The COS Method: An Efficient Fourier Method for Pricing Financial Derivatives

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof.ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties, in het openbaar te verdedigen op
woensdag 8 december 2010 om 12:30 uur

door

Fang FANG
Master of Science Computational Engineering
Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

geboren te Shizuishan, P.R.China

Dit proefschrift is goedgekeurd door de promotor:
Prof.dr.ir. C.W. Oosterlee

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof.dr.ir. C.W. Oosterlee,	Technische Universiteit Delft, promotor
Prof.dr. R. Cont,	Universitè Paris VI-VII, France
Prof.dr.ir. A.W. Heemink,	Technische Universiteit Delft
Prof.dr. K. In't Hout,	Universiteit Antwerpen, Belgium
Prof.dr. A. Pelsser,	Universiteit Maastricht
Prof.dr. W. Schoutens,	Katholieke Universiteit Leuven, Belgium
Prof.dr.ir. C. Vuik,	Technische Universiteit Delft

The COS Method: An Efficient Fourier Method for Pricing Financial Derivatives.
Dissertation at Delft University of Technology.

Acknowledgements

This dissertation concludes my Ph.D work that started in June 2006 and ended in June 2010. In the first two and half years the research was financially supported by CICAD and was carried out in the Numerical Analysis group of TU Delft. From Jan. 2009 till June 2010, the research was carried out in my spare time.

My most sincere gratitude goes to my advisor prof. Kees Oosterlee. The research topic, Fourier-based methods for option valuation, is a direct continuation of my M.Sc. thesis that was accomplished in Delft as well under his supervision. During these years I've benefited greatly from his guidance and encouragement. Without his help, the research work would have never been finished.

I'd also like to thank Roger Lord and Frank Bervoets. Together we developed the CONV method in 2005 and 2006, which is one of the first efficient methods that employs FFT to price early-exercise options. The joint-work has been a great pleasure and from the experience I gradually gained understanding of option pricing theories and the key insight for the COS method, the method presented in this dissertation.

My gratitude also goes to Henrik Jönsson and prof. Wim Schoutens for the nice cooperation in generalizing the applicability of the COS method to pricing CDSs and recovering survival(default) probabilities.

Besides, I am indebted to Jiayuan Li, who helped me with the implementation of the very first version of the COS-quadrature method that values early-exercise options and barrier options under the Heston stochastic volatility model. That version was definitely a good starting point of the final, robust and efficient method (presented in Chapter 5).

Special thanks also go to the members of my doctoral examination committee for putting their efforts on this thesis and providing helpful comments and suggestions.

Also, I thank all the colleagues in the Numerical Analysis group of TU Delft, for the pleasant research atmosphere and for their help and care, and the secretary of CICAT, Mrs. T. Olsder, for her constant help.

Finally, my greatest debts are to my husband, Chenggang. Were it not for his help and full support for all these years, I'd have never gone so far.

Summary

The COS Method: An Efficient Fourier Method for Pricing Financial Derivatives

Fang Fang

When valuing and risk-managing financial derivatives, practitioners demand fast and accurate prices and sensitivities. Aside from the pricing of non-standard exotic financial derivatives, so-called *plain vanilla* European options form the basis for the calibration of financial models. As any pricing and risk management system has to be able to calibrate to these plain vanilla options, it is important to be able to value these options quickly and accurately.

By means of the risk-neutral valuation formula the price of any option, without early exercise features, can be written as an expectation of the discounted payoff of this option. Starting from this representation one can apply three types of numerical methods to calculate the price itself: Monte Carlo simulation, numerical solution of the corresponding partial-(integro) differential equation (P(I)DE) and numerical integration.

An important aspect of research in Computational Finance is to continuously improve the performance of the pricing methods. In this dissertation we present a *novel and efficient* option pricing method based on the integration representation for various financial derivatives. The method is called *the COS method*, because the key idea is to replace the probability density function, appearing in the risk-neutral valuation formula, by its Fourier-cosine series expansion. Fourier-cosine series coefficients have an elegant closed-form relation with the characteristic function (i.e. the Fourier transform of the underlying density function), which is readily known, for example for exponential Lévy processes, and more general for the broader class of regular affine processes.

For European options, presented in Chapter 2 of this thesis, the risk-neutral valuation formula appears as an inner product of the probability density function and the payoff function. Approximating the density function by its Fourier-cosine series expansion transforms the inner product into combinations of products of cosine basis functions and the (payoff) function which is known analytically. The method's convergence is therefore directly connected to the convergence of the Fourier-cosine series approximation of the density function. For *smooth density functions*, we show that the convergence is exponential. Since the computational complexity grows only linearly with the number of terms in the expansion, the COS method is optimal in

error convergence and in computational complexity for European options.

In other chapters of this thesis, the applicability of the COS method has been generalized to pricing options with early-exercise features and discretely-monitored barrier options, as well as to the calibration of Credit Default Swaps, under exponential Lévy processes. Furthermore, an efficient pricing method based on the COS method has been developed for pricing early-exercise options under the (two-dimensional) Heston stochastic volatility dynamics.

The main insight for generalizing to the options with early-exercise features under Lévy processes, as presented in Chapter 3, is that between any two adjacent early-exercise dates the COS formula for European options can be applied, and that the series coefficients of the option values at each time lattice can be recovered recursively from those of the payoff function. This recursion can be written as a matrix-vector product with the matrix being the sum of two special matrices: a Hankel matrix and a Toeplitz matrix. Multiplication of a vector to these special matrices can be written as circular convolution of two vectors and can therefore be computed rapidly by means of the Fast Fourier Transform (FFT) algorithm. The overall computational complexity for early-exercise (and discretely monitored barrier) options with M exercise dates is $O((M-1)N\log_2 N)$, with N being the number of terms in the cosine expansion.

It is then shown in Chapter 4 that the COS method can be used to calibrate Credit Default Swaps (CDSs), that are the basic building blocks of the credit risk market. The (bid-ask) spread for a CDS depends on the default probability of the underlying reference entity. In the approach adopted here, the credit default spreads are related to a series of survival/default probabilities with different maturities. These survival probabilities can be viewed as prices of binary down-and-out barrier options (without discounting). As such, the COS method for discrete barrier options can be used to recover the probabilities. A special scheme has been developed to reduce the computational time by computing several survival probabilities simultaneously. The method's capabilities have been demonstrated by the calibration to quotes of the constituents of the iTraxx Series.

Finally, in Chapter 5, the COS method has been extended by quadrature rules to efficiently deal with two-dimensional pricing problems, originating for early-exercise options under the Heston stochastic volatility model. We focus especially on parameter values for which the volatility can reach zero. This is a nontrivial situation for which many other numerical schemes fail, including the finite difference PDE schemes. The problem is related to parameter values for which the Feller condition is not satisfied, and for which the pricing problem is *close to singular*. The variance of the stock process follows a noncentral chi-square distribution, and, for some combinations of the relevant parameters, the density function values tend to infinitely large numbers when the variance approaches zero. We handle this problem by a transformation from the variance domain to the log-variance domain. The two-dimensional discrete pricing formula is then obtained by applying the Fourier-cosine series expansion approximation in the log-stock dimension and a high-order quadrature rule in the log-variance dimension. The overall computational complexity is almost-linear in the log-stock dimension and quadratic in the log-variance dimension with very satisfactory error convergence.

In this thesis we show that the COS method can price various financial derivatives under exponential Lévy and Heston stochastic volatility models. Highly efficient pricing results are presented, due to a fast error convergence and a lean computational

complexity. We further determine the stability of the pricing method by a rigorous error analysis and by many numerical experiments under extreme parameter values.

Samenvatting

The COS Method: An Efficient Fourier Method for Pricing Financial Derivatives

Fang Fang

Voor de waardering en risico-inschatting van financiële derivaten, is het nodig om productprijzen en de bijbehorende Grieken snel en accuraat door te rekenen. Tijdens het prijzen van de zogeheten exotische derivaten, die niet standaard op beurzen verhandeld worden, vormen de standaard *plain vanilla* Europese opties de basis voor de calibratie van de onderliggende financiële aandelenmodellen. Ieder prijs- en risicobeheerssysteem dient in zo kort mogelijke tijd gecalibreerd te worden op basis van deze Europese optieprijzen.

Met behulp van de risico-neutrale waarderingsformule kan de prijs van een financiële optie, zonder vervroegde uitoefeningsmogelijkheden, geschreven worden als de verdisconteerde verwachtingswaarde van de uitbetaling van deze optie op de uitoefendatum. Uitgaande van deze beschrijving kan men drie soorten numerieke methoden toepassen om de prijs berekenen: Monte Carlo simulatie, numerieke oplossing van de bijbehorende partiële-(integro) differentiaalvergelijking en numerieke integratie.

Een belangrijk aspect van onderzoek in *Computational Finance* is het continu verbeteren van de prestaties van de rekenmethoden. In dit proefschrift presenteren we een *nieuwe en efficiënte* optiewaarderingsmethode, gebaseerd op de integraalvoorstelling, voor de verschillende financiële derivaten. De methode is genaamd *de COS methode*, omdat het centrale idee de benadering van de kansdichtheidsfunctie in de risico-neutrale waarderingsformule met behulp van een Fourier cosinus expansie is. Fourier cosinuscoëfficiënten hebben een elegante relatie met de karakteristieke functie (is de Fourier getransformeerde van de onderliggende kansdichtheidsfunctie). Deze karakteristieke functie kan analytisch bekend verondersteld worden, bijvoorbeeld voor exponentiële Lévy processen, en meer algemeen voor de klasse van reguliere affine processen.

Voor de waardering van Europese opties, gepresenteerd in Hoofdstuk 2 van dit proefschrift, wordt de risico-neutrale waarderingsformule gepresenteerd als de integraal van het inwendige product van de kansdichtheidsfunctie en de uitoefenfunctie. De benadering van de dichtheidsfunctie door de Fourier cosinusreeks leidt tot vermenigvuldigingen van de cosinus basisfuncties en de uitoefenfunctie, die analytisch bekend zijn. De convergentie van de reeks is rechtstreeks gerelateerd aan de convergentie van de Fourier cosinusreeksen. Voor *gladde functies* tonen we aan dat de

convergentie exponentieel is. Aangezien de rekencomplexiteit slechts lineair met het aantal termen in de reeks groeit, is de COS methode optimaal met betrekking tot convergentie en rekencomplexiteit voor Europese opties.

