



The Two Dimensional COS Method for Pricing Early-Exercise and Discrete Barrier Options Under the Heston Model

Author:
Richard Moir

Supervisor:
Prof. R. Becker

May 21, 2014

Minor Dissertation presented in partial fulfilment of the requirements for the degree of
Master of Philosophy in Mathematical Finance

Abstract

We focus on the pricing of Bermudan and barrier options under the dynamics of the Heston stochastic volatility model. The two-dimensional nature of the Heston model makes the pricing of these options problematic, as the risk-neutral expectations need to be calculated at each exercise/observation date along a continuum of the two state spaces. We examine the 2D-COS method, which makes use of Fourier-cosine expansions in each of the two dimensions in order to approximate the integrals. Using the fast Fourier transform, we are able to efficiently calculate the cosine series coefficients at each exercise/observation date. A construction of this method is provided and we conduct numerical experiments to evaluate its speed and accuracy.

The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

Acknowledgements

I owe many thanks to my supervisor, Professor Ronald Becker, for stepping in to supervise at a time when I really needed it. I am very grateful for the help which he provided.

Plagiarism declaration

- I know that plagiarism is wrong. Plagiarism is to use another's work and to pretend that it is one's own.
- Each significant contribution to, and quotation in, this report from the work of other people has been attributed, and has been cited and referenced.
- This dissertation is my own work.
- I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as his or her own work.

Signature:

Contents

1	Introduction	1
2	The Heston Model	3
2.1	Distribution of the variance process	3
2.2	Characteristic Function	4
2.2.1	The Bivariate ChF	4
2.2.2	Cumulants of the log-stock Process	7
3	The Two-Dimensional COS Method for Pricing European Options	8
3.1	Evaluating the V_{k_1, k_2} terms	11
3.2	Truncation of the computational domain	14
4	The Two-Dimensional COS Method for Pricing Bermudan Options	15
4.1	Evaluating the V_{k_1, k_2} terms	16
4.1.1	Finding the G_{k_1, k_2} terms	17
4.1.2	Finding the C_{k_1, k_2} terms	17
4.1.3	Calculating the H^\pm terms	20
4.1.4	The early-exercise and continuation regions	21
4.1.5	Finding the early-exercise points, y_m^*	22
4.1.6	Finding $\mathcal{M}_{k_1, j_1}(x_1, x_2, a_1, b_1)$	23
4.2	Use of the Fast Fourier Transform	26
4.2.1	Calculation of \hat{C} using convolutions	27
4.2.2	Reducing the computational complexity	29
4.3	Algorithm for pricing Bermudan options	31
5	Discrete Barrier Options	33
5.1	Evaluating the V_{k_1, k_2} terms	34
5.2	Finding the C_{k_1, k_2} terms	36
5.3	Algorithm for pricing barrier options	38
6	Numerical Results	39
6.1	European Options	39
6.1.1	European Options within the Bermudan Framework	40
6.2	Bermudan Options	40
6.3	Barrier Options	43
7	Conclusion	46
A	A Basic Outline of Fourier Analysis	47
A.1	The Fourier Transform Pair	47
A.2	Characteristic Functions	47

B Results Used in the Development of the 2D-COS Method 48
 B.1 Representing Hankel matrix-vector products in terms of a circular convolution 48
 B.2 Verification of the Discrete Fourier Transform’s Shifting Property 50

References 52

1 Introduction

In financial markets, a considerable amount of effort has been spent on improving the speed and accuracy of option pricing techniques. One of the main motivations for this is that efficient methods are required for the calibration of financial models as well as to price complex contracts rapidly.

As stated by Fang and Oosterlee [2008], the existing numerical pricing methods fall into three major groups: partial (integro) differential equation (PIDE) methods, Monte Carlo simulation and numerical integration methods.

The current state-of-the-art numerical integration methods are based on a transformation to the Fourier domain, and are thus collectively known as the *transform methods*. These methods are computationally very efficient as a result of the availability of the Fast Fourier Transform (FFT). In the Fourier domain, a wide range of derivative contracts can be priced, provided that the characteristic function of the asset price process is available. This is the case for a number of underlying models, such as those in the class of Lévy processes.

A number of transform methods have been developed to price European options for a variety of asset price processes. Examples of such methods are those of Carr and Madan [1999] and Fang and Oosterlee [2008]. More recently, transform methods have been generalized in order to price more complicated options, such as Bermudan, American and barrier options. Examples of these methods are Lord et al. [2008] and Fang and Oosterlee [2009].

Although these techniques for option pricing using the FFT are very fast and accurate, they are based on the one-dimensional characteristic function of a single stochastic process. This means that they are not suitable for the pricing of options with multiple underlying assets, such as rainbow options, or for the pricing of options written on one asset which is governed by two or more stochastic processes, as is the case for models such as the stochastic volatility model of Heston [1993]. Because of the multi-dimensional nature of the dynamics being considered, it is non-trivial to apply the transform methods to the pricing of Bermudan and barrier options in such situations.

As a result of both the multi-dimensionality and the path dependency of such options, Monte Carlo simulation methods are often used to price these contracts, and hence much of the recent advances in this area of study have been obtained for these simulation methods. In Fang and Oosterlee [2011], the pricing of Bermudan and barrier options under the Heston model by transform methods was considered and is handled by applying the COS formula for the log-stock dimension and a quadrature rule for the log-variance dimension.

In this paper, we consider a two-dimensional FFT-based method for the pricing of options under the Heston model. The 2D-COS method of Ruijter and Oosterlee [2012] is an extension of the COS method of Fang and Oosterlee [2009] to two dimensions, and thus is also based on Fourier-cosine series expansions. This method differs from that used in Fang and Oosterlee

[2011] in that Fourier-cosine series expansions are applied in both dimensions, rather than just in the log-stock dimension. As a result, this approach requires the availability of the *bivariate characteristic function* of the log-asset price and the variance. Since the Heston model is in the class of affine jump diffusions (Duffie et al. [2000]), it is fairly straightforward to obtain the bivariate characteristic function in closed form. However, the Heston model is not in the class of Lévy processes, and as a result, the FFT algorithm can only be applied in one dimension, similarly to the cases dealt with in Fang and Oosterlee [2011] and Zhang et al. [2012].

In their paper, Ruijter and Oosterlee [2012] focused predominantly on the pricing of options on multiple underlying assets, but did also cover the pricing of Bermudan options under Heston's model. We will focus exclusively on the pricing of options on a single underlying asset which follows the dynamics of the Heston model and will apply the 2D-COS method to the pricing of European, Bermudan and discrete barrier options.

The dissertation is organised as follows: in Section 2 we outline the basic Heston dynamics and present a few relevant facts about the processes involved, as well as deriving the bivariate characteristic function; in Section 3 we introduce and explain the 2D-COS Method for European options and in Section 4 we progress to the 2D-COS Method for Bermudan options of Ruijter and Oosterlee [2012]; in Section 5 we extend the 2D-COS method to enable the pricing of discretely-monitored barrier options; in Section 6 we will test and discuss the speed and accuracy of the 2D-COS Method for the pricing of European, Bermudan and barrier options under the Heston dynamics.

2 The Heston Model

The Heston stochastic volatility model defines the dynamics of the process $\mathbf{X}_t = (X_t^1, X_t^2) = (S_t, v_t)$, where S_t is the stock price and v_t is the variance, by the following system of stochastic differential equations:

$$\begin{aligned} dS_t &= (r - q)S_t dt + \sqrt{v_t}S_t dW_t^1 \\ dv_t &= \lambda(\bar{v} - v_t)dt + \eta\sqrt{v_t} dW_t^2 \\ dW_t^1 dW_t^2 &= \rho dt \end{aligned}$$

where r is the risk-free interest rate, q is the continuously compounded dividend yield, the three (non-negative) parameters, λ , \bar{v} and η , represent the speed of mean reversion, the mean level of variance and the volatility of the volatility process, respectively, and ρ is the constant correlation between the stock price and the variance processes. W_t^1 and W_t^2 are thus correlated Brownian motions. v_t represents the instantaneous variance of relative changes in S_t , in the sense that the quadratic variation of $\frac{dS_t}{S_t}$ over the interval $[t, t + dt]$ is $v_t dt$.

A more common representation of the Heston model is in terms of the logarithm of the stock price, $x_t = \ln S_t$. This representation is easily obtained by applying Itô's formula to get

$$\begin{aligned} dx_t &= \left(r - q - \frac{1}{2}v_t \right) dt + \sqrt{v_t} dW_t^1 \\ dv_t &= \lambda(\bar{v} - v_t)dt + \eta\sqrt{v_t} dW_t^2 \\ dW_t^1 dW_t^2 &= \rho dt \end{aligned}$$

We can also define the Heston model in terms of independent Brownian motions $(\tilde{W}_t^1, \tilde{W}_t^2)$, in which case it will then take the following form:

$$dx_t = \left(r - q - \frac{1}{2}v_t \right) dt + \rho\sqrt{v_t}d\tilde{W}_t^1 + \sqrt{1 - \rho^2}\sqrt{v_t} d\tilde{W}_t^2 \quad (2.1)$$

$$dv_t = \lambda(\bar{v} - v_t)dt + \eta\sqrt{v_t} d\tilde{W}_t^1 \quad (2.2)$$

2.1 Distribution of the variance process

The variance process in (2.2) is modelled as a mean-reverting square-root diffusion, with dynamics which are similar to those used for the interest rate model of Cox et al. [1985]. This process for the variance precludes negative values for v_t , and if v_t reaches zero it can subsequently return to being positive. If the Feller condition ($2\lambda\bar{v} \geq \eta^2$) is satisfied, then v_t is guaranteed to stay positive. If this condition is not satisfied, then v_t may reach zero.

In Cox et al. [1985] and van Haastrecht and Pelsser [2010] it is shown that, with

$$q_{Feller} := \frac{2\lambda\bar{v}}{\eta^2} - 1 \quad \text{and} \quad \zeta := \frac{2\lambda}{(1 - e^{-\lambda(t-s)})\eta^2}$$

(for $0 < s < t$) the process $2\zeta v_t$ is governed by the non-central chi-square distribution with $2(q_{Feller} + 1)$ degrees of freedom and non-centrality parameter $2\zeta v_s e^{-\lambda(t-s)}$. i.e:

$$2\zeta v_t \sim \chi^2 \left(2(q_{Feller} + 1), 2\zeta v_s e^{-\lambda(t-s)} \right) \quad (2.3)$$

Note that satisfying the Feller condition is now equivalent to having $q_{Feller} \geq 0$.

In cases where the Feller condition is not satisfied, the cumulative distribution of the variance will display “near-singular behaviour”, with the left tail of the density growing rapidly. This type of behaviour may lead to significant errors, particularly for integration-based option-pricing methods, which require truncation of the integration range. In Andersen [2008] it was reported that the Feller condition is often found not to be satisfied by parameters obtained from market data.

2.2 Characteristic Function

In Heston [1993] it was shown that under the log transform of the stock price, $x_t = \log(S_t)$, the Heston model is in the class of affine jump diffusions. As a result of this, the characteristic function of the process $\mathbf{X}_t = (x_t, v_t)$ will be given by

$$\begin{aligned} \phi(\mathbf{u}, \mathbf{X}_t, t, T) &= \mathbb{E}[e^{i\mathbf{u} \cdot \mathbf{X}_T} | \mathbf{X}_t] \\ &= e^{A(\mathbf{u}, T-t) + \mathbf{B}^T(\mathbf{u}, T-t) \mathbf{X}_t} \end{aligned} \quad (2.4)$$

where $A(\mathbf{u}, \tau)$ and $\mathbf{B}(\mathbf{u}, \tau)$ can be obtained in closed form.

2.2.1 The Bivariate ChF

For the two-dimensional COS method, we will require the *bivariate characteristic function*, which is given by

$$\phi(u_1, u_2, x_t, v_t, t, T) = \mathbb{E}[e^{iu_1 x_T + iu_2 v_T} | x_t, v_t] \quad (2.5)$$

Applying Itô's lemma then gives us

$$\begin{aligned}
d\phi = & \left(\frac{\partial\phi}{\partial t} + \left(r - q - \frac{1}{2}v_t \right) \frac{\partial\phi}{\partial x} + \lambda(\bar{v} - v_t) \frac{\partial\phi}{\partial v} + \frac{1}{2}v_t \frac{\partial^2\phi}{\partial x^2} + \rho\eta v_t \frac{\partial^2\phi}{\partial x\partial v} + \frac{1}{2}\eta^2 v_t \frac{\partial^2\phi}{\partial v^2} \right) dt \\
& + \left(\rho\sqrt{v_t} \frac{\partial\phi}{\partial x} + \eta\sqrt{v_t} \frac{\partial\phi}{\partial v} \right) d\tilde{W}_t^1 + \sqrt{v_t(1-\rho^2)} \frac{\partial\phi}{\partial x} d\tilde{W}_t^2
\end{aligned} \tag{2.6}$$

Now, by applying iterated expectations we can see that ϕ must be a martingale:

For $s < t < T$

$$\begin{aligned}
\mathbb{E}[\phi(u_1, u_2, x_t, v_t, t, T) | x_s, v_s] &= \mathbb{E} \left[\mathbb{E}[e^{iu_1 x_T + iu_2 v_T} | x_t, v_t] \Big| x_s, v_s \right] \\
&= \mathbb{E}[e^{iu_1 x_T + iu_2 v_T} | x_s, v_s] \\
&= \phi(u_1, u_2, x_s, v_s, s, T)
\end{aligned}$$

Hence, we must have $\mathbb{E}[d\phi] = 0$. Applying this condition to (2.6) then gives us

$$\frac{\partial\phi}{\partial t} + \left(r - q - \frac{1}{2}v_t \right) \frac{\partial\phi}{\partial x} + \lambda(\bar{v} - v_t) \frac{\partial\phi}{\partial v} + \frac{1}{2} \left[v_t \frac{\partial^2\phi}{\partial x^2} + 2\eta\rho v_t \frac{\partial^2\phi}{\partial x\partial v} + \eta^2 v_t \frac{\partial^2\phi}{\partial v^2} \right] = 0 \tag{2.7}$$

Since \mathbf{X}_t is an affine jump diffusion, it follows from (2.4) that the bivariate characteristic function has the form

$$\phi(u_1, u_2, x_t, v_t, t, T) = \exp \left(A(u_1, u_2, T-t) + B_1(u_1, u_2, T-t)x_t + B_2(u_1, u_2, T-t)v_t \right) \tag{2.8}$$

Using the fact that ϕ is of this form, one can easily find the necessary partial derivatives. Inserting these into (2.7) then results in the following:

$$\begin{aligned}
& \frac{\partial A(u_1, u_2, T-t)}{\partial t} + \frac{\partial B_1(u_1, u_2, T-t)}{\partial t} x + \frac{\partial B_2(u_1, u_2, T-t)}{\partial t} v + \left(r - q - \frac{1}{2}v \right) B_1(u_1, u_2, T-t) \\
& + \lambda(\bar{v} - v)B_2(u_1, u_2, T-t) + \eta\rho v B_1(u_1, u_2, T-t)B_2(u_1, u_2, T-t) + \frac{1}{2}v B_1^2(u_1, u_2, T-t) \\
& + \frac{1}{2}\eta^2 v B_2^2(u_1, u_2, T-t) = 0
\end{aligned} \tag{2.9}$$

It is, however, more convenient to use the time to maturity, $\tau = T - t$, as the parameter rather than the time, t . As a result of this, the partial derivatives with respect to t will be replaced by the negative of the partial derivatives with respect to τ :

$$\begin{aligned} \frac{\partial A(u_1, u_2, \tau)}{\partial \tau} + \frac{\partial B_1(u_1, u_2, \tau)}{\partial \tau} x + \frac{\partial B_2(u_1, u_2, \tau)}{\partial \tau} v - \left(r - q - \frac{1}{2}v \right) B_1(u_1, u_2, \tau) - \lambda(\bar{v} - v)B_2(u_1, u_2, \tau) \\ - \eta\rho v B_1(u_1, u_2, \tau)B_2(u_1, u_2, \tau) - \frac{1}{2}vB_1^2(u_1, u_2, \tau) - \frac{1}{2}\eta^2 v B_2^2(u_1, u_2, \tau) = 0 \end{aligned} \quad (2.10)$$

This expression can then be rearranged to obtain the following:

$$\begin{aligned} \frac{\partial A(u_1, u_2, \tau)}{\partial \tau} - (r - q)B_1(u_1, u_2, \tau) - \lambda\bar{v}B_2(u_1, u_2, \tau) + \frac{\partial B_1(u_1, u_2, \tau)}{\partial \tau} x + \left[\frac{\partial B_2(u_1, u_2, \tau)}{\partial \tau} \right. \\ \left. + \frac{1}{2}B_1(u_1, u_2, \tau) + (\lambda - \eta\rho B_1(u_1, u_2, \tau))B_2(u_1, u_2, \tau) - \frac{1}{2}B_1^2(u_1, u_2, \tau) - \frac{1}{2}\eta^2 B_2^2(u_1, u_2, \tau) \right] v = 0 \end{aligned} \quad (2.11)$$

Now, with the terminal condition $\phi(u_1, u_2, x_T, v_T, T, T) = e^{iu_1 x_T + iu_2 v_T}$, the complex-valued functions $A(u_1, u_2, \tau)$, $B_1(u_1, u_2, \tau)$ and $B_2(u_1, u_2, \tau)$ must satisfy the following system of ordinary differential equations (ODEs):

$$\frac{\partial B_1(u_1, u_2, \tau)}{\partial \tau} = 0 \quad (2.12)$$

$$\frac{\partial B_2(u_1, u_2, \tau)}{\partial \tau} = \frac{1}{2}\eta^2 B_2^2(u_1, u_2, \tau) - \beta B_2(u_1, u_2, \tau) - \frac{1}{2}u_1(i + u_1) \quad (2.13)$$

$$\frac{\partial A(u_1, u_2, \tau)}{\partial \tau} = iu_1(r - q) + \lambda\bar{v}B_2(u_1, u_2, \tau) \quad (2.14)$$

with the initial conditions $B_1(u_1, u_2, 0) = iu_1$, $B_2(u_1, u_2, 0) = iu_2$ and $A(u_1, u_2, 0) = 0$, and where β is defined as follows:

$$\beta := \lambda - i\rho\eta u_1 \quad (2.15)$$

This is a system of ODEs of Riccati type, and the solution of these ODEs can then be derived as

$$B_1(u_1, u_2, \tau) = iu_1 \quad (2.16)$$

$$B_2(u_1, u_2, \tau) = \frac{\beta - D - (\beta + D)he^{-D\tau}}{\eta^2(1 - he^{-D\tau})} \quad (2.17)$$

$$A(u_1, u_2, \tau) = iu_1(r - q)\tau + \frac{\lambda\bar{v}}{\eta^2} \left[(\beta - D)\tau - 2 \ln \left(\frac{he^{-D\tau} - 1}{h - 1} \right) \right] \quad (2.18)$$

where we further define

$$D := \sqrt{\beta^2 + \eta^2 u_1(i + u_1)} \quad (2.19)$$

$$h := \frac{\beta - D - iu_2\eta^2}{\beta + D - iu_2\eta^2} \quad (2.20)$$

Note that for D we use the common convention that the real part of the square root is non-negative. This is not a restriction, since the characteristic function is even in D .

