does error  $\varepsilon_{k_1, k_2}(t_m)$ .*

The proof of this result is similar to that for pricing Bermudan options with one underlying asset, which can be found in [12]. The convergence is algebraically for non-smooth density functions or if we approximate  $G_{k_1, k_2}(\mathcal{G}^p)$  by an algebraically converging method, see Section 3.2.2.

## 5.2 Computational domain

The performance of the (2D-)COS method is sensitive to the choice of the computational domain  $[a_1, b_1] \times [a_2, b_2]$ . If the domain size is set too small, then the resulting option values may be too low. However, the larger the domain, the more terms in the series expansions are required to reach a certain accuracy.

For the tests on rainbow options, we will take  $a_1 = a_2 = a$  and  $b_1 = b_2 = b$ , where

$$a := \min_i \left[ x_0^i + \xi_1^i - L\sqrt{\xi_2^i + \sqrt{\xi_4^i}} \right], \quad b := \max_i \left[ x_0^i + \xi_1^i + L\sqrt{\xi_2^i + \sqrt{\xi_4^i}} \right], \quad L = 10. \quad (64)$$

$\xi_j^i$  denotes the  $j$ -th cumulant of the stochastic variable  $X_T^i$ . For the cumulants  $\xi_1, \xi_2$ , and  $\xi_4$  of the Brownian motion and the log-jump-diffusion process we refer to [11]. The choice of equal domains for both dimensions facilitates the calculation of the coefficients  $V_{k_1, k_2}(T)$ .

For the Heston dynamics, we base the interval  $[a_1, b_1]$  on the cumulants  $\xi_j$  of  $X_T$ , similar as in [11],

$$[a_1, b_1] := \left[ x_0 + \xi_1 - L\sqrt{\xi_2}, x_0 + \xi_1 + L\sqrt{\xi_2} \right], \quad L = 8. \quad (65)$$

The second cumulant can be approximated by  $\xi_2 \approx \bar{\nu}(1 + \eta)T$ . For  $a_2 \geq 0$  and  $b_2$ , however, we take a tolerance level  $\text{TOL} = 10^{-4}$  and determine the integration range so that

$$F_{\nu_T|\nu_0}(a_2|\nu_0) = \text{TOL} = 1 - F_{\nu_T|\nu_0}(b_2|\nu_0), \quad (66)$$

where  $F_{\nu_T|\nu_0}$  represents the cumulative distribution function of the variance at the terminal time. This is a general strategy in the case of non-smooth densities.

If  $N_1$  and  $N_2$  are chosen sufficiently large, then a larger size of the computational domain should not affect the option price.

## 6 Numerical experiments rainbow options

In this section we perform several numerical experiments to test the 2D-COS method for pricing European and Bermudan rainbow options. We will validate the algorithm and show its efficiency.

The following options are studied: geometric basket, arithmetic basket, put-on-minimum, and call-on-maximum options. The asset price paths are modelled by either correlated geometric Brownian motions (GBM) or by Merton's jump-diffusion processes (but in principle we can use other Lévy processes). MATLAB 7.11.0 is used for the computations, with an Intel(R) Quad-Core 2.83 GHz and 8 GB RAM. For the tests in this section, we take an equal number of terms in both series expansions, that is  $N_2 = N_1$ .

**Geometric Brownian motion** Under GBM the risk-neutral asset prices evolve according to the following dynamics

$$dS_t^i = (r - \delta_i)S_t^i dt + S_t^i \sigma_i dZ_t^i, \quad i = 1, 2, \quad (67)$$

with correlation  $dZ_t^i dZ_t^j = \rho_{ij} dt$ ,  $r$  the risk-free rate,  $\delta_i$  the dividend rate, and  $\sigma_i$  the volatility of asset  $i$ . We switch to the log-processes  $X_t^i := \log S_t^i$ ,

$$dX_t^i = (r - \delta_i - \frac{1}{2}\sigma_i^2)dt + \sigma_i dZ_t^i. \quad (68)$$

The log-asset prices at time  $t_m$ , given the values at time  $t_{m-1}$ , are bivariate normally distributed,

$$\mathbf{X}_{t_m} \sim \mathcal{N}(\mathbf{X}_{t_{m-1}} + \boldsymbol{\mu}, \Sigma), \quad (69)$$

with  $\mu_i = (r - \delta_i - \frac{1}{2}\sigma_i^2)\Delta t$  and covariance matrix  $\Sigma_{ij} = \sigma_i \sigma_j \rho_{ij} \Delta t$ . The characteristic function reads  $\varphi(\mathbf{u}|\mathbf{x}) = e^{i\mathbf{x}'\mathbf{u}} \varphi_{levy}(\mathbf{u})$  with

$$\varphi_{levy}(\mathbf{u}) = \exp(i\boldsymbol{\mu}'\mathbf{u} - \frac{1}{2}\mathbf{u}'\Sigma\mathbf{u}). \quad (70)$$

**Jump-diffusion process** Under jump-diffusion the asset prices follow

$$dS_t^i = (r - \lambda \varkappa_i)S_t^i dt + S_t^i \sigma_i dZ_t^i + (e^{J_i} - 1)S_t^i dq_t, \quad i = 1, 2, \quad (71)$$

with  $\varkappa_i := \mathbb{E}[e^{J_i} - 1]$ ,  $q_t$  a Poisson process with mean arrival rate  $\lambda$ , and  $\mathbf{J} = (J_1, J_2)$  bivariate normally distributed jumps, with mean  $\boldsymbol{\mu}^J = [\mu_1^J, \mu_2^J]'$  and covariance matrix  $\Sigma_{ij}^J = \sigma_i^J \sigma_j^J \rho_{ij}^J$ . The log-processes  $X_t^i := \log S_t^i$  read

$$dX_t^i = (r - \lambda \varkappa_i - \frac{1}{2}\sigma_i^2)dt + \sigma_i dZ_t^i + J_i dq_t. \quad (72)$$

The characteristic function reads  $\varphi(\mathbf{u}|\mathbf{x}) = e^{i\mathbf{x}'\mathbf{u}} \varphi_{levy}(\mathbf{u})$  with

$$\varphi_{levy}(\mathbf{u}) = \exp(i\boldsymbol{\mu}'\mathbf{u} - \frac{1}{2}\mathbf{u}'\Sigma\mathbf{u}) \exp\left(\lambda \Delta t (\exp(i\boldsymbol{\mu}^J'\mathbf{u} - \frac{1}{2}\mathbf{u}'\Sigma^J\mathbf{u}) - 1)\right), \quad (73)$$

where  $\mu_i = (r - \lambda \varkappa_i - \frac{1}{2}\sigma_i^2)\Delta t$  and  $\Sigma_{ij} = \sigma_i \sigma_j \rho_{ij} \Delta t$ .