Een belangrijk inzicht voor generatisatie van de COS methode naar opties die vervroegd uitgeoefend kunnen worden onder Lévy processen, gepresenteerd in Hoofdstuk 3, is dat tussen twee aan elkaar grenzende uitoefendata de COS formule voor Europese opties kan worden toegepast. Verder kunnen de coëfficiënten van optiewaarden op elk uitoefentijdstip recursief worden afgeleid van de coëfficienten van de uitoefenfunctie. Deze recursie kan worden geschreven als een matrix-vector product waarbij de matrix de som is van twee speciale matrices, een Hankel en een Toeplitz matrix. Vermenigvuldiging van een vector met deze bijzondere matrices kan worden geschreven als een circulaire convolutie van twee vectoren en kan daarom zeer efficiënt worden uitgerekend met behulp van de Fast Fourier Transformatie. De rekencomplexiteit voor deze opties die vervroegd uitgeoefend kunnen worden, met M uitoefendata, is $O((M-1)N\log_2(N))$, waarin N het aantal termen in de cosinusexpansie representeert.

Vervolgens wordt in Hoofdstuk 4 de COS methode gebruikt om *credit default swaps* (CDS), i.e. de elementaire bouwstenen van de markt voor kredietrisico, te calibreren. De spreiding van de bied- en laatprijzen voor een CDS hangt af van de standaard-deviatie van een onderliggende referentie-entiteit. Bij de hier gekozen aanpak zijn deze credit default spreidingen gerelateerd aan een reeks van overlevingswaarschijnlijkheden van de referentie-entiteit met verschillende looptijden. Deze overlevingskansen kunnen worden geïnterpreteerd als de waarden van financiële binaire down-and-out barrière opties. Het is derhalve mogelijk om de COS methode voor discrete barrière opties, ook gepresenteerd in Hoofdstuk 3, te gebruiken om de gewenste waarschijnlijkheden te berekenen. Speciale technieken zijn verder ontwikkeld om de rekentijd te verminderen door verschillende overlevingskansen tegelijkertijd door te rekenen. In het hoofdstuk wordt aangetoond dat het mogelijk is om op basis van verschillende reeksen met data van de iTraxx Series (afkomstig uit de kredietmarkt) te calibreren.

Tenslotte is, in Hoofdstuk 5, de COS methode uitgebreid met kwadra-tuurregels met als doel twee-dimensionale optieproblemen efficiënt te berekenen. In het bijzonder richten we ons op opties die vervroegd uitgeoefend kunnen worden onder het Heston stochastische volatiliteitmodel. We focusseren op parameterwaarden waarvoor de volatiliteit de waarde nul kan bereiken. Dit is een niet-triviale situatie waarvoor vele bestaande numerieke schema's falen, eindige differentie discretisaties van de bijbehorende partiële differentiaalvergelijking incluis. Het probleem is gerelateerd aan parameterwaarden die niet aan de Feller-conditie voldoen. De variantie van het aandeel volgt een niet-centraal Chi-kwadraat proces, en voor sommige combinaties van de relevante parameters, als de variantie naar nul gaat, kan de dichtheidsfunctie enorm grote waarden aannemen.

We behandelen dit probleem door een transformatie van het variantiedomein naar het *log-variantiedomein*. De twee-dimensionale discrete optieprijsformule wordt dan verkregen door toepassing van een Fourier cosinus expansie in de log-aandeel dimensie en een hogere orde kwadratuurformule in de log-variantie dimensie. De rekencomplexiteit is vrijwel lineair in de log-aandeel dimensie en kwadratisch in de log-variantie dimensie, met een snelle foutconvergentie.

In dit proefschrift laten we zien dat door middel van de COS methode verschil-

lende financiële derivaten geprijsd kunnen worden onder exponentiële Lévy en ook onder Heston stochastische volatiliteit modellen. Uiterst efficiënte rekenmethoden worden gepresenteerd, op basis van snelle foutconvergentie en een lage rekencomplexiteit. We analyseren verder de stabiliteit en consistentie van de methodes door een rigoureuze foutenanalyse en door vele numerieke experimenten onder extreme parameterwaarden.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

In the field of Computational Finance, efficient numerical methods are required to rapidly price complex contracts and calibrate financial models. Whereas the former topic requires especially robust numerical techniques, the latter also relies on efficiency and speed of computation. During calibration, i.e., fitting model parameters of the stochastic asset processes to market data, we typically need to price (liquid) options at a single spot price, with many different strike prices, very quickly. Particular examples of where this is important would be processes with several parameters, like the Heston model [41] or the infinite activity Lévy processes (see, for example, [23]), since there the pricing problem (for many strikes) is used inside an optimization method.

It is the famous Feynman–Kac theorem that relates the conditional expectation of the value of a contract payoff function under the risk-neutral measure to the solution of a partial differential equation. In the research areas covered by this theorem, various numerical pricing techniques have been developed.

Existing numerical methods in literature can be categorized into three groups: partial-(integro) differential equation (PIDE) methods, Monte Carlo simulation, and *numerical integration methods*. The distinction between the PIDE and the integration methods is, however, subtle: Given the option pricing PIDE, one can formally write down the solution as a Green's function integral. Often the Fourier transform of the Green's function is known; hence the problem reduces to evaluating the integral numerically. The Green's function, modulo a discounting term, is the risk-neutral probability density in finance-speak.

Among the three, numerical integration methods are traditionally very efficient and typically faster than PIDE and Monte Carlo methods for pricing single asset European options. They are also referred as "transform methods" as the Fast Fourier Transform (FFT) algorithm is very often combined with numerical integration[20, 61]. The fast computational speed, especially for *plain vanilla* options, makes these integration methods state of the art for calibration at financial institutions.

In addition, numerical integration methods can readily cope with various asset price dynamics for which the characteristic function (i.e., the Fourier transform of the probability density function) is available. This is the case for models from the

class of regular affine processes of [27], which also includes the exponentially affine jump-diffusion class of [28], and, in particular, the exponentially Lévy models, and some stochastic volatility models.

In recent literature, integration-based methods are generalized to pricing options with early-exercise features. The key idea is to set up a time lattice on each early-exercise date and view the option as of European type between two adjacent lattices. Pricing an early-exercisable option usually involves two steps: recovery of the density and solving the integration that appears in the risk-neutral valuation formula. Some of existing methods employ quadrature rules in both steps. We will elaborate more on the details in the next section. Quadrature rule based techniques are not of high efficiency especially when the integrand is highly oscillatory and therefore a relatively fine grid has to be used for a satisfactory accuracy.

An important aspect of research in computational finance is to further increase the performance of the pricing methods. It is the focus of this dissertation to develop a *novel and highly efficient* option pricing method for various financial derivatives, called *the COS method*. The key idea is to apply *Fourier-cosine expansions* as an alternative for the methods based on quadrature rules.

Financial derivatives covered in this dissertation include European options, Bermudan options, American options, (discrete) barrier options and credit default swaps.

1.2 Asset Dynamics

The asset dynamics considered in this dissertation are mainly exponential Lévy processes (e.g. Geometric Brownian Motion, the Variance Gamma (VG) model[57], the CGMY model[19], the Normal Inverse Gaussian model[8]) and stochastic volatility models (like Heston's stochastic volatility model [41]).

1.2.1 Exponential Lévy Processes

With exponential Lévy processes the asset price is modelled as an exponential function of a Lévy process $L(t)$:

$$S(t) = S_0 \exp(L(t)). \tag{1.1}$$

For ease of exposure we assume that the asset pays a continuous stream of dividends, measured by the dividend rate q. In addition, we assume the existence of a bank account $B(t)$ which evolves according to $dB(t) = rB(t)dt$, r being the (deterministic) risk-free rate. Recall that a process $L(t)$ on $(\Omega, \mathcal{J}, P)$, with $L(0) = 0$, is a Lévy process if:

1 it has independent increments;

2 it has stationary increments;

3 it is stochastically continuous, i.e., for any $t \geq 0$ and $\epsilon > 0$ we have

$$\lim_{s \to t} \mathbb{P}(|L(t) - L(s)| > \epsilon) = 0. \tag{1.2}$$

Each Lévy process can be characterised by a triplet (μ, σ, ν) with $\mu \in \mathbb{R}, \sigma \geq 0$ and ν a measure satisfying $\nu(0) = 0$ and

$$\int_{\mathbb{R}} \min(1, |x|^2)\nu(dx) < \infty. \tag{1.3}$$

In terms of this triplet the characteristic function of the Lévy process equals:

$$\begin{aligned} \phi(u) &= \mathbb{E}[\exp(iuL(t))] \\ &= \exp\left(t(i\mu u - \frac{1}{2}\sigma^2 u^2 + \int_{\mathbb{R}} (e^{iux} - 1 - iux\mathbf{1}_{[|x|<1]}\nu(dx))\right), \end{aligned} \tag{1.4}$$

the celebrated Lévy-Khinchine formula. As is common in most models nowadays we assume that (1.1) is formulated directly under the risk-neutral measure. To ensure that the reinvested relative price $e^{qt}S(t)/B(t)$ is a martingale under the risk-neutral measure, we need to ensure that

$$\phi(-i) = \mathbb{E}[\exp(L(t))] = e^{(r-q)t}, \tag{1.5}$$

which is satisfied if we choose the drift μ as:

$$\mu = r - q - \frac{1}{2}\sigma^2 - \int_{\mathbb{R}} (e^x - 1 - x\mathbf{1}_{[|x|<1]})\nu(dx) \tag{1.6}$$

The simplest and widely used exponential Lévy process is the Geometric Brownian Motion (GBM) model, whereby the logarithm of the asset price follows a Brownian motion (also called a Wiener process) [65]. The asset price S_t is said to follow a GBM if it satisfies the following stochastic differential equation:

$$dS_t = \mu S_t dt + \sigma S_t dW_t$$

where W_t is a Wiener process or Brownian motion, μ is the percentage drift and σ is the percentage volatility. Both μ and σ are constants. This is also referred to as the Black-Scholes model.