2.2.2 Cumulants of the log-stock Process

Now that we have an analytic representation of the bivariate ChF, we can obtain the univariate ChF simply by setting $u_2 = 0$. We can then apply (A.4) with this ChF in order to obtain the cumulants. With the aid of Maple, the first and second cumulants, c_1 and c_2 respectively, are given by

$$c_1 = (r - q)\tau + (1 - e^{-\lambda\tau})\frac{\bar{v} - v_0}{2\lambda} - \frac{1}{2}\bar{v}\tau \quad (2.21)$$

$$c_2 = \frac{1}{8\lambda^3} \left[\eta\tau\lambda e^{-\lambda\tau}(v_0 - \bar{v})(8\lambda\rho - 4\eta) + \lambda\rho\eta(1 - e^{-\lambda\tau})(16\bar{v} - 8v_0) + 2\bar{v}\lambda\tau(-4\lambda\rho\eta + \eta^2 + 4\lambda^2) \right. \\ \left. + \eta^2((\bar{v} - 2v_0)e^{-2\lambda\tau} + \bar{v}(6e^{-\lambda\tau} - 7) + 2v_0) + 8\lambda^2(v_0 - \bar{v})(1 - e^{-\lambda\tau}) \right] \quad (2.22)$$

3 The Two-Dimensional COS Method for Pricing European Options

The two-dimensional COS Method is an extension of the COS Method of Fang and Oosterlee [2008], and as such is based on the Fourier cosine series expansions of the payoff function and the density.

We begin with the discounted risk neutral option valuation formula, which is obtained by taking the discounted risk neutral expectation at time t_0 of the European option payoff, $v(T, \mathbf{y})$:

$$v(t_0, \mathbf{x}) = e^{-r\Delta t} \mathbb{E}_{t_0}^{\mathbb{Q}} [v(T, \mathbf{y})] = e^{-r\Delta t} \iint_{\mathbb{R}^2} v(T, \mathbf{y}) f(\mathbf{y}|\mathbf{x}; \Delta t) d\mathbf{y} \quad (3.1)$$

where $\mathbf{x} = (x_1, x_2)$ is the current value of the asset price process, $v(t, x_1, x_2)$ is the value of the option at time t , $f(y_1, y_2|x_1, x_2)$ is the conditional density function, r is the risk-free interest rate and the time to expiration is denoted by $\Delta t := T - t_0$.

We first consider the Fourier transform pair, which consists of the conditional characteristic function and the conditional transition density

$$\phi(\mathbf{u}|\mathbf{x}; \Delta t) = \iint_{\mathbb{R}^2} e^{i\mathbf{u}\cdot\mathbf{y}} f(\mathbf{y}|\mathbf{x}; \Delta t) d\mathbf{y} \quad (3.2)$$

$$f(\mathbf{y}|\mathbf{x}; \Delta t) = \frac{1}{2\pi} \iint_{\mathbb{R}^2} e^{-i\mathbf{u}\cdot\mathbf{y}} \phi(\mathbf{u}|\mathbf{x}; \Delta t) d\mathbf{u} \quad (3.3)$$

We assume that f is integrable. We can therefore 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.

Define

$$\sum'_{k=0}^{\infty} B_k := \frac{1}{2} B_0 + \sum_{k=1}^{\infty} B_k$$

where $\{B_k\}_k$ is some set of coefficients. The two-dimensional Fourier-cosine series expansion of $f(\mathbf{y}|\mathbf{x})$ on $[a_1, b_1] \times [a_2, b_2]$ is then given by

$$f(\mathbf{y}|\mathbf{x}; \Delta t) = \sum'_{k_1=0}^{+\infty} \sum'_{k_2=0}^{+\infty} A_{k_1, k_2}(\mathbf{x}) \cos\left(k_1 \pi \frac{y_1 - a_1}{b_1 - a_1}\right) \cos\left(k_2 \pi \frac{y_2 - a_2}{b_2 - a_2}\right) \quad (3.4)$$

where the cosine series coefficients are defined by

$$A_{k_1, k_2}(\mathbf{x}) := \frac{2}{b_1 - a_1} \frac{2}{b_2 - a_2} \int_{a_2}^{b_2} \int_{a_1}^{b_1} f(\mathbf{y}|\mathbf{x}; \Delta t) \cos\left(k_1 \pi \frac{y_1 - a_1}{b_1 - a_1}\right) \cos\left(k_2 \pi \frac{y_2 - a_2}{b_2 - a_2}\right) dy_1 dy_2 \quad (3.5)$$

Define $k_j^* := \frac{k_j \pi}{b_j - a_j}$ for $j = 1, 2$. The two-dimensional Fourier cosine expansion formulation will then be:

$$v_1(t_0, \mathbf{x}) = e^{-r\Delta t} \int_{a_2}^{b_2} \int_{a_1}^{b_1} v(T, \mathbf{y}) f(\mathbf{y} | \mathbf{x}; \Delta t) dy_1 dy_2 \quad (3.6)$$

$$= e^{-r\Delta t} \int_{a_2}^{b_2} \int_{a_1}^{b_1} v(T, \mathbf{y}) \sum_{k_1=0}^{+\infty} \sum_{k_2=0}^{+\infty} A_{k_1, k_2}(\mathbf{x}) \cos(k_1^*(y_1 - a_1)) \cos(k_2^*(y_2 - a_2)) dy_1 dy_2 \quad (3.7)$$

where the notation v_i is used to indicate the successive approximations of v . We can then interchange the summation and integration as follows:

$$v_1(t_0, \mathbf{x}) = e^{-r\Delta t} \int_{a_2}^{b_2} \int_{a_1}^{b_1} v(T, \mathbf{y}) \sum_{k_1=0}^{+\infty} \sum_{k_2=0}^{+\infty} A_{k_1, k_2}(\mathbf{x}) \cos(k_1^*(y_1 - a_1)) \cos(k_2^*(y_2 - a_2)) dy_1 dy_2 \quad (3.8)$$

$$= e^{-r\Delta t} \sum_{k_1=0}^{+\infty} \sum_{k_2=0}^{+\infty} A_{k_1, k_2}(\mathbf{x}) \int_{a_2}^{b_2} \int_{a_1}^{b_1} v(T, \mathbf{y}) \cos(k_1^*(y_1 - a_1)) \cos(k_2^*(y_2 - a_2)) dy_1 dy_2 \quad (3.9)$$

$$= e^{-r\Delta t} \sum_{k_1=0}^{+\infty} \sum_{k_2=0}^{+\infty} A_{k_1, k_2}(\mathbf{x}) V_{k_1, k_2} \quad (3.10)$$

where

$$V_{k_1, k_2} := \int_{a_2}^{b_2} \int_{a_1}^{b_1} v(T, \mathbf{y}) \cos(k_1^*(y_1 - a_1)) \cos(k_2^*(y_2 - a_2)) dy_1 dy_2 \quad (3.11)$$

are the Fourier cosine series coefficients of $v(T, \mathbf{y})$ on $[a_1, b_1] \times [a_2, b_2]$. Truncation of the series summations then gives

$$v_2(t_0, \mathbf{x}) = e^{-r\Delta t} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} A_{k_1, k_2}(\mathbf{x}) V_{k_1, k_2} \quad (3.12)$$

Now, the coefficients $A_{k_1, k_2}(\mathbf{x})$ can be approximated in the following manner (letting $\omega_j = \frac{2}{b_j - a_j}$):

$$A_{k_1, k_2}(\mathbf{x}) = \omega_1 \omega_2 \int_{a_2}^{b_2} \int_{a_1}^{b_1} f(\mathbf{y}|\mathbf{x}; \Delta t) \cos(k_1^*(y_1 - a_1)) \cos(k_2^*(y_2 - a_2)) dy_1 dy_2 \quad (3.13)$$

$$\approx \omega_1 \omega_2 \iint_{\mathbb{R}^2} f(\mathbf{y}|\mathbf{x}; \Delta t) \cos(k_1^*(y_1 - a_1)) \cos(k_2^*(y_2 - a_2)) dy_1 dy_2 \quad (3.14)$$

$$\begin{aligned} &= \omega_1 \omega_2 \iint_{\mathbb{R}^2} f(\mathbf{y}|\mathbf{x}; \Delta t) \frac{1}{2} [\cos(k_1^*(y_1 - a_1) + k_2^*(y_2 - a_2)) \\ &\quad + \cos(k_1^*(y_1 - a_1) - k_2^*(y_2 - a_2))] dy_1 dy_2 \end{aligned} \quad (3.15)$$

$$\begin{aligned} &= \frac{1}{2} \omega_1 \omega_2 \left[\iint_{\mathbb{R}^2} f(\mathbf{y}|\mathbf{x}; \Delta t) \cos(k_1^*(y_1 - a_1) + k_2^*(y_2 - a_2)) dy_1 dy_2 \right. \\ &\quad \left. + \iint_{\mathbb{R}^2} f(\mathbf{y}|\mathbf{x}; \Delta t) \cos(k_1^*(y_1 - a_1) - k_2^*(y_2 - a_2)) dy_1 dy_2 \right] \end{aligned} \quad (3.16)$$

$$= \frac{1}{2} [F_{k_1, k_2}^+(\mathbf{x}) + F_{k_1, k_2}^-(\mathbf{x})] \quad (3.17)$$

where

$$F_{k_1, k_2}^\pm(\mathbf{x}) := \omega_1 \omega_2 \iint_{\mathbb{R}^2} f(\mathbf{y}|\mathbf{x}; \Delta t) \cos(k_1^*(y_1 - a_1) \pm k_2^*(y_2 - a_2)) dy_1 dy_2 \quad (3.18)$$

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

$$F_{k_1, k_2}^\pm(\mathbf{x}) = \omega_1 \omega_2 \iint_{\mathbb{R}^2} f(\mathbf{y}|\mathbf{x}; \Delta t) \cos(k_1^*(y_1 - a_1) \pm k_2^*(y_2 - a_2)) dy_1 dy_2 \quad (3.19)$$

$$\begin{aligned} &= \omega_1 \omega_2 \operatorname{Re} \left\{ \iint_{\mathbb{R}^2} f(\mathbf{y}|\mathbf{x}; \Delta t) \cos(k_1^*(y_1 - a_1) \pm k_2^*(y_2 - a_2)) dy_1 dy_2 \right. \\ &\quad \left. + i \iint_{\mathbb{R}^2} f(\mathbf{y}|\mathbf{x}; \Delta t) \sin(k_1^*(y_1 - a_1) \pm k_2^*(y_2 - a_2)) dy_1 dy_2 \right\} \end{aligned} \quad (3.20)$$

$$= \omega_1 \omega_2 \operatorname{Re} \left\{ \iint_{\mathbb{R}^2} f(\mathbf{y}|\mathbf{x}; \Delta t) \exp[ik_1^*(y_1 - a_1) \pm ik_2^*(y_2 - a_2)] dy_1 dy_2 \right\} \quad (3.21)$$

$$= \omega_1 \omega_2 \operatorname{Re} \left\{ \iint_{\mathbb{R}^2} f(\mathbf{y}|\mathbf{x}; \Delta t) \exp [ik_1^* y_1 \pm ik_2^* y_2] dy_1 dy_2 \cdot \exp [-ik_1^* a_1 \mp ik_2^* a_2] \right\} \quad (3.22)$$

$$= \omega_1 \omega_2 \operatorname{Re} \left\{ \phi \left(k_1^*, \pm k_2^* \mid \mathbf{x}; \Delta t \right) \exp [-ik_1^* a_1 \mp ik_2^* a_2] \right\} \quad (3.23)$$

which can be simplified further for the Heston model to

$$F_{k_1, k_2}^\pm(\mathbf{x}) = \omega_1 \omega_2 \operatorname{Re} \left\{ \phi \left(k_1^*, \pm k_2^* \mid x_2; \Delta t \right) \exp [ik_1^*(x_1 - a_1) \mp ik_2^* a_2] \right\} \quad (3.24)$$

where we use the fact that for the Heston model we have

$$\phi(u_1, u_2 | x, v; \Delta t) = \phi(u_1, u_2 | 0, v; \Delta t) e^{iu_1 x} \quad (3.25)$$

We can now replace $A_{k_1, k_2}(\mathbf{x})$ in (3.12) by $F_{k_1, k_2}(\mathbf{x})$ to obtain the final expression for the 2-D COS method value for European options:

$$v_3(t_0, \mathbf{x}) = e^{-r\Delta t} \sum'_{k_1=0}^{N_1-1} \sum'_{k_2=0}^{N_2-1} \frac{1}{2} \left[F_{k_1, k_2}^+(\mathbf{x}) + F_{k_1, k_2}^-(\mathbf{x}) \right] V_{k_1, k_2} \quad (3.26)$$

3.1 Evaluating the V_{k_1, k_2} terms

For European call and put options we have

$$v(T, \mathbf{y}) = (\alpha K (e^{y_1} - 1))^+, \quad \text{where } \alpha = \begin{cases} 1 & \text{for a call,} \\ -1 & \text{for a put.} \end{cases} \quad (3.27)$$

and we can then see that for a European call option, when $y_1 > 0$, $v(T, \mathbf{y})$ is non-zero. Hence, if we assume that $a_1 \leq 0 \leq b_1$, we then have

$$V_{k_1, k_2}^{\text{call}} = \int_{a_2}^{b_2} \int_{a_1}^{b_1} (K(e^{y_1} - 1))^+ \cos(k_1^*(y_1 - a_1)) \cos(k_2^*(y_2 - a_2)) dy_1 dy_2 \quad (3.28)$$

$$= K \int_{a_2}^{b_2} \int_0^{b_1} (e^{y_1} - 1) \cos(k_1^*(y_1 - a_1)) \cos(k_2^*(y_2 - a_2)) dy_1 dy_2 \quad (3.29)$$

$$= K \int_{a_2}^{b_2} \cos(k_2^*(y_2 - a_2)) \left(\int_0^{b_1} (e^{y_1} - 1) \cos(k_1^*(y_1 - a_1)) dy_1 \right) dy_2 \quad (3.30)$$

Similarly, for European puts we have

$$V_{k_1, k_2}^{\text{put}} = K \int_{a_2}^{b_2} \cos(k_2^*(y_2 - a_2)) \left(\int_{a_1}^0 (1 - e^{y_1}) \cos(k_1^*(y_1 - a_1)) dy_1 \right) dy_2 \quad (3.31)$$

In Stewart [2006, Table of Integrals, pp. 10] we can see that

$$\int_{c_1}^{c_2} e^{au} \cos(bu) du = \frac{e^{au}}{a^2 + b^2} (a \cos(bu) + b \sin(bu)) \Big|_{u=c_1}^{u=c_2} \quad (3.32)$$

Letting (for $j = 1, 2$) $c_1 = x_1 - a_j$, $c_2 = x_2 - a_j$, $a = 1$ and $b = k_j^*$ (for $k_j \neq 0$), (3.32) then becomes

$$\int_{x_1 - a_j}^{x_2 - a_j} e^u \cos(k_j^* u) du = \frac{e^u}{1 + (k_j^*)^2} [\cos(k_j^* u) + k_j^* \sin(k_j^* u)] \Big|_{u=x_1 - a_j}^{u=x_2 - a_j} \quad (3.33)$$

If we then change the variable of integration to $y = u + a$, we have

$$\int_{x_1}^{x_2} e^{y - a_j} \cos(k_j^*(y - a_j)) dy = \frac{e^u}{1 + (k_j^*)^2} [\cos(k_j^* u) + k_j^* \sin(k_j^* u)] \Big|_{u=x_1 - a_j}^{u=x_2 - a_j} \quad (3.34)$$