The bivariate density functions of the correlated Brownian motion and jump-diffusion process are both in  $C^\infty$ , which will result in exponentially converging errors in  $N_1$  and  $N_2$ . *Density recovery* using Fourier cosine expansions and the characteristic function gives us

$$\hat{f}(\mathbf{y}|\mathbf{x}) := \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} F_{k_1, k_2}(\mathbf{x}) \cos\left(k_1 \pi \frac{y_1 - a_1}{b_1 - a_1}\right) \cos\left(k_2 \pi \frac{y_2 - a_2}{b_2 - a_2}\right). \quad (74)$$

The approximated density function of a log-jump-diffusion process is presented in Figure 3.

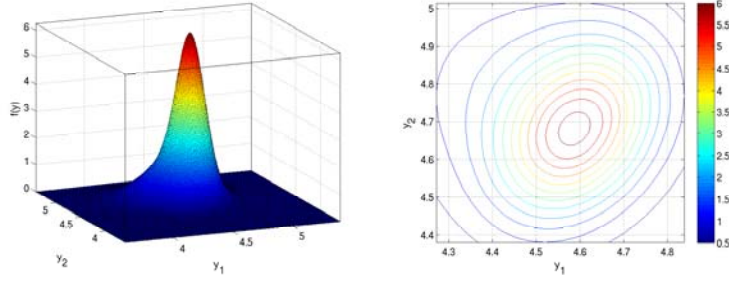


Figure 3: Density recovery jump-diffusion process  $\hat{f}(\mathbf{X}_T|\mathbf{x}_0)$  (Parameter set II).

**Parameter sets** The following five parameters sets are used for the rainbow options.

Set I:

$$\mathbf{S}_0 = [90 \quad 110]', r = 0.04, \delta_i = 0, \sigma = [0.2 \quad 0.3], \rho = \begin{bmatrix} 1 & 0.25 \\ 0.25 & 1 \end{bmatrix}, t_0 = 0, T = 1, K = 100.$$

Set II:

$$\mathbf{S}_0 = [90 \quad 110]', r = 0.05, \sigma = [0.12 \quad 0.15], \rho = \begin{bmatrix} 1 & 0.30 \\ 0.30 & 1 \end{bmatrix}, t_0 = 0, T = 1, K = 100,$$

$$\lambda = 0.60, \boldsymbol{\mu}^J = [-0.10, 0.10]', \boldsymbol{\sigma}^J = [0.17 \quad 0.13], \boldsymbol{\rho}^J = \begin{bmatrix} 1 & -0.20 \\ -0.20 & 1 \end{bmatrix}.$$

Set III:

$$\mathbf{S}_0 = [40 \quad 40]', r = 0.048790, \delta_i = 0, \sigma = [0.2 \quad 0.3], \rho = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}, t_0 = 0, T = 7/12, K = 40.$$

Set IV:

$$\mathbf{S}_0 = [100 \quad 100 \quad 100]', r = 0.04, \delta_i = 0, \sigma = [0.3 \quad 0.35 \quad 0.4], \rho = \begin{bmatrix} 1 & 0.5 & 0.5 \\ 0.5 & 1 & 0.5 \\ 0.5 & 0.5 & 1 \end{bmatrix}, t_0 = 0$$

$$T = 1, K = 100.$$

Set V:

$$r = 0.05, \delta_i = 0.1, \sigma = [0.2 \quad 0.2], \rho = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, t_0 = 0, T = 3, K = 100.$$

## 6.1 European rainbow options

We start in Sections 6.1.1 and 6.1.2 by pricing two types of basket options with two underlying asset prices, namely,

- geometric basket call options, with payoff  $g(\mathbf{x}) = (\sqrt{e^{x_1}}\sqrt{e^{x_2}} - K)^+$ ,
- arithmetic basket call options, with payoff  $g(\mathbf{x}) = (\frac{1}{2}e^{x_1} + \frac{1}{2}e^{x_2} - K)^+$ .

Then, Section 6.1.3 discusses call-on-maximum and put-on-minimum options.

### 6.1.1 Geometric basket call option

The price of a geometric basket call option under GBM equals the price of a Black-Scholes call option with initial option price  $\hat{S}_0 = \sqrt{S_0^1} \sqrt{S_0^2}$ , dividend rate  $\hat{\delta}$ , and volatility  $\hat{\sigma}$  ([21]), where

$$\hat{\sigma} = \sqrt{\left(\frac{1}{2}\right)^2 \sum_{i,j} \sigma_i \sigma_j \rho_{ij}} \quad \text{and} \quad \hat{\delta} = \frac{1}{2} \sum_i (\delta_i + \frac{1}{2} \sigma_i^2) - \frac{1}{2} \hat{\sigma}^2. \quad (75)$$

So, we can compare our results with the analytic option values. The Fourier cosine coefficients of the payoff function are given by

$$V_{k_1, k_2}(T) = \frac{2}{b_1 - a_1} \frac{2}{b_2 - a_2} \int_{a_2}^{b_2} \int_{a_1}^{b_1} (\sqrt{e^{y_1}} \sqrt{e^{y_2}} - K)^+ \cos\left(k_1 \pi \frac{y_1 - a_1}{b_1 - a_1}\right) \cos\left(k_2 \pi \frac{y_2 - a_2}{b_2 - a_2}\right) dy_1 dy_2.$$

An analytic solution is available and can be found using, for instance, Maple 14. We use parameter set I. Besides, we test the method for a deep out-of-the-money option with strike price  $K = 200$ . The option values are  $v(t_0, \mathbf{x}_0) = 8.8808$  ( $K = 100$ ) and  $v(t_0, \mathbf{x}_0) = 2.8 \times 10^{-3}$  ( $K = 200$ ). The results in Table 2 and Figure 4 are highly satisfactory and show exponential convergence. Convergence is reached in milliseconds. For both strike prices, the Fourier coefficients  $V_{k_1, k_2}$  are of the same order-of-magnitude. The last coefficients retained in the series expansion truncation determine the order-of-magnitude of the truncation error. Because of that, the error results are similar, however, the relative error is larger for the out-of-the-money option.

Table 2: Results geometric basket call (GBM).

$N_1 (= N_2)$	10	20	40	80
Error ( $K = 100$ )	-7.60e-1	-4.07e-2	-1.42e-5	2.34e-13
Error ( $K = 200$ )	-7.23e-1	-4.06e-2	-1.42e-5	3.08e-13
CPU (ms)	1.65	1.99	3.15	7.46

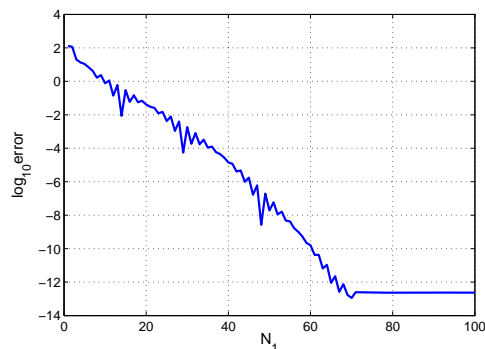


Figure 4: Error geometric basket call ( $K = 100$ ) (GBM).