One problem with the GBM model is that it is not able to reproduce the volatility skew or smile present in most financial markets. This is the main motivation for the practitioners to use more general Lévy processes. Over the past few years it has been shown that several exponential Lévy models are, at least to some extent, able to reproduce the skew or smile. The particular model we will consider is the extended CGMY model. The underlying Lévy process is characterized by the triple $(\mu, \sigma, \nu_{CGMY})$, where the Lévy density is specified as:

$$\nu_{CGMY}(x) = \begin{cases} C\dfrac{\exp\left(-G|x|\right)}{|x|^{1+Y}} & \text{if} \quad x < 0 \\ C\dfrac{\exp\left(-M|x|\right)}{|x|^{1+Y}} & \text{if} \quad x > 0. \end{cases} \tag{1.7}$$

The parameters satisfy $C \geq 0$, $G \geq 0$, $M \geq 0$, and $Y < 2$. The condition $Y < 2$ is induced by the requirement that Lévy densities integrate x^2 in the neighbourhood

of 0. Conveniently, the characteristic function of the log-asset price can be found in closed-form as:

$$\begin{aligned}\phi(u) &= S(0)^{iu}\exp\Big(iu\mu t-\frac{1}{2}u^2\sigma^2 t\\ &\quad +tC\Gamma(-Y)[(M-iu)^Y-M^Y+(G+iu)^Y-G^Y]\Big),\end{aligned}\tag{1.8}$$

where $\Gamma(x)$ is the gamma function. One can verify that the parameters G and M represent respectively the smallest and largest finite moment in the model, as $\phi(-iu)=\mathbb{E}[S(t)^u]$ is infinite for $u<-G$ and for $u>M$. The model encompasses several models. When $\sigma=0$ and $Y=0$ we obtain the VG model, which is often parameterised slightly differently with parameters [1] σ,θ and ν, related to C,G and M through:

$$C=\frac{1}{\nu},\ G=\frac{1}{\sqrt{\frac{1}{4}\theta^2\nu^2+\frac{1}{2}\sigma^2\nu-\frac{1}{2}\theta\nu}},\ M=\frac{1}{\sqrt{\frac{1}{4}\theta^2\nu^2+\frac{1}{2}\sigma^2\nu+\frac{1}{2}\theta\nu}}.\tag{1.9}$$

Finally, when $C=0$ the model reduces to the GBM model. The reader is referred to [23, 67] for the usage of Lévy processes in a financial context and to [66] for a detailed analysis of Lévy processes in general.

Characteristic functions for several exponential Lévy processes are summarized in Table 1.1. Notations of the parameters in Table 1.1 basically follow the books [23, 67]. Note that"GBM" stands for Geometric Brownian Motion model; "NIG" represents the Normal Inverse Gaussian (NIG) distribution, which is a variance-mean mixture of a Gaussian distribution with an inverse Gaussian [8]; "VG" and "CGMY" stand for VG model and CGMY model, respectively; "Kou" and "Merton" denote the jump-diffusion models developed in [50] and [59], respectively.

Table 1.1: Characteristic functions of $\ln(S_t/K)$ for various models.

GBM	$\varphi(\omega,t)=\exp\left(i\omega\mu t-\frac{1}{2}\sigma^2\omega^2 t\right)$
NIG	$\varphi(\omega,t)=\exp\left(i\omega\mu t-\frac{1}{2}\sigma^2\omega^2 t\right)\phi_{NIG}(\omega,t;\alpha,\beta,\delta)$ $\phi_{NIG}(\omega,t;\alpha,\beta,\delta)=\exp\left[\delta t\left(\sqrt{\alpha^2-\beta^2}-\sqrt{\alpha^2-(\beta+i\omega)^2}\right)\right]$
Kou	$\varphi(\omega,t)=\exp\left(i\omega\mu t-\frac{1}{2}\sigma^2\omega^2 t\right)\phi_{Kou}(\omega,t;\lambda,p,\eta_1,\eta_2)$ $\phi_{Kou}(\omega,t;\lambda,p,\eta_1,\eta_2)=\exp\left[\lambda t\left(\frac{p\eta_1}{\eta_1-i\omega}-\frac{(1-p)\eta_2}{\eta_2+i\omega}-1\right)\right]$
Merton	$\varphi(\omega,t)=\exp\left(i\omega\mu t-\frac{1}{2}\sigma^2\omega^2 t\right)\phi_{Merton}(\omega,t;\lambda,\bar{\mu},\bar{\sigma})$ $\phi_{Merton}(\omega,t)=\exp\left[\lambda t\left(\exp(i\bar{\mu}\omega-\frac{1}{2}\bar{\sigma}^2\omega^2)-1\right)\right]$
VG	$\varphi(\omega,t)=\exp\left(i\omega\mu t\right)\phi_{VG}(\omega,t;\sigma,\nu,\theta)$ $\phi_{VG}(\omega,t;\sigma,\nu,\theta)=(1-i\omega\theta\nu+\frac{1}{2}\sigma^2\nu\omega^2)^{-t/\nu}$
CGMY	$\varphi(\omega,t)=\exp\left(i\omega\mu t-\frac{1}{2}\sigma^2\omega^2 t\right)\phi_{CGMY}(\omega,t;C,G,M,Y)$ $\phi_{CGMY}(\omega,t;C,G,M,Y)=\exp(Ct\Gamma(-Y)[(M-i\omega)^Y-M^Y+(G+i\omega)^Y-G^Y])$

Given the characteristic functions, the cumulants, defined in [23], can be computed via

$$\xi_n(X)=\frac{1}{i^n}\frac{\partial^n(t\Psi(\omega))}{\partial\omega^n}\bigg|_{\omega=0},$$

[1] The parameters σ and ν should not be confused with the volatility and Lévy density of the Lévy triplet.

where $t\Psi(\omega)$ is the exponent of the characteristic function $\varphi(\omega, t)$, i.e.

$$\varphi(\omega, t) = e^{t\Psi(\omega)}, \quad t \geq 0.$$

The formulas for the cumulants are summarized in Table 1.2. They have been verified with the help of Mathematica. Cumulants are used in defining the proper truncation range in the following chapters.

Table 1.2: Cumulants of $\ln(S_t/K)$ for various models.

GBM	$\xi_1 = (\mu - \frac{1}{2}\sigma^2)t, \quad \xi_2 = \sigma^2 t, \quad \xi_4 = 0$
NIG	$\xi_1 = (\mu - \frac{1}{2}\sigma^2 + w)t + \delta t\beta/\sqrt{\alpha^2 - \beta^2}$ $\xi_2 = \delta t\alpha^2(\alpha^2 - \beta^2)^{-3/2}$ $\xi_4 = 3\delta t\alpha^2(\alpha^2 + 4\beta^2)(\alpha^2 - \beta^2)^{-7/2}$ $w = -\delta(\sqrt{\alpha^2 - \beta^2} - \sqrt{\alpha^2 - (\beta + 1)^2})$
Kou	$\xi_1 = t\left(\mu + \frac{\lambda p}{\eta_1} + \frac{\lambda(1-p)}{\eta_2}\right)$ $\quad \xi_2 = t\left(\sigma^2 + 2\frac{\lambda p}{\eta_1^2} + 2\frac{\lambda(1-p)}{\eta_2^2}\right)$ $\xi_4 = 24t\lambda\left(\frac{p}{\eta_1^4} + \frac{1-p}{\eta_2^4}\right)$ $\quad w = \lambda\left(\frac{p}{\eta_1+1} - \frac{1-p}{\eta_2-1}\right)$
Merton	$\xi_1 = t(\mu + \lambda\bar{\mu})$ $\quad \xi_2 = t\left(\sigma^2 + \lambda\bar{\mu}^2 + \bar{\sigma}^2\lambda\right)$ $\xi_4 = t\lambda\left(\bar{\mu}^4 + 6\bar{\sigma}^2\bar{\mu}^2 + 3\bar{\sigma}^4\lambda\right)$
VG	$\xi_1 = (\mu + \theta)t$ $\quad \xi_2 = (\sigma^2 + \nu\theta^2)t$ $\xi_4 = 3(\sigma^4\nu + 2\theta^4\nu^3 + 4\sigma^2\theta^2\nu^2)t$ $\quad w = \frac{1}{\nu}\ln(1 - \theta\nu - \sigma^2\nu/2)$
CGMY	$\xi_1 = \mu t + Ct\Gamma(1-Y)\left(M^{Y-1} - G^{Y-1}\right)$ $\xi_2 = \sigma^2 t + Ct\Gamma(2-Y)\left(M^{Y-2} + G^{Y-2}\right)$ $\xi_4 = Ct\Gamma(4-Y)\left(M^{Y-4} + G^{Y-4}\right)$ $w = -C\Gamma(-Y)[(M-1)^Y - M^Y + (G+1)^Y - G^Y]$

where w is the drift correction term that satisfies $\exp(-wt) = \varphi(-i, t)$.

1.2.2 Heston's Stochastic Volatility Model

Like Lévy processes, stochastic volatility models have also been developed to capture the volatility smiles and skews present in the market quotes. Within this class, the Heston stochastic volatility model[41], whereby the variance of (the logarithm of) the stock price is modeled by a square-root process, has become popular in industrial practice.

The Heston stochastic volatility model defines the dynamics of the logarithm of the stock price (*log-stock*), x_t, and the variance, ν_t, by the following stochastic differential equations (SDEs) [41]:

$$dx_t = \left(\mu - \frac{1}{2}\nu_t\right)dt + \rho\sqrt{\nu_t}dW_{1,t} + \sqrt{1-\rho^2}\sqrt{\nu_t}dW_{2,t} \tag{1.10}$$

$$d\nu_t = \lambda\left(\bar{\nu} - \nu_t\right)dt + \eta\sqrt{\nu_t}dW_{1,t}, \tag{1.11}$$

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

We will pay special attention to the properties of this model, as well as difficulties arisen in option pricing under this model, in Chapter 5.

1.3 Existing Integration-based Methods

1.3.1 European Options

By means of the risk-neutral valuation formula the price of any option without early exercise features can be written as an expectation of the discounted payoff of this option:

$$v(t, S_t) = e^{-r\tau}\mathbb{E}\left[v(T, S_T)\right], \tag{1.12}$$

where v denotes the value of the option, r is the risk-neutral interest rate[2], t is the current time point, T is the maturity of the option and $\tau = T - t$. The variable S_τ denotes the asset price at time τ. The expectation is taken with respect to the risk-neutral probability measure.

As (1.12) is an expectation, it can be calculated via numerical integration provided that the probability density is known in closed-form, which is, however, not the case for many models. What is usually available or easier to derive is the characteristic function.

The characteristic function (Ch.f.) is in essence the continuous Fourier transform of the density function, and thus, one needs to apply inverse continuous Fourier transform to recover the density. The fast Fourier transform (FFT) algorithm that computes discrete Fourier transforms in $O(N \log_2(N))$ operations can be employed after the continuous Fourier integral is "discretized".

Different inversion methods exist.

