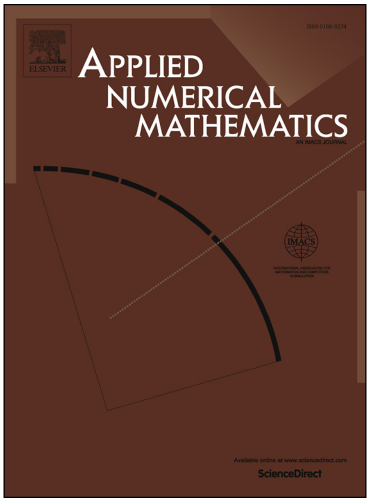

Two-dimensional Shannon wavelet inverse Fourier technique for pricing European options

G. Colldeforns-Papiol, L. Ortiz-Gracia, C.W. Oosterlee

PII: S0168-9274(17)30062-4
DOI: http://dx.doi.org/10.1016/j.apnum.2017.03.002
Reference: APNUM 3179

To appear in: *Applied Numerical Mathematics*

Received date: 3 June 2016
Revised date: 7 January 2017
Accepted date: 9 March 2017

Please cite this article in press as: G. Colldeforns-Papiol et al., Two-dimensional Shannon wavelet inverse Fourier technique for pricing European options, *Appl. Numer. Math.* (2017), http://dx.doi.org/10.1016/j.apnum.2017.03.002

Two-dimensional Shannon wavelet inverse Fourier technique for pricing European options

G. Colldeforns-Papiol L. Ortiz-Gracia C. W. Oosterlee

Abstract

The SWIFT method for pricing European-style options on one underlying asset was recently published and presented as an accurate, robust and highly efficient technique. The purpose of this paper is to extend the method to higher dimensions by pricing exotic option contracts, called rainbow options, whose payoff depends on multiple assets. The multidimensional extension inherits the properties of the one-dimensional method, being the exponential convergence one of them. Thanks to the nature of local Shannon wavelets basis, we do not need to rely on a-priori truncation of the integration range, we have an error bound estimate and we use fast Fourier transform (FFT) algorithms to speed up computations. We test the method for similar examples with state-of-the-art methods found in the literature, and we compare our results with analytical expressions when available.

Key words. Option pricing, European options, two-color rainbow options, basket options, spread options, Lévy process, Shannon wavelets, cardinal sine function, Fourier transform inversion

AMS subject classifications. 62P05, 60E10, 65T60

1 Introduction

Financial derivatives such as options are traded all over the world. In the general class of exotic options that are not listed on regulated exchanges, multi-asset options form a class for which efficient solution methods are not easily obtained.

Analytical formulae to price multi-asset derivatives are only available for the most simple cases. Hence, there is a need to develop numerical methods to approximate their prices and develop efficient algorithms to implement them, so that they provide useful information in a market that changes rapidly. Recently, multidimensional option pricing has become an important topic, but this is an area with high computational demands. Some examples of multidimensional options are exotic option contracts called multicolour rainbow options whose payoff depends on multiple assets.

One of the most commonly used methods for pricing options is Monte Carlo simulation. This method has the advantage of scaling linearly with the number of dimensions. However, convergence is slow and a large number of simulations is needed if accurate results are desired. Different approaches are based on partial differential equations (PDE) and Fourier methods. This last class of approximations relies on a transformation to the Fourier domain. The probability density function appears in the time domain, and it is not known for many relevant multidimensional processes. However, its characteristic function, this is its Fourier transform, is often available in closed form. Nevertheless, in both PDE and Fourier-based methods the curse of dimensionality plays a prominent role. The curse of dimensionality is the exponential growth of the complexity of the problem when the dimension increases, and modern computer systems cannot handle this huge amount of computations. For this reason, and despite their drawbacks, Monte Carlo methods are the most commonly used alternatives when the dimension is bigger than four or five, depending on the specific product.

One well-developed multidimensional Fourier-based method is the multidimensional COS method presented in [11] and called 2D-COS when the dimension is two. However, it may exhibit problems in the vicinity of the integration boundaries because of the periodic behaviour of cosines. This problem becomes evident for long maturity options, where round-off errors may accumulate near domain boundaries. Also for short maturity options, typically governed by a highly peaked density function, many cosine terms may be needed for an accurate representation. In addition, an accurate integration interval is important to capture the whole mass of the recovered density, but the choice of the interval is entirely based on the cumulants, which sometimes are not easily available or do not provide a good truncation range.

In the one-dimensional case, local wavelets bases have been considered in [9, 10], which overcome some of the problems of the well known one-dimensional COS method [2]. Wavelets give flexibility and enhance robustness when pricing long maturity options and heavy tailed asset processes. In particular, Shannon wavelets used in the SWIFT method [10], are smooth wavelets based on the cardinal sine (sinc in short) function. The SWIFT method has been used for European option pricing and it is based on a wavelet expansion of the underlying density function recovered from its Fourier transform. Haar wavelets and B-splines of order one were used in [9]. These wavelets have compact support and the pricing formula is particularly easy to implement with the Haar basis, although the method converges slower than SWIFT. Higher order B-splines are considered in [3].

The aim of the present work is to extend the one-dimensional SWIFT method to a higher dimension to be able to price European-style financial contracts with a payoff depending on more than one asset. For two-dimensional contracts we call it 2D-SWIFT. The unknown density function is approximated in terms of a finite combination of multidimensional Shannon scaling functions and the coefficients of the approximation are recovered by inverting its Fourier transform. Then, the payoff coefficients are calculated by means of the approximated density within the discounted expected payoff pricing formula setting, and the final price of the contract is readily obtained. Central to this two-step process is the use of an appropriate approximation of the sinc function. As opposed to the one-dimensional case, where the Vieta's formula was employed, we approximate the sinc function in terms of a sum of complex exponentials. Proceeding this way, the algebraic manipulation to obtain the pricing formulae is drastically simplified. Further, we provide an error analysis which facilitates the choice of the parameters of the new method and enhance the overall speed with an FFT algorithm. We test our method with examples and methods from the literature, like for instance the well-known 2D-COS in [11]. We price basket (both geometric and arithmetic), spread, call-on-max, put-on-min, and correlation options. It is worth mentioning that we can perform a consistency check of our method, since there exists a closed form solution for spread options with strike zero (the Margrabe formula [6]), as well as an analytical method for the valuation of a geometric basket option (since it can be transformed into a one-dimensional European option). The 2D-SWIFT method appears to be a very competitive pricing machinery, showing exponential convergence with a very short run time. We also benefit from the local behaviour of Shannon wavelets. Due to that fact, the accurate treatment of options with long maturities is possible since we can remove the density coefficients affected by the exponential growth of the payoff without changing the remaining part of the approximation. Finally, it is not necessary to rely on a-priori truncation of the integration range. We use an initial guess of the truncation range, which allows to compute the density coefficients much faster with an FFT algorithm, and we adaptively compute the final integration range if necessary. It is worth remarking that the number of terms needed in the expansion is automatically calculated once the scale of approximation has been fixed, which is a central step in our method.

The outline of this paper is as follows. We start with the presentation of the two-dimensional option pricing problem in Section 2 and the related theory of the multidimensional wavelets framework. In Section 3 the 2D-SWIFT pricing formula is derived and the respective extension to higher dimensions is also introduced. In Section 4 we present an error analysis along with a study on how to select the parameters of the method. Numerical tests are performed in Section 5 and a specific study of 2D-SWIFT strengths is carried out in Section 6, such as the behaviour for extreme maturities or the automatic computation of the number of coefficients. Section 7 concludes.

2 Motivation: rainbow option pricing

In this section we define the two-dimensional pricing formula as a discounted expectation of the option value at expiration. From now on, bold letters will denote vectors.

Assume $(\Omega, \mathcal{F}, P)$ is a probability space, $T > 0$ is the finite terminal time, and $\mathbb{F} = (\mathcal{F}_S)_{0 \leq S \leq T}$ is a filtration with the usual conditions. Then, 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 of the underlying. We assume that the bivariate characteristic function of the stochastic process is known.

The value of a European rainbow option with payoff function $g(\cdot)$, which depends on the underlying asset price, is given by the risk-neutral option valuation formula,

$$v(t_0, \mathbf{x}) = e^{-r\Delta t}\mathbb{E}\left[g\left(\mathbf{X}_T\right)\right] = e^{-r\Delta t} \iint_{\mathbb{R}^2} g(\mathbf{y}) f(\mathbf{y}|\mathbf{x})\, d\mathbf{y}, \tag{2.1}$$

where $\mathbf{x} = (x_1, x_2)$ is the current state, $f(y_1, y_2|x_1, x_2)$ is the underlying conditional density function, r is the risk-free rate and $\Delta t := T - t_0$ denotes time to expiration.

Whereas f is typically not known, its characteristic function is often available, this is, the Fourier transform of f. The strategy followed to determine the price of the option consist of approximating the density function f in (2.1) by a finite combination of Shannon scaling functions and recovering the coefficients of the approximation from its Fourier transform. A brief review of the basic theory of wavelets is given in the next section.

2.1 Multidimensional wavelets framework

We introduce some useful definitions for the two-dimensional framework which can be easily extended to more dimensions.

The space $L^2(\mathbb{R}^2)$ is the vector space of measurable, square-integrable two dimensional functions $f(x, y)$. For any two functions $h(x, y)$, $p(x, y) \in L^2(\mathbb{R}^2)$, their inner product in $L^2\left(\mathbb{R}^2\right)$ is defined by

$$\langle h\left(x,y\right), p\left(x,y\right)\rangle := \iint_{\mathbb{R}^2} h\left(x,y\right) \bar{p}\left(x,y\right) dxdy, \tag{2.2}$$

where $\bar{p}$ is the complex conjugate of p. The bivariate Fourier transform of a function $h(x, y) \in L^2(\mathbb{R}^2)$ is defined by,

$$\hat{h}(u_1, u_2) := \iint_{\mathbb{R}^2} h(x,y) e^{-i(u_1 x + u_2 y)}\, dxdy. \tag{2.3}$$

Now, we can introduce the wavelet theory in two dimensions. To begin with, we give a brief description of the one-dimensional case because the multidimensional derivations are based on that approach. There are two functions that play a primary role in wavelet analysis, the scaling function (or father wavelet) ϕ and the wavelet (or mother wavelet) ψ. By the following definition one can define a general structure for wavelets in $L^2(\mathbb{R})$, which is called a multiresolution analysis (MRA).

Definition 1. *Let V_j, $j = \cdots, -2, -1, 0, 1, 2, \cdots$ be a sequence of subspaces of functions in $L^2(\mathbb{R})$. The collection of spaces $(V_j)_{j \in \mathbb{Z}}$ is called a multiresolution analysis of $L^2(\mathbb{R})$ with scaling function ϕ, if the following conditions hold*

1. *(nested)* $V_j \subset V_{j+1}$,
2. *(dense)* $\overline{\cup V_j} = L^2(\mathbb{R})$,

3. *(separation)* $\cap V_j = \{0\}$,

4. *(scaling) The function* $f(x)$ *belongs to* V_j *if and only if the function* $f(2x)$ *belongs to* V_{j+1},

5. *(orthonormal basis) The function* ϕ *belongs to* V_0 *and the set* $\{\phi(x-k), k \in \mathbb{Z}\}$ *is an orthonormal basis (using the* L^2 *inner product) for* V_0.

The V_j are called approximation spaces. Different choices of ϕ yield different multiresolution analysis. If $(V_j)_{j\in\mathbb{Z}}$ is an MRA with scaling function ϕ, then for any $j \in \mathbb{Z}$, the set of functions $\{\phi_{j,k}(x) = 2^{j/2}\phi(2^j x - k); k \in \mathbb{Z}\}$ is an orthonormal basis for V_j. We define $W_j \subset V_{j+1}$ as the orthogonal complement of V_j in V_{j+1}, and it is called the detail space. Furthermore, there exist $\psi \in W_0$ such that $\left\{\psi_{j,k}(x) := 2^{j/2}\psi(2^j x - k), k \in \mathbb{Z}\right\}$ is an orthornormal basis for W_j. In addition, we have that

$$L^2(\mathbb{R}) = \cdots \oplus W_{-1} \oplus W_0 \oplus W_1 \oplus \cdots . \tag{2.4}$$

Multiresolution analysis which defines general wavelet structures in L^2 spaces, can be generalized to any dimension $d \in \mathbb{N}$ (see [7] for details). We illustrate the two-dimensional case.

A multiresolution approximation of $L^2(\mathbb{R}^2)$ is a sequence of subspaces of $L^2(\mathbb{R}^2)$ which satisfies a straightforward two-dimensional extension of the properties of MRA presented in Definition 1. Let $\left(V_j^2\right)_{j\in\mathbb{Z}}$ be such a multiresolution approximation of $L^2(\mathbb{R}^2)$. One can show that there exists a unique two-dimensional scaling function $\Phi(x,y)$ whose dilation and translation give an orthonormal basis of each space V_j^2 (see [1, 4]).

We stay in the particular case of separable wavelet bases and separable multiresolutions, due to the direct connection to the one-dimensional case, and because it avoids mixing of information at two different scales. For such multiresolution approximations, each vector space V_j^2 is decomposed as a tensor product of two identical subspaces of $L^2(\mathbb{R})$ whose elements are products of functions dilated at the same scale,

$$V_j^2 = V_j \otimes V_j. \tag{2.5}$$

The sequence of vector spaces $\left(V_j^2\right)_{j\in\mathbb{Z}}$ forms a multiresolution approximation of $L^2(\mathbb{R}^2)$ if and only if $(V_j)_{j\in\mathbb{Z}}$ is a multiresolution analysis of $L^2(\mathbb{R})$. It is easy to see that the two-dimensional scaling function $\Phi(x,y)$ can be written as

$$\Phi(x,y) := \phi(x)\phi(y), \tag{2.6}$$

where $\phi(x)$ is the one-dimensional scaling function of V_j.

As stated, $\{\phi_{j,m}; m \in \mathbb{Z}\}$ is an orthonormal basis of V_j. Since $V_j^2 = V_j \otimes V_j$, for $\boldsymbol{x} = (x_1, x_2)$ and $\boldsymbol{k} = (k_1, k_2)$,

$$\{\Phi_{j,\boldsymbol{k}}(\boldsymbol{x}) = \phi_{j,k_1}(x_1)\phi_{j,k_2}(x_2)\}_{\boldsymbol{k}\in\mathbb{Z}^2} \tag{2.7}$$

is an orthonormal basis of V_j^2.