In Section 6.1.2, we will price arithmetic basket options, for which the terminal coefficients  $V_{k_1, k_2}(T)$  need to be approximated by means of two-dimensional discrete Fourier cosine transforms, as explained in Section 3.2.1. Since there is an analytic solution available for the *geometric* basket options, we can analyze the error of the discretization approach for this option. For each spatial dimension we now take  $Q \geq N_1$  grid-points. So we replace the payoff coefficients by a discrete approximation.

Firstly, we analyze the approximate coefficients  $V_{k_1, k_2}^{DCT}(T)$  for the geometric basket option and calculate the maximum absolute error. This error converges quadratically in  $Q$ , see Figure 5. Secondly, we calculate the errors of the COS formula, see Section 5.1. The computational domain is chosen sufficiently large, so that we can neglect errors  $\epsilon_1$  and  $\epsilon_3$ , in (58) and (60), respectively. Results are shown in Table 3 and Figure 6. The left plot shows the series truncation error  $\epsilon_2$ , which indeed converges exponentially in  $N_1$ . The mid-plot confirms the algebraically converging error  $\epsilon_{DCT}$ , and the right plot shows the total error  $\epsilon$ . If  $N_1$  is chosen sufficiently large, then the error of using discrete Fourier transforms dominates the total error. The computation time of the coefficients is of order  $O(Q^2 \log_2 Q)$  and is the most time-consuming part. The CPU time is however still less than one second.

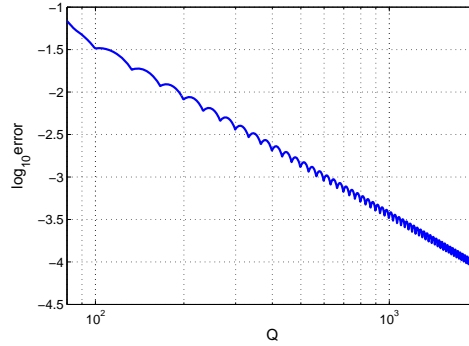


Figure 5: Max. abs. error  $V_{k_1, k_2}^{DCT}(T)$  ( $k_1, k_2 = 1, \dots, 80$ ).

Table 3: Results geometric basket call option with payoff coefficients approximated by DCTs.

(a) Abs.error					(b) CPU time (in sec.)				
Q	$N_1 (= N_2)$				Q	$N_1 (= N_2)$			
	10	20	40	80		10	20	40	80
250	1.09e+1	7.61e-1	4.18e-2	1.14e-3	250	0.007	0.007	0.008	0.014
500	7.59e-1	4.02e-2	3.91e-4	4.05e-4	500	0.04	0.04	0.04	0.06
1000	7.60e-1	4.06e-2	4.96e-5	6.38e-5	1000	0.17	0.17	0.17	0.20
2000	7.60e-1	4.07e-2	1.36e-5	5.75e-7	2000	0.71	0.71	0.73	0.75

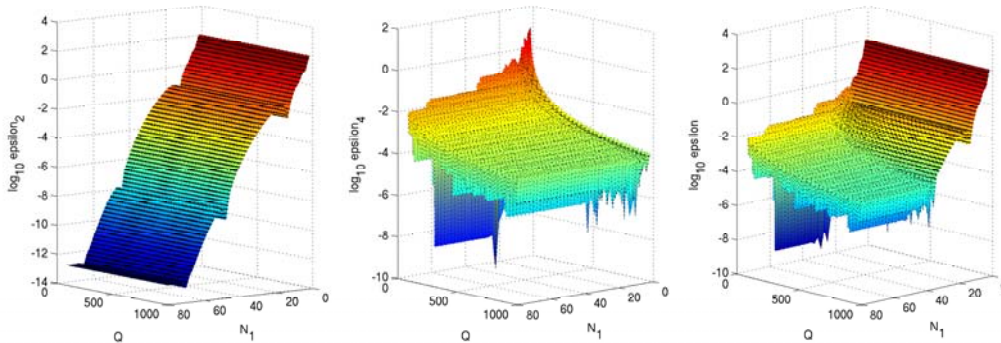


Figure 6: Error geometric basket call option, with payoff coefficients approximated by DCTs.

### 6.1.2 Arithmetic basket call option

For arithmetic basket options under geometric Brownian motion, there is no analytic solution to the option price. Instead, we use the reference option value  $v(t_0, \mathbf{x}_0) \approx 10.173230$ , obtained by using  $Q = 5000$ ,  $N_1 = N_2 = 100$  (Set I). This value is validated by a plain Monte Carlo simulation with  $10^6$  runs, which results in  $v(t_0, \mathbf{x}_0) \approx 10.1714$  with standard deviation 0.017.

The Fourier cosine coefficients of the payoff function are given by

$$V_{k_1, k_2}(T) = \frac{2}{b_1 - a_1} \frac{2}{b_2 - a_2} \int_{a_2}^{b_2} \int_{a_1}^{b_1} \left( \frac{1}{2} e^{y_1} + \frac{1}{2} e^{y_2} - K \right)^+ \cos \left( k_1 \pi \frac{y_1 - a_1}{b_1 - a_1} \right) \cos \left( k_2 \pi \frac{y_2 - a_2}{b_2 - a_2} \right) dy_1 dy_2. \quad (76)$$

No analytic representation is available and we approximate the coefficients by using discrete Fourier cosine transforms. The results in Table 4 are satisfactory and the error converges exponentially in  $N_1$ . The CPU times are the same as in Table 3 and convergence is achieved within two seconds.

Table 4: Abs.error arithmetic basket call option.

		$N_1 (= N_2)$			
		10	20	40	80
$Q$	250	2.97e0	1.85e-1	2.65e-3	2.70e-3
	500	2.98e0	1.91e-1	1.97e-5	2.97e-5
	1000	2.98e0	1.91e-1	4.82e-5	1.29e-6
	2000	2.98e0	1.91e-1	5.08e-5	1.28e-6

### 6.1.3 Call-on-max and put-on-min option

In this section, we discuss the following two-color rainbow options,

- call-on-maximum option, with payoff  $g(\mathbf{x}) = \left( \max(e^{x_1}, e^{x_2}) - K \right)^+$ ,
- put-on-minimum option, with payoff  $g(\mathbf{x}) = \left( K - \min(e^{x_1}, e^{x_2}) \right)^+$ .

The Fourier coefficients of the payoff functions can be calculated analytically. Next, we summarize the results. For the European call-on-max and put-on-min options under GBM we use parameters set III ([5]) and compare the results with the analytic solutions from [27]. Figure 7 shows the results and confirms again exponential convergence.

Next we consider a put-on-minimum option under a jump-diffusion process. Under jump-diffusion asset price processes, there is no analytic solution to the option values available. We use the model parameters from set II and set  $N_1 = N_2 = 125$ , for which machine precision is reached. The results in Table 5 are achieved in a few milliseconds and correspond well to the reference prices in [8].