Pioneering articles applying Fourier transform techniques are concerned with the pricing of European-style options, like the Gil-Pelaez inversion [38] or the Carr-Madan inversion by Carr and Madan [20].

Gil-Palaez inversion relates the cumulative distribution function $F(s)$ to the characteristic function $\phi(u)$ as follows:

$$F(s) = \frac{1}{2} - \frac{1}{\pi}\int_0^\infty \mathrm{Re}\left(\phi(u)\frac{e^{-ius}}{iu}\right)du. \tag{1.13}$$

Carr and Madan [20] considered another approach. Note that L^1-integrability is a sufficient condition for the Fourier transform of a function to exist. A call option is not L^1-integrable with respect to the logarithm of the strike price, as:

$$\lim_{k\to-\infty} v(t, S(t)) = S(t),$$

Pre-multiplying the option price with $\exp(\alpha k)$ for $\alpha > 0$ solves this however, and Carr and Madan ended up with this method by pre-multiplying a damping function to the option value so ensuring the existence of the Fourier transform. They presented

[2] Although we assume throughout the paper that interest rates are deterministic, this assumption can be relaxed at the cost of increasing the dimensionality of some of the methods.

the following highly efficient pricing formula,

$$\begin{aligned}\mathcal{F}\{e^{\alpha k}v(t,k)\} &= e^{-r\tau}\int_{-\infty}^{\infty} e^{iuk}e^{\alpha k}\mathbb{E}\left[(S(T)-e^{k})^{+}\right]dk \\ &= \frac{e^{-r\tau}\phi(u-(\alpha+1)i)}{-(u-\alpha i)(u-(\alpha+1)i)},\end{aligned} \tag{1.14}$$

where we now consider the option price v as a function of time and k.

A necessary and sufficient condition for (1.14) to exist is that

$$|\phi(u-(\alpha+1)i)| \leq \phi(-(\alpha+1)i) = \mathbb{E}[S(T)^{\alpha+1}] < \infty,$$

i.e., that the $(\alpha+1)^{th}$ moment of the asset price exists. The option price can be recovered by inverting (1.14) and undamping

$$v(t,k) = \frac{1}{2\pi}e^{-r\tau-\alpha k}\int_{-\infty}^{\infty} e^{-iuk}\frac{\phi(u-(\alpha+1)i)}{-(u-\alpha i)(u-(\alpha+1)i)}du \tag{1.15}$$

Pricing requires only one numerical integration and the numerical stability of (1.15) can be controlled by means of the damping coefficient α. Finally we note that if we discretize (1.15) with Newton-Côtes quadrature the option price can be efficiently evaluated by means of the FFT, yielding option prices over a whole range of strike prices.

1.3.2 Bermudan Options

Define the set of exercise dates as $\mathcal{T} = \{t_1, \ldots, t_M\}$ and $0 = t_0 \leq t_1$. For ease of exposure, assume the exercise dates are equally spaced, so that $t_{m+1} - t_m = \Delta t$. The best known examples of options with early exercise are American and Bermudan options. American options can be exercised at any time prior to the option's expiry, whereas Bermudan options can only be exercised at certain dates in the future. If the option is exercised at some time $t \in \mathcal{T}$ the holder of the option obtains the exercise payoff $g(t, S(t))$. The Bermudan option price can then be found via backward induction as

$$\begin{cases} v(t_M, S(t_M)) &= g(t_M, S(t_M)) \\ c(t_m, S(t_m)) &= e^{-r\Delta t}\mathbb{E}_{t_m}\left[v(t_{m+1}, S(t_{m+1}))\right] \\ & \quad (m = M-1, \ldots, 1), \\ v(t_m, S(t_m)) &= \max\{c(t_m, S(t_m)), g(t_m, S(t_m))\}, \\ v(t_0, S(t_0)) &= c(t_0, S(t_0)) \end{cases} \tag{1.16}$$

with c being the continuation value of the option and v the value of the option immediately before the exercise opportunity.

The dynamic programming problem in (1.16) is a successive application of the risk-neutral valuation formula, and we can write the continuation value as

$$c(t_m, S(t_m)) = e^{-r\Delta t}\int_{-\infty}^{\infty} v(t_{m+1}, y)f(y|S(t_m))dy, \tag{1.17}$$

where $f(y|S(t_m))$ represents the probability density describing the transition from $S(t_m)$ at t_m to y at t_{m+1}.

Based on (1.16) and (1.17) the QUAD method was introduced in [6]. The method requires the transition density to be known in closed-form, which is the case in e.g. the GBM model and Merton's jump-diffusion model. This requirement is relaxed in [61], where the QUAD-FFT method is introduced. The underlying idea is that the transition density can be recovered by inverting the characteristic function, so that the QUAD method can be used for a wider range of models. As such the QUAD-FFT method effectively combines the QUAD method with the early transform methods. The overall complexity of both methods is $O(MN^2)$ for an M-times exercisable Bermudan option with N grid points used to discretise the price of the underlying asset.

In a presentation by Reiner [64], it was recognized that for the GBM model the risk-neutral valuation formula in (1.17) can be seen as a convolution or cross-correlation of the continuation value with the transition density. As convolutions can be handled very efficiently by means of the FFT, an overall complexity of $O(MN \log_2 N)$ can be achieved. By working forward instead of backward in time a number of discrete path-dependent options can also be treated, such as lookbacks, barriers, Asian options and cliquets. Building on Reiner's idea, Broadie and Yamamoto [15] reduced the complexity to $O(MN)$ for the GBM model by combining the double-exponential integration formula and the Fast Gauss Transform. Their technique is applicable to any model in which the transition density can be written as a weighted sum of Gaussian densities, which is the case in e.g. Merton's jump-diffusion model.

As one of the defining properties of a Lévy process is that its increments are independent of each other, the insight of Reiner has a much wider applicability than only to the GBM model. This is especially appealing since the usage of Lévy processes in finance has become more established nowadays.

By combining Reiner's ideas with the work of Carr and Madan, the Convolution method, the CONV method for short was introduced [54].The difference with the Carr–Madan approach is that the transform is with respect to the log-spot price in the CONV method instead of the log-strike price (something which [52] and [63] also consider). It avoids a "density-recovery step" as in the QUAD-FFT method, since it is based solely on the price process's characteristic function. An overal computational complexity is therefore $O(MN \log_2 N)$ for an M-times exercisable Bermudan option.

1.3.3 American Options

For the valuation of American options by integration-based methods, there are basically two approaches. One way is to approximate an American option by a Bermudan option with many exercise opportunities, the other is to use Richardson extrapolation on a series of Bermudan options with an increasing number of exercise opportunities. This method has been described in detail in [21], though the approach in finance dates back to [37]. The QUAD method in [6] also uses the same technique to price American options. We restrict ourselves to the essentials here. Let $v(\Delta t)$ be the price of a Bermudan option with a maturity of T years where the exercise dates are Δt years apart. It is assumed that $v(\Delta t)$ can be expanded as

$$v(\Delta t) = v(0) + \sum_{i=1}^{\infty} a_i (\Delta t)^{\gamma_i}, \tag{1.18}$$

with $0 < \gamma_i < \gamma_{i+1}$. $v(0)$ is the price of the American option. Classical extrapolation procedures assume that the exponents γ_i are known, which means that we can use $n+1$ Bermudan prices with varying Δt in order to eliminate n of the leading order terms in (1.18). The only theoretical paper considering an expansion of the Bermudan option price in terms of Δt we are aware of is of Howison [43], who shows that $\gamma_1 = 1$ for the GBM model. Numerical tests indicate that the assumption $\gamma_i = 1$ produces satisfactory results also for more general Lévy models.

1.4 The CONV Method

1.4.1 Derivation of the CONV method

The main premise of the CONV method is that the conditional probability density $f(y|x)$ in (1.17) only depends on x and y via their difference, i.e.

$$f(y|x) = f(y - x). \tag{1.19}$$

Note that x and y do not have to represent the asset price directly, they could be monotone functions of the asset price. The assumption made in (1.19) therefore certainly holds when the asset price is modeled as a monotone function of a Lévy process, since one of the defining properties of a Lévy process is that its increments are independent of each other. For the time being, let x and y in (1.19) represent the log-spot price on two adjacent time points, t_m and t_{m+1}, respectively. By including (1.19) in (1.17) and changing variables $z = y - x$ the continuation value can be expressed as

$$c(t_m, x) = e^{-r\Delta t} \int_{-\infty}^{\infty} v(t_{m+1}, x + z) f(z) dz, \tag{1.20}$$

which is a cross-correlation[3] of the option value at time t_{m+1} and the density $f(z)$, or equivalently, a convolution of $v(t_{m+1})$ and the conjugate of $f(z)$. For many exponential Lévy models we either do not have a closed-form expression for the density (e.g. the CGMY/KoBoL model of [12] and [19] and many Exponential Affine Jump Diffusion (EAJD) models), or if we have, it involves one or more special functions (e.g. the VG model). In contrast, the characteristic function of the log-spot price can typically be obtained in closed-form or, in case of the EAJD models, via the solution of a system of ordinary differential equations (ODEs).