Let W_j^2 be the detail space equal to the orthogonal complement of the lower resolution approximation space V_j^2 in V_{j+1}^2,

$$V_{j+1}^2 = V_j^2 \oplus W_j^2. \tag{2.8}$$

The following theorem builds a wavelet basis of each detail space W_j^2, to construct later a wavelet orthonormal basis of $L^2(\mathbb{R}^2)$. A separable orthonormal wavelet basis of $L^2(\mathbb{R}^2)$ is constructed with separable products of a scaling function ϕ and a wavelet ψ.

Theorem 1 (Theorem 7.24 of [4]). *Let* ϕ *be a scaling function and* ψ *be the corresponding wavelet generating a wavelet orthonormal wavelet basis of* $L^2(\mathbb{R})$. *We define the three wavelets*

$$\Psi^1(\boldsymbol{x}) = \phi(x_1)\psi(x_2),\ \Psi^2(\boldsymbol{x}) = \psi(x_1)\phi(x_2),\ \Psi^3(\boldsymbol{x}) = \psi(x_1)\psi(x_2), \tag{2.9}$$

and denote, for $1 \le n \le 3$,

$$\Psi_{j,k}^n(\boldsymbol{x}) = 2^j \Psi^n\left(2^j x_1 - k_1, 2^j x_2 - k_2\right). \tag{2.10}$$

Then, the wavelet family

$$\left\{\Psi^1_{j,\boldsymbol{k}},\ \Psi^2_{j,\boldsymbol{k}},\ \Psi^3_{j,\boldsymbol{k}}\right\}_{\boldsymbol{k}\in\mathbb{Z}^2} \tag{2.11}$$

forms an orthonormal basis of W^2_j*. And*

$$\left\{\Psi^1_{j,\boldsymbol{k}},\ \Psi^2_{j,\boldsymbol{k}},\ \Psi^3_{j,\boldsymbol{k}}\right\}_{(j,\boldsymbol{k})\in\mathbb{Z}\times\mathbb{Z}^2} \tag{2.12}$$

forms an orthonormal basis of $L^2(\mathbb{R}^2)$.

For any function $f(x_1, x_2) \in L^2(\mathbb{R}^2)$, a projection map of $L^2(\mathbb{R}^2)$ onto V^2_m, $\mathcal{P}_m : L^2(\mathbb{R}^2) \to V^2_m$, is defined as

$$\begin{aligned}\mathcal{P}_m f(x_1, x_2) :&= \sum_{j=-\infty}^{m-1} \sum_{n=1,2,3} \sum_{k_1\in\mathbb{Z}} \sum_{k_2\in\mathbb{Z}} d^n_{j,k_1,k_2} \Psi^n_{j,k_1,k_2}(x_1, x_2)\\ &= \sum_{k_1\in\mathbb{Z}} \sum_{k_2\in\mathbb{Z}} c_{m,k_1,k_2} \Phi_{m,k_1,k_2}(x_1, x_2)\\ &= \sum_{k_1\in\mathbb{Z}} \sum_{k_2\in\mathbb{Z}} c_{m,k_1,k_2} \phi_{m,k_1}(x_1)\phi_{m,k_2}(x_2),\end{aligned} \tag{2.13}$$

where

$$c_{m,k_1,k_2} := \iint_{\mathbb{R}^2} f(x_1, x_2)\Phi_{m,k_1,k_2}(x_1, x_2)\, dx_1 dx_2 \tag{2.14}$$

are the scaling coefficients, and

$$d^n_{j,k_1,k_2} := \iint_{\mathbb{R}^2} f(x_1, x_2)\Psi^n_{m,k_1,k_2}(x_1, x_2)\, dx_1 dx_2 \tag{2.15}$$

are the wavelets coefficients. And the convergence is in the $L^2(\mathbb{R}^2)$ space, this is,

$$\|f - \mathcal{P}_m f\|_2 \underset{m\to+\infty}{\longrightarrow} 0. \tag{2.16}$$

In this work, we consider the Shannon father function rather than the mother wavelet, due to its tractability and simplicity. Hence, our wavelet bases are a set of Shannon scaling functions in the subspace V_m, given by, for $x \in \mathbb{R}$,

$$\phi_{m,k}(x) = 2^{m/2}\operatorname{sinc}\left(2^m x - k\right), \tag{2.17}$$

where the sinc function is defined as

$$\operatorname{sinc}(x) := \frac{\sin(\pi x)}{\pi x}. \tag{2.18}$$

The two-dimensional Shannon scaling function is given by, see Figure 1,

$$\Phi_{m,k_1,k_2}(x_1, x_2) := \phi_{m,k_1}(x_1)\phi_{m,k_2}(x_2) = 2^m \operatorname{sinc}(2^m x_1 - k_1)\operatorname{sinc}(2^m x_2 - k_2). \tag{2.19}$$

2.1.1 Wavelets bases in higher dimensions

Separable orthonormal wavelet bases of $L^2(\mathbb{R}^p)$ can be constructed for any $p \geq 2$ with a procedure similar to the two-dimensional extension. Let ϕ be a scaling function and ψ a wavelet that yields an orthogonal basis of $L^2(\mathbb{R})$. Now, $V^p_j = V_j \otimes \cdots \otimes V_j$ and W^p_j is the detail space, i.e. the orthogonal complement of the lower resolution approximation space.

We denote $\theta^0 := \phi$ and $\theta^1 := \psi$. For an integer $0 \leq \epsilon \leq 2^p$ written in binary form, $\epsilon = \epsilon_1 \cdots \epsilon_p$, we associate p-dimensional functions defined in $\boldsymbol{x} = (x_1, \cdots, x_p)$ by

$$\psi^\epsilon(\boldsymbol{x}) := \theta^{\epsilon_1}(x_1) \cdots \theta^{\epsilon_n}(x_p), \tag{2.20}$$

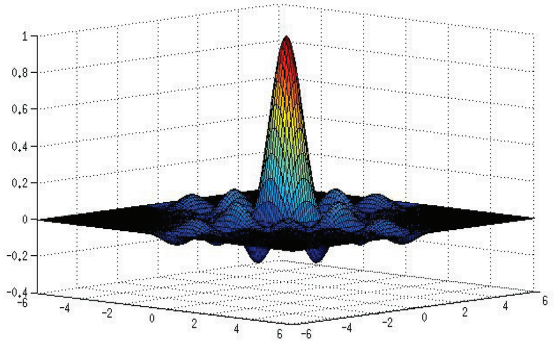

Figure 1: Two-dimensional Shannon wavelet $\Phi_{0,0,0}(x_1, x_2)$, with $(x_1, x_2) \in [-6, 6]^2$.

For $\epsilon = 0$, we obtain a $p-$dimensional scaling function

$$\psi^0(\boldsymbol{x}) = \phi(x_1) \cdots \phi(x_p). \tag{2.21}$$

Non-zero indexes ϵ correspond to $2^p - 1$ wavelets. At any scale 2^j and for $\boldsymbol{n} = (n_1, \cdots, n_p)$ we denote

$$\psi^{\epsilon}_{j,\boldsymbol{n}}(\boldsymbol{x}) := 2^{pj/2} \psi^{\epsilon}\left(2^j x_1 - n_1, \cdots, 2^j x_p - n_p\right). \tag{2.22}$$

Theorem 2 (Theorem 7.25 in [4])**.** *The family obtained by dilating and translating the $2^p - 1$ wavelets for $\epsilon \neq 0$*

$$\left\{\psi^{\epsilon}_{j,\boldsymbol{n}}\right\}_{1 \leq \epsilon < 2^p, \boldsymbol{n} \in \mathbb{Z}^p} \tag{2.23}$$

is an orthonormal basis of W^p_j. And the family

$$\left\{\psi^{\epsilon}_{j,\boldsymbol{n}}\right\}_{1 \leq \epsilon < 2^p, (j,\boldsymbol{n}) \in \mathbb{Z}^{p+1}} \tag{2.24}$$

is an orthonormal basis of $L^2(\mathbb{R}^p)$.

For any function $f \in L^2(\mathbb{R}^p)$, a projection map of $L^2(\mathbb{R}^p)$ onto V^p_m, $\mathcal{P}_m : L^2(\mathbb{R}^p) \to V^p_m$, is defined as

$$\begin{aligned} \mathcal{P}_m f(\boldsymbol{x}) :&= \sum_{j=-\infty}^{m-1} \sum_{0<\epsilon\leq 2^p} \sum_{k_1 \in \mathbb{Z}} \sum_{k_2 \in \mathbb{Z}} \cdots \sum_{k_p \in \mathbb{Z}} d^{\epsilon}_{j,k_1,\cdots,k_p} \psi^{\epsilon}_{j,k_1,\cdots,k_p}(\boldsymbol{x}) \\ &= \sum_{k_1 \in \mathbb{Z}} \sum_{k_2 \in \mathbb{Z}} \cdots \sum_{k_p \in \mathbb{Z}} c_{m,k_1,\cdots,k_p} \phi_{m,k_1}(x_1) \phi_{m,k_2}(x_2) \cdots \phi_{m,k_p}(x_p), \end{aligned} \tag{2.25}$$

where $c_{m,k_1,\cdots,k_p}$ are the scaling coefficients and d^n_{j,k_1,k_2} the wavelets coefficients.

The p-dimensional Shannon scaling function is given by,

$$\Phi_{m,k_1,\cdots,k_p}(\boldsymbol{x}) := \phi_{m,k_1}(x_1) \cdots \phi_{m,k_p}(x_p) = 2^{\frac{pm}{2}} \operatorname{sinc}(2^m x_1 - k_1) \cdots \operatorname{sinc}(2^m x_p - k_p). \tag{2.26}$$

Further details about wavelets in high dimensions can be found in Section 7.7.4 of [4].

2.1.2 Approximation of the sinc function

Central to the option pricing process, that will be presented later, is a convenient approximation of the sinc function by means of a midpoint quadrature rule. For $t, x \in \mathbb{R}$,

$$\operatorname{sinc}(t) = \frac{1}{2\pi} \int_{\mathbb{R}} \widehat{\operatorname{sinc}}(x) e^{itx}\, dx = \frac{1}{2\pi} \int_{\mathbb{R}} \operatorname{rect}\left(\frac{x}{2\pi}\right) e^{itx}\, dx = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{itx}\, dx \approx \operatorname{sinc}^*_N(t) := \frac{1}{N} \sum_{j=1}^{N} e^{i\Omega_j t}, \tag{2.27}$$

where $\widehat{\text{sinc}}$ represents the Fourier transform of sinc and rect is the rectangle function,

$$\operatorname{rect}(x) = \begin{cases} 1, & \text{if } |x| < \frac{1}{2}, \\ \frac{1}{2}, & \text{if } |x| = \frac{1}{2}, \\ 0, & \text{if } |x| > \frac{1}{2}, \end{cases} \tag{2.28}$$

being N the number of points in the midpoint quadrature and $\Omega_j := -\pi + \frac{2j-1}{N}\pi$.

The approximation presented in the one-dimensional case in [10] reads,

$$\operatorname{sinc}(t) \approx \operatorname{sinc}^*(t) := \frac{1}{2^{J-1}} \sum_{j=1}^{2^{J-1}} \cos\left(\frac{2j-1}{2^J}\pi t\right) = \frac{1}{2^{J-1}} \sum_{j=1}^{2^{J-1}} \cos\left(\omega_j t\right), \tag{2.29}$$

for $\omega_j = \frac{2j-1}{2^J}\pi$, where the authors approximate the cardinal sine function by a finite combination of cosines through the application of Vieta's formula.

It can be easily shown that the approximation (2.29) is a particular case of the approximation (2.27) for $N = 2^J$, indeed,

$$\begin{aligned} \operatorname{sinc}^*(t) &= \frac{2}{N} \sum_{j=1}^{N/2} \cos\left(\frac{2j-1}{N}\pi t\right) = \frac{1}{N} \sum_{j=1}^{N/2} \left(e^{i\left(\frac{2j-1}{N}\right)\pi t} + e^{-i\left(\frac{2j-1}{N}\right)\pi t}\right) = \frac{1}{N} \sum_{j=1}^{N/2} e^{i\left(\frac{2j-1}{N}\right)\pi t} \\ &+ \frac{1}{N} \sum_{j=1-N/2}^{0} e^{i\left(\frac{2j-1}{N}\right)\pi t} = \frac{1}{N} \sum_{j=1-N/2}^{N/2} e^{i\left(\frac{2j-1}{N}\right)\pi t} = \frac{1}{N} \sum_{j=1}^{N} e^{i\left(\frac{2j-N-1}{N}\right)\pi t} = \operatorname{sinc}^*_N(t). \end{aligned} \tag{2.30}$$

The complex exponential form (2.27) will facilitate the derivation of the density and payoff coefficients along the option pricing procedure presented in the next section.

3 European rainbow option pricing

We present the 2D-SWIFT formula to recover the density function f in expression (2.1) assuming that its Fourier transform $\hat{f}$ is known. The density function of the asset price process at terminal time T is usually not known, but often its characteristic function is known. The method is based on the approximation of the density function by a finite combination of Shannon wavelets in dimension two. Once the density has been recovered, we replace f by its approximation in formula (2.1) to get the final price of the financial contract. This two-step procedure is explained in detail in Section 3.1, while the general case for higher dimensions is briefly exposed in Section 3.2.