Now define $\chi_{k_j}(x_1, x_2)$ as

$$\chi_{k_j}(x_1, x_2) \quad (3.35)$$

$$:= \int_{x_1}^{x_2} e^y \cos(k^*(y - a_j)) dy \quad (3.36)$$

$$= \frac{e^{u+a_j}}{1 + (k_j^*)^2} [\cos(k_j^* u) + k_j^* \sin(k_j^* u)] \Big|_{u=x_1 - a_j}^{u=x_2 - a_j} \quad (3.37)$$

$$= \frac{1}{1 + (k_j^*)^2} [e^{x_2} \cos(k_j^*(x_2 - a_j)) + e^{x_2} k_j^* \sin(k_j^*(x_2 - a_j)) - e^{x_1} \cos(k_j^*(x_1 - a_j)) - e^{x_1} k_j^* \sin(k_j^*(x_1 - a_j))] \quad (3.38)$$

Next, define $\psi_{k_j}(x_1, x_2)$ (for $k_j \neq 0$) as

$$\psi_{k_j}(x_1, x_2) := \int_{x_1}^{x_2} \cos(k_j^*(y - a_j)) dy \quad (3.39)$$

$$= \frac{1}{k_j^*} \sin(k_j^*(y - a_j)) \Big|_{y=x_1}^{y=x_2} \quad (3.40)$$

$$= \frac{1}{k_j^*} [\sin(k_j^*(x_2 - a_j)) - \sin(k_j^*(x_1 - a_j))] \quad (3.41)$$

and for $k_j = 0$:

$$\psi_0(x_1, x_2) = \int_{x_1}^{x_2} \cos(0) dy \quad (3.42)$$

$$= \int_{x_1}^{x_2} 1 dy \quad (3.43)$$

$$= x_2 - x_1 \quad (3.44)$$

Using (3.38), (3.41) and (3.44), the coefficients V_{k_1, k_2} for European calls can then be calculated as follows:

$$V_{k_1, k_2}^{\text{call}} = K \int_{a_2}^{b_2} \cos(k_2^*(y_2 - a_2)) \left(\int_0^{b_1} (e^{y_1} - 1) \cos(k_1^*(y_1 - a_1)) dy_1 \right) dy_2 \quad (3.45)$$

$$= K \int_{a_2}^{b_2} \cos(k_2^*(y_2 - a_2)) [\chi_{k_1}(0, b_1) - \psi_{k_1}(0, b_1)] dy_2 \quad (3.46)$$

$$= K [\chi_{k_1}(0, b_1) - \psi_{k_1}(0, b_1)] \int_{a_2}^{b_2} \cos(k_2^*(y_2 - a_2)) dy_2 \quad (3.47)$$

$$= K [\chi_{k_1}(0, b_1) - \psi_{k_1}(0, b_1)] \psi_{k_2}(a_2, b_2) \quad (3.48)$$

Similarly, for European puts we have

$$V_{k_1, k_2}^{\text{put}} = K [\psi_{k_1}(a_1, 0) - \chi_{k_1}(a_1, 0)] \psi_{k_2}(a_2, b_2) \quad (3.49)$$

Notice that (3.48) and (3.49) both include a multiplication by $\psi_{k_2}(a_2, b_2)$. Now, for $k_2 \neq 0$, we have

$$\psi_{k_2}(a_2, b_2) = \frac{1}{k_2^*} [\sin(k_2^*(b_2 - a_2)) - \sin(k_2^*(a_2 - a_2))] \quad (3.50)$$

$$= \frac{1}{k_2^*} \left[\sin\left(\frac{k_2\pi}{b_2 - a_2}(b_2 - a_2)\right) - \sin(0) \right] \quad (3.51)$$

$$= \frac{1}{k_2^*} \sin(k_2\pi) \quad (3.52)$$

$$= 0 \quad \text{since } k_2 \in \mathbb{Z} \quad (3.53)$$

Thus, for $k_2 > 0$ we have $V_{k_1, k_2} = 0$, which means that increasing N_2 beyond 1 will offer no increases in accuracy over the (1D)COS method of Fang and Oosterlee [2008] and serves only to increase the computational time.

3.2 Truncation of the computational domain

The choice of the computational domain is an important factor in the performance of the 2D-COS method. A domain that is too small will result in low accuracy, however, larger domains will require more terms in the expansions in order to reach a certain level of accuracy.

The computational domain $[a_1, b_1] \times [a_2, b_2]$ proposed by Ruijter and Oosterlee [2012] is based on the truncation used for the one-dimensional COS method by Fang and Oosterlee [2008]. For the log-stock dimension, $[a_1, b_1]$ the interval is determined with the use of the cumulants, c_j , of X_T :

$$[a_1, b_1] := \left[x_0 + c_1 - L\sqrt{|c_2|}, x_0 + c_1 + L\sqrt{|c_2|} \right] \quad (3.54)$$

where $x_0 := \ln \frac{S_0}{K}$ and $L = 12$. For c_1 and c_2 we use the analytical formulae (2.21) and (2.22) respectively.

A truncation rule which includes the fourth cumulant, c_4 , would be more accurate for short maturities, however the analytical formula for c_4 in the Heston model is involved and lengthy (although it can also be obtained with the aid of Maple).

We take the absolute value of cumulant c_2 as it may become negative if the parameter set does not satisfy the Feller condition.

For $a_2 \geq 0$ and b_2 an integration range is determined such that

$$F_{v_T|v_0}(a_2|v_0) = \text{TOL} = 1 - F_{v_T|v_0}(b_2|v_0) \quad (3.55)$$

where $F_{v_T|v_0}(\cdot)$ represents the cumulative distribution function of the variance at terminal time and TOL is some tolerance level, for which a value $\text{TOL} = 10^{-4}$ is suggested.

4 The Two-Dimensional COS Method for Pricing Bermudan Options

In this section we extend the 2-D COS method for European options to the case of Bermudan options with a two-dimensional underlying asset price process. A Bermudan option can be exercised at a fixed set of M early exercise times, $\mathcal{T} = \{t_0, t_1, \dots, t_M\}$ where $t_0 < t_1 < \dots < t_m < \dots < t_M = T$, with $\Delta t := t_{m+1} - t_m$. Let the payoff received by the option holder at any time $t \in \mathcal{T}$ be denoted as $I(S_t)$, where S_t is the value of the underlying at that time. $I(S_t)$ is known as the intrinsic value of the option at time t . The Bermudan option pricing formula reads as

$$v(t_m, \mathbf{x}) = \begin{cases} I(\mathbf{x}) & \text{for } m = M, \\ \max[c(t_m, \mathbf{x}), I(\mathbf{x})] & \text{for } m = 1, 2, \dots, M - 1, \\ c(t_m, \mathbf{x}) & \text{for } m = 0. \end{cases} \quad (4.1)$$

where $c(\tau, \mathbf{x})$ is the continuation value at time τ , defined as the value of the option if it is not exercised at time τ . The continuation value is given by (using a simplified notation of \mathbf{x}_m for \mathbf{x}_{t_m})

$$c(t_m, \mathbf{x}_m) = e^{-r\Delta t} \mathbb{E}_{t_m}^{\mathbb{Q}} [v(t_{m+1}, \mathbf{x}_{m+1})] \quad (4.2)$$

Now, similarly to the European option value, we have

$$c(t_m, \mathbf{x}_m) = e^{-r\Delta t} \mathbb{E}_{t_m}^{\mathbb{Q}} [v(t_{m+1}, \mathbf{x}_{m+1})] \quad (4.3)$$

$$= e^{-r\Delta t} \iint_{\mathbb{R}^2} v(t_{m+1}, \mathbf{y}) f(\mathbf{y} | \mathbf{x}_m; \Delta t) d\mathbf{y} \quad (4.4)$$

$$\approx e^{-r\Delta t} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} \frac{1}{2} \left[F_{k_1, k_2}^+(\mathbf{x}_m) + F_{k_1, k_2}^-(\mathbf{x}_m) \right] V_{k_1, k_2}(t_{m+1}) \quad (4.5)$$

where (4.5) is obtained by using the two-dimensional Fourier-cosine series expansion to approximate the double integral and

$$F_{k_1, k_2}^{\pm}(\mathbf{x}) := \operatorname{Re} \left\{ \phi \left(k_1^*, \pm k_2^* \mid \mathbf{x}; \Delta t \right) \exp \left[-ik_1^* a_1 \mp ik_2^* a_2 \right] \right\} \quad (4.6)$$

$$\text{and} \quad (4.7)$$

$$V_{k_1, k_2}(t_{m+1}) := \frac{2}{b_1 - a_1} \frac{2}{b_2 - a_2} \int_{a_2}^{b_2} \int_{a_1}^{b_1} v(t_{m+1}, \mathbf{y}) \cos(k_1^*(y_1 - a_1)) \cos(k_2^*(y_2 - a_2)) dy_1 dy_2 \quad (4.8)$$

where, as in Section 3, we define $k_j^* := \frac{k_j \pi}{b_j - a_j}$.

Notice that the coefficients $V_{k_1, k_2}(t_{m+1})$ are now time dependent. For $m = M-1, M-2, \dots, 1$ we have, by (4.1) (again, letting $\omega_j = \frac{2}{b_j - a_j}$):

$$V_{k_1, k_2}(t_{m+1}) = \omega_1 \omega_2 \int_{a_2}^{b_2} \int_{a_1}^{b_1} \max[c(t_m, \mathbf{x}), I(\mathbf{x})] \cos(k_1^*(y_1 - a_1)) \cos(k_2^*(y_2 - a_2)) dy_1 dy_2 \quad (4.9)$$

4.1 Evaluating the V_{k_1, k_2} terms

In this section, we show that the coefficients V_{k_1, k_2} can be recovered recursively and derive an algorithm using the Fast Fourier Transform (FFT) to do this.

Starting with the final time point, for the coefficients $V_{k_1, k_2}(t_M)$ we have the terminal condition $v(t_M, \mathbf{y}) = I(\mathbf{y})$. This is the same as the European case, and can therefore be calculated using (3.48) and (3.49) for calls and puts respectively. For the V_{k_1, k_2} coefficients that are used to approximate the continuation values at times t_{M-2}, \dots, t_0 , the value function $v(t_m, \mathbf{y}) = \max[I(\mathbf{y}), c(t_m, \mathbf{y})]$ appears, and we need to find an optimal policy for all state values $\mathbf{y} \in [a_1, b_1] \times [a_2, b_2]$.

In order to do this, we divide the domain $[a_1, b_1] \times [a_2, b_2]$ into rectangular subdomains, \mathcal{C}^q and \mathcal{G}^p , such that it is optimal to continue for all states $\mathbf{y} \in \mathcal{C}^q$ and it is optimal to exercise for all states $\mathbf{y} \in \mathcal{G}^p$. This then allows us to split the integral in (4.9) into different parts:

$$\begin{aligned} V_{k_1, k_2}(t_m) &= \omega_1 \omega_2 \sum_p \iint_{\mathcal{G}^p} I(\mathbf{y}) \cos(k_1^*(y_1 - a_1)) \cos(k_2^*(y_2 - a_2)) d\mathbf{y} \\ &\quad + \omega_1 \omega_2 \sum_q \iint_{\mathcal{C}^q} c(t_m, \mathbf{y}) \cos(k_1^*(y_1 - a_1)) \cos(k_2^*(y_2 - a_2)) d\mathbf{y} \\ &:= \sum_p G_{k_1, k_2}(\mathcal{G}^p) + \sum_q C_{k_1, k_2}(t_m, \mathcal{C}^q) \end{aligned} \quad (4.10)$$

where

$$G_{k_1, k_2}(\mathcal{G}^p) := \omega_1 \omega_2 \iint_{\mathcal{G}^p} I(\mathbf{y}) \cos(k_1^*(y_1 - a_1)) \cos(k_2^*(y_2 - a_2)) d\mathbf{y} \quad (4.11)$$

and

$$C_{k_1, k_2}(t_m, \mathcal{C}^q) := \omega_1 \omega_2 \iint_{\mathcal{C}^q} c(t_m, \mathbf{y}) \cos(k_1^*(y_1 - a_1)) \cos(k_2^*(y_2 - a_2)) d\mathbf{y} \quad (4.12)$$

Assume that a method for finding the regions \mathcal{G}^p and \mathcal{C}^q is known (the details are discussed in section 4.1.4). We then need a way to evaluate the two terms $G_{k_1, k_2}(\mathcal{G}^p)$ and $C_{k_1, k_2}(t_m, \mathcal{C}^q)$ for $1 \leq m \leq M - 1$.

4.1.1 Finding the G_{k_1, k_2} terms

Consider now the terms $G_{k_1, k_2}([z_p, z_{p+1}] \times [w_p, w_{p+1}])$, where the variables z_p, z_{p+1}, w_p and w_{p+1} denote the corner points of the rectangular exercise region \mathcal{G}^p . We then have

$$\begin{aligned} & G_{k_1, k_2}([z_p, z_{p+1}] \times [w_p, w_{p+1}]) \\ &= \omega_1 \omega_2 \int_{w_p}^{w_{p+1}} \int_{z_p}^{z_{p+1}} I(\mathbf{y}) \cos(k_1^*(y_1 - a_1)) \cos(k_2^*(y_2 - a_2)) dy_1 dy_2 \\ &= \omega_1 \omega_2 \int_{w_p}^{w_{p+1}} \int_{z_p}^{z_{p+1}} \alpha K(e^{y_1} - 1) \cos(k_1^*(y_1 - a_1)) \cos(k_2^*(y_2 - a_2)) dy_1 dy_2 \end{aligned} \quad (4.13)$$

This integral can be obtained analytically, and using the results obtained in section 3.1 we get

$$G_{k_1, k_2}([z_p, z_{p+1}] \times [w_p, w_{p+1}]) = \omega_1 \omega_2 \alpha K [\chi_{k_1}(z_p, z_{p+1}) - \psi_{k_1}(z_p, z_{p+1})] \psi_{k_2}(w_p, w_{p+1}) \quad (4.14)$$

4.1.2 Finding the C_{k_1, k_2} terms

We will now derive an approximation for the terms $C_{k_1, k_2}(t_m, [z_q, z_{q+1}] \times [w_q, w_{q+1}])$, where the corner points of the rectangular continuation region \mathcal{C}^q are denoted by z_q, z_{q+1}, w_q and w_{q+1} .

First, note that the ChF of the Heston model can be written as follows (taking fixed time steps $\Delta t := t_{m+1} - t_m$):

$$\phi(u_1, u_2, x, v, t_m, t_{m+1}) := e^{iu_1 x} e^{B_2(u_1, u_2, \Delta t)v} \varphi_A(u_1, u_2) \quad (4.15)$$

where $\varphi_A(u_1, u_2) := e^{A(u_1, u_2, \Delta t)}$.

Then, beginning with the definition of $C_{k_1, k_2}(t_m, [z_q, z_{q+1}] \times [w_q, w_{q+1}])$ given in (4.12) and defining $j_n^* := \frac{jn\pi}{b_n - a_n}$ for $n = 1, 2$, we have (omitting the Δt arguments from B_2 for brevity):

$$\begin{aligned} & C_{k_1, k_2}(t_m, [z_q, z_{q+1}] \times [w_q, w_{q+1}]) \\ &:= \omega_1 \omega_2 \int_{w_q}^{w_{q+1}} \int_{z_q}^{z_{q+1}} c(t_m, \mathbf{y}) \cos(k_1^*(y_1 - a_1)) \cos(k_2^*(y_2 - a_2)) d\mathbf{y} \end{aligned}$$

$$\begin{aligned} &\approx \omega_1 \omega_2 \int_{w_q}^{w_{q+1}} \int_{z_q}^{z_{q+1}} e^{-r\Delta t} \sum'_{j_1=0}^{N_1-1} \sum'_{j_2=0}^{N_2-1} \frac{1}{2} \left[F_{j_1, j_2}^+(\mathbf{y}) + F_{j_1, j_2}^-(\mathbf{y}) \right] V_{j_1, j_2}(t_{m+1}) \cos(k_1^*(y_1 - a_1)) \\ &\quad \cdot \cos(k_2^*(y_2 - a_2)) d\mathbf{y} \end{aligned} \quad (4.16)$$

$$\begin{aligned} &= \omega_1 \omega_2 \int_{w_q}^{w_{q+1}} \int_{z_q}^{z_{q+1}} e^{-r\Delta t} \sum'_{j_1=0}^{N_1-1} \sum'_{j_2=0}^{N_2-1} \frac{1}{2} \left[\operatorname{Re} \left\{ \phi \left(j_1^*, j_2^* \mid y_2; \Delta t \right) \exp [ij_1^*(y_1 - a_1) - ij_2^*a_2] \right\} \right. \\ &\quad \left. + \operatorname{Re} \left\{ \phi \left(j_1^*, -j_2^* \mid y_2; \Delta t \right) \exp [ij_1^*(y_1 - a_1) + ij_2^*a_2] \right\} \right] V_{j_1, j_2}(t_{m+1}) \\ &\quad \cdot \cos(k_1^*(y_1 - a_1)) \cos(k_2^*(y_2 - a_2)) d\mathbf{y} \end{aligned} \quad (4.17)$$

$$\begin{aligned} &= \frac{1}{2} \omega_1 \omega_2 e^{-r\Delta t} \int_{w_q}^{w_{q+1}} \int_{z_q}^{z_{q+1}} \sum'_{j_1=0}^{N_1-1} \sum'_{j_2=0}^{N_2-1} \operatorname{Re} \left\{ \varphi_A(j_1^*, j_2^*) \exp [B_2(j_1^*, j_2^*) y_2] \right. \\ &\quad \left. \cdot \exp [ij_1^*(y_1 - a_1) - ij_2^*a_2] + \varphi_A(j_1^*, -j_2^*) \exp [B_2(j_1^*, -j_2^*) y_2] \cdot \exp [ij_1^*(y_1 - a_1) + ij_2^*a_2] \right\} \\ &\quad \cdot V_{j_1, j_2}(t_{m+1}) \cos(k_1^*(y_1 - a_1)) \cos(k_2^*(y_2 - a_2)) d\mathbf{y} \end{aligned} \quad (4.18)$$

$$\begin{aligned} &= \operatorname{Re} \left\{ \frac{1}{2} \omega_1 \omega_2 e^{-r\Delta t} \sum'_{j_1=0}^{N_1-1} \sum'_{j_2=0}^{N_2-1} \varphi_A(j_1^*, j_2^*) e^{-ij_2^*a_2} V_{j_1, j_2}(t_{m+1}) \right. \\ &\quad \left. \int_{w_q}^{w_{q+1}} \exp [B_2(j_1^*, j_2^*) y_2] \cos(k_2^*(y_2 - a_2)) dy_2 \int_{z_q}^{z_{q+1}} e^{ij_1^*(y_1 - a_1)} \cos(k_1^*(y_1 - a_1)) dy_1 \right\} \\ &+ \operatorname{Re} \left\{ \frac{1}{2} \omega_1 \omega_2 e^{-r\Delta t} \sum'_{j_1=0}^{N_1-1} \sum'_{j_2=0}^{N_2-1} \varphi_A(j_1^*, -j_2^*) e^{ij_2^*a_2} V_{j_1, j_2}(t_{m+1}) \right. \\ &\quad \left. \int_{w_q}^{w_{q+1}} \exp [B_2(j_1^*, -j_2^*) y_2] \cos(k_2^*(y_2 - a_2)) dy_2 \int_{z_q}^{z_{q+1}} e^{ij_1^*(y_1 - a_1)} \cos(k_1^*(y_1 - a_1)) dy_1 \right\} \end{aligned} \quad (4.19)$$