Therefore, the Fourier transform of (1.20) is taken. The insight that the continuation value can be seen as a convolution is useful here, as the Fourier transform of a convolution is the product of the Fourier transforms of the two functions being convolved. In the remainder the following definitions for the continuous Fourier

[3] The cross-correlation of two functions $f(t)$ and $g(t)$, denoted $f \star g$, is defined by

$$f \star g \equiv \bar{f}(-t) * g(t) = \int_{-\infty}^{\infty} f(\tau) g(t + \tau) d\tau,$$

where '$*$' denotes the convolution operator.

transform and its inverse are employed:

$$\hat{h}(u) := \mathcal{F}\{h(t)\}(u) = \int_{-\infty}^{\infty} e^{iut} h(t) dt, \tag{1.21}$$

$$h(t) := \mathcal{F}^{-1}\{\hat{h}(u)\}(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-iut} \hat{h}(u) du. \tag{1.22}$$

If we dampen the continuation value (1.20) by a factor $\exp(\alpha x)$ and subsequently take its Fourier transform, we obtain

$$\begin{aligned} e^{r\Delta t} \mathcal{F}\{c_\alpha(t_m, x)\}(u) &= \int_{-\infty}^{\infty} e^{iux} e^{\alpha x} \int_{-\infty}^{\infty} v(t_{m+1}, x+z) f(z) dz dx \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{iu(x+z)} v(t_{m+1}, x+z) e^{-iz(u-i\alpha)} f(z) dz dx. \end{aligned} \tag{1.23}$$

where in the first step we used the risk-neutral valuation formula from (1.20). We introduced the convention that $(\cdot)_\alpha$ indicates damping by the damping factor, i.e., $c_\alpha(t_m, x) = e^{\alpha x} c(t_m, x)$ and $v_\alpha(t_m, x+z) = e^{\alpha(x+z)} v(t_m, x+z)$. Changing the order of integration and remembering that $x = y - z$, we obtain

$$\begin{aligned} e^{r\Delta t} \mathcal{F}\{c_\alpha(t_m, x)\}(u) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{iuy} v_\alpha(t_{m+1}, y) dy \, e^{-i(u-i\alpha)z} f(z) dz \\ &= \int_{-\infty}^{\infty} e^{iuy} v_\alpha(t_{m+1}, y) dy \int_{-\infty}^{\infty} e^{-i(u-i\alpha)z} f(z) dz \\ &= \mathcal{F}\{e^{\alpha y} v(t_{m+1}, y)\}(u) \; \phi(-(u - i\alpha)). \end{aligned} \tag{1.24}$$

In the last step we used the fact that the complex-valued Fourier transform of the density is the extended characteristic function

$$\phi(x + yi) = \int_{-\infty}^{\infty} e^{i(x+yi)z} f(z) dz, \tag{1.25}$$

which is well-defined when $\phi(yi) < \infty$, as $|\phi(x + yi)| \leq |\phi(yi)|$. As such (1.24) puts a condition on the damping coefficient α, because $\phi(\alpha i)$ must be finite.

The Fourier transform of the damped continuation value can thus be calculated as the product of two functions, one of which, the extended characteristic function, is readily available for exponential Lévy models. We now recover the continuation value by taking the inverse Fourier transform of the right-hand side of (1.24), and calculate $v(t_m)$ as the maximum of the continuation and the exercise value at t_m. We repeat (1.16) recursively until we have obtained the option price at time t_0. In pseudo-code the CONV algorithm is presented as follows:

Algorithm 1.4.1 (The CONV algorithm for Bermudan options).

$v(t_M, x) = g(t_M, x)$ for all x
$g(t_0, x) = 0$ for all x
For $m = M - 1$ to 0
    Dampen $v(t_{m+1}, x)$ with $\exp(\alpha x)$ and take its Fourier transform
    Calculate the right-hand side of (1.24)
    Calculate $c(t_m, x)$ by applying Fourier inversion to (1.24) and undamping
    $v(t_m, x) = \max\{g(t_m, x), c(t_m, x)\}$

The essence of the CONV method is the calculation of a convolution[4]:

$$c_\alpha(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-iux}\hat{v}_\alpha(u)\phi\left(-(u-i\alpha)\right)du, \tag{1.26}$$

where $\hat{v}_\alpha(u)$ is the Fourier transform of v_α:

$$\hat{v}_\alpha(u) = \int_{-\infty}^{\infty} e^{iuy}v_\alpha(y)dy. \tag{1.27}$$

Both integrals in (1.26) and (1.27) are approximated by a discrete sum, so that the FFT algorithm can be employed for a faster computation. This necessitates the use of uniform grids for u, x and y:

$$u_j = u_0 + j\Delta u,\ x_j = x_0 + j\Delta x,\ y_j = y_0 + j\Delta y, \tag{1.28}$$

where $j = 0, \ldots, N-1$. Though they may be centered around a different point, the x- and y-grids have the same mesh size: $\Delta x = \Delta y$. Further, *the Nyquist relation* must be satisfied, i.e.,

$$\Delta u \cdot \Delta y = \frac{2\pi}{N}. \tag{1.29}$$

Details about the exact location of x_0 and y_0 are given in[54]. Inserting (1.27) into (1.26), and approximating (1.27) and (1.26) with the composite Trapezoidal rule yields:

$$c_\alpha(x_p) \approx \frac{\Delta u\Delta y}{2\pi}\sum_{j=0}^{N-1} e^{-iu_jx_p}\phi\left(-(u_j-i\alpha)\right)\sum_{n=0}^{N-1} w_n e^{iu_jy_n}v(y_n), \tag{1.30}$$

for $p = 0, \ldots, N-1$. Inserting the definitions of the grids into (1.30) yields:

$$c_\alpha(x_p) \approx \frac{e^{-iu_0(x_0+p\Delta y)}}{2\pi}\Delta u\sum_{j=0}^{N-1} e^{-ijp2\pi/N}e^{ij(y_0-x_0)\Delta u}\phi\left(-(u_j-i\alpha)\right)\hat{v}_\alpha(u_j), \tag{1.31}$$

where the Fourier transform of v_α is approximated by:

$$\hat{v}_\alpha(u_j) \approx e^{iu_0y_0}\Delta y\sum_{n=0}^{N-1} e^{ijn2\pi/N}e^{inu_0\Delta y}w_nv_\alpha(y_n). \tag{1.32}$$

Let us now define the DFT and its inverse of a sequence $x_p,\ p = 0, \ldots, N-1$, as:

$$\mathcal{D}_j\{x_n\} := \sum_{n=0}^{N-1} e^{ijn2\pi/N}x_n,\ \ \mathcal{D}_n^{-1}\{x_j\} = \frac{1}{N}\sum_{j=0}^{N-1} e^{-ijn2\pi/N}x_j. \tag{1.33}$$

Set $u_0 = -N/2\Delta u$ and recall $e^{inu_0\Delta y} = (-1)^n$. This finally leads us to write (1.31), (1.32) as:

$$c_\alpha(x_p) \approx e^{iu_0(y_0-x_0)}(-1)^p\mathcal{D}_p^{-1}\{e^{ij(y_0-x_0)\Delta u}\phi\left(-(u_j-i\alpha)\right)\mathcal{D}_j\{(-1)^nw_nv_\alpha(y_n)\}\}. \tag{1.34}$$

As a result, in the CONV algorithm the FFT algorithm is used twice.

The method has also been generalized to multi-dimensional cases like pricing basket options.

[4]For notational convenience we have dropped the discounting term out of the equation.

1.4.2 Numerical results

Herewith we demonstrate the performance of the CONV method by a few numerical tests. The parameter sets for the experiments are summarized in Table 1.3, where we included one VG test and two CGMY tests (one with $Y < 1$ and the other with $Y > 1$, as the latter is considered a hard test case when numerically solving the corresponding PIDE). The reference results are calculated with the CONV method using 2^{20} grid points.

Table 1.3: Parameter sets for the numerical experiments in this sectio

T1-VG:	$S_0 = 100$,	$r = 0.1$,	$q = 0$,	$\sigma = 0.12$,
	$\theta = -0.14$,	$\nu = 0.2$;		
T2-CGMY:	$S_0 = 1$,	$r = 0.1$,	$q = 0$,	$\sigma = 0$,
	$C = 1$,	$G = 5$,	$M = 5$,	$Y = 0.5$;
T3-CGMY:	$S_0 = 90$,	$r = 0.06$,	$q = 0$,	$\sigma = 0$
	$C = 0.42$,	$G = 4.37$,	$M = 191.2$,	$Y = 1.0102$;

As shown in Table 1.4, error convergence for a Bermudan option is quadratic and the computational complexity is almost linear. Within fraction of a second, the accuracy is up to the 6th decimal place.

Table 1.4: CPU time, error and convergence rate pricing a 10-times exercisable Bermudan put under T1-VG; $K = 110, T = 1$ with reference value $V_{ref}(0, S_0) = 9.040646119$.

n ($N = 2^n$)	time(sec)	error	error(n)/ error($n+1$)
8	0.007	1.09e-2	-
9	0.008	2.50e-3	4.4
10	0.010	6.51e-4	3.8
11	0.016	1.65e-4	4.0
12	0.026	4.15e-5	4.0
13	0.090	1.99e-06	4.0

The CONV method can also rapidly price American options with reasonable accuracy. The efficiency can be superior to PIDE methods in valuing American options especially under sophisticated Lévy dynamics such as VG and CGMY. For example, as demonstrated in Table 1.5, using the CONV method with 2-times Richardson extrapolation on 512-, 256- and 128-times exercisable Bermudans, one obtains American option prices that are accurate up to 5 decimal places, within half a second. Both CGMY tests stem from the PIDE literature, where reference values for the same American puts were reported as 0.112171 for T2-CGMY [1], and 9.2254842 for T3-CGMY [73].

The results indicate that the CONV method is able to price American options under a wide variety of Lévy processes. A reasonable accuracy can be obtained rapidly.

Table 1.5: CPU time and errors of the CONV method for American puts under VG and CGMY

	T1-VG		T2-CGMY		T3-CGMY	
	$K = 110, T = 1$		$K = 1, T = 1$		$K = 98, T = 0.25$	
$(N = 2^n)$	$V_{ref}(0, S(0)) = 10.0000$		$V_{ref}(0, S(0) = 0.112152$		$V_{ref}(0, S(0) = 9.225439$	
n	time(sec)	error	time(sec)	error	time(sec)	error
7	0.073	-3.49e-1	0.062	-6.35e-3	0.074	-1.89e-1
8	0.096	4.13e-2	0.097	1.38e-4	0.093	2.93e-2
9	0.115	1.37e-2	0.116	1.16e-4	0.118	-1.30e-3
10	0.157	-6.17e-3	0.160	1.10e-5	0.162	-3.97e-4
11	0.270	6.03e-3	0.275	1.18e-5	0.278	2.89e-4
12	0.466	1.31e-3	0.482	-2.35e-6	0.483	9.59e-5

In summary, the overall error convergence is quadratic due to the usage of the Trapezoidal rule, and the computational complexity is $O(MN \log_2(N))$.

In the rest of this thesis, we will show that the COS method improves the error convergence to exponential for many cases while $O(MN \log_2(N))$ complexity remains.

1.5 Organization of This Thesis

The remainder of this thesis is as follows.

In Chapter 2 the COS method for recovering probability density functions from characteristic functions and for valuing European options is developed. The key insight lies in the close relation between the characteristic function with the series coefficients of the Fourier-cosine expansion of the density function. In most cases, the convergence rate of the COS method is exponential and the computational complexity is linear. Its range of application covers underlying asset processes for which the characteristic function is known and various types of option contracts. This chapter contains essentially the contents of the paper [32].

In Chapter 3 the method is generalized for early-exercise options and discretely-monitored barrier options under Lévy asset price models. The error convergence is exponential for processes characterized by smooth ($C^\infty[a, b] \in \mathbb{R}$) transitional probability density functions. The computational complexity is $O((M - 1)N \log N)$ with N a (small) number of terms from the series expansion, and M, the number of early-exercise/monitoring dates. This chapter contains essentially the contents of the paper [33].