3.1 Derivation of the 2D-SWIFT method

As mentioned before, the first step during the derivation of the 2D-SWIFT pricing formula is the recovery of the density function. We distinguish three main steps,

Step 1. We approximate the conditional density function by a finite combination of Shannon scaling functions,

$$f(\mathbf{y}|\mathbf{x}) \approx f_1(\mathbf{y}|\mathbf{x}) := \mathcal{P}_m f(\mathbf{y}|\mathbf{x}) = \sum_{k_1 \in \mathbb{Z}} \sum_{k_2 \in \mathbb{Z}} D_{m,k_1,k_2}(\mathbf{x}) \Phi_{m,k_1,k_2}(\mathbf{y}), \tag{3.1}$$

where D_{m,k_1,k_2}, the density coefficients, are defined by,

$$D_{m,k_1,k_2}(\mathbf{x}) := \iint_{\mathbb{R}^2} f(\mathbf{y}|\mathbf{x}) \Phi_{m,k_1,k_2}(\mathbf{y})\, d\mathbf{y}. \tag{3.2}$$

Step 2. Next, we truncate the summation range such that $k_1 \in \{l_1, \ldots, u_1\}$ and $k_2 \in \{l_2, \ldots, u_2\}$, and thus, the density approximation becomes,

$$f_1(\mathbf{y}|\mathbf{x}) \approx f_2(\mathbf{y}|\mathbf{x}) := \sum_{k_1=l_1}^{u_1} \sum_{k_2=l_2}^{u_2} D_{m,k_1,k_2}(\mathbf{x}) \Phi_{m,k_1,k_2}(\mathbf{y}). \tag{3.3}$$

Step 3. Using the complex exponential formula (2.27), we approximate the two-dimensional Shannon scaling function as,

$$\Phi_{m,k_1,k_2}(\mathbf{y}) \approx \Phi^*_{m,k_1,k_2}(\mathbf{y}) := \frac{2^m}{N^2} \sum_{j_1=1}^{N} e^{i\Omega_{j_1}(2^m y_1 - k_1)} \sum_{j_2=1}^{N} e^{i\Omega_{j_2}(2^m y_2 - k_2)}, \tag{3.4}$$

where $\Omega_j = -\pi + \frac{2j-1}{N}\pi$. For convenience, we use in both dimensions the same discretization in the sinc approximation, this is, N points.

Then, from (3.2) and (3.4), we approximate the density coefficients $D_{m,k_1,k_2}(\mathbf{x})$ by,

$$\begin{aligned} D^*_{m,k_1,k_2}(\mathbf{x}) :&= \iint_{\mathbb{R}^2} f(\mathbf{y}|\mathbf{x}) \Phi^*_{m,k_1,k_2}(\mathbf{y}) d\mathbf{y} \\ &= \frac{2^m}{N^2} \sum_{j_1=1}^{N} \sum_{j_2=1}^{N} e^{-i\sum_{n=1,2} \Omega_{j_n} k_n} \iint_{\mathbb{R}^2} f(\mathbf{y}|\mathbf{x}) e^{i\sum_{n=1,2} \Omega_{j_n} 2^m y_n} dy_1 dy_2 \\ &= \frac{2^m}{N^2} \sum_{j_1=1}^{N} \sum_{j_2=1}^{N} e^{-i\sum_{n=1,2} \Omega_{j_n} k_n} \hat{f}\left(-\Omega_{j_1} 2^m, -\Omega_{j_2} 2^m | \mathbf{x}\right). \end{aligned} \tag{3.5}$$

Finally, from (3.3) and (3.5) we obtain the density approximation,

$$f_2(\mathbf{y}|\mathbf{x}) \approx f^*(\mathbf{y}|\mathbf{x}) := \sum_{k_1=l_1}^{u_1} \sum_{k_2=l_2}^{u_2} D^*_{m,k_1,k_2}(\mathbf{x}) \Phi_{m,k_1,k_2}(\mathbf{y}). \tag{3.6}$$

The 2D-SWIFT formula for the approximation of $v(t_0, \mathbf{x})$ in (2.1) is obtained first by truncating the integration range,

$$v(t_0, \mathbf{x}) \approx v_1(t_0, \mathbf{x}) := e^{-r\Delta t} \int_{a_1}^{b_1} \int_{a_2}^{b_2} g(\mathbf{y}) f(\mathbf{y}|\mathbf{x})\, d\mathbf{y}, \tag{3.7}$$

for some values a_1, a_2, b_1, b_2, and then replacing the density $f(\mathbf{y}|\mathbf{x})$ by the approximation $f^*(\mathbf{y}|\mathbf{x})$ from (3.6). Finally, the 2D-SWIFT pricing formula is given by,

$$\begin{aligned} v_1(t_0, \mathbf{x}) \approx v^*(t_0, \mathbf{x}) &:= e^{-r\Delta t} \int_{a_1}^{b_1} \int_{a_2}^{b_2} g(\mathbf{y}) f^*(\mathbf{y}|\mathbf{x})\, d\mathbf{y} \\ &= e^{-r\Delta t} \sum_{k_1=l_1}^{u_1} \sum_{k_2=l_2}^{u_2} D^*_{m,k_1,k_2}(\mathbf{x}) \int_{a_1}^{b_1} \int_{a_2}^{b_2} g(\mathbf{y}) \Phi_{m,k_1,k_2}(\mathbf{y})\, d\mathbf{y} \\ &= e^{-r\Delta t} \sum_{k_1=l_1}^{u_1} \sum_{k_2=l_2}^{u_2} D^*_{m,k_1,k_2}(\mathbf{x})\, G_{m,k_1,k_2}, \end{aligned} \tag{3.8}$$

where we define the payoff coefficients as,

$$G_{m,k_1,k_2} := \int_{a_1}^{b_1} \int_{a_2}^{b_2} g(\mathbf{y}) \Phi_{m,k_1,k_2}(\mathbf{y})\, d\mathbf{y}. \tag{3.9}$$

Remark 1. *An alternative method to get the density coefficients would be a straightforward extension of the one-dimensional method [10],which basically considers the approximation in (2.29) for the* sinc *function. We approximate the two-dimensional Shannon scaling function as,*

$$\Phi_{m,k_1,k_2}(\mathbf{y}) \approx \frac{2^m}{2^{2(J-1)}} \sum_{j_1=1}^{2^{J-1}} \sum_{j_2=1}^{2^{J-1}} \cos(\omega_{j_1}(2^m y_1 - k_1))\cos(\omega_{j_2}(2^m y_2 - k_2)), \tag{3.10}$$

where $\omega_j := \frac{2j-1}{2^J}\pi$*. Hence, the density coefficients can also be approximated by,*

$$D^{**}_{m,k_1,k_2}(\mathbf{x}) = \frac{2^m}{2^{2(J-1)}} \sum_{j_1=1}^{2^{J-1}} \sum_{j_2=1}^{2^{J-1}} \iint_{\mathbb{R}^2} f(\mathbf{y}|\mathbf{x}) \prod_{n=1,2} \cos(\omega_{j_n}(2^m y_n - k_n))\, d\mathbf{y}. \tag{3.11}$$

Using the goniometric relation $2\cos(\alpha)\cos(\beta) = \cos(\alpha+\beta) + \cos(\alpha-\beta)$*, we obtain from (3.11),*

$$\begin{aligned} D^{**}_{m,k_1,k_2}(\mathbf{x}) = \frac{2^{m+1}}{2^{2(J-1)}} \sum_{j_1=1}^{2^{J-1}} \sum_{j_2=1}^{2^{J-1}} \Bigg[&\iint_{\mathbb{R}^2} f(\mathbf{y}|\mathbf{x}) \cos(\sum_{n=1,2} \omega_{j_n}(2^m y_n - k_n))\, d\mathbf{y} \\ &+ \iint_{\mathbb{R}^2} f(\mathbf{y}|\mathbf{x}) \cos(\sum_{n=1,2} (-1)^{n+1}\omega_{j_n}(2^m y_n - k_n)))\, d\mathbf{y} \Bigg]. \end{aligned} \tag{3.12}$$

Observe that $-\omega_{j_2} = \omega_{1-j_2}$*. Then,*

$$D^{**}_{m,k_1,k_2}(\mathbf{x}) = \frac{2^{m+1}}{2^{2(J-1)}} \sum_{j_1=1}^{2^{J-1}} \sum_{j_2=1-2^{J-1}}^{2^{J-1}} \iint_{\mathbb{R}^2} f(\mathbf{y}|\mathbf{x}) \cos(\sum_{n=1,2} \omega_{j_n}(2^m y_n - k_n))\, d\mathbf{y}. \tag{3.13}$$

Since $\mathrm{Re}\left\{e^{-i\alpha}\right\} = \cos(\alpha)$*, we find,*

$$\begin{aligned} D^{**}_{m,k_1,k_2}(\mathbf{x}) &= \frac{2^{m+1}}{2^{2(J-1)}} \mathrm{Re}\left\{ \sum_{j_1=1}^{2^{J-1}} \sum_{j_2=1-2^{J-1}}^{2^{J-1}} \iint_{\mathbb{R}^2} f(\mathbf{y}|\mathbf{x}) e^{\left(-i\sum_{n=1,2}\omega_{j_n}(2^m y_n - k_n)\right)}\, d\mathbf{y} \right\} \\ &= \frac{2^{m+1}}{2^{2(J-1)}} \mathrm{Re}\left\{ \sum_{j_1=1}^{2^{J-1}} \sum_{j_2=1-2^{J-1}}^{2^{J-1}} e^{\left(i\sum_{n=1,2}\omega_{j_n} k_n\right)} \iint_{\mathbb{R}^2} f(\mathbf{y}|\mathbf{x}) e^{\left(-i\sum_{n=1,2}\omega_{j_n} 2^m y_n\right)}\, d\mathbf{y} \right\} \\ &= \frac{2^{m+1}}{2^{2(J-1)}} \mathrm{Re}\left\{ \sum_{j_1=1}^{2^{J-1}} \sum_{j_2=1-2^{J-1}}^{2^{J-1}} \hat{f}(\omega_{j_1}2^m, \omega_{j_2}2^m|\mathbf{x}) e^{\left(i\sum_{n=1,2}\omega_{j_n} k_n\right)} \right\}, \end{aligned} \tag{3.14}$$

where $\hat{f}$ *is the Fourier transform of* f*. Observe that, for* $N = 2^J$*, we have* $D^*_{m,k_1,k_2}(\mathbf{x}) = D^{**}_{m,k_1,k_2}(\mathbf{x})$*. We note that Vieta's formula is more tedious to use and we will therefore consider the approximation (2.27) to compute the payoff coefficients in Section 5.*

Remark 2. *In our search for an efficient method, formula (3.5) can be conveniently rearranged to get a new expression where an FFT algorithm can be applied. Using the fact that,*

$$-i\Omega_j k = \frac{-i2\pi k}{N}(j-1) - i\pi k\left(\frac{1}{N} - 1\right), \tag{3.15}$$

we get,

$$D^*_{m,k_1,k_2}(\mathbf{x}) = \frac{2^m}{N^2} e^{-i\pi\left(\frac{1}{N}-1\right)(k_1+k_2)} \sum_{j_1=1}^{N} \sum_{j_2=1}^{N} \hat{f}\left(-\Omega_{j_1}2^m, -\Omega_{j_2}2^m|\,\mathbf{x}\right) e^{\frac{-i2\pi k_2}{N}(j_2-1)} e^{\frac{-i2\pi k_1}{N}(j_1-1)}. \tag{3.16}$$

We apply N *times an FFT for each sum in (3.16) and the computational complexity to perform this task is* $\mathcal{O}(N^2\log_2(N))$*.*

3.2 General multidimensional SWIFT formula

The 2D-SWIFT formula can be easily generalized to higher dimensions. It is clear, however, that if we choose the dimension d to be very large, the curse of dimensionality issue will occur and numerical techniques turn out to be useless in practical terms.

For $x \in \mathbb{R}^d$, the d-dimensional formula reads,

$$v^*(t_0, \mathbf{x}) = e^{-r\Delta t} \sum_{k_1=l_1}^{u_1} \sum_{k_2=l_2}^{u_2} \cdots \sum_{k_d=l_d}^{u_d} D^*_{m,k_1,k_2,\cdots,k_d}(\mathbf{x}) G_{m,k_1,k_2,\cdots,k_d}, \tag{3.17}$$

where,

$$\begin{aligned} D^*_{m,k_1,k_2,\cdots,k_d}(\mathbf{x}) =& \frac{2^{\frac{dm}{2}}}{N^d} e^{-i\pi\left(\frac{1}{N}-1\right)(k_1+k_2+\cdots+k_d)} \sum_{j_1=l_1}^{u_1} \sum_{j_2=l_2}^{u_2} \cdots \sum_{j_d=l_d}^{u_d} \hat{f}(-\Omega_{j_1} 2^m, -\Omega_{j_2} 2^m, \cdots, -\Omega_{j_d} 2^m) \\ & \cdot e^{\frac{-i2\pi k_d}{N}(j_d-1)} \cdots e^{\frac{-i2\pi k_2}{N}(j_2-1)} e^{\frac{-i2\pi k_1}{N}(j_1-1)}, \end{aligned} \tag{3.18}$$

and,

$$G_{m,k_1,k_2,\cdots,k_d} = \int_{a_1}^{b_1} \int_{a_2}^{b_2} \cdots \int_{a_d}^{b_d} g(\mathbf{y}) \Phi_{m,k_1,k_2,\cdots,k_d}(\mathbf{y}) d\mathbf{y}. \tag{3.19}$$

4 Error analysis and parameters selection

In this section we present an error analysis of the 2D-SWIFT method and give a prescription on the selection of the parameters.

4.1 Error analysis

There are two main sources of error. The first one is the error due to the truncation of the integration range in (2.1), while the second one is the error caused by replacing the density function f in (3.7) by f^*. If we define,

$$\mathcal{E}_1 := |v(t_0, \mathbf{x}) - v_1(t_0, \mathbf{x})|, \quad \text{and,} \quad \mathcal{E}_2 := |v_1(t_0, \mathbf{x}) - v^*(t_0, \mathbf{x})|, \tag{4.1}$$

then the overall error $\mathcal{E} := |v(t_0, \mathbf{x}) - v^*(t_0, \mathbf{x})|$ can be bounded by $\mathcal{E} \leq \mathcal{E}_1 + \mathcal{E}_2$. In what follows, we give a detailed analysis of the error. Let us start by considering,

$$\mathcal{E}_1 = |v(t_0, \mathbf{x}) - v_1(t_0, \mathbf{x})| \leq \int_{\mathbb{R}\backslash[a_1,b_1]} \int_{\mathbb{R}\backslash[a_2,b_2]} |g(\mathbf{y}) f(\mathbf{y}|\mathbf{x})| \, d\mathbf{y}. \tag{4.2}$$

Since the mass in the tails of the density function f tends to zero at infinity, for every $\varepsilon_1 > 0$ there exist values $a_1, a_2, b_1, b_2 > 0$ such that,

$$\tau := \int_{\mathbb{R}\backslash[a_1,b_1]} \int_{\mathbb{R}\backslash[a_2,b_2]} f(\mathbf{y}|\mathbf{x}) \, d\mathbf{y} \leq \varepsilon_1. \tag{4.3}$$

If we assume that g is bounded in the domain of integration, then,

$$\mathcal{E}_1 \leq \tau \|g\|_\infty, \tag{4.4}$$

where $\|g\|_\infty$ is the infinity norm of g. This error can be made arbitrarily small by increasing the size of the truncation intervals. Observe that the assumption on the boundedness of function g is satisfied for put options. If we consider call options, then we may impose some assumptions on the decay rate of the density function f to have an estimation of the error. For sake of clarity and simplicity in the exposition, we stay within the assumption of bounded payoffs.