Now, if we define

$$\mathcal{M}_{k,j}(x_1, x_2, a, b) := \frac{2}{b-a} \int_{x_1}^{x_2} e^{ij\pi \frac{y-a}{b-a}} \cos\left(k\pi \frac{y-a}{b-a}\right) dy \quad (4.20)$$

and

$$H_{k,j}^{\pm}(x_1, x_2, a, b, z) := \frac{2}{b-a} \int_{x_1}^{x_2} e^{yB_2(z, \pm \frac{j\pi}{b-a})} e^{\pm ij\pi \frac{y-a}{b-a}} \cos\left(k\pi \frac{y-a}{b-a}\right) dy \quad (4.21)$$

then

$$\begin{aligned} & C_{k_1, k_2}(t_m, [z_q, z_{q+1}] \times [w_q, w_{q+1}]) \\ & \approx \operatorname{Re} \left\{ \sum_{j_1=0}^{N_1-1} \sum_{j_2=0}^{N_2-1} \frac{1}{2} e^{-r\Delta t} \varphi_A(j_1^*, j_2^*) V_{j_1, j_2}(t_{m+1}) \mathcal{M}_{k_1, j_1}(z_q, z_{q+1}, a_1, b_1) H_{k_2, j_2}^+(w_q, w_{q+1}, a_2, b_2, j_1^*) \right\} \\ & + \operatorname{Re} \left\{ \sum_{j_1=0}^{N_1-1} \sum_{j_2=0}^{N_2-1} \frac{1}{2} e^{-r\Delta t} \varphi_A(j_1^*, -j_2^*) V_{j_1, j_2}(t_{m+1}) \mathcal{M}_{k_1, j_1}(z_q, z_{q+1}, a_1, b_1) H_{k_2, j_2}^-(w_q, w_{q+1}, a_2, b_2, j_1^*) \right\} \end{aligned} \quad (4.22)$$

$$\begin{aligned} & \approx \operatorname{Re} \left\{ \sum_{j_1=0}^{N_1-1} \sum_{j_2=0}^{N_2-1} \frac{1}{2} e^{-r\Delta t} \varphi_A(j_1^*, j_2^*) \hat{V}_{j_1, j_2}(t_{m+1}) \mathcal{M}_{k_1, j_1}(z_q, z_{q+1}, a_1, b_1) H_{k_2, j_2}^+(w_q, w_{q+1}, a_2, b_2, j_1^*) \right\} \\ & + \operatorname{Re} \left\{ \sum_{j_1=0}^{N_1-1} \sum_{j_2=0}^{N_2-1} \frac{1}{2} e^{-r\Delta t} \varphi_A(j_1^*, -j_2^*) \hat{V}_{j_1, j_2}(t_{m+1}) \mathcal{M}_{k_1, j_1}(z_q, z_{q+1}, a_1, b_1) H_{k_2, j_2}^-(w_q, w_{q+1}, a_2, b_2, j_1^*) \right\} \end{aligned} \quad (4.23)$$

$$:= \hat{C}_{k_1, k_2}(t_m, [z_q, z_{q+1}] \times [w_q, w_{q+1}]) \quad (4.24)$$

where $\hat{V}_{j_1, j_2}(t_m)$ in (4.23) is defined for $m = M$ by

$$\begin{aligned} \hat{V}_{j_1, j_2}(t_M) &= V_{j_1, j_2}(t_M) \\ &= \begin{cases} G_{j_1, j_2}([0, b_1] \times [a_2, b_2]) & \text{for a call,} \\ G_{j_1, j_2}([a_1, 0] \times [a_2, b_2]) & \text{for a put.} \end{cases} \end{aligned} \quad (4.25)$$

and for $1 \leq m \leq M - 1$ is defined as

$$\hat{V}_{j_1, j_2}(t_m) = \sum_p G_{j_1, j_2}(\mathcal{G}^p) + \sum_q \hat{C}_{j_1, j_2}(t_m, \mathcal{C}^q) \quad (4.26)$$

The approximation \hat{C}_{k_1, k_2} can then be further simplified to

$$\hat{C}_{k_1, k_2}(t_m, [z_q, z_{q+1}] \times [w_q, w_{q+1}]) = \text{Re} \left\{ \sum_{j_1=0}^{N_1-1} \mathcal{M}_{k_1, j_1}(z_q, z_{q+1}, a_1, b_1) \mathcal{A}_{j_1, k_2}^q \right\} \quad (4.27)$$

where

$$\begin{aligned} \mathcal{A}_{j_1, k_2}^q &:= \sum_{j_2=0}^{N_2-1} \frac{1}{2} e^{-r\Delta t} \hat{V}_{j_1, j_2}(t_{m+1}) \left[H_{k_2, j_2}^+(w_q, w_{q+1}, a_2, b_2, j_1^*) \varphi_A(j_1^*, j_2^*) \right. \\ &\quad \left. + H_{k_2, j_2}^-(w_q, w_{q+1}, a_2, b_2, j_1^*) \varphi_A(j_1^*, -j_2^*) \right] \end{aligned} \quad (4.28)$$

4.1.3 Calculating the H^\pm terms

From (4.21) we have

$$\begin{aligned} H_{k, j}^\pm(x_1, x_2, a, b, z) &= \frac{2}{b-a} \int_{x_1}^{x_2} e^{yB_2(z, \pm \frac{j\pi}{b-a})} e^{\pm ij\pi \frac{-a}{b-a}} \cos\left(k\pi \frac{y-a}{b-a}\right) dy \\ &= \frac{2}{b-a} e^{\pm ij\pi \frac{-a}{b-a}} \int_{x_1}^{x_2} e^{yB_2(z, \pm \frac{j\pi}{b-a})} \cos\left(k\pi \frac{y-a}{b-a}\right) dy \end{aligned} \quad (4.29)$$

Now, recall from (3.32) that

$$\int_{c_1}^{c_2} e^{au} \cos(bu) du = \frac{e^{au}}{a^2 + b^2} (a \cos(bu) + b \sin(bu)) \Big|_{u=c_1}^{u=c_2}$$

and so, letting $c_1 = x_1 - a$, $c_2 = x_2 - a$, $b_1 = B_2\left(z, \pm \frac{j\pi}{b-a}\right)$ and $b_2 = \frac{k\pi}{b-a}$ gives us (omitting the arguments of B_2 for brevity)

$$\int_{x_1-a}^{x_2-a} e^{B_2 u} \cos\left(\frac{k\pi}{b-a} u\right) du = \frac{e^{B_2 u}}{B_2^2 + \left(\frac{k\pi}{b-a}\right)^2} \left[B_2 \cos\left(\frac{k\pi}{b-a} u\right) + \frac{k\pi}{b-a} \sin\left(\frac{k\pi}{b-a} u\right) \right] \Big|_{u=x_1-a}^{u=x_2-a} \quad (4.30)$$

Then, applying the methods from Section 3.1, H^\pm is given by

$$H_{k,j}^\pm(x_1, x_2, a, b, z) = \frac{2}{b-a} e^{\pm i j \pi \frac{-a}{b-a}} \chi_k^H \left(x_1, x_2, B_2 \left(z, \pm \frac{j\pi}{b-a} \right) \right) \quad (4.31)$$

where (letting $k^* = \frac{k\pi}{b-a}$)

$$\begin{aligned} \chi_k^H(x_1, x_2, y) = \\ \frac{1}{y^2 + k^{*2}} [e^{x_2 y} (y \cos(k^*(x_2 - a)) + k^* \sin(k^*(x_2 - a))) - e^{x_1 y} (y \cos(k^*(x_1 - a)) + k^* \sin(k^*(x_1 - a)))] \end{aligned} \quad (4.32)$$

4.1.4 The early-exercise and continuation regions

We have seen that in order to price Bermudan options, we need to determine rectangular continuation and early-exercise regions so that the integral obtained in (4.9) can be separated into different parts, as shown in (4.10). To do this, the domain of the second dimension (variance), $[a_2, b_2]$, is divided into J subintervals:

$$[a_2, b_2] = [w_0, w_1] \cup [w_1, w_2] \cup \dots \cup [w_q, w_{q+1}] \cup \dots \cup [w_{J-1}, w_J] \quad (4.33)$$

At the centre of each subinterval, we then find the value y^* for which the intrinsic value, $I(\mathbf{y})$, and the continuation value, $c(t_m, \mathbf{y})$, at time t_m are equal, i.e.

$$I(y^*, \frac{1}{2}(w_q + w_{q+1})) = c(t_m, y^*, \frac{1}{2}(w_q + w_{q+1})) \quad (4.34)$$

For a Bermudan put option we can then define the early-exercise and continuation regions as $\mathcal{G}^q = [a_1, y^*] \times [w_q, w_{q+1}]$ and $\mathcal{C}^q = [y^*, b_1] \times [w_q, w_{q+1}]$, respectively.

For the subintervals $[w_q, w_{q+1}]$ we can use either equidistant or non-equidistant intervals. For *equidistant* intervals we can set $w_q = a_2 + \frac{(b_2 - a_2)q}{J}$ for $q = 0, 1, \dots, J$. For *non-equidistant* intervals we can make use of the quantile function (inverse distribution function) of the variance v_T , which was shown in Section 2.1 to be governed by the non-central chi-square distribution. We set $w_0 = a_2$, $w_J = b_2$ and take $w_q = F_{v_T}^{-1}(\frac{q}{J})$ for $q = 1, \dots, J-1$. Using non-equidistant intervals generally results in more efficient pricing.

4.1.5 Finding the early-exercise points, y_m^*

At time M the option value is the same as for European options, and so from Section 3.1 we can see that $y_M^* = 0$. For $1 \leq m \leq M - 1$, if we define

$$h^q(t_m, y) := c(t_m, y, \frac{1}{2}(w_q + w_{q+1})) - I(y, \frac{1}{2}(w_q + w_{q+1})) \quad (4.35)$$

then $y_{q,m}^*$ will be the unique solution of

$$h^q(t_m, y_{q,m}^*) = 0 \quad (4.36)$$

In Fang and Oosterlee [2011], the Newton-Raphson method is proposed to approximate such a solution. Since there is only an approximation available for $c(t_m, x_1, x_2)$, we define the following function

$$\hat{h}^q(t_m, y) := \hat{c}(t_m, y, \frac{1}{2}(w_q + w_{q+1})) - I(y, \frac{1}{2}(w_q + w_{q+1})) \quad (4.37)$$

and consider $y_{q,m}^*$ to be the value of y for which this function equals zero.

When applying the Newton-Raphson method, the zero of the function is estimated by successive iterations, with the n^{th} iteration being given by

$$y_{q,m,n} = y_{q,m,n-1} - \frac{\hat{h}^q(t_m, y_{q,m,n-1})}{\left. \frac{\partial \hat{h}^q}{\partial y}(t_m, y) \right|_{y=y_{q,m,n-1}}} \quad (4.38)$$

From this expression, it is clear that we need to find the derivative of $\hat{h}^q(t_m, y)$ with respect to y . We have (letting $v^q = \frac{1}{2}(w_q + w_{q+1})$)

$$\begin{aligned} & \hat{h}^q(t_m, y) \\ &= e^{-r\Delta t} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} \frac{1}{2} \left[F_{k_1, k_2}^+(y, v^q) + F_{k_1, k_2}^-(y, v^q) \right] \hat{V}_{k_1, k_2}(t_{m+1}) - \alpha K(e^y - 1) \\ &= e^{-r\Delta t} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} \frac{1}{2} \left[\text{Re} \left\{ \phi \left(k_1^*, k_2^* \mid v^q; \Delta t \right) \exp [ik_1^*(y - a_1) - ik_2^*a_2] \right\} \right. \\ & \quad \left. + \text{Re} \left\{ \phi \left(k_1^*, -k_2^* \mid v^q; \Delta t \right) \exp [ik_1^*(y - a_1) + ik_2^*a_2] \right\} \right] \hat{V}_{k_1, k_2}(t_{m+1}) - \alpha K(e^y - 1) \quad (4.39) \end{aligned}$$

which can be differentiated to obtain the following:

$$\begin{aligned} & \frac{\partial \hat{h}^q}{\partial y}(t_m, y) \\ &= e^{-r\Delta t} \sum'_{k_1=0}^{N_1-1} \sum'_{k_2=0}^{N_2-1} \frac{1}{2} \left[\operatorname{Re} \left\{ \phi \left(k_1^*, k_2^* \mid v^q; \Delta t \right) ik_1^* \exp [ik_1^*(y - a_1) - ik_2^*a_2] \right\} \right. \\ & \quad \left. + \operatorname{Re} \left\{ \phi \left(k_1^*, -k_2^* \mid v^q; \Delta t \right) ik_1^* \exp [ik_1^*(y - a_1) + ik_2^*a_2] \right\} \right] \hat{V}_{k_1, k_2}(t_{m+1}) - \alpha K e^y \quad (4.40) \end{aligned}$$

$$\begin{aligned} &= \frac{e^{-r\Delta t} \pi}{2(b_1 - a_1)} \sum'_{k_1=0}^{N_1-1} \sum'_{k_2=0}^{N_2-1} \operatorname{Re} \left\{ ik_1 \left(\phi \left(k_1^*, k_2^* \mid v^q; \Delta t \right) \exp [ik_1^*(y - a_1) - ik_2^*a_2] \right. \right. \\ & \quad \left. \left. + \phi \left(k_1^*, -k_2^* \mid v^q; \Delta t \right) \exp [ik_1^*(y - a_1) + ik_2^*a_2] \right) \right\} \hat{V}_{k_1, k_2}(t_{m+1}) - \alpha K e^y \quad (4.41) \end{aligned}$$

$$\begin{aligned} &= \frac{-e^{-r\Delta t} \pi}{2(b_1 - a_1)} \sum'_{k_1=0}^{N_1-1} \sum'_{k_2=0}^{N_2-1} \operatorname{Im} \left\{ k_1 \left(\phi \left(k_1^*, k_2^* \mid v^q; \Delta t \right) \exp [ik_1^*(y - a_1) - ik_2^*a_2] \right. \right. \\ & \quad \left. \left. + \phi \left(k_1^*, -k_2^* \mid v^q; \Delta t \right) \exp [ik_1^*(y - a_1) + ik_2^*a_2] \right) \right\} \hat{V}_{k_1, k_2}(t_{m+1}) - \alpha K e^y \quad (4.42) \end{aligned}$$

4.1.6 Finding $\mathcal{M}_{k_1, j_1}(x_1, x_2, a_1, b_1)$

From (4.20) we have

$$\mathcal{M}_{k, j}(x_1, x_2, a, b) := \frac{2}{b - a} \int_{x_1}^{x_2} e^{ij\pi \left(\frac{x-a}{b-a} \right)} \cos \left(k\pi \frac{x-a}{b-a} \right) dx$$

Now, if we use the substitution $u = \frac{x-a}{b-a}$ we get

$$\begin{aligned} \mathcal{M}_{k, j}(x_1, x_2, a, b) &= \frac{2}{b - a} \int_{x'_1}^{x'_2} e^{ij\pi u} \cos(k\pi u) (b - a) du \\ &= 2 \int_{x'_1}^{x'_2} e^{ij\pi u} \cos(k\pi u) du \quad (4.43) \end{aligned}$$

where $x'_\alpha = \frac{x_\alpha - a}{b - a}$.