In Chapter 4 the COS method is generalized to calculating survival/default probabilities and pricing Credit Default Swaps under advanced jump dynamics. We have chosen to use the firm's value approach, modeling the firm's value by an exponential Lévy model. For this approach the default event is defined as a first passage of a barrier and it is therefore possible to exploit a numerical technique developed to price barrier options under Lévy models to calculate the default probabilities. This chapter contains essentially the contents of the paper [31].

In Chapter 5 the COS method is combined with a quadrature rule to solve the two-dimentional pricing problem under Heston's stochastic volatility model. Error analysis and experiments demonstrate fast error convergence along both dimensions, which, together with lean computational complexity, is verified by numerical experiments. This chapter contains essentially the contents of the paper [34].

In Chapter 6 conclusions are presented, as well as an outlook of topics for future research.

Chapter 2

Density Recovery and European Option Valuation

2.1 Introduction

In this chapter, we derive the highly efficient COS method for recovering density functions as well as for pricing European options. The error convergence is exponential for many processes and the computational complexity is linear. This chapter contains essentially the contents of the paper [32].

Other highly efficient techniques for pricing plain vanilla options include the fast Gauss transform [15] and the double-exponential transformation [60, 75]. The COS method can, however, handle more general dynamics for the underlying compared to these methods. In fact, we can price a vector of strike prices simultaneously. Furthermore, the COS method offers a highly efficient way to recover the density from the characteristic function, which is of importance for several financial applications, like calibration, the computation of forward starting options, or static hedging.

This chapter is organized as follows. In Section 2.2, we introduce the Fourier-cosine expansion for solving inverse Fourier integrals. Based on this, we derive, in Section 2.3, the formulas for pricing European options and the Greeks. We focus on the Lévy and the Heston processes for the underlying. An error analysis is presented in Section 2.4, and numerical results are given in Section 2.5.

2.2 Fourier Integrals and Cosine Series

The point of departure for pricing European options with numerical integration techniques is the risk-neutral valuation formula:

$$v(x,t_0) = e^{-r\Delta t}\mathbb{E}^{\mathbb{Q}}\left[v(y,T)|x\right] = e^{-r\Delta t}\int_{\mathbb{R}} v(y,T)f(y|x)dy, \tag{2.1}$$

where v denotes the option value, Δt is the difference between the maturity, T, and the initial date, t_0, and $\mathbb{E}^{\mathbb{Q}}[\cdot]$ is the expectation operator under risk-neutral measure $\mathbb{Q}$. x and y are state variables at times t_0 and T, respectively; $f(y|x)$ is the probability density of y given x, and r is the risk-neutral interest rate.

As indicated in Section 1.3, in the Carr–Madan approach [20] and its variants, the Fourier transform of a version of valuation formula (2.1) is taken with respect to the log-strike price. Damping of the payoff is then necessary as a call option is not L^1-integrable with respect to the logarithm of the strike price. The method's accuracy depends on the correct value of the damping parameter. A closed-form expression for the resulting integral is available in Fourier space. To return to the log-price domain, quadrature rules have to be applied to the inverse Fourier integral for which the application of the FFT algorithm is appropriate.

The CONV method [54] can also be efficiently used for European options, and numerical experiments in [54] have shown that the accuracy is not influenced by the choice of the damping parameter. In the derivation of the CONV method (as presented in Chapter 1) the risk-neutral valuation formula is rewritten as a cross-correlation between the option value and the transition density. The cross-correlation is handled numerically by replacing the option value by its Fourier series expansion so that the cross-correlation is transformed into an inner product of series coefficients. The coefficients are recovered by applying quadrature rules, combined with the FFT algorithm. Error analysis and experimental results have demonstrated second order accuracy and $O(N \log_2(N))$ computational complexity for European options.

These numerical integration methods have to numerically solve forward or inverse[1] Fourier integrals. The density and its characteristic function, $f(x)$ and $\phi(\omega)$, form an example of a Fourier pair,

$$\phi(\omega) = \int_{\mathbb{R}} e^{ix\omega} f(x) dx, \tag{2.2}$$

$$f(x) = \frac{1}{2\pi} \int_{\mathbb{R}} e^{-i\omega x} \phi(\omega) d\omega. \tag{2.3}$$

Existing numerical integration methods in finance typically compute the Fourier integrals by applying equally spaced numerical integration rules and then employing the FFT algorithm by imposing the Nyquist relation to the grid sizes in the x- and ω-domains,

$$\Delta x \cdot \Delta \omega \equiv 2\pi / N,$$

with N representing the number of grid points. The grid values can then be obtained in $O(N \log_2 N)$ operations. However, there are disadvantages: The error convergence of equally spaced integration rules, except for the Clenshaw–Curtis rule, is not very high; N has to be a power of two; finally, the relation imposed on the grid sizes prevents one from using coarse grids in both domains.

Remark 2.2.1. *In principle we could use the fractional FFT algorithm (FrFT), which does not require the Nyquist relation to be satisfied, as in [22]. However, numerical tests for several options indicated that this advantage of the FrFT did not outweigh the speed of the FFT in our applications.*

Remark 2.2.2. *Alternative methods for the forward Fourier integral, based on replacing $f(x)$ in (2.2) by its Chebyshev [62] or Legendre [29] polynomial expansion, can achieve a high accuracy with only a limited number of terms in the expansion. However, the resulting computational complexity is typically at least quadratic.*

[1] Here we use the convention of the Fourier transform definition often seen in the financial engineering literature. Other conventions can also be used, and modifications to the methods are then straightforward.

2.2.1 Inverse Fourier integral via cosine expansion

In this section, as a first step, we present a different methodology for solving, in particular, the inverse Fourier integral in (2.3). The main idea is to reconstruct the whole integral—not just the integrand—from its Fourier-cosine series expansion (also called "cosine expansion"), extracting the series coefficients directly from the integrand. Fourier-cosine series expansions usually give an optimal approximation of functions with a finite support[2] [13]. In fact, the cosine expansion of $f(x)$ in x equals the Chebyshev series expansion of $f(\cos^{-1}(t))$ in t.

For a function supported on $[0, \pi]$, the cosine expansion reads

$$f(\theta) = \sum_{k=0}^{\infty}{}' A_k \cdot \cos(k\theta) \quad \text{with} \quad A_k = \frac{2}{\pi}\int_0^{\pi} f(\theta)\cos(k\theta)d\theta, \tag{2.4}$$

where $\sum'$ indicates that the first term in the summation is weighted by one-half. For functions supported on any other finite interval, say $[a, b] \in \mathbb{R}$, the Fourier-cosine series expansion can easily be obtained via a change of variables:

$$\theta := \frac{x-a}{b-a}\pi, \quad x = \frac{b-a}{\pi}\theta + a.$$

It then reads

$$f(x) = \sum_{k=0}^{\infty}{}' A_k \cdot \cos\left(k\pi\frac{x-a}{b-a}\right), \tag{2.5}$$

with

$$A_k = \frac{2}{b-a}\int_a^b f(x)\cos\left(k\pi\frac{x-a}{b-a}\right)dx. \tag{2.6}$$

Since any real function has a cosine expansion when it is finitely supported, the derivation starts with a truncation of the infinite integration range in (2.3). Due to the conditions for the existence of a Fourier transform, the integrands in (2.3) have to decay to zero at $\pm\infty$ and we can truncate the integration range in a proper way without losing accuracy.

Suppose $[a, b] \in \mathbb{R}$ is chosen such that the truncated integral approximates the infinite counterpart very well, i.e.,

$$\phi_1(\omega) := \int_a^b e^{i\omega x} f(x)dx \approx \int_{\mathbb{R}} e^{i\omega x} f(x)dx = \phi(\omega). \tag{2.7}$$

By subscripts for variables, like i in ϕ_i, we denote subsequent numerical approximations (not to be confused with subscripted series coefficients, A_k and F_k).

Comparing (2.7) with the cosine series coefficients of $f(x)$ on $[a, b]$ in (2.6), we find that

$$A_k \equiv \frac{2}{b-a}\mathrm{Re}\left\{\phi_1\left(\frac{k\pi}{b-a}\right)\cdot\exp\left(-i\frac{ka\pi}{b-a}\right)\right\}, \tag{2.8}$$

where $\mathrm{Re}\{\cdot\}$ denotes taking the real part of the argument. It then follows from (2.7) that $A_k \approx F_k$ with

[2] The usual Fourier series expansion is actually superior when a function is periodic.

$$F_k \equiv \frac{2}{b-a}\text{Re}\left\{\phi\left(\frac{k\pi}{b-a}\right)\cdot\exp\left(-i\frac{ka\pi}{b-a}\right)\right\}. \tag{2.9}$$

We now *replace* A_k by F_k in the series expansion of $f(x)$ on $[a,b]$, i.e.,

$$f_1(x) = \sum_{k=0}^{\infty}{}' F_k \cos\left(k\pi\frac{x-a}{b-a}\right), \tag{2.10}$$

and truncate the series summation such that

$$f_2(x) = \sum_{k=0}^{N-1}{}' F_k \cos\left(k\pi\frac{x-a}{b-a}\right). \tag{2.11}$$

The resulting error in $f_2(x)$ consists of two parts: a series truncation error from (2.10) to (2.11) and an error originating from the approximation of A_k by F_k. An error analysis that takes these different approximations into account is presented in Section 2.4.

Since the cosine series expansion of *entire functions* (i.e., functions without any singularities[3] anywhere in the complex plane, except at ∞) exhibits an *exponential convergence* [13], we can expect (2.11) to give highly accurate approximations to functions that have no singularities on $[a,b]$, with a small N.

To demonstrate this, here we evaluate (2.11), where

$$f(x) = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}x^2},$$

and determine the accuracy for different values of N. We choose $[a,b] = [-10,10]$, and the maximum absolute error is measured at $x = \{-5,-4,\ldots,4,5\}$.

Table 2.1 indicates that a very small error is obtained with only a small number of terms, N, in the expansion. From the differences in the CPU times in the table, defined as "time(N)-time($N/2$)," we can observe a linear complexity. This technique is thus highly efficient for the recovery of the density function; see also Section 2.5.

Table 2.1: Maximum error when recovering $f(x)$ from $\phi(\omega)$ by Fourier-cosine expansion.