As mentioned in Section 3.1, there are three sources of error when approximating f by f^*. If we define,

(i) the projection error, given by,

$$\epsilon_p := |f(\mathbf{y}|\mathbf{x}) - f_1(\mathbf{y}|\mathbf{x})| = \left| f(\mathbf{y}|\mathbf{x}) - \sum_{k_1\in\mathbb{Z}} \sum_{k_2\in\mathbb{Z}} D_{m,k_1,k_2}(\mathbf{x})\Phi_{m,k_1,k_2}(\mathbf{y}) \right|, \tag{4.5}$$

where D_{m,k_1,k_2} and Φ_{m,k_1,k_2} are in (3.2) and (2.19) respectively,

(ii) the sum truncation error,

$$\epsilon_t := |f_1(\mathbf{y}|\mathbf{x}) - f_2(\mathbf{y}|\mathbf{x})| = \left| \sum_{k_1\notin\{l_1,\cdots,u_1\}} \sum_{k_2\notin\{l_2,\cdots,u_2\}} D_{m,k_1,k_2}(\mathbf{x})\Phi_{m,k_1,k_2}(\mathbf{y}) \right|, \tag{4.6}$$

(iii) and the coefficients approximation error,

$$\epsilon_c := |f_2(\mathbf{y}|\mathbf{x}) - f^*(\mathbf{y}|\mathbf{x})| = \left| \sum_{k_1=l_1}^{u_1} \sum_{k_2=l_2}^{u_2} \left(D_{m,k_1,k_2}(\mathbf{x}) - D^*_{m,k_1,k_2}(\mathbf{x})\right)\Phi_{m,k_1,k_2}(\mathbf{y}) \right|, \tag{4.7}$$

then,

$$\begin{aligned} \mathcal{E}_2 = |v_1(t_0,\mathbf{x}) - v^*(t_0,\mathbf{x})| &= \left| e^{-r\Delta t} \int_{a_1}^{b_1}\int_{a_2}^{b_2} g(\mathbf{y})(f(\mathbf{y}|\mathbf{x}) - f^*(\mathbf{y}|\mathbf{x}))d\mathbf{y} \right| \\ &\le e^{-r\Delta t}\,|b_1-a_1|\,|b_2-a_2|\,\|g\|_\infty(\epsilon_p+\epsilon_t+\epsilon_c), \end{aligned} \tag{4.8}$$

and,

$$\mathcal{E} \le \left(\tau + e^{-r\Delta t}\,|b_1-a_1|\,|b_2-a_2|\,(\epsilon_p+\epsilon_t+\epsilon_c)\right)\|g\|_\infty. \tag{4.9}$$

The projection error in one dimension is studied in [5], and we use a similar procedure to give an estimation of that error for the two-dimensional case.

Lemma 1. *Consider an MRA generated by Shannon scaling function Φ defined in (2.6). With the same notation as before,*

$$\epsilon_p \le K(2^m\pi, 2^m\pi), \tag{4.10}$$

where,

$$K(v_1,v_2) := \frac{1}{4\pi^2}\int_{|\omega_1|>v_1}\int_{|\omega_2|>v_2} |\hat{f}(\boldsymbol{\omega})|d\boldsymbol{\omega}, \tag{4.11}$$

and $\hat{f}$ is the bivariate Fourier transform of f defined in (2.3).

Proof. See Appendix A. □

The sum truncation error ϵ_t depends on the size of the scaling coefficients D_{m,k_1,k_2}, since from (4.6) we have,

$$\epsilon_t \le 2^m \cdot \sum_{k_1\notin\{l_1,\cdots,u_1\}} \sum_{k_2\notin\{l_2,\cdots,u_2\}} |D_{m,k_1,k_2}(\mathbf{x})|, \tag{4.12}$$

and by Lemma 1,

$$\left| f(\mathbf{y}|\mathbf{x}) - \sum_{k_1\in\mathbb{Z}} \sum_{k_2\in\mathbb{Z}} D_{m,k_1,k_2}(\mathbf{x})\Phi_{m,k_1,k_2}(\mathbf{y}) \right| \le K(2^m\pi, 2^m\pi). \tag{4.13}$$

In particular, if we evaluate expression (4.13) in $\mathbf{y} = \left(\frac{k_1}{2^m}, \frac{k_2}{2^m}\right)$ then,

$$\left| f\left(\frac{k_1}{2^m}, \frac{k_2}{2^m} \;\middle|\; \mathbf{x}\right) - 2^m D_{m,k_1,k_2}(\mathbf{x}) \right| \le K(2^m\pi, 2^m\pi), \tag{4.14}$$

which means that the coefficients D_{m,k_1,k_2} are very well approximated by,

$$D_{m,k_1,k_2}(\mathbf{x}) \approx \frac{1}{2^m} f\left(\frac{k_1}{2^m}, \frac{k_2}{2^m} \,\middle|\, \mathbf{x}\right), \tag{4.15}$$

when $|\hat{f}(\boldsymbol{\omega})|$ in (4.11) decays very fast, as typically happens with the densities considered in this work. Finally, if we assume that the density f tends to zero at minus and plus infinity, then the error ϵ_t can be neglected for sufficiently big truncation values l_1, l_2, u_1, u_2 in (3.3). In Section 4.2 we give a detailed explanation on how to select these values.

The last error we need to estimate is ϵ_c defined in (4.7). For this purpose, we present the following two lemmas.

Lemma 2 (Lemma 2 of [10])**.** *Define the absolute error $\mathcal{E}_V(t) := sinc(t) - sinc^*(t)$. Then,*

$$|\mathcal{E}_V(t)| \leq \frac{(\pi c)^2}{2^{2(J+1)} - (\pi c)^2}, \tag{4.16}$$

for $t \in [-c, c]$, where $c \in \mathbb{R}, c > 0$ and $J \geq \log_2(\pi c)$.

Lemma 3. *Define the error $\bar{\mathcal{E}}_V(t_1, t_2) := sinc(t_1)sinc(t_2) - sinc^*(t_1)sinc^*(t_2)$. Then,*

$$\left|\bar{\mathcal{E}}_V(t_1, t_2)\right| \leq |\mathcal{E}_V(t_1)| + |\mathcal{E}_V(t_2)|. \tag{4.17}$$

Proof. We observe that,

$$\begin{aligned} \bar{\mathcal{E}}_V(t_1, t_2) &= \mathrm{sinc}(t_1)\mathrm{sinc}(t_2) - \mathrm{sinc}^*(t_1)\mathrm{sinc}^*(t_2) - \mathrm{sinc}^*(t_1)\mathrm{sinc}(t_2) + \mathrm{sinc}^*(t_1)\mathrm{sinc}(t_2) \\ &= \mathrm{sinc}(t_2)[\mathrm{sinc}(t_1) - \mathrm{sinc}^*(t_1)] + \mathrm{sinc}^*(t_1)[\mathrm{sinc}(t_2) - \mathrm{sinc}^*(t_2)] \\ &= \mathrm{sinc}(t_2)\mathcal{E}_V(t_1) + \mathrm{sinc}^*(t_1)\mathcal{E}_V(t_1). \end{aligned} \tag{4.18}$$

The proof concludes by noting that $|\mathrm{sinc}(t_2)| = |\mathrm{sinc}^*(t_1)| = 1$ for all $t_1, t_2 \in \mathbb{R}$. □

Theorem 3. *Let $F(\mathbf{x})$ be the distribution function of a two-dimensional random variable $\boldsymbol{X}$ and define $H(\mathbf{x}) := F(-\mathbf{x}) + 1 - F(\mathbf{x})$. Let $c_1, c_2 > 0$ be constants such that $H(c_1, c_2) < \epsilon$, for $\epsilon > 0$, and let $c := \max(c_1, c_2)$. Define,*

$$M^1_{m,k_1} := \max\left(|2^m c - k_1|, |2^m c + k_1|\right), \quad M^2_{m,k_2} := \max\left(|2^m c - k_2|, |2^m c + k_2|\right),$$

$$M_{m,k_1,k_2} := \max\left(M^1_{m,k_1}, M^2_{m,k_2}\right),$$

and consider $J \geq \log_2\left(\pi M_{m,k_1,k_2}\right)$. Then,

$$|D_{m,k_1,k_2}(\mathbf{x}) - D^*_{m,k_1,k_2}(\mathbf{x})| \leq 2^m \left(2\epsilon + 4c\|f(\,\cdot \mid \mathbf{x})\|_2 \frac{\left(\pi M_{m,k_1,k_2}\right)^2}{2^{2(J+1)} - \left(\pi M_{m,k_1,k_2}\right)^2}\right), \tag{4.19}$$

and $\lim_{J \to +\infty} D^*_{m,k_1,k_2}(\mathbf{x}) = D_{m,k_1,k_2}(\mathbf{x})$.

Proof. See Appendix B. □

Finally, from (4.7) and Theorem 3,

$$\begin{aligned} \epsilon_c &\leq 2^m(u_1 - l_1 + 1)(u_2 - l_2 + 1)\left|D_{m,k_1,k_2}(\mathbf{x}) - D^*_{m,k_1,k_2}(\mathbf{x})\right| \\ &\leq 2^{2m}(u_1 - l_1 + 1)(u_2 - l_2 + 1)\left(2\epsilon + 4c\|f(\,\cdot \mid \mathbf{x})\|_2 \frac{\left(\pi M_{m,k_1,k_2}\right)^2}{2^{2(J+1)} - \left(\pi M_{m,k_1,k_2}\right)^2}\right). \end{aligned} \tag{4.20}$$

4.2 Parameters selection

The parameters of the method are the scale of approximation m in (3.1), the range of coefficients $l_1 \leq k_1 \leq u_1$, $l_2 \leq k_2 \leq u_2$ in (3.3), the number of terms $N = 2^J$ to approximate the cardinal sine function in (3.4), and the truncation of the integration domain $[a_1, b_1] \times [a_2, b_2]$ in expression (3.7).

Concerning the scale of approximation m, we know from Lemma 1 that the approximation error decreases with m. Further, this error is even smaller when the modulus of the characteristic function decays rapidly. We propose an adaptive computation of m by following this strategy. We select m such that given a tolerance $\varepsilon_2 > 0$,

$$|\hat{f}(-2^m\pi, -2^m\pi)| + |\hat{f}(-2^m\pi, 2^m\pi)| + |\hat{f}(2^m\pi, -2^m\pi)| + |\hat{f}(2^m\pi, 2^m\pi)| < \varepsilon_2. \tag{4.21}$$

An initial guess of the integration domain $[a_1, b_1] \times [a_2, b_2]$ is given (for example) by the nth cumulant[1] $c_{i,n}$ of $X_t^i := \log S_t^i$ and the parameter $L = 10$, like for the 2D-COS method in [11],

$$\begin{aligned}[a_1, b_1] \times [a_2, b_2] = &\left[c_{1,1} - L\sqrt{c_{2,1} + \sqrt{c_{4,1}}},\ c_{1,1} + L\sqrt{c_{2,1} + \sqrt{c_{4,1}}}\right] \\ &\times \left[c_{1,2} - L\sqrt{c_{2,2} + \sqrt{c_{4,2}}},\ c_{1,2} + L\sqrt{c_{2,2} + \sqrt{c_{4,2}}}\right].\end{aligned} \tag{4.22}$$

We compute the coefficients D^*_{m,k_1,k_2} by means of an FFT algorithm, with k_i ranging from $l_i = \lfloor 2^m a_i \rfloor$ to $u_i = \lceil 2^m b_i \rceil$ and $i = 1, 2$. It is worth underlining that this a-priori truncation facilitates the application of an FFT algorithm as mentioned in Remark 2 to get the stated computational complexity. Then, in order to know whether the initial truncated range is accurate or not, we measure the size of the density coefficients at the boundaries. For this purpose, we compute $D^*_{m,l_1,0}, D^*_{m,u_1,0}, D^*_{m,0,l_2}, D^*_{m,0,u_2}$, since as we have seen in (4.15) the size of those coefficients is closely related to the value of the density at the boundary points. If necessary, we can compute extra coefficients until the desired precision is reached. Moreover, we can also calculate the volume underneath the surface represented by the density f as a byproduct and verify that the volume is close to 1. Considering the partition of the domain $[a_1, b_1] \times [a_2, b_2]$ given by points of the form $\left(\frac{h_1}{2^m}, \frac{h_2}{2^m}\right)$ for $h_1, h_2 \in \mathbb{Z}$, the two-dimensional composite trapezoidal rule gives us the following approximation,

$$\begin{aligned}\mathrm{V}(f^*) := \iint_{\mathbb{R}^2} f^*(\mathbf{y}|\mathbf{x})\, d\mathbf{y} \approx \mathcal{S} := \frac{1}{2^{m+2}} \Bigg[& D^*_{m,l_1,l_2} + D^*_{m,u_1,l_2} + D^*_{m,l_1,u_2} + D^*_{m,u_1,u_2} \\ & + 2\sum_{i=1}^{K_1-1} \left(D^*_{m,l_1+i,l_2} + D^*_{m,l_1+i,u_2}\right) + 2\sum_{j=1}^{K_2-1} \left(D^*_{m,l_1,l_2+j} + D^*_{m,u_1,l_2+j}\right) \\ & + 4\sum_{j=1}^{K_2-1} \left(\sum_{i=1}^{K_1-1} D^*_{m,l_1+i,l_2+j}\right)\Bigg],\end{aligned} \tag{4.23}$$

with $K_i = u_i - l_i + 1$. For sake of simplicity, we will consider $a = \min(a_1, a_2)$ and $b = \max(b_1, b_2)$, and work with a, b instead of a_1, a_2, b_1 and b_2, being this selection more conservative.