## 6.2 Bermudan rainbow options

For pricing Bermudan options we need to determine rectangular continuation and early-exercise regions. For this, we divide the domain of the second dimension,  $[a_2, b_2]$ , into  $J$  subintervals,

$$[a_2, b_2] = [w_0, w_1] \cup [w_1, w_2] \dots [w_q, w_{q+1}] \dots [w_{J-1}, w_J]. \quad (77)$$

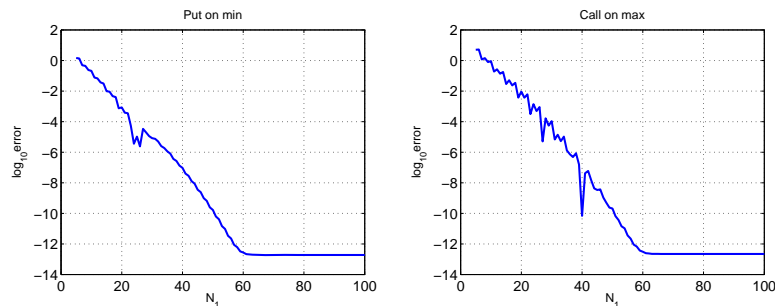


Figure 7: Error put-on-min and call-on-max option.

Table 5: Put-on-min option values  $\hat{v}(t_0, \mathbf{x}_0)$  (jump-diffusion).

$S_0^2 \backslash S_0^1$	90	100	110
90	15.6916	13.4073	12.1305
100	12.1918	9.1360	7.5175
110	10.3853	6.7274	4.8337

At the center of each subinterval, we determine the value(s)  $y^*$  for which the optimal exercise policy changes to the optimal continuation, in other words

$$g(y^*, \frac{1}{2}(w_q + w_{q+1})) = c(t_m, y^*, \frac{1}{2}(w_q + w_{q+1})). \quad (78)$$

For, for instance, a basket put option, we can then define an early-exercise region  $\mathcal{G}^q = [a_1, y^*] \times [w_q, w_{q+1}]$  and a continuation region  $\mathcal{C}^q = [y^*, b_1] \times [w_q, w_{q+1}]$ . So, the computational domain is divided into  $J$  early-exercise and continuation regions. In the case of a call-on-max option there is a multiply connected exercise region, see Figure 1, and for each subinterval  $[w_q, w_{q+1}]$  we have one or two early-exercise regions  $\mathcal{G}^q$ . As long as we select a technique by which we can distinguish multiple continuation and early-exercise regions from each other, this does not affect the complexity of the method.

**Remark 4** *We elaborate on the error in the determination of the early-exercise regions and the convergence in parameter  $J$ . We assume that  $N_1$  and  $N_2$  are chosen sufficiently large, so that the approximated option values are sufficiently accurate. Besides, we presume that a root-finding method is available which is able to accurately determine the value(s)  $y^*$ , see (78). For example, the Newton method with an accurate initial guess suffices. In this setting, the early-exercise regions converge, quadratically in  $J$ , to the true regions by means of the rectangle-rule-based method. However, if we would choose  $N_1, N_2$  too small, the COS formula may give inaccurate, oscillatory options values, resulting in inaccurate early-exercise regions and the convergence result only holds for sufficiently large  $N_1, N_2$ .*

For *equidistant* intervals  $[w_q, w_{q+1}]$  we take  $w_q = a_2 + (b_2 - a_2)q/J$ , which gives the exercise regions  $\mathcal{G}^q$  shown in Figure 8a. A *non-equidistant* grid, as in Figure 8b, is based on the quantile function (the inverse distribution function) of asset price  $X_T^2$ . We use  $w_0 = a_2, w_J = b_2$ , and  $w_q = F_{X_T^2}^{-1}(q/J)$ . The large rectangle at the bottom of the domain is not an accurate representation of the true region. However, it is located near the boundary of the computational domain, outside the domain of interest, and therefore does not affect the option values in the middle of the domain. The usage of non-equidistant grids typically leads to more efficient pricing.

**Remark 5** *If the quantile function is not known analytically, then we can approximate the random variable by a normally distributed variable with the same mean (first cumulant) and variance (second cumulant), for which the quantile function is known.*

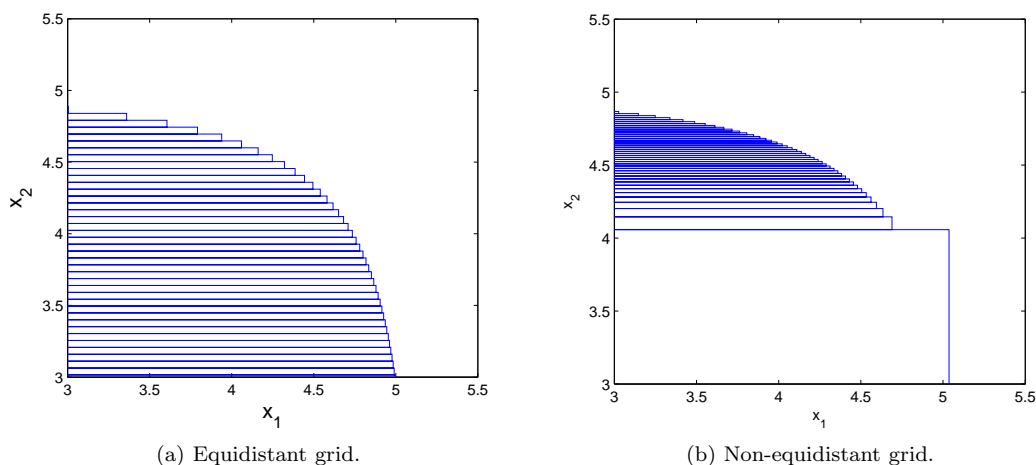


Figure 8: Rectangular early-exercise regions  $\mathcal{G}^q$  (arithmetic basket put).

### 6.2.1 Geometric basket put option

We start by pricing Bermudan-style geometric basket put options under geometric Brownian motion. For the tests we use parameter Set I and take  $\mathcal{M} = 10$  early-exercise dates. The reference solution to the put option,  $v(t_0, \mathbf{x}) = 6.95904$ , equals a Bermudan put option in 1D ([3]) and is computed with the 1D-COS method. The terms  $G_{k_1, k_2}(\mathcal{G}^p)$  admit an analytic solution. The results are presented in Table 6. Convergence is exponentially in  $N_1$  and algebraically in  $J$ , with order two, see Section 5.1. The use of non-equidistant grids improves the convergence significantly. The computation time is linear in  $J$  and the results for, for example,  $N_1 = 80$  and  $J = 160$  are obtained in about 30 seconds.

Table 6: Abs.error Bermudan geometric put basket option.

(a) Equidistant grid.				(b) Non-equidistant grid.					
$N_1 (= N_2)$				$N_1 (= N_2)$					
40      80      160				40      80      160					
	20	6.65e-2	7.77e-2	7.79e-2		20	7.07e-4	2.23e-3	2.20e-3
	40	1.22e-2	1.56e-2	1.56e-2		40	2.63e-3	5.13e-4	4.83e-4
$J$	80	1.69e-4	3.54e-3	3.51e-3	$J$	80	2.97e-3	1.46e-4	1.16e-4
	160	2.51e-3	8.86e-4	8.56e-4		160	3.17e-3	5.85e-5	2.88e-5
	320	3.15e-3	2.42e-4	2.13e-4		320	3.18e-3	3.70e-5	7.17e-6

In Section 6.2.2, we will price Bermudan arithmetic basket options, for which the coefficients  $V_{k_1, k_2}(T)$  are approximated by means of two-dimensional discrete Fourier cosine transforms and the terms  $G_{k_1, k_2}$  are approximated by  $\hat{G}_{k_1, k_2}$ , as explained in Section 3.2.2 (with equation (37)). We examine here the error of this approach by using the Bermudan *geometric* basket, for which an accurate reference price is available. We take  $Q = 2000$ . Table 7 shows satisfactory results.