N	4	8	16	32	64
Error	0.25	0.11	0.0072	4.04e-07	3.33e-16
CPU time (msec.)	0.046	0.061	0.088	0.16	0.29
Diff. in CPU (msec.)	–	0.015	0.027	0.072	0.13

2.3 Pricing European Options

In this section, we derive the COS formula for European-style options by replacing the density function by its Fourier-cosine series. We make use of the fact that a density function tends to be smooth and therefore only a few terms in the expansion may already give a good approximation.

[3] By "singularity" we mean [13] poles, fractional powers, logarithms, other branch points, and discontinuities in a function or in any of its derivatives.

Since the density rapidly decays to zero as $y \to \pm\infty$ in (2.1), we truncate the infinite integration range without losing significant accuracy to $[a, b] \subset \mathbb{R}$, and we obtain approximation v_1:

$$v_1(x, t_0) = e^{-r\Delta t} \int_a^b v(y, T) f(y|x) dy. \tag{2.12}$$

We will give insight into the choice of $[a, b]$ in Section 2.5.

In the second step, since $f(y|x)$ is usually not known whereas the characteristic function is, we replace the density by its cosine expansion in y,

$$f(y|x) = \sum_{k=0}^{+\infty}{}' A_k(x) \cos\left(k\pi \frac{y-a}{b-a}\right) \tag{2.13}$$

with

$$A_k(x) := \frac{2}{b-a} \int_a^b f(y|x) \cos\left(k\pi \frac{y-a}{b-a}\right) dy, \tag{2.14}$$

so that

$$v_1(x, t_0) = e^{-r\Delta t} \int_a^b v(y, T) \sum_{k=0}^{+\infty}{}' A_k(x) \cos\left(k\pi \frac{y-a}{b-a}\right) dy. \tag{2.15}$$

We interchange the summation and integration, and insert the definition

$$V_k := \frac{2}{b-a} \int_a^b v(y, T) \cos\left(k\pi \frac{y-a}{b-a}\right) dy, \tag{2.16}$$

resulting in

$$v_1(x, t_0) = \frac{1}{2}(b-a) e^{-r\Delta t} \cdot \sum_{k=0}^{+\infty}{}' A_k(x) V_k. \tag{2.17}$$

Note that the V_k are the cosine series coefficients of payoff function $v(y, T)$ in y. Thus, from (2.12) to (2.17) we have transformed the product of two real functions, $f(y|x)$ and $v(y, T)$, into that of their Fourier-cosine series coefficients.

Due to the rapid decay rate of these coefficients, we further truncate the series summation to obtain approximation v_2:

$$v_2(x, t_0) = \frac{1}{2}(b-a) e^{-r\Delta t} \cdot \sum_{k=0}^{N-1}{}' A_k(x) V_k. \tag{2.18}$$

Similar to Section 2.2, coefficients $A_k(x)$ defined in (2.14) can be approximated by $F_k(x)$ as defined in (2.9). Replacing $A_k(x)$ in (2.18) by $F_k(x)$, we obtain

$$v(x, t_0) \approx v_3(x, t_0) = e^{-r\Delta t} \sum_{k=0}^{N-1}{}' \text{Re}\left\{\phi\left(\frac{k\pi}{b-a}; x\right) e^{-ik\pi \frac{a}{b-a}}\right\} V_k, \tag{2.19}$$

with characteristic function ϕ. This is the COS formula for general underlying processes. We will show that the V_k can be obtained analytically for plain vanilla and digital options, and that (2.19) can be simplified for the Lévy and the Heston models, so that many strikes can be handled simultaneously.

The key step in obtaining this semianalytic formula (2.19) for option pricing is the replacement of the probability density function by its Fourier-cosine series expansion.

The advantage is that the product of the density and the payoff is transformed into a linear combination of products of cosine basis functions and a (payoff) function which is known analytically.

Important for convergence is therefore the convergence of the density function's cosine series, not the cosine series of the payoff, which appears only because we interchanged the summation and the integration in (2.17).

Heuristically speaking, we decompose the probability density into a weighted sum of many "density-like basis functions" with which option values can be obtained analytically. What matters for the accuracy and the computational speed is how well this probability density function is approximated.

2.3.1 Coefficients V_k for plain vanilla options

Before we can use (2.19) for pricing options, the payoff series coefficients, V_k, have to be recovered. We can find analytic solutions for V_k for several contracts.

As we assume here that the characteristic function of the log-asset price is known, we represent the payoff as a function of the log-asset price. Let us denote the log-asset prices by

$$x := \ln(S_0/K) \quad \text{and} \quad y := \ln(S_T/K),$$

with S_t the underlying price at time t and K the strike price. The payoff for European options, in log-asset price, reads

$$v(y,T) \equiv [\varrho \cdot K(e^y - 1)]^+ \quad \text{with} \quad \varrho = \begin{cases} 1 & \text{for a call,} \\ -1 & \text{for a put.} \end{cases}$$

Before deriving V_k from its definition in (2.16), we need two mathematical results.

Result 2.3.1. *The cosine series coefficients, χ_k, of $g(y) = e^y$ on $[c,d] \subset [a,b]$,*

$$\chi_k(c,d) := \int_c^d e^y \cos\left(k\pi \frac{y-a}{b-a}\right) dy, \tag{2.20}$$

and the cosine series coefficients, ψ_k, of $g(y) = 1$ on $[c,d] \subset [a,b]$,

$$\psi_k(c,d) := \int_c^d \cos\left(k\pi \frac{y-a}{b-a}\right) dy, \tag{2.21}$$

are known analytically.

Proof. Basic calculus shows that

$$\begin{aligned} \chi_k(c,d) := \frac{1}{1+\left(\frac{k\pi}{b-a}\right)^2} \Bigg[& \cos\left(k\pi\frac{d-a}{b-a}\right) e^d - \cos\left(k\pi\frac{c-a}{b-a}\right) e^c \\ & + \frac{k\pi}{b-a} \sin\left(k\pi\frac{d-a}{b-a}\right) e^d - \frac{k\pi}{b-a} \sin\left(k\pi\frac{c-a}{b-a}\right) e^c \Bigg] \end{aligned} \tag{2.22}$$

and

$$\psi_k(c,d) := \begin{cases} \left[\sin\left(k\pi\frac{d-a}{b-a}\right) - \sin\left(k\pi\frac{c-a}{b-a}\right)\right] \frac{b-a}{k\pi}, & k \neq 0, \\ (d-c), & k = 0. \end{cases} \tag{2.23}$$

Focusing, for example, on a call option, we obtain

$$V_k^{call} = \frac{2}{b-a}\int_0^b K(e^y-1)\cos\left(k\pi\frac{y-a}{b-a}\right)dy = \frac{2}{b-a}K\left(\chi_k(0,b)-\psi_k(0,b)\right), \tag{2.24}$$

where χ_k and ψ_k are given by (2.22) and (2.23), respectively. Similarly, for a vanilla put, we find

$$V_k^{put} = \frac{2}{b-a}K\left(-\chi_k(a,0)+\psi_k(a,0)\right). \tag{2.25}$$

Analytic expressions of V_k can also be obtained for some exotic options.

2.3.2 Coefficients V_k for digital and gap options

Whereas for European products (2.19) always applies, the coefficients V_k are different for different payoff functions. With analytic expressions for these coefficients, the convergence of the COS does not depend on the continuity of the payoff.

Digital options are popular in the financial markets for hedging and speculation. They are also important to financial engineers as building blocks for constructing more complex option products. Here we consider the payoff of a cash-or-nothing call option as an example, which is 0 if $S_T \le K$ and K if $S_T > K$. For this contract the cash-or-nothing call coefficients, V_k^{cash}, can be obtained analytically:

$$V_k^{cash} = \frac{2}{b-a}K\int_0^b \cos\left(k\pi\frac{y-a}{b-a}\right)dy = \frac{2}{b-a}K\psi_k(0,b).$$

We also give the formula for a so-called gap call option [40], whose payoff reads

$$v(y,T) = [K(e^y-1)^+ - r_b]\cdot\mathbf{1}_{\{S_T<H\}} + r_b,$$

where $\mathbf{1}_\Psi$ equals 0 if Ψ is empty and 1 otherwise, and r_b is a so-called rebate and is paid if the barrier is hit. The time-dependent version of this payoff represents a barrier option, which will be discussed in the follow-up chapter. The integral that defines V_k^{gap} for such payoff functions can be split into two parts:

$$V_k^{gap} = \frac{2}{b-a}\int_0^h K(e^y-1)\cos\left(k\pi\frac{y-a}{b-a}\right)dy + \frac{2}{b-a}\int_h^b r_b\cdot\cos\left(k\pi\frac{y-a}{b-a}\right)dy,$$

where $h := \ln(H/K)$. It then follows that

$$V_k^{gap} = \frac{2}{b-a}K\left(\chi_k(0,h)-\psi_k(0,h)\right) + \frac{2}{b-a}r_b\cdot\psi_k(h,b). \tag{2.26}$$

For those contracts, however, for which the V_k can be obtained only numerically, the error convergence is dominated by the numerical rules employed.

2.3.3 Formula for exponential Lévy processes and the Heston model

It is worth mentioning that (2.19) is greatly simplified for the Lévy and the Heston models, so that options *for many strike prices* can be computed simultaneously. Here we use boldfaced values to distinguish vectors.

For Lévy processes, whose characteristic functions can be represented by

$$\phi(\omega;\mathbf{x}) = \varphi_{levy}(\omega)\cdot e^{i\omega\mathbf{x}} \quad \text{with} \quad \varphi_{levy}(\omega) := \phi(\omega;0), \tag{2.27}$$

the pricing formula is simplified to

$$v(\mathbf{x},t_0) \approx e^{-r\Delta t}\sum_{k=0}^{N-1}{}' \operatorname{Re}\left\{\varphi_{levy}\left(\frac{k\pi}{b-a}\right)e^{ik\pi\frac{\mathbf{x}-a}{b-a}}\right\}\mathbf{V}_k. \tag{2.28}$$

Recalling the V_k-formulas for vanilla European options in (2.24) and (2.25), we can now present them as a vector multiplied by a scalar,

$$\mathbf{V}_k = U_k\mathbf{K},$$

where

$$U_k = \begin{cases} \frac{2}{b-a}\left(\chi_k(0,b)-\psi_k(0,b)\right) & \text{for a call,}\\ \frac{2}{b-a}\left(-\chi_k(a,0)+\psi_k(a,0)\right) & \text{for a put.}\end{cases} \tag{2.29}$$

As a result, the pricing formula reads[4]

$$v(\mathbf{x},t_0) \approx \mathbf{K}e^{-r\Delta t}\cdot\operatorname{Re}\left\{\sum_{k=0}^{N-1}{}'\varphi_{levy}\left(\frac{k\pi}{b-a}\right)U_k\cdot e^{ik\pi\frac{\mathbf{x}-a}{b-a}}\right\}, \tag{2.30}$$

where the summation can be written as a matrix-vector product if $\mathbf{K}$ (and therefore $\mathbf{x}$) is a vector. In the section with numerical results, we will show that with very small N we can achieve highly accurate results.