Finally, the selection of J is related to the error studied in Theorem 3. Although a different J can be selected for each pair (k_1, k_2), we prefer to consider a constant J, defined here by $J = \lceil \log_2(\pi \max_{k_1,k_2} M_{m,k_1,k_2}) \rceil$ where $\lceil \cdot \rceil$ stands for the ceiling function. The reason is that, in practice, the computationally most involved part in (3.5) is the evaluation of $\hat{f}$ at the grid points. Those values can be computed only once and used in the FFT algorithm mentioned in Remark 2.

[1] The cumulants of a random variable X are the power series coefficients of the cumulant generating function $c(s) = \log \mathbb{E}\left(e^{sX}\right)$.

5 Numerical experiments

We present a wide variety of examples to test the 2D-SWIFT method for pricing European rainbow options. We consider arithmetic basket call options in Section 5.1, spread options in Section 5.2, call-on-max and put-on-min options in Section 5.3, and correlation options in Section 5.4. In order to perform a consistency check, we also consider the pricing of a geometric basket put option as well as the valuation of a spread option with strike equal to zero, for which a closed form solution exists. These two examples are presented in Appendix C.

For simplicity, and without loss of generality, in all examples we assumed $t_0 = 0$. The asset price is modelled by either correlated geometric Brownian motions (GBM) or by Merton's jumps-diffusion (JD) process (but any other Lévy process or process with known characteristic function could be used). The cumulants for these dynamics are well-known and we therefore consider them as our initial guess in (4.22). It is worth remarking that this selection of the interval appears to be accurate in all the examples considered in this work, and we do not need to compute extra coefficients. The programs were coded in MATLAB and run on a Dell Vostro 320 with Intel Core 2 Duo E7500 2.93GHz processor and 4GB RAM.

Under GBM dynamics the risk-neutral asset prices evolve according to the following dynamics,

$$dS_t^i = rS_t^i dt + \sigma_i S_t^i dW_t^i, \quad i = 1, 2, \tag{5.1}$$

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

$$dX_t^i = (r - \frac{1}{2}\sigma_i^2)dt + \sigma_i dW_t^i. \tag{5.2}$$

The log-asset prices at time t given the current state at $t_0 = 0$ are bivariate normally distributed, i.e.,

$$\mathbf{X}_t \sim \mathcal{N}(\mathbf{X}_0 + \boldsymbol{\mu}\Delta t, \boldsymbol{\Sigma}), \tag{5.3}$$

with $\mu_i = r - \frac{1}{2}\sigma_i^2$ and covariance matrix $\Sigma_{ij} = \sigma_i \sigma_j \rho_{ij} \Delta t$. The Fourier transform function (defined in (2.3)) reads $\hat{f}(\mathbf{u}|\mathbf{x}) = e^{-i\mathbf{x}'\mathbf{u}} \hat{f}_{\text{Lévy}}(-\mathbf{u})$, with,

$$\hat{f}_{\text{Lévy}}(\mathbf{u}) = \exp(i\boldsymbol{\mu}'\Delta t\mathbf{u} - \frac{1}{2}\mathbf{u}'\boldsymbol{\Sigma}\mathbf{u}). \tag{5.4}$$

Under a Merton's jump-diffusion process the asset prices follow the dynamics described by the stochastic differential equation,

$$dS_t^i = (r - \lambda\kappa_i)S_t^i dt + \sigma_i S_t^i dW_t^i + J_i S_t^i dq_t, \quad i = 1, 2, \tag{5.5}$$

with $\kappa_i := \mathbb{E}\left[e^{J_i} - 1\right]$, q_t a Poisson process with mean arrival date λ, and $\mathbf{J} = (J_1, J_2)$ bivariate normally distributed jump sizes 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,

$$dX_t^i = (r - \lambda\kappa_i - \frac{1}{2}\sigma_i^2)dt + \sigma_i dW_t^i + J_i dq_t. \tag{5.6}$$

The Fourier transform function is $\hat{f}(\mathbf{u}|\mathbf{x}) = e^{-i\mathbf{x}'\mathbf{u}} \hat{f}_{\text{Lévy}}(-\mathbf{u})$, with,

$$\hat{f}_{\text{Lévy}}(\mathbf{u}) = \exp\left(i\boldsymbol{\mu}'\Delta t\mathbf{u} - \frac{1}{2}\mathbf{u}'\boldsymbol{\Sigma}\mathbf{u}\right) \exp\left(\lambda\Delta t\left(\exp\left(i\boldsymbol{\mu}'^J\mathbf{u}\frac{1}{2}\mathbf{u}'\boldsymbol{\Sigma}^{\boldsymbol{J}}\mathbf{u}\right) - 1\right)\right), \tag{5.7}$$

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

	Arithmetic	Geometric
Call	$g(y_1,y_2)=\left(\frac{1}{2}e^{y_1}+\frac{1}{2}e^{y_2}-K\right)^+$	$g(y_1,y_2)=(\sqrt{e^{y_1}}\sqrt{e^{y_2}}-K)^+$
Put	$g(y_1,y_2)=\left(K-(\frac{1}{2}e^{y_1}+\frac{1}{2}e^{y_2})\right)^+$	$g(y_1,y_2)=(K-\sqrt{e^{y_1}}\sqrt{e^{y_2}})^+$

Table 1: Payoffs for basket options on two underlyings.

5.1 Basket options on two underlyings

We present the pricing of arithmetic basket call options driven by two-dimensional correlated GBM. Table 1 contains the payoff of arithmetic and geometric basket options on two underlyings.

In the arithmetic case, the payoff coefficients (3.9) are given by,

$$G_{m,k_1,k_2}=\iint_{\mathbb{R}^2}\left(\frac{1}{2}e^{y_1}+\frac{1}{2}e^{y_2}-K\right)^+\Phi_{m,k_1,k_2}(\mathbf{y})\,d\mathbf{y}. \tag{5.8}$$

If we truncate the integration range, use the exponential approximation (2.27) of the sinc function and interchange sums with integrals, then $G_{m,k_1,k_2}\approx G^*_{m,k_1,k_2}$, where,

$$G^*_{m,k_1,k_2}:=\frac{2^m}{N^2}\sum_{j_2=1}^{N}\sum_{j_1=1}^{N}e^{-i\Omega_{j_1}k_1}e^{-i\Omega_{j_2}k_2}\int_a^b\int_a^b\left(\frac{1}{2}e^{y_1}+\frac{1}{2}e^{y_2}-K\right)^+e^{i\Omega_{j_1}2^my_1}e^{i\Omega_{j_2}2^my_2}\,dy_1\,dy_2, \tag{5.9}$$

and $\Omega_{j_1}=-\pi+\frac{2j_1-1}{N}\pi, \Omega_{j_2}=-\pi+\frac{2j_2-1}{N}\pi$. The double integral in (5.9) cannot be solved analytically, and we therefore apply numerical integration. We approximate it by a midpoint quadrature with Q terms, obtaining,

$$G^*_{m,k_1,k_2}\approx\frac{2^m\,(b-a)^2}{N^2Q^2}\sum_{j_2=1}^{N}\sum_{j_1=1}^{N}\sum_{l_2=1}^{Q}\sum_{l_1=1}^{Q}\left(\frac{1}{2}e^{\eta_{l_1}}+\frac{1}{2}e^{\eta_{l_2}}-K\right)^+e^{i\Omega_{j_1}2^m\eta_{l_1}}e^{i\Omega_{j_2}2^m\eta_{l_2}}e^{-i\Omega_{j_1}k_1}e^{-i\Omega_{j_2}k_2}, \tag{5.10}$$

where $\eta_{l_i}=a+\frac{b-a}{2Q}(2l_i-1)$. After rearranging terms, we end up with the following formula,

$$\begin{aligned}G^*_{m,k_1,k_2}\approx&\frac{2^m\,(b-a)^2}{N^2Q^2}e^{2i2^m\pi\left(1+\frac{1}{N}\right)\left(\frac{b-a}{2Q}-a\right)}e^{i\pi\left(1+\frac{1}{N}\right)(k_1+k_2)}\sum_{j_2=1}^{N}\sum_{j_1=1}^{N}e^{i2^m\frac{2\pi}{N}\left(a+\frac{b-a}{2Q}\right)(j_1+j_2)}\\&\cdot\sum_{l_2=1}^{M}\sum_{l_1=1}^{M}X(l_1,l_2)e^{\frac{i2\pi}{M}j_1(l_1-1)}e^{\frac{i2\pi}{M}j_2(l_2-1)}e^{\frac{-i2\pi}{N}k_1(j_1-1)}e^{\frac{-i2\pi}{N}k_2(j_2-1)},\end{aligned} \tag{5.11}$$

for $M:=\frac{NQ}{2^m(b-a)}$. In order to ensure that $M\in\mathbb{Z}$, we choose Q of the form $Q=2^n(b-a)$, for $n\geq m$, and,

$$X(l_1,l_2):=\begin{cases}\left(\frac{1}{2}e^{\eta_{l_1}}+\frac{1}{2}e^{\eta_{l_2}}-K\right)^+e^{-\frac{i2^m\pi(b-a)\left(1+\frac{1}{N}\right)(l_1+l_2)}{Q}}, & \text{if } l_1\leq Q \,\text{and}\, l_2\leq Q,\\[2ex] 0, & \text{otherwise.}\end{cases} \tag{5.12}$$

Clearly we can apply a combination of FFT and inverse FFT algorithms to compute the sums efficiently in (5.11). Figure 2 shows the log-scale plots of the errors (left) and CPU time (right), where n is chosen such that $Q=2^n(b-a)$ and m is the level of resolution. We take the parameters and reference price from [11]. We observe that 2D-SWIFT converges exponentially.

5.2 Spread options

Next we price a two-dimensional European spread call option with strike K on two assets driven by a two-dimensional correlated GBM. Because of their generic nature, spread options are used in

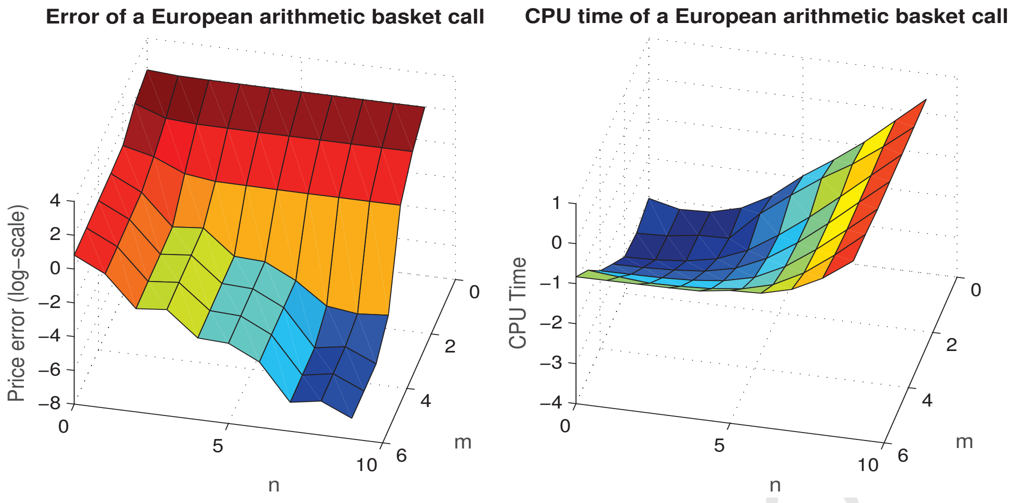

Figure 2: Error (left) and CPU time (right) in seconds of a two-dimensional arithmetic basket call option under the GBM dynamics. The parameter values are $S_0 = (90, 110)$, $r = 0.04$, $\sigma_1 = 0.2$, $\sigma_2 = 0.3$, $\rho = 0.25$, $T = 1$, $K = 100$ and $L = 10$. Reference: 10.173230.

markets like the fixed income markets, the currency and foreign exchange markets, the commodity futures markets and the energy markets. The payoff function for the European call spread is given by,

$$g(y_1, y_2) = \max(e^{y_1} - e^{y_2} - K, 0). \tag{5.13}$$

We distinguish two cases, when $K = 0$ (this case is known as the exchange of assets) the payoff coefficients in (3.9) can be obtained analytically, while for $K > 0$ we use the midpoint quadrature, as presented in the previous example. We give results for $K = 0$ in Appendix C, and the general case with $K > 0$ is considered here.

We report in Figure 3 the error and CPU time for 2D-SWIFT method when the underlying processes follow GBM dynamics. We observe exponential convergence and competitive CPU time for 2D-SWIFT method. We take as reference price the result of the 2D-SWIFT with $m = 12$ and $n = 12$.

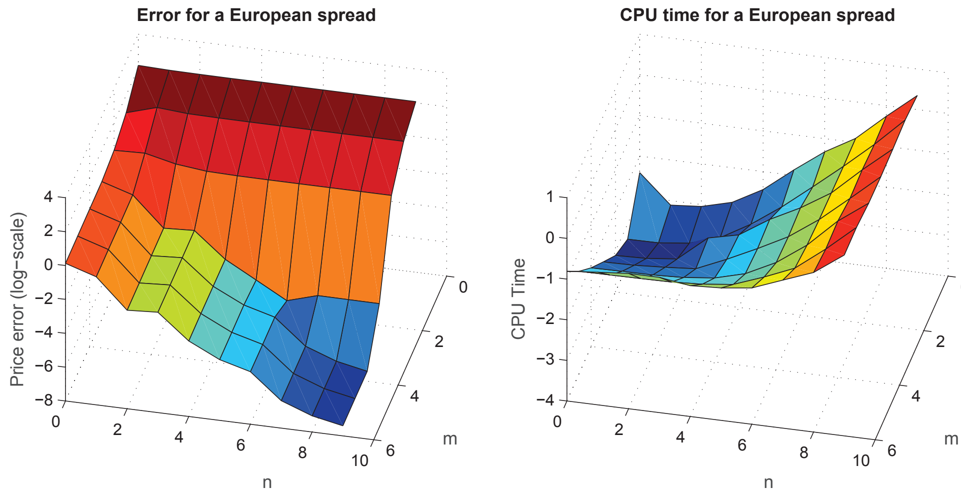

Figure 3: Error (left) and CPU time in seconds (right) corresponding to the pricing of a two-dimensional European call spread under the GBM dynamics. The parameters are $S_0 = (90, 110), r = 0.04, \sigma_1 = 0.2, \sigma_2 = 0.3, \rho = 0.25, K = 20, L = 10$ and $T = 1$. Reference: 1.352591908717933.