Then, by applying Euler's formula:

$$\begin{aligned}
\mathcal{M}_{k,j}(x_1, x_2, a, b) &= 2 \int_{x'_1}^{x'_2} [\cos(j\pi u) + i \sin(j\pi u)] \cos(k\pi u) du \\
&= 2 \int_{x'_1}^{x'_2} \cos(j\pi u) \cos(k\pi u) du + 2i \int_{x'_1}^{x'_2} \sin(j\pi u) \cos(k\pi u) du \quad (4.44)
\end{aligned}$$

In Stewart [2006, Table of Integrals, pp. 9] it is shown that (when $a - b \neq 0$ and $a + b \neq 0$)

$$\int_{c_1}^{c_2} \cos(au) \cos(bu) du = \frac{\sin((a-b)u)}{2(a-b)} + \frac{\sin((a+b)u)}{2(a+b)} \Big|_{u=c_1}^{u=c_2} \quad (4.45)$$

and

$$\int_{c_1}^{c_2} \sin(au) \cos(bu) du = -\frac{\cos((a-b)u)}{2(a-b)} - \frac{\cos((a+b)u)}{2(a+b)} \Big|_{u=c_1}^{u=c_2} \quad (4.46)$$

We therefore have that (if $j + k \neq 0$ and $j - k \neq 0$)

$$\begin{aligned}
&\mathcal{M}_{k,j}(x_1, x_2, a, b) \\
&= 2 \left[\frac{\sin((j\pi - k\pi)u)}{2(j\pi - k\pi)} + \frac{\sin((j\pi + k\pi)u)}{2(j\pi + k\pi)} \right]_{u=x'_1}^{u=x'_2} + 2i \left[-\frac{\cos((j\pi - k\pi)u)}{2(j\pi - k\pi)} - \frac{\cos((j\pi + k\pi)u)}{2(j\pi + k\pi)} \right]_{u=x'_1}^{u=x'_2} \\
&= \left[\frac{\sin((j-k)\pi u)}{(j-k)\pi} + \frac{\sin((j+k)\pi u)}{(j+k)\pi} \right]_{u=x'_1}^{u=x'_2} + i \left[-\frac{\cos((j-k)\pi u)}{(j-k)\pi} - \frac{\cos((j+k)\pi u)}{(j+k)\pi} \right]_{u=x'_1}^{u=x'_2} \\
&= \left[\frac{\sin((j+k)\pi u)}{(j+k)\pi} - \frac{i \cos((j+k)\pi u)}{(j+k)\pi} + \frac{\sin((j-k)\pi u)}{(j-k)\pi} - \frac{i \cos((j-k)\pi u)}{(j-k)\pi} \right]_{u=x'_1}^{u=x'_2} \\
&= -\frac{i}{\pi} \left[\frac{\cos((j+k)\pi u)}{(j+k)} - \frac{i \sin((j+k)\pi u)}{(j+k)} + \frac{\cos((j-k)\pi u)}{(j-k)} - \frac{i \sin((j-k)\pi u)}{(j-k)} \right]_{u=x'_1}^{u=x'_2} \\
&= -\frac{i}{\pi} \left[\frac{1}{j+k} \exp[i(j+k)\pi u] + \frac{1}{j-k} \exp[i(j-k)\pi u] \right]_{u=x'_1}^{u=x'_2} \quad (4.47)
\end{aligned}$$

When $j, k = 0$ we have

$$\begin{aligned}
\mathcal{M}_{0,0}(x_1, x_2, a, b) &= 2 \int_{x'_1}^{x'_2} [\cos^2(0) + i \sin(0) \cos(0)] du \\
&= 2 \int_{x'_1}^{x'_2} 1 du \\
&= 2(x'_2 - x'_1) \\
&= 2 \frac{x_2 - x_1}{b - a} \\
&= -\frac{i}{\pi} \left[i\pi \frac{x_2 - x_1}{b - a} + i\pi \frac{x_2 - x_1}{b - a} \right]
\end{aligned} \tag{4.48}$$

and for $j = k$ and $j, k \neq 0$ we have

$$\begin{aligned}
\mathcal{M}_{k,k}(x_1, x_2, a, b) &= 2 \int_{x'_1}^{x'_2} [\cos^2(k\pi u) + i \sin(k\pi u) \cos(k\pi u)] du \\
&= 2 \int_{x'_1}^{x'_2} \left(\frac{1 + \cos(2k\pi u)}{2} + i \frac{\sin(2k\pi u)}{2} \right) du \\
&= 2 \int_{x'_1}^{x'_2} (1 + \exp(2k\pi u)) du \\
&= \frac{x_2 - x_1}{b - a} + \left[\frac{1}{2k\pi i} \exp(2k\pi u) \right]_{u=x'_1}^{u=x'_2} \\
&= -\frac{i}{\pi} \left(i\pi \frac{x_2 - x_1}{b - a} + \left[\frac{1}{j + k} \exp[i(j + k)\pi u] \right]_{u=x'_1}^{u=x'_2} \right)
\end{aligned} \tag{4.49}$$

Considering the above results, we therefore have

$$\mathcal{M}_{k,j}(x_1, x_2, a, b) = -\frac{i}{\pi} [\mathcal{M}_{k,j}^c(x_1, x_2, a, b) + \mathcal{M}_{k,j}^s(x_1, x_2, a, b)] \tag{4.50}$$

where

$$\mathcal{M}_{k,j}^c(x_1, x_2, a, b) = \begin{cases} i\pi \frac{x_2 - x_1}{b - a} & \text{for } k, j = 0, \\ \frac{1}{j+k} \left[\exp[i(j+k) \frac{x_2 - a}{b - a} \pi] - \exp[i(j+k) \frac{x_1 - a}{b - a} \pi] \right] & \text{otherwise.} \end{cases} \quad (4.51)$$

$$\mathcal{M}_{k,j}^s(x_1, x_2, a, b) = \begin{cases} i\pi \frac{x_2 - x_1}{b - a} & \text{for } k = j, \\ \frac{1}{j-k} \left[\exp[i(j-k) \frac{x_2 - a}{b - a} \pi] - \exp[i(j-k) \frac{x_1 - a}{b - a} \pi] \right] & \text{for } k \neq j. \end{cases} \quad (4.52)$$

4.2 Use of the Fast Fourier Transform

Using (4.50), (4.51) and (4.52) we can now re-write $\hat{C}_{k_1, k_2}(t_m, [z_q, z_{q+1}] \times [w_q, w_{q+1}])$ as follows:

$$\begin{aligned} \hat{C}_{k_1, k_2}(t_m, [z_q, z_{q+1}] \times [w_q, w_{q+1}]) &= \operatorname{Re} \left\{ \sum_{j_1=0}^{N_1-1} \mathcal{M}_{k_1, j_1}(z_q, z_{q+1}, a_1, b_1) \mathcal{A}_{j_1, k_2}^q \right\} \\ &= \operatorname{Re} \left\{ -\frac{i}{\pi} \sum_{j_1=0}^{N_1-1} [\mathcal{M}_{k_1, j_1}^c(z_q, z_{q+1}, a_1, b_1) + \mathcal{M}_{k_1, j_1}^s(z_q, z_{q+1}, a_1, b_1)] \mathcal{A}_{j_1, k_2}^q \right\} \\ &= \frac{1}{\pi} \operatorname{Im} \left\{ \sum_{j_1=0}^{N_1-1} [\mathcal{M}_{k_1, j_1}^c(z_q, z_{q+1}, a_1, b_1) + \mathcal{M}_{k_1, j_1}^s(z_q, z_{q+1}, a_1, b_1)] \mathcal{A}_{j_1, k_2}^q \right\} \end{aligned} \quad (4.53)$$

Now, if we define the following matrices for $m = M - 1, M - 2, \dots, 1$:

$$\begin{aligned} \underline{\hat{C}}(t_m, \mathcal{C}^q) &:= \left\{ \hat{C}_{k_1, k_2}(t_m, \mathcal{C}^q) \right\}_{k_1, k_2=0}^{N_1-1, N_2-1} \\ \underline{\mathcal{M}}^c(x_1, x_2, a, b) &:= \left\{ \mathcal{M}_{k_1, j_1}^c(x_1, x_2, a, b) \right\}_{k_1, j_1=0}^{N_1-1} \\ \underline{\mathcal{M}}^s(x_1, x_2, a, b) &:= \left\{ \mathcal{M}_{k_1, j_1}^s(x_1, x_2, a, b) \right\}_{k_1, j_1=0}^{N_1-1} \\ \underline{\mathcal{A}}^q(x_1, x_2, t_{m+1}) &:= \left\{ p_{j_1} \mathcal{A}_{j_1, k_2}^q(x_1, x_2, t_{m+1}) \right\}_{j_1, k_2=0}^{N_1-1, N_2-1} \end{aligned}$$

(where $p_0 = \frac{1}{2}$ and $p_j = 1$ for $j > 0$) we then have

$$\hat{\underline{C}}(t_m, \mathcal{C}^q) = \frac{1}{\pi} \text{Im} \left\{ \left[\underline{\mathcal{M}}^c(z_q, z_{q+1}, a_1, b_1) + \underline{\mathcal{M}}^s(z_q, z_{q+1}, a_1, b_1) \right] \underline{\mathcal{A}}^q(w_q, w_{q+1}, t_{m+1}) \right\} \quad (4.54)$$

4.2.1 Calculation of \hat{C} using convolutions

If we define

$$m_j(x_1, x_2, a, b) := \begin{cases} i\pi \frac{x_2 - x_1}{b - a} & \text{for } j = 0, \\ \frac{1}{j} \left[\exp(iju\pi) \right]_{u=\frac{x_1-a}{b-a}}^{u=\frac{x_2-a}{b-a}} & \text{for } 1 - N_1 \leq j \leq -1, 1 \leq j \leq 2N_1 - 1. \end{cases} \quad (4.55)$$

then the matrices $\underline{\mathcal{M}}^c$ and $\underline{\mathcal{M}}^s$ can be represented in the following manner:

$$\underline{\mathcal{M}}^c(z_q, z_{q+1}, a_1, b_1) = \begin{pmatrix} m_0 & m_1 & \dots & m_{N_1-2} & m_{N_1-1} \\ m_1 & m_2 & \dots & m_{N_1-1} & m_{N_1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ m_{N_1-2} & m_{N_1-1} & \dots & m_{2N_1-4} & m_{2N_1-3} \\ m_{N_1-1} & m_{N_1} & \dots & m_{2N_1-3} & m_{2N_1-2} \end{pmatrix}$$

$$\underline{\mathcal{M}}^s(z_q, z_{q+1}, a_1, b_1) = \begin{pmatrix} m_0 & m_1 & \dots & m_{N_1-2} & m_{N_1-1} \\ m_{-1} & m_0 & \dots & m_{N_1-3} & m_{N_1-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ m_{2-N_1} & m_{3-N_1} & \dots & m_0 & m_1 \\ m_{1-N_1} & m_{2-N_1} & \dots & m_{-1} & m_0 \end{pmatrix}$$

where the arguments for each m_j have been omitted for brevity.

When presented in this manner, it can be seen that $\underline{\mathcal{M}}^c$ and $\underline{\mathcal{M}}^s$ are in the form of a ‘‘Hankel’’ matrix and a ‘‘Toeplitz’’ matrix, respectively. A property of each of these types of matrix is that we can transform the product of a vector with either of these matrices into a circular convolution.

Define the following vectors:

$$\underline{m}^s(x_1, x_2) := [m_0, m_{-1}, \dots, m_{1-N_1}, 0, m_{N_1-1}, m_{N_1-2}, \dots, m_1]^T \quad (4.56)$$

$$\underline{u}_j^s(t_{m+1}) := [\hat{A}_{0,j}^q, \hat{A}_{1,j}^q, \dots, \hat{A}_{N_1-1,j}^q, \underbrace{0, \dots, 0}_{N_1 \text{ terms}}]^T \quad (4.57)$$

$$\underline{m}^c(x_1, x_2) := [m_{2N_1-1}, m_{2N_1-2}, \dots, m_1, m_0]^T \quad (4.58)$$

$$\underline{u}_j^c(t_{m+1}) := [\underbrace{0, \dots, 0}_{N_1 \text{ terms}}, \hat{A}_{0,j}^q, \hat{A}_{1,j}^q, \dots, \hat{A}_{N_1-1,j}^q]^T \quad (4.59)$$

It can then be shown that (for $1 \leq j \leq N_1$)

$$\underline{\mathcal{M}}^s(x_1, x_2, a, b) \underline{\mathcal{A}}_{.,j}^q(y_1, y_2, t_{m+1}) = \text{first } N_1 \text{ elements of } \{\underline{m}^s(x_1, x_2) * \underline{u}_j^s(t_{m+1})\} \quad (4.60)$$

$$\underline{\mathcal{M}}^c(x_1, x_2, a, b) \underline{\mathcal{A}}_{.,j}^q(y_1, y_2, t_{m+1}) = \text{first } N_1 \text{ elements of } \{\underline{m}^c(x_1, x_2) * \underline{u}_j^c(t_{m+1})\} \text{ in reverse order} \quad (4.61)$$

where $\underline{\mathcal{A}}_{.,j}^q(y_1, y_2, t_{m+1})$ denotes the j th column of the matrix $\underline{\mathcal{A}}^q(y_1, y_2, t_{m+1})$. The discrete circular convolution of two vectors is indicated by the binary operator $*$, and is defined as follows:

For two vectors \underline{a} and \underline{b} , both of length N , the n th element of $\underline{a} * \underline{b}$ is given by

$$\{\underline{a} * \underline{b}\}^{[n]} := \sum_{j=1}^N \underline{a}(j) \underline{b}(n - j + 1) \quad (4.62)$$

A proof for the result of (4.60) can be found in Van Loan [1992, §4.2.4], while a verification of the result of (4.61) can be found in Appendix B.1.

Now, if we apply the discrete Time Convolution Theorem from Matsuda [2004, pp. 80], we are then able to use the FFT for efficient calculation of the right-hand sides of (4.60) and (4.61):

$$\underline{m}^s(x_1, x_2) * \underline{u}_j^s(t_{m+1}) = \underline{\mathcal{D}}^{-1} \left\{ \underline{\mathcal{D}}\{\underline{m}^s(x_1, x_2)\} \cdot \underline{\mathcal{D}}\{\underline{u}_j^s(t_{m+1})\} \right\} \quad (4.63)$$

$$\underline{m}^c(x_1, x_2) * \underline{u}_j^c(t_{m+1}) = \underline{\mathcal{D}}^{-1} \left\{ \underline{\mathcal{D}}\{\underline{m}^c(x_1, x_2)\} \cdot \underline{\mathcal{D}}\{\underline{u}_j^c(t_{m+1})\} \right\} \quad (4.64)$$

where $\underline{\mathcal{D}}\{\underline{f}\} = \left\{ \mathcal{D}_j\{\underline{f}\} \right\}_{j=0}^{2N_1-1}$ and $\underline{\mathcal{D}}^{-1}\{\underline{g}\} = \left\{ \mathcal{D}_n^{-1}\{\underline{g}\} \right\}_{n=0}^{2N_1-1}$, with $\mathcal{D}_j\{\}$ and $\mathcal{D}_n^{-1}\{\}$ defined as follows:

$$\mathcal{D}_j\{\underline{f}\} = \sum_{n=0}^{N-1} e^{ijn\frac{2\pi}{N}} f_n \quad (4.65)$$

$$\mathcal{D}_n^{-1}\{\underline{g}\} = \frac{1}{N} \sum_{j=0}^{N-1} e^{-ijn\frac{2\pi}{N}} g_j \quad (4.66)$$

for sequences $\underline{f} := \{f_n\}_{n=0}^{N-1}$ and $\underline{g} := \{g_j\}_{j=0}^{N-1}$.

4.2.2 Reducing the computational complexity

Fang and Oosterlee [2009] observed a few ways in which the computational complexity of calculating the matrix $\underline{\underline{\hat{C}}}$ can be reduced.

From (4.55) we have, for $j \neq 0$:

$$\begin{aligned} m_j(x_1, x_2, a, b) &= \frac{1}{j} \left[\exp(iju\pi) \right]_{u=x'_1}^{u=x'_2} \\ &= \frac{1}{j} \left[\cos(ju\pi) + i \sin(ju\pi) \right]_{u=x'_1}^{u=x'_2} \end{aligned}$$

and for $m_{-j}(x_1, x_2, a, b)$ we then have

$$\begin{aligned} m_{-j}(x_1, x_2, a, b) &= -\frac{1}{j} \left[\cos(-ju\pi) + i \sin(-ju\pi) \right]_{u=x'_1}^{u=x'_2} \\ &= -\frac{1}{j} \left[\cos(ju\pi) - i \sin(ju\pi) \right]_{u=x'_1}^{u=x'_2} \\ &= -\overline{m_j(x_1, x_2, a, b)} \end{aligned} \quad (4.67)$$

If we write m_{j+N_1} in its full form, we have

$$\begin{aligned}
m_{j+N_1}(x_1, x_2, a, b) &= \frac{\exp\left[i(j+N_1)\frac{x_2-a}{b-a}\pi\right] - \exp\left[i(j+N_1)\frac{x_1-a}{b-a}\pi\right]}{j+N_1} \\
&= \frac{\exp\left[ij\frac{x_2-a}{b-a}\pi\right] \exp\left[iN_1\frac{x_2-a}{b-a}\pi\right] - \exp\left[ij\frac{x_1-a}{b-a}\pi\right] \exp\left[iN_1\frac{x_1-a}{b-a}\pi\right]}{j+N_1} \quad (4.68)
\end{aligned}$$

Since the multiplication operation is computationally cheaper than exponentiation, it will be more efficient to perform a single computation for the following terms:

$$\exp\left(ij\frac{x_1-a}{b-a}\pi\right) \text{ and } \exp\left(ij\frac{x_2-a}{b-a}\pi\right) \quad \text{for } 1 \leq j \leq N_1 \quad (4.69)$$

Then, using these properties we can calculate $m_j(\cdot)$ as follows:

- directly from the definition in (4.55) for $0 \leq j \leq N_1$
- using the result above with (4.68) for $N_1 + 1 \leq j \leq 2N_1 - 1$
- applying (4.67) for $1 - N_1 \leq j \leq -1$

It can also be shown, using the shifting property of the discrete Fourier transform, that

$$\underline{\mathcal{D}}\{\underline{u}_j^c(t_m + 1)\} = \underline{\zeta} \cdot \underline{\mathcal{D}}\{\underline{u}_j^s(t_m + 1)\} \quad (4.70)$$

where

$$\underline{\zeta} = \{(-1)^k\}_{k=0}^{2N_1-1} \quad (4.71)$$

A proof of this result can be found in Appendix B.2.