The results are similar to Table 6, with analytic terms  $G_{k_1, k_2}$ . However, for higher values of  $N_1$  the usage of analytic terms is more accurate.

Table 7: Abs.error Bermudan geometric put basket option, with  $\hat{G}_{k_1, k_2}$  ( $Q = 2000$ ).

(a) Equidistant grid.				(b) Non-equidistant grid.					
		$N_1 (= N_2)$					$N_1 (= N_2)$		
		40	80	160			40	80	160
	20	7.05e-02	7.78e-02	7.79e-02		20	4.55e-03	2.45e-03	2.25e-03
	40	1.70e-02	1.58e-02	1.56e-02		40	2.69e-03	7.39e-04	5.54e-04
$J$	80	5.56e-03	3.69e-03	3.54e-03	$J$	80	2.37e-03	3.74e-04	1.92e-04
	160	2.96e-03	1.09e-03	9.13e-04		160	2.23e-03	2.87e-04	1.06e-04
	320	2.33e-03	4.66e-04	2.86e-04		320	2.21e-03	2.66e-04	8.52e-05

### 6.2.2 Arithmetic basket put option

For the arithmetic basket option, we take the same model parameters (Set I) as in the previous section, with  $\mathcal{M} = 10$  early-exercise dates. We approximate the Fourier coefficients at the terminal time by using the discrete Fourier cosine transform and the coefficients  $G_{k_1, k_2}$  are approximated by  $\hat{G}_{k_1, k_2}$ . The results in Table 8 show converging option values.

Table 8: Option values Bermudan arithmetic put basket option, with  $\hat{G}_{k_1, k_2}$  ( $Q = 2000$ ).

		Non-equidistant grid		
		$N_1 (= N_2)$		
		40	80	160
	20	6.6086	6.6067	6.6077
	40	6.6102	6.6091	6.6102
$J$	80	6.6106	6.6096	6.6108
	160	6.6109	6.6097	6.6109
	320	6.6108	6.6097	6.6109

### 6.2.3 Call-on-max and put-on-min option

Here, we consider Bermudan call-on-maximum and put-on-minimum options. They are discussed in, among others, [2], [8], [16], and [15]. The authors in [7] prove properties of the continuation and early-exercise regions for the American-style option.

The convergence results for a call-on-max option under GBM, Set V with  $\mathcal{M} = 9$ , are presented in Table 9a. In Table 9b, option prices for different initial asset values are shown. They correspond to the values in the third and fourth column, which are obtained in [2].

At last, we price put-on-minimum options under a jump-diffusion process (parameters set II). The results in Table 10a show converging values. This verifies the applicability of the 2D-COS method to the broad class of Lévy processes. With increasing the number of early-exercise dates,  $\mathcal{M}$ , the option prices converge to the American prices in [8] (Table 10b).

**Remark 6** *Other rainbow options that can be priced easily and efficiently with the 2D-COS formula are, among others, double binary (digital) cash-or-nothing options, spread options, two asset correlation options, and exchange options.*

Table 9: Bermudan call-on-max option values  $\hat{v}(t_0, \mathbf{x})$ .

(a) Convergence in $N_1$ and $J$ .				(b) Option prices for different initial values ( $S_0^2 = S_0^1$ ) ( $N_1 = 160, J = 320$ ).				
	$J$	$N_1 (= N_2)$			$S_0^1$	2D-COS	95% CI ([2])	Binomial ([2])
		40	80	160				
	40	14.2946	13.8919	13.8921	90	8.0727	[8.053, 8.082]	8.075
	80	14.3032	13.8993	13.8994	100	13.9017	[13.892, 13.934]	13.902
	160	14.3025	13.9011	13.9013	110	21.3437	[21.316, 21.359]	21.345
	320	14.3029	13.9015	13.9017				
	640	14.3037	13.9016	13.9018				

Table 10: Bermudan put-on-min option values  $\hat{v}(t_0, \mathbf{x})$  (jump-diffusion).

(a) Convergence in $N_1$ and $J$ ( $\mathcal{M} = 9$ ).					(b) Convergence in $\mathcal{M}$ ( $J = 320, N_1 = 160$ ).	
	$J$	$N_1 (= N_2)$			$\mathcal{M}$	
		40	80	160		
	40	9.5495	9.5570	9.5586	2	9.3577
	80	9.5597	9.5580	9.5597	4	9.4863
	160	9.5606	9.5582	9.5599	8	9.5526
	320	9.5621	9.5584	9.5600	16	9.5862
	640	9.5632	9.5584	9.5600	32	9.6033

## 7 Numerical experiments Heston model

In this section, we test the performance of the 2D-COS method for pricing Bermudan put options under the Heston dynamics. We use three different parameter sets:

Set A :  $\rho = 0.1, \nu_0 = 0.0625, \bar{\nu} = 0.16, S_0 = 10, K = 10, r = 0.1, \eta = 0.9, \kappa = 5, T = 0.25$ ,

Set B :  $\rho = -0.64, \nu_0 = 0.0348, \bar{\nu} = 0.0348, S_0 = 100, K = 100, r = 0.04, \eta = 0.9, \kappa = 1.15, T = 0.25$ ,

Set C :  $\rho = -0.9, \nu_0 = 0.04, \bar{\nu} = 0.04, S_0 = 100, K = 100, r = 0.04, \eta = 0.5, \kappa = 0.5, T = 1$ .

The Feller condition satisfied is equivalent to  $2\kappa\bar{\nu}/\eta^2 - 1 := q_F \geq 0$ . For Set A we have  $q_F = 0.98$  and the variance process remains strictly positive. For sets B and C we have  $q_F = -0.47$  and  $q_F = -0.84$ , respectively, and the Feller condition is not satisfied.

Figure 9 shows the approximated bivariate density function  $\hat{f}(X_T, \nu_T | x_0, \nu_0)$ , as obtained by density recovery, equation (74), for set A.

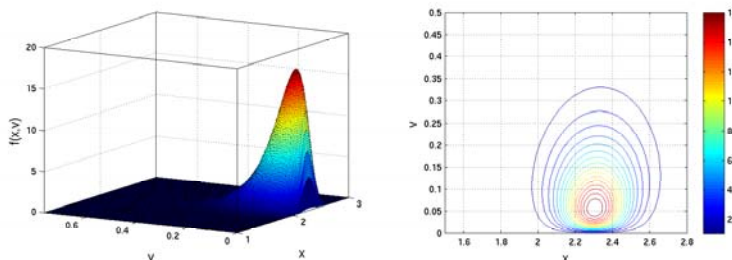


Figure 9: Bivariate density function (Set A).