5.3 Options on the minimum or the maximum of two risky assets

Here we consider a two-dimensional European option either on the minimum or on the maximum of two assets. Table 2 shows the corresponding payoffs.

	on minimum	on maximum
Call	$g(y_1, y_2) = (\min(e^{y_1}, e^{y_2}) - K)^+$	$g(y_1, y_2) = (\max(e^{y_1}, e^{y_2}) - K)^+$
Put	$g(y_1, y_2) = (K - \min(e^{y_1}, e^{y_2}))^+$	$g(y_1, y_2) = (K - \max(e^{y_1}, e^{y_2}))^+$

Table 2: Payoffs for options on the minimum/maximum of two assets.

For the payoffs described in Table 2, the double integral of the payoff coefficients in (3.9) can be solved analytically. In Figure 4 we present the error of some of these options driven by different dynamics and compare them with 2D-COS method. Again the 2D-SWIFT method converges exponentially.

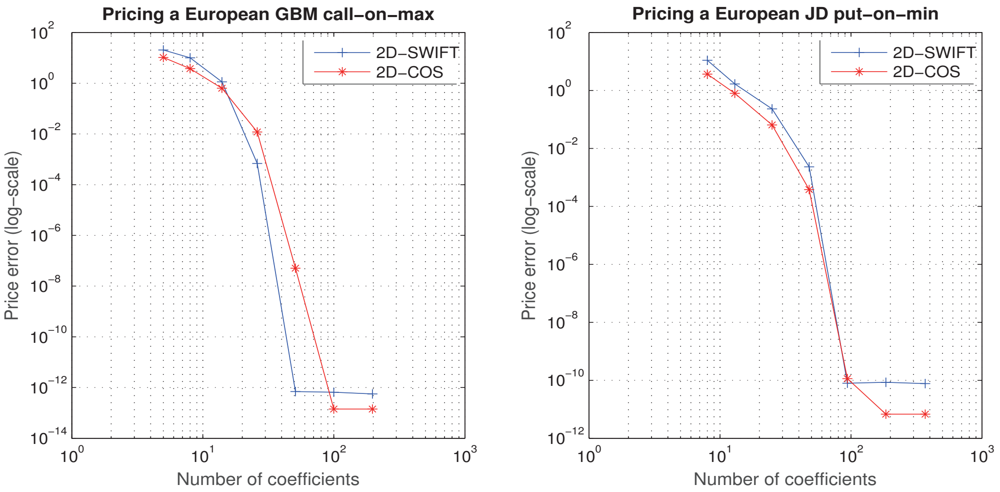

Figure 4: Pricing errors by means of 2D-SWIFT and 2D-COS methods under different dynamics. The left plot corresponds to a call-on-max driven by GBM dynamics, with parameters $S_0 = (40, 40), r = 0.048799, \sigma_1 = 0.2, \sigma_2 = 0.3, \rho = 0.5, K = 40, L = 10$ and $T = 7/12$. The right plot stands for a put-on-min driven by JD dynamics, with parameters $S_0 = (90, 110), r = 0.05, \sigma = (0.12, 0.15), \rho = 0.3, K = 100, \lambda = 0.6, \mu_J = (-0.1, 0.1), \sigma_J = (0.17, 0.13), \rho_J = 0.2, L = 10$ and $T = 1$.

5.4 Correlation options

A correlation option is an extension of the plain vanilla European call to two dimensions. Its payoff is the product of two European calls with different strikes. Similar to spread options, correlation options allow the purchaser to speculate on how asset prices will move together, as the option requires both assets to move in the same direction in order to have at maturity time a non-zero value. The payoff of a correlation option is given by,

$$g(y_1, y_2) = (e^{y_1} - K_1)^+ (e^{y_2} - K_2)^+ . \tag{5.14}$$

The payoff coefficients in this case are computed analytically. We present in Table 3 the relative error when pricing with 2D-SWIFT method under GBM and JD dynamics. The parameters corresponding to the GBM process are $S_0 = (90, 100), r = 0.04, \sigma_1 = 0.2, \sigma_2 = 0.3, \rho = 0.25, T = 1, K_1 = 90, K_2 = 110, L = 10$. The parameters for JD dynamics are the same as GBM with the jump component $\lambda = 0.6, \mu_J = (-0.1, 0.1), \sigma_J = (0.17, 0.13)$ and correlation $\rho_J = -0.2$. The reference value for the GBM case is computed by means of Monte Carlo simulation with one million paths and the 95% confidence interval is given. Regarding the JD dynamics, the reference price is given by 2D-SWIFT with scale of approximation $m = 10$. We select the scale of approximation m by means of formula (4.21) where we set $\epsilon_2 = 1.0e-04$.

Dynamics	m	Relative error
GBM	3	$1.9e-03$
JD	3	$8.5e-07$

Table 3: Reference price for GBM: 204.2355, 95% confidence interval: $[203.0214, 205.4495]$, price given by 2D-SWIFT: 204.6172. Reference price for JD: 212.9888744552966.

With this example we can see how correlation options behave in the presence of jumps. We study the dependence on the parameter λ, i.e. the mean arrival rate in the Merton's jump-diffusion model. We see in Figure 5 the evolution of the option value according to this parameter. The rest of parameters of the model are the same as before. We can clearly observe the increasing value of the option when λ increases its value.

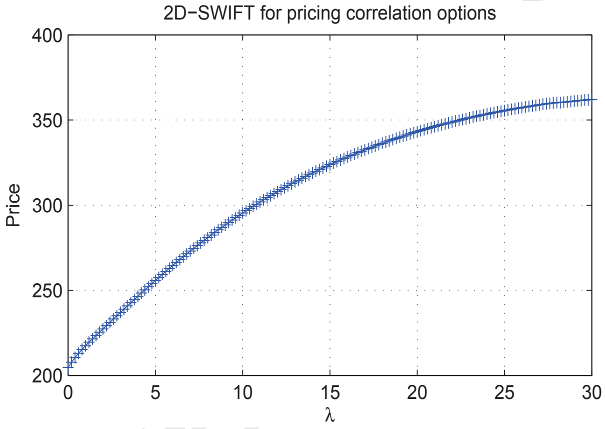

Figure 5: Correlation option prices with respect to the parameter λ under JD dynamics.

6 Strengths of 2D-SWIFT

For the cases shown in the numerical experiments section, the convergence is similar for 2D-SWIFT and 2D-COS. The difference is that 2D-SWIFT is computationally a bit slower than 2D-COS, although it is still very competitive. It is worth mentioning that 2D-SWIFT could be speeded up by running in different threads (i.e. in parallel) the density and payoff coefficients. In this section we present the strengths of 2D-SWIFT with respect to 2D-COS, so that we can appreciate the advantages of using the new method presented in this work.

6.1 Integration range and scale of approximation

The 2D-COS method has a strong dependence on the integration range based on the cumulants in expression (4.22). On the contrary, as explained in Section 4.2, the 2D-SWIFT method considers the integration range as an initial guess, being able to adapt it if necessary.

Once the integration range has been calculated, the accuracy of the 2D-COS method depends on the number of coefficients used in the approximation of the density. If the number of terms in the expansion is not properly chosen, the 2D-COS method does not perform well. The larger the interval the more terms we should consider, although it is not a-priori clear how many coefficients should be used. In regards to the 2D-SWIFT method, the scale of approximation m is a-priori fixed with the help of formula (4.21) and the number of coefficients is determined automatically. In Figure 6 we have fixed the number of terms employed for the 2D-COS approximation and m for 2D-SWIFT, and we have changed the size of the integration range by modifying the parameter L in both methods. As we can observe, the approximation deteriorates for the 2D-COS while it remains very accurate for the 2D-SWIFT method, showing that 2D-SWIFT is not sensitive with respect to this parameter.

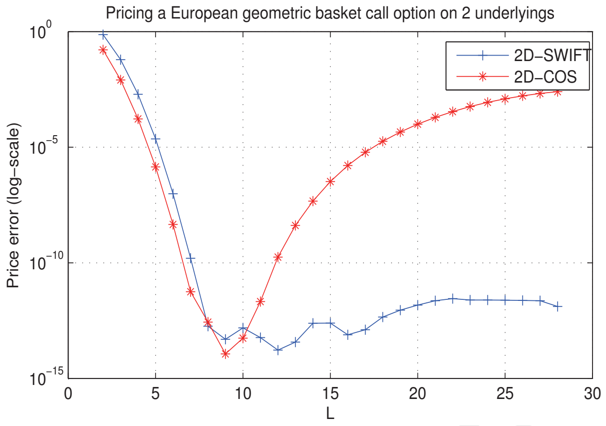

Figure 6: Absolute errors corresponding to the 2D-COS (red) and 2D-SWIFT (blue).

6.2 Behaviour for extreme maturities

Small maturities

Small maturity options are important in high-frequency trading, also short term binary options are well-known in the markets. The density function for small maturities is highly peaked. Thus, the characteristic function is very smooth with fat tails as we see in Figure 7.

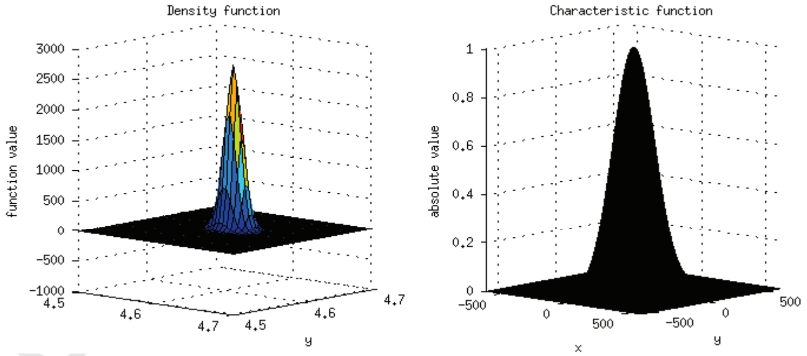

Figure 7: Density (left) and characteristic function (right) for GBM dynamics when $T = 0.001$.

In these situations, the length of the interval goes to zero, and the scale m increases when T tends to 0, because of the shape of the density. It can be seen numerically that the interval length goes to zero quicker than the scale goes to infinity. Thus, the number of coefficients needed by the 2D-SWIFT method tends to be very low, providing advantage in the use of the method for high dimensions. For example, when pricing with 2D-SWIFT a geometric call option on two assets driven by GBM dynamics with parameters $S_0 = (100, 100), r = 0.1, \sigma_1 = 0.2, \sigma_2 = 0.3, \rho = 0.2, T = 0.001$ and $K = 100$, using a scale of $m = 6$ with just 6 coefficients we obtain an error of 3.32e−03.

Long maturities

Long maturity options are present in insurance markets. For example in [8], insurance contracts of a call spread up to 50 years are considered. Moreover, in recent years the long-dated FX option's market has grown considerably. Currently, some traded and liquid long-dated FX hybrid products are Power-Reverse Dual-Currency swaps (PRDC) as well as vanilla or exotic long-dated products such as barrier options. In the case of large maturities, the density function has fat tails and the respective characteristic function is very peaked as shown in Figure 8.

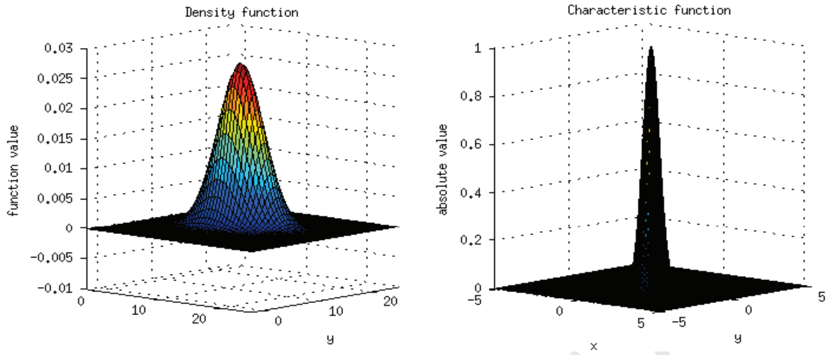

Figure 8: Density (left) and characteristic function (right) for GBM dynamics when $T = 100$.

Thus, the interval length increases and the scale value decreases when T takes large values. As we show in Figure 9, the choice of L when working with large maturities is again a problem. If L is set around 10 as suggested in [11] for moderate maturities, then the results are not accurate.

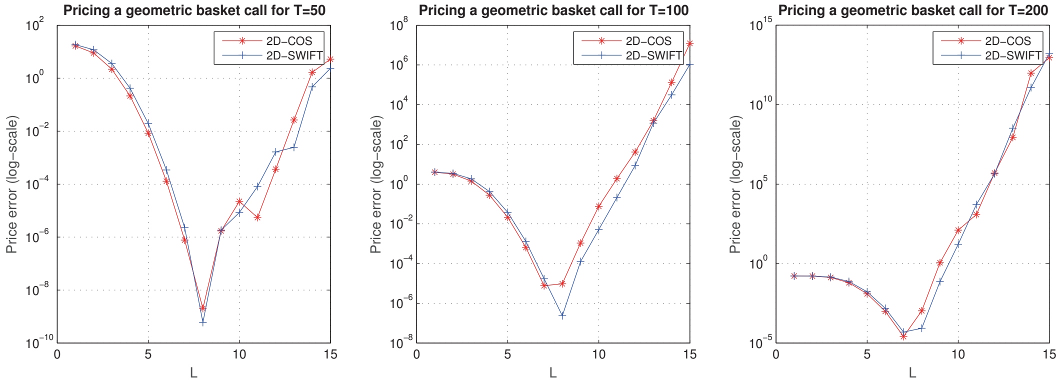

Figure 9: Absolute errors when pricing by means of the 2D-COS (with 200 terms) and 2D-SWIFT methods (at scale $m = 1$) a European geometric basket call option on two underlyings with parameters $S_0 = (100, 100), r = 0.5, \sigma = (0.4, 0.4), \rho = 0.2, K = 90$. The reference price is 20.189651798215621.