4.3 Algorithm for pricing Bermudan options

Using all of the results obtained in this section, we are now able to present an algorithm for the pricing of Bermudan options using the 2D-COS method:

Initialisation

- Calculate the truncation ranges, $[a_1, b_1]$ and $[a_2, b_2]$, using (3.54) and (3.55), respectively.
- For $q = 0, 1, 2, \dots, J$, calculate w_q as discussed in Section 4.1.4.
- For $q = 0, 1, \dots, J-1$, $k_2 = 0, 1, \dots, N_2$, and $j_i = 0, 1, \dots, N_i$, calculate $H_{k_2, j_2}^\pm(w_q, w_{q+1}, \frac{j_1 \pi}{b_1 - a_1})$ using (4.21).
- For $k_i = 0, 1, \dots, N_i - 1$, calculate

$$V_{k_1, k_2}(t_M) = \begin{cases} G_{k_1, k_2}([0, b_1] \times [a_2, b_2]) & \text{for a call} \\ G_{k_1, k_2}([0, b_1] \times [a_2, b_2]) & \text{for a put} \end{cases} \quad (4.72)$$

where $G_{k_1, k_2}([x_1, x_2] \times [y_1, y_2])$ is calculated using (4.14).

Loop To recover $\hat{V}_{k_1, k_2}(t_m)$ recursively for $m = M - 1$ to 1

- Determine the early-exercise regions, \mathcal{G}^p , and continuation regions, \mathcal{C}^q , using the procedure outlined in Section 4.1.5.
- For $q = 0, 1, \dots, J - 1$, $j_1 = 0, 1, \dots, N_1$ and $k_2 = 0, 1, \dots, N_2$, calculate \mathcal{A}_{j_1, k_2}^q using (4.28).
- Compute $m_j(x_1, x_2)$ for $j = 0, 1, \dots, N_1 - 1$ using (4.55).
- Construct $\underline{m}^s(x_1, x_2)$ and $\underline{m}^c(x_1, x_2)$ using the properties of the m_j 's as outlined in Section 4.2.2.
- Construct $\underline{u}^s(t_m)$ by padding $N_1 \times N_2$ zeroes to $\hat{\underline{A}}^q$
- Calculate $\xi_j^s = \mathcal{D}^{-1}\{\mathcal{D}(\underline{m}^s) \cdot \mathcal{D}(\underline{u}_{\cdot, j}^s)\}$ and $\xi_j^c = \mathcal{D}^{-1}\{\mathcal{D}(\underline{m}^c) \cdot \underline{\zeta} \cdot \mathcal{D}(\underline{u}_{\cdot, j}^s)\}$ for $j = 1, 2, \dots, N_2$
- $\underline{M}su_{\cdot, j} =$ the first N_1 elements of ξ_j^s
- $\underline{M}cu_{\cdot, j} =$ the first N_1 elements of ξ_j^c in reverse order.
- $\hat{\underline{C}}(t_m, \mathcal{C}^q) = \frac{1}{\pi} \text{Im} \left\{ \underline{M}su + \underline{M}cu \right\}$

- Calculate $G_{k_1, k_2}(\mathcal{G}^p)$ using (4.14)
- Calculate $\hat{V}_{k_1, k_2}(t_m)$ by

$$\hat{V}_{k_1, k_2}(t_m) = \sum_p G_{k_1, k_2}(\mathcal{G}^p) + \sum_q C_{k_1, k_2}(t_m, \hat{\mathcal{C}}^q) \quad (4.73)$$

Final Step

- Calculate $\hat{v}(x, v_0, t_0)$ by inserting $\hat{V}_{k_1, k_2}(t_1)$ into (4.5)

5 Discrete Barrier Options

In this section we adapt the 2D-COS Method for Bermudan options from Section 4 so as to price discretely monitored barrier options under the dynamics of the Heston model. A discretely-monitored “out” barrier option is an option that ceases to exist if the underlying asset price hits a certain barrier level, H , at any of the pre-specified observation dates. If $H > S_0$, it is called an “up-and-out” option, and if $H < S_0$ it is called a “down-and-out” options. Other variants include “in” barrier options and double barrier options (with barriers both above and below S_0). We will focus on “out” barrier options with a single barrier for the development of the pricing method, however, it can easily be adapted to handle the other variants.

The payoff for an up-and-out option is given by

$$v(T, S_T) = \left((\alpha(S_T - K))^+ - Rb \right) \mathbf{1}_{\{S(t_i) < H\}} + Rb \quad (5.1)$$

where $\alpha = 1$ for a call and $\alpha = -1$ for a put, Rb is a rebate, the t_i 's are the observation dates and $\mathbf{1}_A$ is the indicator function, given by

$$\mathbf{1}_A = \begin{cases} 1 & \text{if } A \text{ is not empty,} \\ 0 & \text{otherwise.} \end{cases} \quad (5.2)$$

Using the (two-dimensional) Heston model in terms of the logarithm of the stock price as given by (2.1) and (2.2), the payoff can then be written as

$$v(T, x_1(T), x_2(T)) = \left((\alpha K(e^{x_1(T)} - 1))^+ - Rb \right) \mathbf{1}_{\{x_1(t_i) < h\}} + Rb \quad (5.3)$$

where $h := \ln(\frac{H}{K})$.

Thus, at the maturity date, t_M , the option price equals the payoff of the option if the barrier has not been reached at any of the observation dates; otherwise the option price equals the rebate.

For $m = 0, 1, \dots, M - 1$, the pricing formula is

$$v(t_m, \mathbf{x}) = \begin{cases} e^{-r\Delta t} Rb & \text{if } x_1 \geq h, \\ c(t_m, \mathbf{x}) & \text{if } x_1 < h. \end{cases} \quad (5.4)$$

where $c(t_m, \mathbf{x})$ is the continuation value, which is defined in the same way as for Bermudan options in (4.2):

$$\begin{aligned} c(t_m, \mathbf{x}_m) &= e^{-r\Delta t} \mathbb{E}_{t_m}^{\mathbb{Q}} [v(t_{m+1}, \mathbf{x}_{m+1})] \\ &\approx e^{-r\Delta t} \sum'_{k_1=0}^{N_1-1} \sum'_{k_2=0}^{N_2-1} \frac{1}{2} \left[F_{k_1, k_2}^+(\mathbf{x}_m) + F_{k_1, k_2}^-(\mathbf{x}_m) \right] V_{k_1, k_2}(t_{m+1}) \end{aligned}$$

5.1 Evaluating the V_{k_1, k_2} terms

As with Bermudan options, the coefficients V_{k_1, k_2} can be recovered recursively for barrier options.

We begin with the final time point, where we have the terminal condition given in (5.3). This is similar to the European and Bermudan cases, and can thus be calculated using the same methods:

For an up-and-out option:

$$\begin{aligned} V_{k_1, k_2}(t_M) &= \int_{a_2}^{b_2} \int_{a_1}^{b_1} v(t_M, y_1, y_2) \cos(k_1^*(y_1 - a_1)) \cos(k_2^*(y_2 - a_2)) dy_1 dy_2 \\ &= \int_{a_2}^{b_2} \int_{a_1}^{b_1} \left[\left((\alpha K(e^{y_1} - 1))^+ - Rb \right) \mathbf{1}_{\{y_1 < h\}} + Rb \right] \cos(k_1^*(y_1 - a_1)) \cos(k_2^*(y_2 - a_2)) dy_1 dy_2 \\ &= \int_{a_2}^{b_2} \cos(k_2^*(y_2 - a_2)) dy_2 \int_{a_1}^{b_1} \left[\left((\alpha K(e^{y_1} - 1))^+ - Rb \right) \mathbf{1}_{\{y_1 < h\}} + Rb \right] \cos(k_1^*(y_1 - a_1)) dy_1 \\ &= \int_{a_2}^{b_2} \cos(k_2^*(y_2 - a_2)) dy_2 \left[\int_{a_1}^h \left((\alpha K(e^{y_1} - 1))^+ - Rb \right) \cos(k_1^*(y_1 - a_1)) dy_1 \right. \\ &\quad \left. + \int_{a_1}^{b_1} Rb \cos(k_1^*(y_1 - a_1)) dy_1 \right] \\ &= \int_{a_2}^{b_2} \cos(k_2^*(y_2 - a_2)) dy_2 \left[\int_{a_1}^h (\alpha K(e^{y_1} - 1))^+ \cos(k_1^*(y_1 - a_1)) dy_1 \right. \\ &\quad \left. + Rb \int_h^{b_1} \cos(k_1^*(y_1 - a_1)) dy_1 \right] \end{aligned} \tag{5.5}$$

Now, using the results obtained in Section 3.1, this can be simplified as follows:

$$V_{k_1, k_2}^{\text{call}}(t_M) = \begin{cases} \omega_1 \omega_2 R b \psi_{k_1}(h, b) \psi_{k_2}(a_2, b_2) & \text{if } h < 0, \\ \omega_1 \omega_2 K [\chi_{k_1}(0, h) - \psi_{k_1}(0, h)] \psi_{k_2}(a_2, b_2) + \omega_1 \omega_2 R b \psi_{k_1}(h, b) \psi_{k_2}(a_2, b_2) & \text{if } h \geq 0. \end{cases} \quad (5.6)$$

and

$$V_{k_1, k_2}^{\text{put}}(t_M) = \begin{cases} \omega_1 \omega_2 K [\psi_{k_1}(a, h) - \chi_{k_1}(a, h)] \psi_{k_2}(a_2, b_2) + \omega_1 \omega_2 R b \psi_{k_1}(h, b) \psi_{k_2}(a_2, b_2) & \text{if } h < 0, \\ \omega_1 \omega_2 K [\psi_{k_1}(a, 0) - \chi_{k_1}(a, 0)] \psi_{k_2}(a_2, b_2) + \omega_1 \omega_2 R b \psi_{k_1}(h, b) \psi_{k_2}(a_2, b_2) & \text{if } h \geq 0. \end{cases} \quad (5.7)$$

Similarly, for down-and-out options we have

$$V_{k_1, k_2}^{\text{call}}(t_M) = \begin{cases} \omega_1 \omega_2 K [\chi_{k_1}(0, b_1) - \psi_{k_1}(0, h)] \psi_{k_2}(a_2, b_2) + \omega_1 \omega_2 R b \psi_{k_1}(a, h) \psi_{k_2}(a_2, b_2) & \text{if } h < 0, \\ \omega_1 \omega_2 K [\chi_{k_1}(h, b_1) - \psi_{k_1}(0, h)] \psi_{k_2}(a_2, b_2) + \omega_1 \omega_2 R b \psi_{k_1}(a, h) \psi_{k_2}(a_2, b_2) & \text{if } h \geq 0. \end{cases} \quad (5.8)$$

and

$$V_{k_1, k_2}^{\text{put}}(t_M) = \begin{cases} \omega_1 \omega_2 K [\psi_{k_1}(h, 0) - \chi_{k_1}(a, h)] \psi_{k_2}(a_2, b_2) + \omega_1 \omega_2 R b \psi_{k_1}(a, h) \psi_{k_2}(a_2, b_2) & \text{if } h < 0, \\ \omega_1 \omega_2 R b \psi_{k_1}(a, h) \psi_{k_2}(a_2, b_2) & \text{if } h \geq 0. \end{cases} \quad (5.9)$$

For the $V_{k_1, k_2}(t_m)$ coefficients required to approximate the continuation values at times t_{M-2}, \dots, t_0 , we use (5.4) to obtain

$$\begin{aligned} V_{k_1, k_2}(t_m) &= \omega_1 \omega_2 \int_{a_2}^{b_2} \int_{a_1}^{b_1} v(t_m, y_1, y_2) \cos(k_1^*(y_1 - a_1)) \cos(k_2^*(y_2 - a_2)) dy_1 dy_2 \\ &= \omega_1 \omega_2 \int_{a_2}^{b_2} \left[\int_{a_1}^h c(t_m, y_1, y_2) \cos(k_1^*(y_1 - a_1)) dy_1 + \int_h^{b_1} e^{-r(T-t_m)} R b \cos(k_1^*(y_1 - a_1)) dy_1 \right] \\ &\quad \cdot \cos(k_2^*(y_2 - a_2)) dy_2 \end{aligned}$$

$$\begin{aligned}
&= \omega_1 \omega_2 \int_{a_2}^{b_2} \int_{a_1}^h c(t_m, y_1, y_2) \cos(k_1^*(y_1 - a_1)) \cos(k_2^*(y_2 - a_2)) dy_1 dy_2 \\
&\quad + \omega_1 \omega_2 e^{-r(T-t_m)} Rb \psi_{k_1}(h, b_1) \psi_{k_2}(a_2, b_2) \\
&= C_{k_1, k_2}(t_m, a_1, h) + \omega_1 \omega_2 e^{-r(T-t_m)} Rb \psi_{k_1}(h, b_1) \psi_{k_2}(a_2, b_2)
\end{aligned} \tag{5.10}$$

where

$$C_{k_1, k_2}(t_m, x_1, x_2) := \omega_1 \omega_2 \int_{a_2}^{b_2} \int_{x_1}^{x_2} c(t_m, y_1, y_2) \cos(k_1^*(y_1 - a_1)) \cos(k_2^*(y_2 - a_2)) dy_1 dy_2 \tag{5.11}$$

Notice that for barrier options there is no need for a root-finding algorithm, as the barrier points are already known.

5.2 Finding the C_{k_1, k_2} terms

Recall from (4.15) that we can write the ChF for the Heston model as

$$\phi(u_1, u_2, x, v, t_m, t_{m+1}) := e^{iu_1 x} e^{B_2(u_1, u_2, \Delta t)v} \varphi_A(u_1, u_2)$$

where $\varphi_A(u_1, u_2) := e^{A(u_1, u_2, \Delta t)}$.

Now, in Section 4.1.2 we showed that

$$\begin{aligned}
&C_{k_1, k_2}(t_m, x_1, x_2) \\
&:= \omega_1 \omega_2 \int_{a_2}^{b_2} \int_{x_1}^{x_2} c(t_m, y_1, y_2) \cos(k_1^*(y_1 - a_1)) \cos(k_2^*(y_2 - a_2)) dy_1 dy_2 \\
&\approx \text{Re} \left\{ \sum_{j_1=0}^{N_1-1} \sum_{j_2=0}^{N_2-1} \frac{1}{2} e^{-r\Delta t} \varphi_A(j_1^*, j_2^*) \hat{V}_{j_1, j_2}(t_{m+1}) \mathcal{M}_{k_1, j_1}(x_1, x_2, a_1, b_1) H_{k_2, j_2}^+(a_2, b_2, a_2, b_2, j_1^*) \right\} \\
&\quad + \text{Re} \left\{ \sum_{j_1=0}^{N_1-1} \sum_{j_2=0}^{N_2-1} \frac{1}{2} e^{-r\Delta t} \varphi_A(j_1^*, -j_2^*) \hat{V}_{j_1, j_2}(t_{m+1}) \mathcal{M}_{k_1, j_1}(x_1, x_2, a_1, b_1) H_{k_2, j_2}^-(a_2, b_2, a_2, b_2, j_1^*) \right\} \\
&:= \hat{C}_{k_1, k_2}(t_m, x_1, x_2)
\end{aligned}$$

where

$$\mathcal{M}_{k,j}(x_1, x_2, a, b) := \frac{2}{b-a} \int_{x_1}^{x_2} e^{ij\pi\left(\frac{y-a}{b-a}\right)} \cos\left(k\pi\frac{y-a}{b-a}\right) dy$$

$$H_{k,j}^{\pm}(x_1, x_2, a, b, z) := \frac{2}{b-a} \int_{x_1}^{x_2} e^{yB_2(z, \pm\frac{j\pi}{b-a})} e^{\pm ij\pi\frac{-a}{b-a}} \cos\left(k\pi\frac{y-a}{b-a}\right) dy$$

and

$$\hat{V}_{j_1, j_2}(t_m) = \begin{cases} V_{j_1, j_2}(t_M) & \text{for } m = M, \\ \hat{C}_{k_1, k_2}(t_m, a_1, h) + \omega_1 \omega_2 e^{-r(T-t_m)} Rb \psi_{k_1}(h, b_1) \psi_{k_2}(a_2, b_2) & \text{for } 1 \leq m \leq M-1. \end{cases} \quad (5.12)$$

If we further define

$$\mathcal{A}_{j_1, k_2} := \sum_{j_2=0}^{N_2-1} \frac{1}{2} e^{-r\Delta t} \hat{V}_{j_1, j_2}(t_{m+1}) \left[H_{k_2, j_2}^+ (a_2, b_2, a_2, b_2, j_1^*) \varphi_A(j_1^*, j_2^*) + H_{k_2, j_2}^- (a_2, b_2, a_2, b_2, j_1^*) \varphi_A(j_1^*, -j_2^*) \right] \quad (5.13)$$

then we can simplify \hat{C}_{k_1, k_2} further to

$$\hat{C}_{k_1, k_2}(t_m, x_1, x_2) = \text{Re} \left\{ \sum_{j_1=0}^{N_1-1} \mathcal{M}_{k_1, j_1}(x_1, x_2, a_1, b_1) \mathcal{A}_{j_1, k_2} \right\} \quad (5.14)$$

The $H_{k,j}^{\pm}$ and $\mathcal{M}_{k,j}$ terms can then be calculated as shown in Sections 4.1.3 and 4.1.6, respectively. We are also able to apply the computational improvements, including the use of the Fast Fourier Transform, of Section 4.2. However, because the barrier points are known and a root-finding algorithm is not necessary, $\psi_{k_1, k_2}(h, b)$, \mathcal{M}_c and \mathcal{M}_s can be calculated in the initialisation, rather than during the recursion loops.