We find

$$|F_{k_1, k_2}^\pm(\mathbf{x})| \leq \frac{2}{b_1 - a_1} \frac{2}{b_2 - a_2} \left| \varphi \left( \frac{k_1 \pi}{b_1 - a_1}, \pm \frac{k_2 \pi}{b_2 - a_2} \middle| \mathbf{x} \right) \right| \sim O \left( k_2^{-(q_F + 1)} \right), \quad (79)$$

based on the univariate characteristic function of the variance process. The coefficients  $F_{k_1, k_2}$  decrease exponentially in  $k_1$  and algebraically in  $k_2$  with order  $q_F + 1$ . For set C this yields convergence order 0.16 which will result in very slow convergence of the @D COS method in this case.

## 7.1 European options with Bermudan framework

To test the algorithm for pricing under the Heston stochastic volatility model, we calculate *European option prices with the Bermudan framework*, with  $\mathcal{M} = 12$  time steps. At every time step  $t_m$  we take only a continuation region,  $C^1 = [a_1, b_1] \times [a_2, b_2]$ , which then corresponds to a European option and no early-exercise opportunities.

Reference prices are obtained by the 1D-COS method for European options ([11]). The error results are shown in Table 11. The error converges exponentially in  $N_1$  and algebraically in  $N_2$ , as expected. Sets A and B give very accurate prices. However, the convergence for Set C is relatively slow and the obtained values are not highly accurate. The reason for this is the peaked density function, with coefficients  $F_{k_1, k_2}$  only slowly decreasing in  $k_2$ .

Table 11: Error European option prices.

(a) Set A,  $v(t_0, \mathbf{x}) = 0.5015$  ( $q_F = 0.98$ ).  $-0.47$ . (b) Set B,  $v(t_0, \mathbf{x}) = 3.1325$  ( $q_F =$

$N_1$	$N_2$			$N_1$	$N_2$		
	50	100	200		50	100	200
50	-1.01e-4	-1.03e-5	1.02e-6	50	3.83e-4	2.05e-4	8.94e-5
100	-1.07e-4	-1.63e-5	-4.90e-6	100	3.75e-4	1.95e-4	7.90e-5
200	-1.07e-4	-1.62e-5	-4.86e-6	200	3.75e-4	1.95e-4	7.90e-5

(c) Set C,  $v(t_0, \mathbf{x}) = 6.2711$  ( $q_F = -0.84$ ).

$N_1$	$N_2$		
	250	500	1000
50	7.8787e-02	4.3008e-02	2.6131e-02
100	6.8267e-02	3.1137e-02	1.3630e-02
200	6.7979e-02	3.0760e-02	1.3220e-02

## 7.2 Bermudan put options under Heston dynamics

In this section, we consider Bermudan put options with  $\mathcal{M} = 10$  early-exercise dates. For the first test, we select  $J = 2^7$  continuation and early-exercise regions. The results in Table 12 show convergence in  $N_1$  and  $N_2$  and the prices for sets A and B match the results in [13] very well. Convergence for set C, with  $q_F = -0.84$ , is somewhat slow because of the slowly decreasing Fourier coefficients in this case. The quadrature-COS method ([13]) gives the reference value 5.3982.

In a second test we vary the number of continuation and exercise regions. The convergence is quadratic in  $J$ , see Table 13.

The computation times are significantly longer than for the Bermudan rainbow options under Lévy processes in Section 6.2, because efficient matrix-vector multiplication with the FFT algorithm is

applicable in only one direction, see Section 4. However, accuracy of four digits can still be obtained within seconds for set A and set B.

Table 12: Results Bermudan put option ( $J = 2^7$ ).

(a) Option values, Set A ( $q_F = 0.98$ ).					(b) Option values, Set B ( $q_F = -0.47$ ).				
$N_1$	$N_2$				$N_1$	$N_2$			
	40	60	80	100		40	60	80	100
40	0.517765	0.517869	0.517894	0.517903	40	3.20083	3.20077	3.20071	3.20066
60	0.517176	0.517285	0.517311	0.517320	60	3.19909	3.19903	3.19897	3.19893
80	0.517021	0.517130	0.517156	0.517165	80	3.19912	3.19907	3.19901	3.19897
100	0.517008	0.517116	0.517143	0.517152	100	3.19910	3.19905	3.19899	3.19894

(c) Option values, Set C ( $q_F = -0.84$ ).					(d) CPU time (s).				
$N_1$	$N_2$				$N_1$	$N_2$			
	40	60	80	100		40	60	80	100
40	5.70464	5.57416	5.50068	5.45386	40	6.7	10.5	14.8	22.8
60	5.77720	5.67101	5.61174	5.57399	60	8.8	14.0	20.2	31.2
80	5.75689	5.64811	5.58898	5.55195	80	11.2	17.9	25.8	42.9
100	5.75808	5.64673	5.58552	5.54697	100	13.5	21.4	31.2	51.2

Table 13: Bermudan option values  $\hat{v}(t_0, \mathbf{x})$ , convergence in  $J$  ( $N_1 = 100$ ,  $N_2 = 100$ ).

$\log_2 J$	3	4	5	6	7
Set A	0.515991	0.516784	0.517060	0.517133	0.517152
Set B	3.186917	3.196282	3.198380	3.198837	3.198944
Set C	5.42431	5.51038	5.53874	5.54523	5.54697
CPU time (s)	3.4	6.3	12.4	25.1	51.2

## 8 Higher-dimensional COS method

The 2D-COS formula can be generalized to higher dimensions easily. In this section we elaborate on the possibilities and difficulties of the higher-dimensional COS method. In Section 8.1, we demonstrate an example with three underlying assets.

Suppose we have an  $n$ -dimensional asset prices process, with  $N_1 = N_2 = \dots = N_n = N$  terms in the series summations. Then, with the methodology described in Section 3, the computational complexity of pricing a Bermudan option is  $O(2^n N^n \log_2 N)$ . It is clear that we cannot choose  $n$  too large, as then the curse of dimensionality sets in. Besides, the data storage grows exponentially in  $n$ . For example, in seven dimensions, with  $N = 100$ , the storage of the Fourier coefficients  $V(t_m)$  requires  $8 \cdot 10^5$  GB memory. Dimensions  $n \in \{2, \dots, 5\}$  should however still be fine. For higher dimensions, efficient aggregation of states or sparse grid methods are needed. However, most existing methods for a dimensionality of ten and higher are based on Monte Carlo methods.

For pricing Bermudan options we need to determine  $n$ -dimensional continuation and early-exercise boxes. In three-dimensional space, we get rectangular cuboids. For four or higher dimensions it gets harder to visualize the regions. Special algorithms to determine the regions efficiently, or even adaptively, may be helpful.