As studied in [9], when dealing with long maturities roundoff errors appear for unbounded payoff options. We see in Figure 10 the behaviour of the payoff functions that we consider and the domains where they can grow rapidly. The payoff functions for put options are bounded. However, for calls and spreads roundoff errors may appear. Due to the local nature of Shannon wavelets, we can remove part of the sum in the final pricing formula (3.8) to avoid roundoff errors by eliminating some coefficients. We note that this is possible because each coefficient is only relevant for a very small interval within the integration range, while in the case of the 2D-COS method, all coefficients intervene in the approximation along the whole integration range. Table 4 confirms the quality of the 2D-SWIFT method. We use 50 (respectively 100) terms in the 2D-COS expansion, and the same number of coefficients for 2D-SWIFT, corresponding to the scale $m = 0$ (respectively

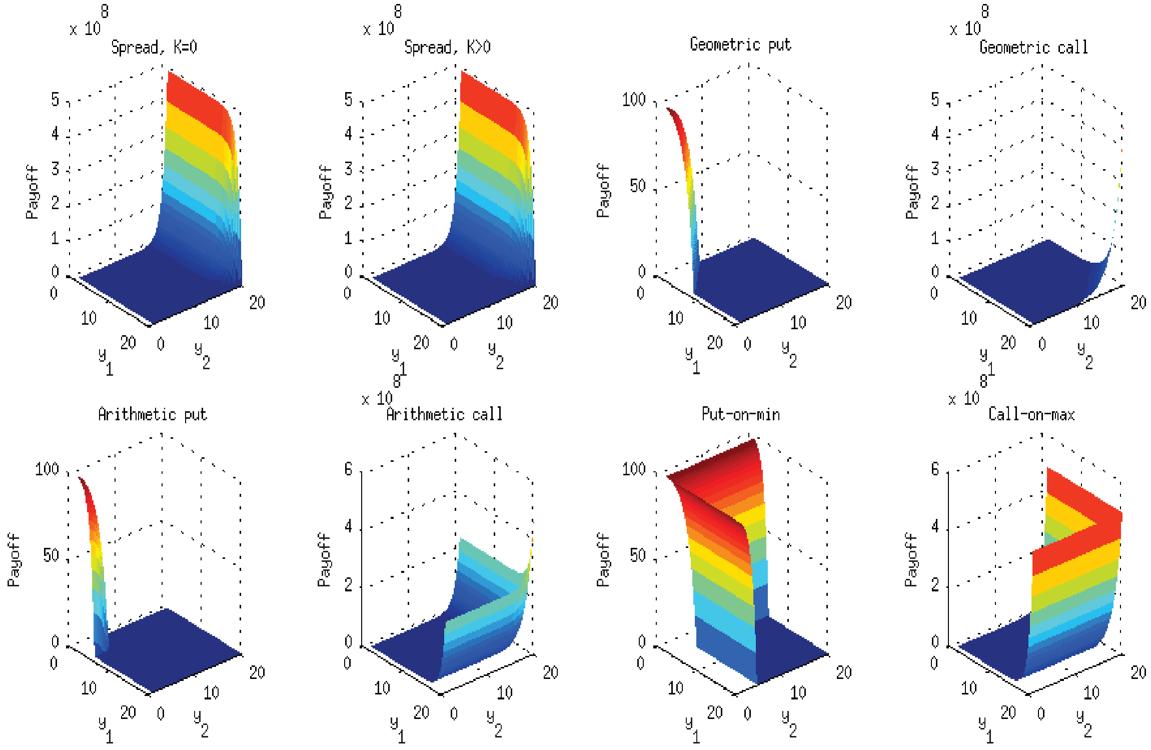

Figure 10: Some two-dimensional payoff functions.

$m = 1$). Although a direct implementation of both methods gives inaccurate results, 2D-SWIFT performs much better when we remove some terms at the boundaries.

Method	Error ($m = 0$)	Range	Error ($m = 1$)	Range
2D-COS	1.18e+01	–	4.48e−07	–
2D-SWIFT	1.59e+01	$-14 \leq k_1, k_2 \leq 37$	5.99e−07	$-28 \leq k_1, k_2 \leq 73$
2D-SWIFT (terms removed)	5.40e−02	$-14 \leq k_1, k_2 \leq 24$	4.60e−07	$-28 \leq k_1, k_2 \leq 65$

Table 4: Absolute errors when pricing a geometric basket call option with parameters $S_0 = (100, 100), r = 0.1, \sigma = (0.25, 0.25), \rho = 0.8, K = 120, T = 100, L = 10$. Reference price is 73.156120362425582.

7 Conclusions

In this paper we have presented the multidimensional SWIFT method motivated by multidimensional option pricing of European rainbow options.

First we presented the multiresolution analysis framework in two and higher dimensions for separable spaces as well as the Shannon wavelets representation, which theoretically supports our method. We also presented pricing formulas for European options in two and higher dimensions using a more convenient approximation than the one previously followed in the literature for the cardinal sine function. We give a complete error analysis of the new method and provide a prescription on how to select the parameters appearing in the method according to the precision required.

We tested with a wide variety of numerical examples the efficiency of the method in the two-dimensional case for different kinds of European rainbow options of assets driven by different dynamics like GBM or JD. Basket options, spread options, options on the minimum or the maximum of two risky assets and correlation options are considered. We compared 2D-SWIFT results with the state-of-the-art 2D-COS method, with closed-form solutions when available, with Mote Carlo simulation or with 2D-SWIFT with a large scale of approximation. Finally, we presented the strengths of the 2D-SWIFT machinery, which includes the domain truncation issue, the calculation of the scale of approximation and the number of coefficients used as well as the results when dealing with extreme maturities. As it has been mentioned, extending the method to large dimensions, more than 4 or 5 depending on the specific kind of product, will not be useful because the curse of dimensionality appears.

We have shown that 2D-SWIFT inherits the strengths of the one-dimensional technique presented in [10] for European-style options, like for instance, the a-priori knowledge of the approximation scale. Therefore, this fact opens the door to future applications of 2D-SWIFT to other financial contracts in two dimensions. For instance, previous work in [5] shows the valuation of one-dimensional barrier and early-exercise options using SWIFT, addressing important issues like boundary errors within the recursion backwards in time. We expect similar benefits in two dimensions as well. Further, either multidimensional path dependent options pricing or Heston's stochastic volatility dynamics seem plausible to be considered, although these challenging topics will be treated in future work.

Acknowledgements

The research leading to these results has received funding from La Caixa Foundation. G. C.-P. acknowledges AGAUR-Generalitat de Catalunya for funding under its doctoral scholarship programme. G. C.-P. and L. O.-G. acknowledge the Spanish Ministry of Economy and Competitiveness (MINECO) for funding under grant MTM2013-40782-P. Finally, G. C.-P. wishes to thank S.C. Maree for helpful discussions.

Appendix A. Proof of Lemma 1

From (3.1), (3.2) and (4.5) we write,

$$\epsilon_p = \left| f(\mathbf{y}|\mathbf{x}) - \mathcal{P}_m f(\mathbf{y}|\mathbf{x}) \right|, \tag{7.1}$$

where,

$$\mathcal{P}_m f(\mathbf{y}|\mathbf{x}) = \sum_{k_1 \in \mathbb{Z}} \sum_{k_2 \in \mathbb{Z}} D_{m,k_1,k_2}(\mathbf{x}) \Phi_{m,k_1,k_2}(\mathbf{y}), \tag{7.2}$$

and,

$$D_{m,k_1,k_2}(\mathbf{x}) := \iint_{\mathbb{R}^2} f(\mathbf{y}|\mathbf{x}) \Phi_{m,k_1,k_2}(\mathbf{y})\, d\mathbf{y}. \tag{7.3}$$

By Parseval's identity,

$$D_{m,k_1,k_2}(\mathbf{x}) = \frac{1}{4\pi^2} \iint_{\mathbb{R}^2} \hat{f}(\boldsymbol{\omega}) \hat{\Phi}_{m,k_1,k_2}(\boldsymbol{\omega}) d\boldsymbol{\omega}, \tag{7.4}$$

where $\boldsymbol{\omega} = (\omega_1, \omega_2)$ and $\hat{f}$ and $\hat{\Phi}_{m,k_1,k_2}$ denote the Fourier transform of f and Φ_{m,k_1,k_2} respectively. We observe that by (2.19) we can write,

$$\hat{\Phi}_{m,k_1,k_2} = \iint_{\mathbb{R}^2} \Phi_{m,k_1,k_2}(\mathbf{y}) e^{-i(w_1 y_1 + w_2 y_2)}\, d\mathbf{y} = \hat{\phi}_{m,k_1}(w_1) \hat{\phi}_{m,k_2}(w_2), \tag{7.5}$$

where, as pointed out in [10],

$$\hat{\phi}_{m,k}(w) = \frac{e^{-i\frac{k}{2^m}w}}{2^{m/2}} \operatorname{rect}\left(\frac{w}{2^{m+1}\pi}\right), \tag{7.6}$$

and rect is the rectangle function, defined as,

$$\operatorname{rect}(x) = \begin{cases} 1, & \text{if } |x| < 1/2, \\ 1/2, & \text{if } |x| = 1/2, \\ 0, & \text{if } |x| > 1/2. \end{cases} \tag{7.7}$$

Now by (7.4), (7.5) and (7.6),

$$D_{m,k_1,k_2}(\mathbf{x}) = \frac{1}{4\pi^2}\frac{1}{2^m}\int\int_{\mathcal{C}} \hat{f}(\boldsymbol{\omega}) e^{i\frac{k_1}{2^m}w_1} e^{i\frac{k_2}{2^m}w_2} d\boldsymbol{\omega}, \tag{7.8}$$

where $\mathcal{C} := [-2^m\pi, 2^m\pi] \times [-2^m\pi, 2^m\pi]$. If we replace this last expression of coefficients in (7.2) and interchange the summation and integration we get,

$$\mathcal{P}_m f(\mathbf{y}|\mathbf{x}) = \frac{1}{4\pi^2}\frac{1}{2^m}\int\int_{\mathcal{C}} \hat{f}(\boldsymbol{\omega}) \left[\sum_{k_1 \in \mathbb{Z}} \sum_{k_2 \in \mathbb{Z}} \Phi_{m,k_1,k_2}(\mathbf{y}) e^{i\frac{k_1}{2^m}w_1} e^{i\frac{k_2}{2^m}w_2} \right] d\boldsymbol{\omega}, \tag{7.9}$$

It we have into account that by Theorem 1.2.1 of [12],

$$\sum_{k \in \mathbb{Z}} \phi_{m,k}(y) e^{i\frac{k}{2^m}w} = 2^{\frac{m}{2}} e^{iwy}, \quad \text{when} \quad w \in (-2^m\pi, 2^m\pi), \tag{7.10}$$

then the projection $\mathcal{P}_m f$ can be written as,

$$\mathcal{P}_m f(\mathbf{y}|\mathbf{x}) = \frac{1}{4\pi^2}\int\int_{\mathcal{C}} \hat{f}(\boldsymbol{\omega}) e^{i\boldsymbol{\omega}'\mathbf{y}} d\boldsymbol{\omega}. \tag{7.11}$$

Finally, by (7.11) and the definition of the inverse Fourier transform[2] of f, we have,

$$\epsilon_p = |f(\mathbf{y}|\mathbf{x}) - \mathcal{P}_m f(\mathbf{y}|\mathbf{x})| = \frac{1}{4\pi^2}\left| \int\int_{\mathbb{R}^2 \setminus \mathcal{C}} \hat{f}(\boldsymbol{\omega}) e^{i\boldsymbol{\omega}'\mathbf{y}} d\boldsymbol{\omega} \right| \leq \frac{1}{4\pi^2}\int\int_{\mathbb{R}^2 \setminus \mathcal{C}} |\hat{f}(\boldsymbol{\omega})| d\boldsymbol{\omega}, \tag{7.13}$$

and this concludes the proof.

Appendix B. Proof of Theorem 3

From (2.19), (3.2), (3.4), (3.5) and having into account the equivalence (2.30), we write,

$$\begin{aligned}
&\left|D_{m,k_1,k_2}(\mathbf{x}) - D^*_{m,k_1,k_2}(\mathbf{x})\right| \\
&= 2^m \left| \iint_{\mathbb{R}^2} f(\mathbf{y}|\mathbf{x}) \left[\operatorname{sinc}(2^m y_1 - k_1)\operatorname{sinc}(2^m y_2 - k_2) \,-\, \operatorname{sinc}^*(2^m y_1 - k_1)\operatorname{sinc}^*(2^m y_2 - k_2)\right] d\mathbf{y} \right| \\
&\leq 2^m \Bigg[\int\int_{\mathcal{D}^c} f(\mathbf{y}|\mathbf{x}) \left|\operatorname{sinc}(2^m y_1 - k_1)\operatorname{sinc}(2^m y_2 - k_2) \,-\, \operatorname{sinc}^*(2^m y_1 - k_1)\operatorname{sinc}^*(2^m y_2 - k_2)\right| d\mathbf{y} \\
&+ \left| \int\int_{\mathcal{D}} f(\mathbf{y}|\mathbf{x}) \left(\operatorname{sinc}(2^m y_1 - k_1)\operatorname{sinc}(2^m y_2 - k_2) \,-\operatorname{sinc}^*(2^m y_1 - k_1)\operatorname{sinc}^*(2^m y_2 - k_2)\right) d\mathbf{y} \right| \Bigg].
\end{aligned} \tag{7.14}$$

where $\mathcal{D} := [-c_1, c_1] \cup [-c_2, c_2]$ and $\mathcal{D}^c := \mathbb{R}^2 \setminus \mathcal{D}$.

[2] The inverse Fourier transform f of $\hat{f}$ is by definition,

$$f(\mathbf{y}|\mathbf{x}) = \frac{1}{4\pi^2}\iint_{\mathbb{R}^2} \hat{f}(\boldsymbol{\omega}) e^{i\boldsymbol{\omega}'\mathbf{y}} d\boldsymbol{\omega}. \tag{7.12}$$

Since the mass in the tails of the density f tends to zero when c_1 and c_2 tend to infinity, for all $\epsilon > 0$ there exist $c_1, c_2 > 0$ such that $H(c_1, c_2) < \epsilon$. Further,

$$|\mathrm{sinc}(2^m y_1 - k_1)\,\mathrm{sinc}(2^m y_2 - k_2) - \mathrm{sinc}^*(2^m y_1 - k_1)\,\mathrm{sinc}^*(2^m y_2 - k_2)| \leq 2,$$

for all $\mathbf{y} \in \mathbb{R}^2$, and therefore the first integral at the right hand-side of inequality (7.14) is bounded by 2ϵ.