5.3 Algorithm for pricing barrier options

Initialisation

- Calculate the truncation ranges, $[a_1, b_1]$ and $[a_2, b_2]$, using (3.54) and (3.55), respectively.
- For $k_2 = 0, 1, \dots, N_2$, and $j_i = 0, 1, \dots, N_i$, calculate $H_{k_2, j_2}^\pm(a_2, b_2, \frac{j_1 \pi}{b_1 - a_1})$ using (4.21).
- For $k_i = 0, 1, \dots, N_i - 1$, calculate $V_{k_1, k_2}(t_M)$ using (5.6) or (5.7).
- For up-and-out options: $x_1 = a_1, x_2 = h, c = h$ and $d = b_1$.
For down-and-out options: $x_1 = h, x_2 = b_1, c = a_1$ and $d = h$.
- Compute $m_j(x_1, x_2)$ for $j = 0, 1, \dots, N_1 - 1$ using (4.55).
- Construct $\underline{m}^s(x_1, x_2)$ and $\underline{m}^c(x_1, x_2)$ using the properties of the m_j 's as outlined in Section 4.2.2.
- Calculate $d_1 = \mathcal{D}\{\underline{m}^s(x_1, x_2)\}$ and $d_2 = \underline{\zeta} \cdot \mathcal{D}\{\underline{m}^c(x_1, x_2)\}$
- For $k_i = 0, 1, \dots, N_i - 1$, calculate $G_{k_1, k_2} = \omega_1 \omega_2 R b \psi_{k_1}(h, b_1) \psi_{k_2}(a_2, b_2)$

Loop To recover $\hat{V}_{k_1, k_2}(t_m)$ recursively for $m = M - 1$ to 1

- For $j_1 = 0, 1, \dots, N_1$ and $k_2 = 0, 1, \dots, N_2$, calculate \mathcal{A}_{j_1, k_2} using (5.13).
- Construct $\underline{u}^s(t_m)$ by padding $N_1 \times N_2$ zeroes to \hat{A}
- Calculate $\xi_j^s = \mathcal{D}^{-1}\{d_1 \cdot \mathcal{D}(\underline{u}^s_{\cdot, j})\}$ and $\xi_j^c = \mathcal{D}^{-1}\{d_2 \cdot \mathcal{D}(\underline{u}^s_{\cdot, j})\}$ for $j = 1, 2, \dots, N_2$
- $\underline{Msu}_{\cdot, j}$ = the first N_1 elements of ξ_j^s
- $\underline{Mcu}_{\cdot, j}$ = the first N_1 elements of ξ_j^c in reverse order.
- $\hat{C}(t_m, x_1, x_2) = \frac{1}{\pi} \text{Im} \left\{ \underline{Msu} + \underline{Mcu} \right\}$
- Calculate $\hat{V}_{k_1, k_2}(t_m)$ by

$$\hat{V}_{k_1, k_2}(t_m) = G_{k_1, k_2} + \hat{C}_{k_1, k_2}(t_m, x_1, x_2) \quad (5.15)$$

Final Step

- Calculate $\hat{v}(x, v_0, t_0)$ by inserting $\hat{V}_{k_1, k_2}(t_1)$ into (4.5)

6 Numerical Results

In this section, a number of numerical tests will be performed in order to evaluate the accuracy and efficiency of the 2D-COS method for the pricing of European, Bermudan and barrier options under the dynamics of the Heston model.

The tests will be performed using the following parameter sets:

- Set A: $\rho = 0.1$, $v_0 = 0.0625$, $\bar{v} = 0.16$, $S_0 = 10$, $K = 10$, $r = 0.1$, $q = 0$, $\eta = 0.9$, $\lambda = 5$,
 $T = 0.25$
- Set B: $\rho = -0.64$, $v_0 = 0.0348$, $\bar{v} = 0.0348$, $S_0 = 100$, $K = 100$, $r = 0.04$, $q = 0$,
 $\eta = 0.39$, $\lambda = 1.15$, $T = 0.25$
- Set C: $\rho = -0.9$, $v_0 = 0.04$, $\bar{v} = 0.04$, $S_0 = 100$, $K = 100$, $r = 0$, $q = 0$, $\eta = 0.5$, $\lambda = 0.5$,
 $T = 1$

We have already seen in Section 2 that satisfying the Feller condition is equivalent to having $\frac{\lambda \bar{v}}{\eta^2} - 1 := q_{Feller} \geq 0$. For Set A we have $q_{Feller} = 0.98$, and thus the variance remains strictly positive and efficient pricing performance is expected. For Sets B and C we have $q_{Feller} = -0.47$ and $q_{Feller} = -0.84$, respectively, which means that the Feller condition is not satisfied and we expect less efficiency and slower convergence as a result of a peaked density function.

The programs were written and computations performed using MATLAB on a computer with an Intel Core i3 CPU, 3.30GHz with cache size 3072KB and 8-GB of memory.

6.1 European Options

The first set of results we present are the absolute error and CPU times achieved by applying the 2D-COS method to price European options. The reference values reported were generated using the 1D-COS method from Fang and Oosterlee [2008], utilizing a substantial quantity of evaluation points ($N = 2^{20}$).

The results in Table 1 are satisfactory and show exponential convergence in N_1 , although increasing the number of terms used in the variance dimension (N_2) has no effect on the accuracy. This is because of the fact that for integers $k_2 > 0$, we will have $\psi_{k_2}(a_2, b_2) = 0$, and hence $V_{k_1, k_2} = 0$. As a result of this, the 2D-COS method therefore offers no improvements over the 1D-COS method for the pricing of European options.

The timings in Table 1(d) show that the option prices were computed fairly quickly, although the requirement of two dimensions means that the computation time is much longer than the 1D-COS method (see Fang and Oosterlee [2008]).

Table 1: Absolute error for European put options priced with the 2D-COS method

(a) Set A ($q_{Feller} = 0.98$) $V_{ref} = 0.501465691$					(b) Set B ($q_{Feller} = -0.47$) $V_{ref} = 3.132502167$				
N_1	N_2				N_1	N_2			
	50	100	150	200		50	100	150	200
50	7.32e-07	7.32e-07	7.32e-07	7.32e-07	50	1.42e-04	1.42e-04	1.42e-04	1.42e-04
100	1.28e-12	1.28e-12	1.28e-12	1.28e-12	100	4.77e-08	4.77e-08	4.77e-08	4.77e-08
150	2.22e-16	2.22e-16	2.22e-16	2.22e-16	150	1.02e-11	1.02e-11	1.02e-11	1.02e-11
200	2.22e-16	2.22e-16	2.22e-16	2.22e-16	200	2.66e-14	2.66e-14	2.66e-14	2.66e-14

(c) Set C ($q_{Feller} = -0.84$) $V_{ref} = 6.270867786$					(d) Average CPU time (s) over Sets A,B & C				
N_1	N_2				N_1	N_2			
	50	100	150	200		50	100	150	200
50	3.23e-02	3.23e-02	3.23e-02	3.23e-02	50	0.016	0.019	0.023	0.026
100	8.48e-04	8.48e-04	8.48e-04	8.48e-04	100	0.019	0.022	0.025	0.030
150	5.10e-06	5.10e-06	5.10e-06	5.10e-06	150	0.023	0.025	0.032	0.038
200	5.37e-06	5.37e-06	5.37e-06	5.37e-06	200	0.027	0.030	0.038	0.046

6.1.1 European Options within the Bermudan Framework

In order to test the algorithm developed in Section 4 for pricing Bermudan options under the Heston stochastic volatility model, we first use the algorithm to value European options. This is done by only taking a continuation region $\mathcal{C}^1 = [a_1, b_1] \times [a_2, b_2]$ at each time step t_m , which corresponds to having no early-exercise opportunities for the Bermudan option which is therefore equivalent to a European option. For these tests we use $M = 12$ time steps. As for the previous set of tests, reference values were obtained using the 1D-COS method over a substantial amount of evaluation points.

The results of these tests are shown in Table 2. From these results it can be seen that the results obtained for parameter sets A and B are quite accurate. However the convergence for set C is very slow, with results which are far less accurate than those obtained for Sets A and B, despite being evaluated over a much larger number of grid points. This is because the q_{Feller} value for set C is much closer to -1 and thus the density for the variance is more “peaked” near the origin.

6.2 Bermudan Options

We now consider the algorithm from Section 4.3 for the valuation of Bermudan options. For these experiments we use parameter sets A and B with $M = 10$ early-exercise dates. Parameter set C is not used as the convergence is very slow, as seen from Table 2(c).

The values are compared to those obtained in Ruijter and Oosterlee [2012, Table 7.2].

Table 2: Absolute error for European put options priced in the Bermudan framework

(a) Set A ($q_{Feller} = 0.98$)					(b) Set B ($q_{Feller} = -0.47$)				
N_1	N_2				N_1	N_2			
	50	100	150	200		50	100	150	200
50	9.69e-05	1.20e-05	4.11e-06	2.40e-06	50	3.53e-04	1.72e-04	9.43e-05	5.61e-05
100	9.73e-05	1.25e-05	4.56e-06	2.85e-06	100	3.55e-04	1.75e-04	9.75e-05	5.61e-05
150	9.73e-05	1.25e-05	4.56e-06	2.85e-06	150	3.55e-04	1.75e-04	9.75e-05	5.61e-05
200	9.73e-05	1.25e-05	4.56e-06	2.85e-06	200	3.55e-04	1.75e-04	9.75e-05	5.61e-05

(c) Set C ($q_{Feller} = -0.84$)					
N_1	N_2				
	200	400	600	800	1000
50	1.21e-01	5.62e-02	3.53e-02	2.51e-02	1.92e-02
100	1.15e-01	5.01e-02	2.91e-02	1.89e-02	1.30e-02
150	1.15e-01	5.01e-02	2.90e-02	1.89e-02	1.29e-02
200	1.15e-01	5.01e-02	2.90e-02	1.89e-02	1.30e-02

For our first test we use $J = 2^7$ continuation and early-exercise regions and examine the results for varying values of N_1 and N_2 . The results in Table 3 match quite well with those of Fang and Oosterlee [2011] and Ruijter and Oosterlee [2012], although the computation time is significantly slower.

Next, we investigate the effects of varying the number of early-exercise and continuation regions, J , while keeping N_1 and N_2 fixed at 120 and 100, respectively. The results of this test are presented in Table 4 and show that while increasing J does improve the accuracy, it also has a significant impact on the computation time as the root-finding algorithm is applied J times at each time step t_m .

As a final test, we examine the prices obtained with increasing values for the number of exercise dates, M . These results should converge towards the value of the equivalent American option. We use only Set A for this test as there are no accurate results available in the literature for American options under Set B. The reference value for Set A is obtained from Ito and Toivanen [2009] and is accurate up to the sixth digit. For these tests we keep the grid sizes fixed with $N_1 = 120$, $N_2 = 100$ and $J = 128$. The results of this test can be seen in Table 5, where the convergence of the Bermudan options towards the American option reference value can be seen clearly.

Table 3: Bermudan put option values ($J = 2^7$)

(a) Set A ($q_{Feller} = 0.98$)					(b) Set B ($q_{Feller} = -0.47$)				
N_1	N_2				N_1	N_2			
	40	60	80	100		40	60	80	100
40	0.517489	0.517591	0.517615	0.517624	40	3.198638	3.198525	3.198433	3.198372
60	0.517063	0.517167	0.517192	0.517200	60	3.199478	3.199427	3.199369	3.199327
80	0.517019	0.517123	0.517148	0.517157	80	3.199059	3.199002	3.198942	3.198899
100	0.517015	0.517119	0.517144	0.517153	100	3.199063	3.199008	3.198948	3.198906
120	0.517013	0.517117	0.517142	0.517151	120	3.199038	3.198984	3.198924	3.198882

(c) Average CPU time (s) over Sets A & B

N_1	N_2			
	40	60	80	100
40	10.5	20.4	33.3	48.5
60	15.7	30.8	50.3	73.6
80	21.5	41.0	66.9	98.5
100	27.3	51.3	84.9	124.0
120	33.0	61.7	102.1	147.5

Table 4: Bermudan put option values for increasing J ($N_1 = 120$, $N_2 = 100$)

	$\log_2 J$				
	3	4	5	6	7
Set A	0.516776	0.517034	0.517117	0.517143	0.517151
Set B	3.190984	3.196000	3.197997	3.198679	3.198882
Time (s)	9.1	18.6	37.0	73.6	147.5

Table 5: Bermudan put option values convergence in M to American option value

Set A ($q_{Feller} = 0.98$) $V_{ref} = 0.520030$					
	M				
	20	40	60	80	100
Val	0.518531	0.519244	0.519482	0.519601	0.519673
err	1.50e-03	7.86e-04	5.48e-04	4.29e-04	3.57e-04

6.3 Barrier Options

Since there are no published results available for barrier option prices with parameter sets A, B and C, we will begin by testing the pricing method using parameter sets which enable comparisons with published results. Thus, our first test will be with the parameters used in Griebisch and Wystup [2008]. The two parameter sets used are as follows:

- Set D: $\rho = 0.5$, $v_0 = 0.5$, $\bar{v} = 0.1$, $S_0 = 100$, $K = 90$, $r = 0.05$, $q = 0$, $\eta = 0.5$, $\lambda = 5$,
 $T = 1$, $H = 90$
- Set E: $\rho = 0.5$, $v_0 = 0.1$, $\bar{v} = 0.1$, $S_0 = 100$, $K = 80$, $r = 0.05$, $q = 0.02$, $\eta = 0.1$,
 $\lambda = 5$, $T = 1$, $H = 120$

where Set D is used to price down-and-out call options and Set E is used to price up-and-out call options. For Set D we have $q_{Feller} = 3$ and for Set E we have $q_{Feller} = 99$, which means that efficient pricing performance is expected for both sets of parameters.

The results of the tests on Sets D and E are presented in Table 6 and Table 7, respectively, showing the accuracy and efficiency to be highly satisfactory. The results match very well with those of Griebisch and Wystup [2008], and the timings for the 2D-COS method are significantly faster. The results also converge very quickly, particularly for Set E, as expected from the large q_{Feller} values.

Table 6: Values and timings for down-and-out call options using parameter Set D ($q_{Feller} = 3$)

N_1 (= N_2)	(a) Value					N_1 (= N_2)	(b) Average CPU time (s)				
	M						M				
	2	3	4	5	6		2	3	4	5	6
40	21.40020	20.01266	18.99177	18.19425	17.54393	40	0.055	0.057	0.059	0.062	0.065
60	21.44451	20.10553	19.09846	18.29410	17.62773	60	0.160	0.162	0.169	0.175	0.181
80	21.44459	20.10849	19.10530	18.30413	17.63993	80	0.379	0.390	0.405	0.406	0.416
100	21.44457	20.10851	19.10564	18.30500	17.64139	100	0.683	0.690	0.709	0.726	0.748

Table 7: Values and timings for up-and-out call options using parameter Set E ($q_{Feller} = 99$)

N_1 (= N_2)	(a) Value				N_1 (= N_2)	(b) Average CPU time (s)			
	M					M			
	2	3	4	5		2	3	4	5
40	7.021947	6.324669	5.892730	5.590088	40	0.063	0.169	0.400	0.702
60	7.021714	6.324034	5.893138	5.593800	60	0.066	0.175	0.406	0.722
80	7.021714	6.324033	5.893104	5.593652	80	0.069	0.182	0.408	0.746
100	7.021714	6.324033	5.893104	5.593653	100	0.071	0.188	0.412	0.768

Next, we will use the 2D-COS method to price barrier options using the parameters of Sets A and B. The barrier level, H , is set at 80% of S_0 for down-and-out options and at 120% of S_0 for up-and-out options. The value of the rebate, Rb is set at 0. The results of the tests are presented in Table 8 for down-and-out call options and Table 9 for up-and-out call options.

The results in Table 8 and Table 9 show the absolute errors for varying values of N_1 and N_2 , and the method again proves to be efficient and accurate. The reference values were obtained by applying the 2D-COS method with $N_1 = 800$ and $N_2 = 500$. From these results, it can be seen that the 2D-COS method is effective for the pricing of barrier options, taking a little over half a second to attain 4 significant figures for Set A and 3 significant figures for Set B. Also notice that the times are significantly lower than those for the pricing of Bermudan options. This is as a result of the fact that a root-finding algorithm is not required.