The terminal coefficients  $V_{k_1, k_2, \dots, k_n}(T)$  can be approximated by using discrete Fourier cosine transforms, see Section 3.2.2. Computation or approximation of the multi-D coefficients  $G_{k_1, k_2, \dots, k_n}(\mathcal{G}^p)$

can be time-consuming or no analytic expression may be available, but we can use the terminal coefficients to approximate them. This approach was explained in Section 3.2.2.

## 8.1 3D-COS formula

The three-dimensional version of the COS formula reads

$$\hat{v}(t_0, \mathbf{x}) := \prod_{i=1}^3 \frac{b_i - a_i}{2} e^{-r\Delta t} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} \sum_{k_3=0}^{N_3-1} \frac{1}{4} \left[ F_{k_1, k_2, k_3}^{++}(\mathbf{x}) + F_{k_1, k_2, k_3}^{+-}(\mathbf{x}) + F_{k_1, k_2, k_3}^{-+}(\mathbf{x}) + F_{k_1, k_2, k_3}^{--}(\mathbf{x}) \right] V_{k_1, k_2, k_3}(T). \quad (80)$$

The coefficients  $F_{k_1, k_2, k_3}^{\pm\pm\pm}$  are given by

$$\begin{aligned} F_{k_1, k_2, k_3}^{\pm\pm\pm}(\mathbf{x}) &= \prod_{i=1}^3 \frac{2}{b_i - a_i} \operatorname{Re} \left( \varphi \left( \frac{k_1 \pi}{b_1 - a_1}, \pm \frac{k_2 \pi}{b_2 - a_2}, \pm \frac{k_3 \pi}{b_3 - a_3} \mid \mathbf{x} \right) \exp \left( -ik_1 \pi \frac{a_1}{b_1 - a_1} \mp ik_2 \pi \frac{a_2}{b_2 - a_2} \mp ik_3 \pi \frac{a_3}{b_3 - a_3} \right) \right) \\ &= \prod_{i=1}^3 \frac{2}{b_i - a_i} \operatorname{Re} \left( \varphi_{\text{levy}} \left( \frac{k_1 \pi}{b_1 - a_1}, \pm \frac{k_2 \pi}{b_2 - a_2}, \pm \frac{k_3 \pi}{b_3 - a_3} \right) \exp \left( ik_1 \pi \frac{x_1 - a_1}{b_1 - a_1} \pm ik_2 \pi \frac{x_2 - a_2}{b_2 - a_2} \pm ik_3 \pi \frac{x_3 - a_3}{b_3 - a_3} \right) \right), \end{aligned} \quad (81)$$

where the last equality holds for Lévy processes, with  $\varphi_{\text{levy}}(u_1, u_2, u_3) := \varphi(u_1, u_2, u_3 | 0, 0, 0)$ .

We test the 3D-COS formula by pricing European basket call options, with parameter set IV. The results in [20] are used as reference prices. For the coefficients  $V_{k_1, k_2, k_3}(T)$  we use discrete Fourier cosine transformations to approximate them. The results are presented in Table 14. They show that the multi-dimensional-COS formula is very accurate. The CPU time for  $N_1 = 40$  and  $Q = 200$  is less than two seconds.

Table 14: Abs.error geometric basket call option,  $v(t_0, \mathbf{x}_0) = 11.9791$ , and arithmetic basket call option,  $v(t_0, \mathbf{x}_0) = 13.2449$  ( $N_1 = N_2 = N_3$ ).

Q \ N <sub>1</sub>	Geometric basket				Arithmetic basket			
	10	20	40	80	10	20	40	80
50	1.78	3.12e-2	4.03e-3	n/a	2.11e+1	4.59e-1	6.94e-3	n/a
100	1.79	4.53e-2	7.92e-3	7.91e-3	2.14e+1	4.80e-1	9.82e-5	7.82e-5
200	1.81	3.68e-2	9.88e-4	9.89e-4	2.15e+1	4.86e-1	<5.0e-5	2.13e-5
400	1.81	3.77e-2	2.46e-4	2.47e-4	2.15e+1	4.87e-1	<5.0e-5	<5.0e-5

## 9 Conclusion

The one-dimensional COS method for pricing European and Bermudan options was developed in [11] and [12]. The method is based on Fourier cosine expansions and the series coefficients of the value function are recovered backwards in time. In this paper we presented the generalisation of the COS method to higher dimensions. The recursive algorithm can be applied to, for example, pricing rainbow options, but also to pricing under the Heston stochastic volatility model.

The 2D-COS method for valuation of Bermudan options is based on the dynamic programming principle and two-dimensional Fourier cosine series expansions. For smooth density functions, the resulting method converges exponentially in  $N_1$  and  $N_2$ , the number of terms in the series summations. Otherwise we achieve algebraic convergence, as in the 1D case. For multi-dimensional stochastic processes in the class of Lévy processes, we can apply efficient matrix-vector multiplication using the FFT algorithm. The Heston model is not in the Lévy class and the FFT algorithm will be applicable in only one dimension.

We performed extensive numerical experiments with European and Bermudan options. The algorithm for pricing two-color rainbow options can be used for various payoffs and performs highly satisfactorily. The asset prices in the tests are modelled by geometric Brownian motions or correlated jump-diffusions, but the method can be applied to a broad class of multi-dimensional asset price processes for which the characteristic function is available.

The density function of the Heston dynamics may be non-smooth, especially near zero variance if the Feller condition is not satisfied, and peaked densities give rise to slow algebraic convergence. However, for a wide range of parameter values, the 2D-COS method achieves accurate Bermudan put prices.

**Acknowledgment** The authors wish to thank Prof.dr. F.H.J. Redig and dr. J.A.M. van der Weide for helpful suggestions and discussions.

## Appendix

### A Fast Fourier Transform (FFT) algorithm

**Theorem A.1** (Efficient computation of  $\hat{C}$  and  $\mathcal{A}^q$ )

The matrix-vector product  $M^+(u_1, u_2, a, b)\mathbf{v}$  and  $M^-(u_1, u_2, a, b)\mathbf{v}$ , can be computed in  $O(N \log_2 N)$  operations with the help of the Fast Fourier Transform (FFT) algorithm, with  $N$  the size of square-matrices  $M^+$  and  $M^-$ .