Now if we apply the Cauchy-Schwarz inequality to the second integral then,

$$\begin{aligned}
&\left| \int\int_{\mathcal{D}} f(\mathbf{y}|\mathbf{x}) \left(\mathrm{sinc}(2^m y_1 - k_1)\,\mathrm{sinc}(2^m y_2 - k_2) - \mathrm{sinc}^*(2^m y_1 - k_1)\,\mathrm{sinc}^*(2^m y_2 - k_2)\right) d\mathbf{y} \right| \\
&\leq \|f(\cdot|\mathbf{x})\|_2 \left(\int\int_{\mathcal{D}} \left(\bar{\mathcal{E}}_V(2^m y_1 - k_1, 2^m y_2 - k_2)\right)^2 d\mathbf{y} \right)^{\frac{1}{2}} \qquad (7.15) \\
&\leq \|f(\cdot|\mathbf{x})\|_2 \left(\int\int_{\mathcal{D}} \left(\mathcal{E}_V(2^m y_1 - k_1) + \mathcal{E}_V(2^m y_2 - k_2)\right)^2 d\mathbf{y} \right)^{\frac{1}{2}},
\end{aligned}$$

where the last inequality is satisfied by Lemma 3. We observe that if $-c \leq y_i \leq c$, then $-2^m c - k_i \leq 2^m y_i - k_i \leq 2^m c - k_i$ and therefore, $2^m y_i - k_i \in [-M^i_{m,k_i}, M^i_{m,k_i}]$, where by definition $c = \max(c_1, c_2)$ and $i = 1, 2$. We note that,

$$[-M^i_{m,k_i}, M^i_{m,k_i}] \subset [-M_{m,k_1,k_2}, M_{m,k_1,k_2}], \quad i = 1, 2,$$

then by Lemma 2, the integral at the right hand-side of the second inequality in (7.15) is bounded by,

$$2c \cdot \frac{2(\pi M_{m,k_1,k_2})^2}{2^{2(J+1)} - (\pi M_{m,k_1,k_2})^2}, \qquad (7.16)$$

when $J \geq \log_2 (\pi M_{m,k_1,k_2})$.

Finally, by (7.14), (7.15) and (7.16) we end up with the error estimate,

$$|D_{m,k_1,k_2}(\mathbf{x}) - D^*_{m,k_1,k_2}(\mathbf{x})| \leq 2^m \left(2\epsilon + 4c\|f(\,\cdot \mid \mathbf{x})\|_2 \frac{(\pi M_{m,k_1,k_2})^2}{2^{2(J+1)} - (\pi M_{m,k_1,k_2})^2} \right). \qquad (7.17)$$

Appendix C. Other numerical examples

Geometric basket put option

It is worth mentioning that the price of a geometric basket option under GBM equals the Black-Scholes price of the corresponding European option with initial price $\hat{S}_0 = \sqrt{S^1_0}\sqrt{S^2_0}$, volatility $\hat{\sigma}$ and dividend rate $\hat{\delta}$, where,

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

So, we can perform a consistency check and compare the results of our method with the analytical option values.

Here, we price a geometric put option driven by two-dimensional correlated GBM. When pricing any option with 2D-SWIFT we need first to compute payoff coefficients, it can be done analogously as shown in the numerical experiments section. The payoff coefficients in this kind of options can be obtained analytically.

In Figure 11, we compare the error and the CPU time (expressed in seconds) between 2D-COS and 2D-SWIFT methods when using the same number of coefficients. We observe exponential convergence of the 2D-SWIFT method as well as 2D-COS in the left-side graph, and that the CPU time for 2D-SWIFT is higher than 2D-COS, although 2D-SWIFT is a very competitive method.

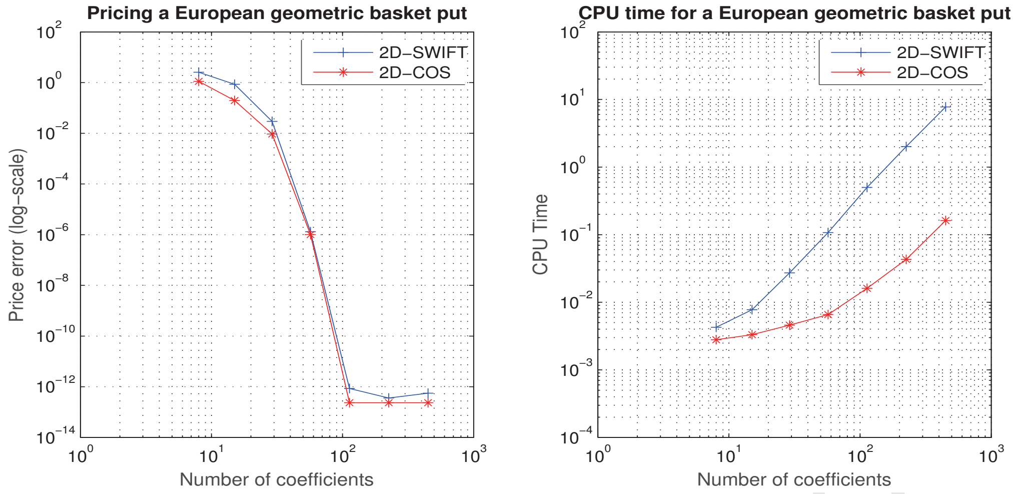

Figure 11: Log-error and CPU time in seconds of a two-dimensional geometric basket put option under the GBM dynamics. The parameter values are $S_0 = (90, 110), r = 0.04, \sigma_1 = 0.2, \sigma_2 = 0.3, \rho = 0.25, K = 100, L = 10$ and $T = 1$.

Depending on the type of option, the double integral in (5.8) can not be solved analytically, and we therefore need to apply a numerical quadrature. In this example we also solve the integral numerically in order to have an insight on the behaviour of the error according to the number of terms used in the integral approximation.

In Figure 12 the surfaces of the errors (left) and CPU time (right) are shown, where n is chosen such that $Q = 2^n(b - a)$ (recall that Q is the number of points in the midpoint quadrature for the double integral), and m is the level of resolution. The parameter values are the same as in Figure 11, and we present the absolute errors in Table 5.

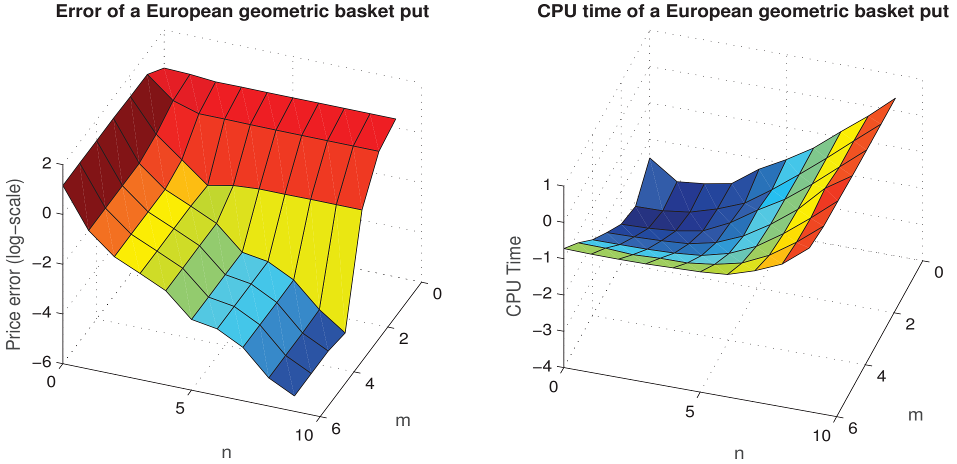

Figure 12: Error and CPU time in seconds (both measured in log-scale) of a two-dimensional geometric basket put option under the GBM dynamics.

From Figure 12 and Table 5 we observe the convergence of the method with respect to the scale of approximation m.

		m						
		0	1	2	3	4	5	6
	1	5.53e−01	2.09e+00	7.12e−01	3.68e−01	3.68e−01	3.68e−01	3.68e−01
	2	8.71e−01	1.05e+00	1.09e−01	5.34e−02	5.34e−02	5.34e−02	5.34e−02
	3	7.92e−01	1.04e+00	1.68e−02	1.84e−02	1.84e−02	1.84e−02	1.84e−02
	4	7.95e−01	1.01e+00	2.79e−02	6.14e−03	6.14e−03	6.14e−03	6.14e−03
n	5	7.98e−01	9.97e−01	3.26e−02	6.95e−04	6.96e−04	6.96e−04	6.96e−04
	6	7.98e−01	9.96e−01	3.27e−02	4.77e−04	4.79e−04	4.79e−04	4.79e−04
	7	7.98e−01	9.95e−01	3.33e−02	1.41e−04	1.40e−04	1.40e−04	1.40e−04
	8	7.98e−01	9.95e−01	3.31e−02	1.46e−05	1.33e−05	1.33e−05	1.33e−05
	9	7.98e−01	9.95e−01	3.31e−02	2.58e−06	4.12e−06	4.12e−06	4.12e−06

Table 5: Absolut errors for a two-dimensional geometric basket put option under GBM dynamics. The reference price is 6.696961159991261.

Exchange of assets

Exchange of assets is the name given to spread options when $K = 0$. If we assume that the processes follow GBM dynamics, we can use the Margrabe formula [6] as the reference price to compare the error and the CPU time between 2D-COS and 2D-SWIFT methods. We use the parameters from problem number 6 of the BENCHOP project [13]. The results are shown in Figure 13, where we observe again exponential convergence and competitive CPU time for 2D-SWIFT method.

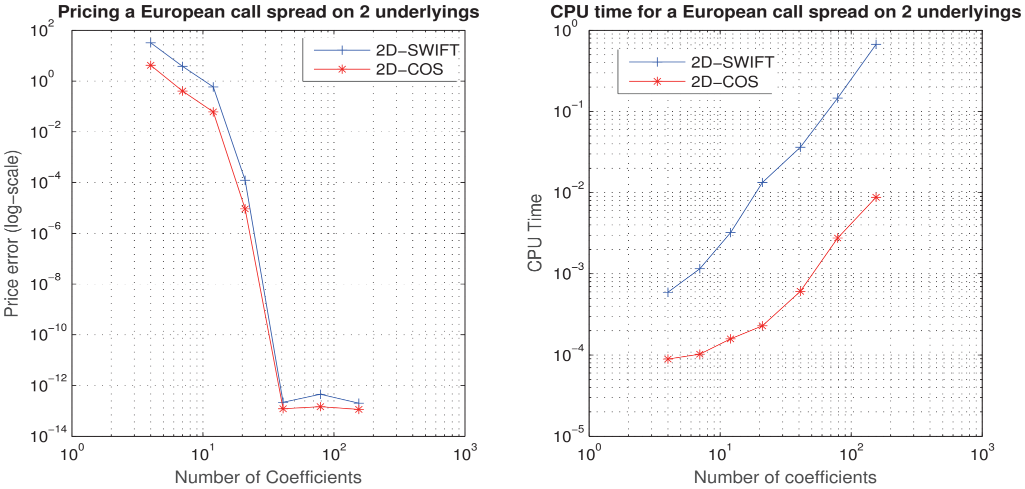

Figure 13: Error (left) and CPU time in seconds (right) corresponding to the pricing of a two-dimensional European call spread under the GBM dynamics. The parameters are $S_0 = (100, 90), r = 0.03, \sigma_1 = \sigma_2 = 0.15, \rho = 0.5, K = 0, L = 10$ and $T = 1$.

References

[1] I. Daubechies. *Ten lectures on wavelets.* Society for Industrial and Applied Mathematics, 1992.

[2] F. Fang and C. W. Oosterlee. A novel option pricing method based on Fourier-cosine series expansions. *SIAM J. Sci. Comput.*, 31(2):826–848, 2008.

[3] J. L. Kirkby. Efficient option pricing by frame duality with the fast Fourier transform. *SIAM J. on Financial Mathematics*, 6(1):713–747, 2015.

[4] S. Mallat. *A wavelet tour of signal processing.* Academic Press, 2009.

[5] S. C. Maree, L. Ortiz-Gracia, and C. W. Oosterlee. Pricing early-exercise and discrete barrier options by Shannon wavelet expansions. *To appear in Numerische Mathematik*, 2016.

[6] W. Margrabe. The value of an option to exchange one asset for another. *Journal of Finance*, 33:177–186, 1978.

[7] Y. Meyer. *Wavelets and operators*. Cambridge University Press, 1993.

[8] T. Moller. On valuation and risk management at the interface of insurance and finance. *British Actuarial Journal*, 8(IV):787–827, 2002.

[9] L. Ortiz-Gracia and C. W. Oosterlee. Robust pricing of European options with wavelets and the characteristic function. *SIAM J. Sci. Comput.*, 35(5):B1055–B1084, 2013.

[10] L. Ortiz-Gracia and C. W. Oosterlee. A highly efficient Shannon wavelet inverse Fourier technique for pricing European options. *SIAM J. Sci. Comput.*, 38(1):B18–B143, 2016.

[11] M. J. Ruiter and C. W. Oosterlee. Two-dimensional Fourier cosine series expansion method for pricing financial options. *SIAM J. on Scientific Computing*, 34(5):B642–B671, 2012.

[12] F. Stenger. *Handbook of Sinc numerical methods*. Chapman & Hall/CRC Numerical Analysis and Scientific Computing. CRC Press, Boca Raton, FL, 2011.

[13] L. von Sydow et al. Benchop - the benchmarking project in option pricing. *International Journal of Computer Mathematics*, 92:2361 – 2379, 2015.

G. Colldeforns-Papiol
Centre de Recerca Matemàtica
Campus de Bellaterra, Edifici C
08193 Bellaterra (Barcelona)
Spain
and
Departament de Matemàtiques
Universitat Autònoma de Barcelona
08193 Bellaterra (Barcelona)
Spain
E-mail address: gcolldeforns@crm.cat, gemma.colldeforns@gmail.com

L. Ortiz-Gracia
Department of Econometrics
University of Barcelona
Diagonal 690
08034 Barcelona, Spain
E-mail address: luis.ortiz-gracia@ub.edu

C. W. Oosterlee
CWI – Centrum Wiskunde & Informatica
NL-1090 GB Amsterdam
The Netherlands
and
Delft University of Technology
Delft Institute of Applied Mathematics
2628 CD Delft
The Netherlands
E-mail address: c.w.oosterlee@cwi.nl