Table 8: Absolute error for Down-and-out call option values

(a) Set A ($q_{Feller} = 0.98$) $V_{ref} = 0.747237813$					(b) Set B ($q_{Feller} = -0.47$) $V_{ref} = 4.127300308$				
N_1	N_2				N_1	N_2			
	40	60	80	100		40	60	80	100
40	3.18e-03	1.99e-03	1.88e-03	1.85e-03	40	1.07e-02	1.08e-02	1.06e-02	1.04e-02
60	2.25e-03	1.06e-03	9.49e-04	9.22e-04	60	5.44e-05	5.96e-04	5.26e-04	4.56e-04
80	1.17e-03	3.18e-05	1.45e-04	1.72e-04	80	2.94e-04	4.02e-04	3.29e-04	2.60e-04
100	1.38e-03	1.77e-04	6.41e-05	3.71e-05	100	3.44e-04	3.48e-04	2.74e-04	2.03e-04
120	1.36e-03	1.58e-04	4.50e-05	1.80e-05	120	3.49e-04	3.44e-04	2.70e-04	1.99e-04

(c) Average CPU time (s) over Sets A & B

N_1	N_2			
	40	60	80	100
40	0.039	0.042	0.070	0.135
60	0.035	0.071	0.136	0.235
80	0.045	0.104	0.198	0.350
100	0.056	0.135	0.258	0.446
120	0.066	0.170	0.325	0.556

Table 9: Absolute error for Up-and-out call option values

(a) Set A ($q_{Feller} = 0.98$) $V_{ref} = 0.255990674$					(b) Set B ($q_{Feller} = -0.47$) $V_{ref} = 3.896249245$				
N_1	N_2				N_1	N_2			
	40	60	80	100		40	60	80	100
40	2.97e-02	3.01e-02	3.02e-02	3.02e-02	40	4.12e-01	4.14e-01	4.14e-01	4.15e-01
60	3.38e-05	3.91e-04	4.16e-04	4.22e-04	60	7.57e-03	7.11e-03	7.08e-03	7.11e-03
80	1.81e-04	4.68e-04	4.87e-04	4.91e-04	80	6.40e-03	5.46e-03	5.44e-03	5.44e-03
100	2.73e-04	3.06e-05	5.08e-05	5.91e-05	100	2.67e-03	1.67e-03	1.58e-03	1.57e-03
120	3.08e-04	2.45e-06	1.80e-05	2.26e-05	120	1.81e-03	8.64e-04	7.91e-04	7.82e-04

(c) Average CPU time (s) over Sets A & B

N_1	N_2			
	40	60	80	100
40	0.097	0.045	0.077	0.145
60	0.035	0.072	0.136	0.237
80	0.046	0.106	0.200	0.349
100	0.056	0.137	0.264	0.443
120	0.067	0.170	0.321	0.557

7 Conclusion

In this dissertation, we have discussed and implemented the 2D-COS method of Ruijter and Oosterlee [2012] for the pricing of European, Bermudan and discrete barrier options under the Heston stochastic volatility model. The model is an extension of the COS method, which was developed in Fang and Oosterlee [2008] and Fang and Oosterlee [2009], and is based on the use of two-dimensional Fourier-cosine expansions. The computation time for this method is reduced significantly by the application of efficient matrix-vector multiplication using the Fast Fourier Transform. However, for the Heston model the FFT algorithm can only be applied in one dimension (the log-stock dimension), as the model is not in the class of Lévy processes.

The 2D-COS method is efficient for pricing European options, however it was shown that this method offers no advantages over the 1D-COS method. The method is effective for the pricing of options with early-exercise features such as Bermudan options, as well as discretely-monitored barrier options, under two-dimensional asset price dynamics. The method works well for a wide range of parameter sets. When the Feller condition is not satisfied, however, the convergence for Bermudan and barrier options can be slow because of the resulting peaked density functions.

A A Basic Outline of Fourier Analysis

A.1 The Fourier Transform Pair

The (direct) Fourier transform of a real-valued function, $g(t)$, $t \in \mathbb{R}$, is defined as

$$\mathcal{F}\{g(t)\}(u) := \int_{-\infty}^{\infty} e^{iut} g(t) dt \quad (\text{A.1})$$

If we define the function $\hat{g}(u) := \mathcal{F}\{g(t)\}(u)$, the corresponding inverse Fourier transform will then be defined as

$$\mathcal{F}^{-1}\{\hat{g}(u)\}(t) := \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-iut} \hat{g}(u) du \quad (\text{A.2})$$

$g(t)$ and $\hat{g}(u)$ then form what is known as a Fourier transform pair, and it is necessarily true that

$$g(t) = \mathcal{F}^{-1}\{\hat{g}(u)\}(t)$$

A.2 Characteristic Functions

Consider a continuous, two-dimensional, real-valued random variable, $\mathbf{X} = (X_1, X_2)$, with joint probability density function $f_{\mathbf{X}}(x_1, x_2)$. The bivariate characteristic function of \mathbf{X} , denoted $\phi_{\mathbf{X}}(u_1, u_2)$, is defined as:

$$\begin{aligned} \phi_{\mathbf{X}}(u_1, u_2) &:= \mathcal{F}\{f_{\mathbf{X}}(x_1, x_2)\}(u_1, u_2) \\ &\equiv \mathbb{E} \left[e^{i(u_1 X_1 + u_2 X_2)} \right] \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{i(u_1 x_1 + u_2 x_2)} f_{\mathbf{X}}(x_1, x_2) dx_1 dx_2 \end{aligned} \quad (\text{A.3})$$

The univariate characteristic function can then be obtained from this by taking

$$\begin{aligned} \phi_{X_1}(u) &= \phi_{\mathbf{X}}(u, 0) \\ &= \int_{-\infty}^{\infty} e^{iux} f_{X_1}(x) dx \end{aligned}$$

where $f_{X_1}(x)$ is the probability density function of the random variable X_1 .

The n^{th} cumulant of X_1 is then given by

$$c_n(X_1) = \frac{1}{i^n} \left. \frac{d^n \ln \phi_{X_1}(u)}{du^n} \right|_{u=0} \quad (\text{A.4})$$

B Verification of Properties Utilized for Developing the 2D-COS Method

B.1 Representing Hankel matrix-vector products in terms of a circular convolution

We begin by defining the vectors

$$\underline{m} := [m_0, m_1, \dots, m_{2N_1-1}]^T$$

and

$$\underline{u} := [u_0, u_1, \dots, u_{N_1-1}]^T$$

and further define $\underline{m}_c(j)$ and $\underline{u}_c(j)$ as in (4.58) and (4.59), respectively.

For the vectors \underline{m} , \underline{m}_c , \underline{u} and \underline{u}_c , we will omit the original arguments in the interest of brevity, and we will use the notation $\underline{m}(j)$, $\underline{m}_c(j)$, $\underline{u}(j)$ and $\underline{u}_c(j)$ to represent the j^{th} element of the relevant vectors.

Now, consider

$$\underline{\underline{\mathcal{M}}}^c \underline{u} = \begin{bmatrix} \sum_{j=1}^{N_1} \underline{m}(j) \underline{u}(j) \\ \sum_{j=1}^{N_1} \underline{m}(j+1) \underline{u}(j) \\ \vdots \\ \sum_{j=1}^{N_1} \underline{m}(j+N_1-1) \underline{u}(j) \end{bmatrix}$$

Considering the vector \underline{m}_c to be periodic, i.e.

$$\underline{m}_c(j) = \underline{m}_c(j + 2N_1n) \quad \text{for } n \in \mathbb{Z}$$

we are then able to use the property

$$\underline{m}(j) = \underline{m}_c(2N_1 - j + 1) \tag{B.1}$$

With these properties we may then treat the first N_1 elements of $\underline{u}_c * \underline{m}_c$ in the following manner:

$$\begin{aligned}
\{\underline{u}_c * \underline{m}_c\}^{[1]} &= \sum_{j=1}^{2N_1} \underline{u}_c(j) \underline{m}_c(1-j+1) \\
&= \sum_{j=N_1+1}^{2N_1} \underline{u}_c(j) \underline{m}_c(1-j+1) \\
&= \sum_{j=1}^{N_1} \underline{u}(j) \underline{m}_c(2-(j+N_1)) \\
&= \sum_{j=1}^{N_1} \underline{u}(j) \underline{m}_c(-j+2+N_1) \\
&= \sum_{j=1}^{N_1} \underline{u}(j) \underline{m}(j+N_1-1)
\end{aligned}$$

Similarly, we then have

$$\begin{aligned}
\{\underline{u}_c * \underline{m}_c\}^{[2]} &= \sum_{j=1}^{N_1} \underline{u}(j) \underline{m}_c(3-(j+N_1)) \\
&= \sum_{j=1}^{N_1} \underline{u}(j) \underline{m}(j+N_1-2)
\end{aligned}$$

and thus

$$\begin{aligned}
\{\underline{u}_c * \underline{m}_c\}^{[N_1-1]} &= \sum_{j=1}^{N_1} \underline{u}(j) \underline{m}_c(N_1-(j+N_1)) \\
&= \sum_{j=1}^{N_1} \underline{u}(j) \underline{m}_c(2N_1-j) \\
&= \sum_{j=1}^{N_1} \underline{u}(j) \underline{m}(j+1)
\end{aligned}$$

Lastly

$$\begin{aligned}
\{\underline{u}_c * \underline{m}_c\}^{[N_1]} &= \sum_{j=1}^{N_1} \underline{u}(j) \underline{m}_c(N_1 + 1 - (j + N_1)) \\
&= \sum_{j=1}^{N_1} \underline{u}(j) \underline{m}_c(2N_1 - j + 1) \\
&= \sum_{j=1}^{N_1} \underline{u}(j) \underline{m}(j)
\end{aligned}$$

Hence, we now have

$$\underline{\mathcal{M}}^c \underline{u} = \begin{bmatrix} \{\underline{u}_c * \underline{m}_c\}^{[N_1]} \\ \{\underline{u}_c * \underline{m}_c\}^{[N_1-1]} \\ \vdots \\ \{\underline{u}_c * \underline{m}_c\}^{[2]} \\ \{\underline{u}_c * \underline{m}_c\}^{[1]} \end{bmatrix}$$

B.2 Verification of the Discrete Fourier Transform's Shifting Property

Using the vectors \underline{u}_s and \underline{u}_c as defined in (4.57) and (4.59) respectively, the k^{th} element ($k \in \{0, 1, \dots, 2N_1 - 1\}$) of $\underline{\mathcal{D}}\{\underline{u}_c\}$ is then obtained as follows:

$$\mathcal{D}_k\{\underline{u}_c\} = \sum_{n=0}^{2N_1-1} \exp\left(ikn \frac{2\pi}{2N_1}\right) \underline{u}_c(n) \tag{B.2}$$

$$= \sum_{n=N_1}^{2N_1-1} \exp\left(ikn \frac{2\pi}{2N_1}\right) \underline{u}_c(n) \tag{B.3}$$

$$= \sum_{n=0}^{N_1-1} \exp\left(ik(n + N_1) \frac{2\pi}{2N_1}\right) \underline{u}_c(n + N_1) \tag{B.4}$$

$$= \exp(ik\pi) \sum_{n=0}^{N_1-1} \exp\left(ikn \frac{2\pi}{2N_1}\right) u_n \tag{B.5}$$

where we again use the notation $\underline{u}_c(n)$ to represent the n^{th} element of the vector \underline{u}_c , as in Appendix B.1.

Now, note that we have

$$\begin{aligned}\exp(i(x + 2\pi n^*)) &= \exp(ix) && \text{for } n^* \in \mathbb{Z} \\ &\text{and} \\ \exp(i2\pi n^*) &= \exp(0) && \text{for } n^* \in \mathbb{Z} \\ &\text{and} \\ \exp(i\pi) &= -1\end{aligned}$$

and so

$$\begin{aligned}\mathcal{D}_k\{\underline{u}_c\} &= \begin{cases} \exp(i2\pi n^*) \sum_{n=0}^{N_1-1} \exp\left(ikn\frac{2\pi}{2N_1}\right) u_n & \text{for } k \text{ even} \\ \exp(i\pi(1 + 2n^*)) \sum_{n=0}^{N_1-1} \exp\left(ikn\frac{2\pi}{2N_1}\right) u_n & \text{for } k \text{ odd} \end{cases} \\ &= \begin{cases} \exp(0) \sum_{n=0}^{N_1-1} \exp\left(ikn\frac{2\pi}{2N_1}\right) u_n & \text{for } k \text{ even} \\ \exp(i\pi) \sum_{n=0}^{N_1-1} \exp\left(ikn\frac{2\pi}{2N_1}\right) u_n & \text{for } k \text{ odd} \end{cases} \\ &= \begin{cases} \sum_{n=0}^{N_1-1} \exp\left(ikn\frac{2\pi}{2N_1}\right) u_n & \text{for } k \text{ even} \\ -\sum_{n=0}^{N_1-1} \exp\left(ikn\frac{2\pi}{2N_1}\right) u_n & \text{for } k \text{ odd} \end{cases} \end{aligned} \tag{B.6}$$

Finally, notice that

$$\begin{aligned}\mathcal{D}_k\{\underline{u}_s\} &= \sum_{n=0}^{2N_1-1} \exp\left(ikn\frac{2\pi}{2N_1}\right) \underline{u}_s(n) \\ &= \sum_{n=0}^{N_1-1} \exp\left(ikn\frac{2\pi}{2N_1}\right) \underline{u}_s(n) \\ &= \sum_{n=0}^{N_1-1} \exp\left(ikn\frac{2\pi}{2N_1}\right) u_n\end{aligned} \tag{B.7}$$

From (B.6) and (B.7) we can then see that

$$\mathcal{D}_k\{\underline{u}_c\} = \begin{cases} \mathcal{D}_k\{\underline{u}_s\} & \text{for } k \text{ even} \\ -\mathcal{D}_k\{\underline{u}_s\} & \text{for } k \text{ odd} \end{cases} \tag{B.8}$$

References

- L. Andersen. Simple and efficient simulation of the Heston stochastic volatility model. *J. Comput. Finance*, 11:1–42, 2008.
- J. P. Boyd. *Chebyshev and Fourier Spectral Methods*. Dover Publications, Inc., 2000.
- P. Carr and D. Madan. Option valuation using the Fast Fourier Transform. *J. Comp. Finance*, 2:61–73, 1999.
- J. Cox, J. Ingersoll, and S. Ross. A theory of the term structure of interest rates. *Econometrica*, 53:217–231, 1985.
- D. Duffie, J. Pan, and K. Singleton. Transform analysis and asset pricing for affine jump-diffusions. *Econometrica*, 68:1343–1376, 2000.
- F. Fang and C. Oosterlee. A novel pricing method for European options based on Fourier-cosine series expansions. *SIAM Journal on Scientific Computing*, 31(2):826–848, 2008.
- F. Fang and C. Oosterlee. Pricing early-exercise and discrete barrier options by Fourier-cosine series expansions. *Numerische Mathematik*, 114(1):27–62, 2009.
- F. Fang and C. Oosterlee. A Fourier-based valuation method for Bermudan and barrier options under Heston’s model. *SIAM Journal on Financial Mathematics*, 2:439–463, 2011.
- W. Feller. Two singular diffusion problems. *Ann. of Math.*, 54:173–182, 1951.
- S. Griebisch and U. Wystup. On the valuation of fader and discrete barrier options in heston’s stochastic volatility model. *Available online*, 2008. URL <http://ssrn.com/abstract=1310422>.
- S. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Review of Financial Studies*, 6(2):327–343, 1993.
- K. Ito and J. Toivanen. Lagrange multiplier approach with optimized finite difference stencils for pricing American options under stochastic volatility. *SIAM J. Sci. Comput.*, 31:2646–2664, 2009.
- C. Kahl and R. Lord. Fourier inversion methods in finance. *Available online*, 2010. URL <http://www.rogerlord.com/fourierinversionmethods.pdf>.
- R. Lord, F. Fang, F. Bervoets, and C. Oosterlee. A fast and accurate FFT-based method for pricing early-exercise options under Lévy processes. *SIAM Journal of Scientific Computing*, 30:1678–1705, 2008.
- K. Matsuda. *Introduction to Option Pricing with Fourier Transform: Option Pricing with Exponential Lévy Models*. PhD thesis, The City University of New York, 2004. URL <http://www.maxmatsuda.com/Papers/2004/Matsuda%20Intro%20FT%20Pricing.pdf>.

- M. Ruijter and C. Oosterlee. Two-dimensional Fourier cosine series expansion method for pricing financial options. *SIAM Journal on Scientific Computing*, 34(5):642–671, 2012.
- J. Stewart. *Calculus: Concepts and Contexts*. Gary W. Ostedt, 2006.
- A. van Haastrecht and A. Pelsser. Efficient, almost exact simulation of the Heston stochastic volatility model. *International Journal of Theoretical and Applied Finance*, 13(1):1–43, 2010.
- C. Van Loan. *Computational Frameworks for the Fast Fourier Transform*, volume 10 of *Frontiers in Applied Mathematics*. SIAM, 1992.
- B. Zhang, L. Grzelak, and C. Oosterlee. Efficient pricing of commodity options with early-exercise under the Ornstein-Uhlenbeck process. *Applied Numerical Mathematics*, 62(2), 2012.