The key insight of this efficient computation are the equalities

$$M_{k,j}^+(z_1, z_2, a, b) = -\frac{i}{\pi} \left( M_{k,j}^{+c}(z_1, z_2, a, b) + M_{k,j}^{+s}(z_1, z_2, a, b) \right), \quad (82)$$

$$M_{k,j}^-(z_1, z_2, a, b) = -\frac{i}{\pi} \left( M_{k,j}^{-c}(z_1, z_2, a, b) + M_{k,j}^{-s}(z_1, z_2, a, b) \right), \quad (83)$$

where

$$M_{k,j}^{\pm c}(z_1, z_2, a, b) = \frac{\exp\left(i(\pm j + k)\frac{(z_2 - a)\pi}{b - a}\right) - \exp\left(i(\pm j + k)\frac{(z_1 - a)\pi}{b - a}\right)}{\pm j + k}, \quad (84)$$

$$M_{k,j}^{\pm s}(z_1, z_2, a, b) = \frac{\exp\left(i(\pm j - k)\frac{(z_2 - a)\pi}{b - a}\right) - \exp\left(i(\pm j - k)\frac{(z_1 - a)\pi}{b - a}\right)}{\pm j - k}, \quad (85)$$

with special cases

$$M_{k,j}^{+c}(z_1, z_2, a, b) = \frac{(z_2 - z_1)\pi i}{b - a}, \quad \text{for } k = j = 0, \quad (86)$$

$$M_{k,j}^{-s}(z_1, z_2, a, b) = \frac{(z_2 - z_1)\pi i}{b - a}, \quad \text{for } k = j = 0, \quad (87)$$

$$M_{k,j}^{-c}(z_1, z_2, a, b) = \frac{(z_2 - z_1)\pi i}{b - a}, \quad \text{for } k = j, \quad (88)$$

$$M_{k,j}^{+s}(z_1, z_2, a, b) = \frac{(z_2 - z_1)\pi i}{b - a}, \quad \text{for } k = j. \quad (89)$$

The matrices  $M^{+c}$  and  $M^{-s}$  are Hankel matrices and  $M^{+s}$  and  $M^{-c}$  are Toeplitz matrices. These special structures make the calculation of the matrix-vector products efficient using FFT algorithms, as described in [12] and [4].

## References

- [1] H. Albrecher, P. Mayer, W. Schoutens, J. Tistaert, The Little Heston Trap, *Wilmott Magazine*, January Issue, 83-92, 2007.
- [2] L. Andersen, M. Broadie, Primal-Dual Simulation Algorithm for Pricing Multidimensional American Options, *Management Science*, Vol. 50(9), 1222-1234, 2004.
- [3] S. J. Berridge, J. M. Schumacher, Pricing High-Dimensional American Options Using Local Consistency Conditions. In: J. Miller, D. Edelman, J. Appleby (eds.), *Numerical Methods for Finance*, 13-52, Chapman & Hall/CRC, London, 2007.
- [4] J. P. Boyd, *Chebyshev and Fourier spectral methods*, Courier Dover Publications, 2001.
- [5] P. P. Boyle, A Lattice Framework for Option Pricing with Two State Variables, *Journal of Financial and Quantitative Analysis*, Vol. 23(1), 1-12, 1988.
- [6] P. P. Boyle, J. Evnine, S. Gibbs, Numerical Evaluation of Multivariate Contingent Claims, *Review of Financial Studies*, Vol. 2(2), 241-250, 1989.
- [7] M. Broadie, J. Detemple, The Valuation of American Options on Multiple Assets, *Mathematical Finance*, Vol. 7(3), 241-286, 1997.
- [8] S. S. Clift, P. A. Forsyth, Numerical solution of two asset jump diffusion models for options valuation, *Applied Numerical Mathematics*, Vol. 58(6), 743-782, 2008.
- [9] D. Duffie, J. Pan, K. J. Singleton, Transform analysis and asset pricing for affine jump-diffusions, *Econometrica*, Vol. 68(6), 1343-1376, 2000.
- [10] D. Dufresne, Bessel processes and Asian options, in Numerical Methods in Finance. In: H. Ben-Ameur and M. Breton (eds.), *Numerical Methods in Finance*, 35-57, Springer-Verlag, New York, 2005.
- [11] F. Fang, C. W. Oosterlee, A novel pricing method for European options based on Fourier-cosine series expansions, *SIAM Journal on Scientific Comp*, Vol. 31(2), 826-848, 2008.
- [12] F. Fang, C. W. Oosterlee, Pricing early-exercise and discrete barrier options by Fourier-cosine series expansions, *Numerische Mathematik*, Vol. 114(1), 27-62, 2009.
- [13] F. Fang, C. W. Oosterlee, A Fourier-Based Valuation Method for Bermudan and Barrier Options under Heston's Model, *SIAM Journal on Financial Mathematics*, Vol. 2, 439-463, 2011.
- [14] S. L. Heston, A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options, *Rev. Financ. Stud.* Vol. 6(2), 327-343. 1993.
- [15] S.-F. Huang, M. Guo, Valuation of Multidimensional Bermudan Options. In: H. K. Wolfgang, N. Hautsch, L. Overbeck (eds.), *Applied Quantitative Finance*, 295-30, Springer, Berlin Heidelberg, 2008.
- [16] S. Jain, C. W. Oosterlee, Pricing higher-dimensional American options using the stochastic grid method, 2010. Available at SSRN: <http://ssrn.com/abstract=1723712>.

- [17] A. Janek, T. Kluge, R. Weron, and U. Wystup, FX smile in the Heston model. In: P. Čížek, W. K. Härdle, R. Weron (eds.), *Statistical Tools for Finance and Insurance*, 133-162, Springer Verlag, 2011.
- [18] C. Kahl, P. Jäckel, Not-so-complex logarithms in the Heston model, *Wilmott Magazine*, 94-103, 2005.
- [19] M. Jeanblanc, M. Yor, and M. Chesney, *Mathematical Methods for Financial Markets*, Springer Finance, 2009.
- [20] C. C. W. Leentvaar, C. W. Oosterlee, On coordinate transformation and grid stretching for sparse grid pricing of basket options, *Journal of Computational and Applied Mathematics*, Vol. 222(1), 193-209, 2008.
- [21] C. C. W. Leentvaar, C. W. Oosterlee, Multi-asset option pricing using a parallel Fourier-based technique, *Journal of Computational Finance*, Vol. 12(1), 1-26, 2008.
- [22] R. Lord, C. Kahl, Why the rotation count algorithm works, Tinbergen Institute Discussion Paper No. 2006-065/2, 2006.
- [23] R. Lord, C. Kahl, Complex logarithms in Heston-like models, *Mathematical Finance*, Vol. 20(4), 671-694, 2010.
- [24] S. Pagliarani, A. Pascucci, C. Riga, Expansion formulae for local Lévy models, 2011. Available at SSRN: <http://ssrn.com/abstract=1937149>.
- [25] M.J. Ruijter, C.W. Oosterlee, Two-Dimensional Fourier Cosine Series Expansion Method for Pricing Financial Options, *SIAM Journal on Scientific Computing*, Vol. 34(5), B642-B671, 2012.
- [26] T. T. Soong, *Random Differential Equations in Science and Engineering*, Academic Press, 1973.
- [27] R. M. Stulz, Options on the minimum or the maximum of two risky assets: Analysis and applications, *Journal of Financial Economics*, Vol. 10(2), 161-185, 1982.
- [28] B. Zhang, L. A. Grzelak, C. W. Oosterlee, Efficient Pricing of Commodity Options with Early-Exercise under the Ornstein-Uhlenbeck process, *Applied Numerical Mathematics*, Vol. 62(2), 2012.
- [29] S. Kehtari, ETH Zürich, private communication.







Publisher:

CPB Netherlands Bureau for Economic Policy Analysis  
P.O. Box 80510 | 2508 GM The Hague  
T (070) 3383 380

November 2012 | ISBN 978-90-5833-572-2