Remark 2.3.1. *Equation (2.30) is an expression with independent variable* $\mathbf{x}$*. It is therefore possible to obtain the option prices for different strikes in one single numerical experiment, by choosing a* $\mathbf{K}$*-vector as the input vector (the same is true for the Carr–Madan formula).*

The characteristic functions for some important exponential Lévy processes have been given in Table 1.1. Next, we elaborate a bit more on the Heston model.

For the Heston model, the COS pricing equation is also simplified, since

$$\phi(\omega;\mathbf{x},u_0) = \varphi_{hes}(\omega;u_0)\cdot e^{i\omega\mathbf{x}}, \tag{2.31}$$

with u_0 the volatility of the underlying at the initial time and $\varphi_{hes}(\omega;u_0) := \phi(\omega;0,u_0)$. We then find

$$v(\mathbf{x},t_0,u_0) \approx \mathbf{K}e^{-r\Delta t}\cdot\operatorname{Re}\left\{\sum_{k=0}^{N-1}{}'\varphi_{hes}\left(\frac{k\pi}{b-a};u_0\right)U_k\cdot e^{ik\pi\frac{\mathbf{x}-a}{b-a}}\right\}. \tag{2.32}$$

The characteristic function of the log-asset price, $\varphi_{hes}(\omega;u_0)$, reads

$$\begin{aligned}\varphi_{hes}(\omega;u_0) = \exp&\left(i\omega\mu\Delta t + \frac{u_0}{\eta^2}\left(\frac{1-e^{-D\Delta t}}{1-Ge^{-D\Delta t}}\right)(\lambda - i\rho\eta\omega - D)\right)\\ &\cdot\exp\left(\frac{\lambda\bar{\nu}}{\eta^2}\left(\Delta t(\lambda-i\rho\eta\omega-D) - 2\log\left(\frac{1-Ge^{-D\Delta t}}{1-G}\right)\right)\right),\end{aligned}$$

[4] Although the U_k values are real, we keep them in the curly brackets. This allows us to interchange $\operatorname{Re}\{\cdot\}$ and $\sum'$, and it simplifies the implementation in MATLAB.

with

$$D = \sqrt{(\lambda - i\rho\eta\omega)^2 + (\omega^2 + i\omega)\eta^2} \quad \text{and} \quad G = \frac{\lambda - i\rho\eta\omega - D}{\lambda - i\rho\eta\omega + D}.$$

This characteristic function is uniquely specified, since we take $\sqrt{(x+yi)}$ such that its real part is nonnegative, and we restrict the complex logarithm to its principal branch. In this case the resulting characteristic function is the correct one for all complex ω in the strip of analycity of the characteristic function, as proven in [56].

Remark 2.3.2 (the Greeks). *Series expansions for the Greeks, e.g., Δ and Γ, can be derived similarly. Since*

$$\Delta = \frac{\partial v}{\partial S_0} = \frac{\partial v}{\partial x}\frac{\partial x}{\partial S_0} = \frac{1}{S_0}\frac{\partial v}{\partial x}, \qquad \Gamma = \frac{\partial^2 v}{\partial S_0^2} = \frac{1}{S_0^2}\left(-\frac{\partial v}{\partial S_0} + \frac{\partial^2 v}{\partial S_0^2}\right),$$

it then follows that

$$\Delta \approx e^{-r\Delta t} \sum_{k=0}^{N-1}{}' Re\left\{\varphi\left(\frac{k\pi}{b-a}; u_0\right) e^{ik\pi\frac{x-a}{b-a}} \frac{ik\pi}{b-a}\right\} \frac{V_k}{S_0} \tag{2.33}$$

and

$$\Gamma \approx e^{-r\Delta t} \sum_{k=0}^{N-1}{}' Re\left\{\varphi\left(\frac{k\pi}{b-a}; u_0\right) e^{ik\pi\frac{x-a}{b-a}} \left[-\frac{ik\pi}{b-a} + \left(\frac{ik\pi}{b-a}\right)^2\right]\right\} \frac{V_k}{S_0^2}. \tag{2.34}$$

It is also easy to obtain the formula for Vega, $\frac{\partial v}{\partial u_0}$, for example, for the Heston model (2.32), as u_0 appears only in the coefficients:

$$\frac{\partial v(x, t_0, u_0)}{\partial u_0} \approx e^{-r\Delta t} \sum_{k=0}^{N-1}{}' \text{Re}\left\{\frac{\partial \varphi_{hes}\left(\frac{k\pi}{b-a}; u_0\right)}{\partial u_0} e^{ik\pi\frac{x-a}{b-a}}\right\} V_k. \tag{2.35}$$

2.4 Error Analysis

In the derivation of the COS formula there are three steps that introduce errors: the truncation of the integration range in the risk-neutral valuation formula, the substitution of the density by its cosine series expansion on the truncated range, and the substitution of the series coefficients by the characteristic function approximation. Therefore, the overall error consists of three parts:

1. The integration range truncation error:

$$\epsilon_1 := v(x, t_0) - v_1(x, t_0) = \int_{\mathbb{R}\setminus[a,b]} v(y, T) f(y|x) dy. \tag{2.36}$$

2. The series truncation error on $[a, b]$:

$$\epsilon_2 := v_1(x, t_0) - v_2(x, t_0) = \frac{1}{2}(b-a)e^{-r\Delta t} \sum_{k=N}^{+\infty} A_k(x) \cdot V_k, \tag{2.37}$$

where $A_k(x)$ and V_k are defined in (2.14) and (2.16), respectively.

3. The error related to approximating $A_k(x)$ by $F_k(x)$ in (2.9):

$$\begin{aligned}\epsilon_3 &:= v_3(x,t_0) - v_2(x,t_0)\\ &= e^{-r\Delta t}{\sum_{k=0}^{N-1}}' \operatorname{Re}\left\{\int_{\mathbb{R}\setminus[a,b]} e^{ik\pi\frac{y-a}{b-a}} f(y|x)dy\right\} V_k. \end{aligned} \tag{2.38}$$

We do not have to take any error in the coefficients V_k into account here, as we have a closed-form solution, at least for the plain vanilla options considered in this chapter.

The key to bound the errors lies in the decay rate of the cosine series coefficients. The convergence rate of the Fourier-cosine series depends on the properties of the functions on the expansion interval. We first give the definitions classifying the rate of convergence of the series for different classes of functions, taken from [13].

Definition 2.4.1 (Algebraic index of convergence)**.** The algebraic index of convergence $n(\geq 0)$ is the largest number for which

$$\lim_{k\to\infty} |A_k|\, k^n < \infty, \qquad k \gg 1,$$

where the A_k are the coefficients of the series. An alternative definition is that if the coefficients of a series, A_k, decay asymptotically as

$$A_k \sim O(1/k^n), \qquad k \gg 1,$$

then n is the algebraic index of convergence.

Definition 2.4.2 (Exponential index of convergence)**.** If the algebraic index of convergence $n(\geq 0)$ is unbounded—in other words, if the coefficients, A_k, decrease faster than $1/k^n$ for any finite n—the series is said to have *exponential convergence*. Alternatively, if

$$A_k \sim O(\exp(-\gamma k^r)), \qquad k \gg 1,$$

with γ, the constant, being the "asymptotic rate of convergence," for some $r > 0$, then the series shows exponential convergence. The exponent r is the index of convergence.

For $r < 1$, the convergence is called *subgeometric*.

For $r = 1$, the convergence is either called *supergeometric* with

$$A_k \sim O(k^{-n}\exp(-(k/j)\ln(k)))$$

(for some $j > 0$) or *geometric* with

$$A_k \sim O(k^{-n}\exp(-\gamma k)). \tag{2.39}$$

The density of the GBM process is a typical function that has a geometrically converging cosine series expansion.

Proposition 2.4.1 (Convergence of Fourier-cosine series [13], pp. 70–71)**.** *If $g(x)$ is infinitely differentiable with nonzero derivatives, then its Fourier-cosine series expansion on $[a,b]$ has geometric convergence. The constant γ in (2.39) is then determined by the location in the complex plane of the singularities nearest to the expansion interval. Exponent n is determined by the type and strength of the singularity.*

Otherwise, the convergence is algebraic. Integration by parts shows that the algebraic index of convergence, n, is at least as large as n', with n' denoting the highest order of derivative that exists or is nonzero.

If the function $g(x)$ has a discontinuity in $[a,b]$, say at x_0, then at the discontinuity the series value converges to $\frac{1}{2}(g(x_0^+) + g(x_0^-))$, as the Fourier-cosine series has in essence the same properties as a Fourier series.

References to the proof of this proposition are available in [13]. Note that in the case of a discontinuous probability density function, we will encounter a very low algebraic convergence order, which can be related to the well-known Gibbs phenomenon observed in Fourier series expansions of discontinuous functions.

The following proposition further bounds the series truncation error of an algebraically converging series.

Proposition 2.4.2 (Series truncation error of algebraically converging series)**.** *It can be shown that the series truncation error of an algebraically converging series behaves like*

$$\sum_{k=N+1}^{\infty} \frac{1}{k^n} \sim \frac{1}{(n-1)N^{n-1}}.$$

The proof can be found in [10].

With the two propositions above, we can state the following lemmas.

Lemma 2.4.1. *Error ϵ_3 consists of integration range truncation errors, and can be bounded by*

$$|\epsilon_3| < Q\,|\epsilon_4|\,, \tag{2.40}$$

where Q is some positive constant and

$$\epsilon_4 := \int_{\mathbb{R}\setminus[a,b]} f(y|x)dy.$$

Proof. Assuming $f(y|x)$ to be a real function, we rewrite (2.38) as

$$\epsilon_3 = e^{-r\Delta t} {\sum_{k=0}^{N-1}}' V_k \int_{\mathbb{R}\setminus[a,b]} \cos\left(k\pi\frac{y-a}{b-a}\right) f(y|x)dy.$$

Applying triangle inequality twice, we yield

$$\begin{aligned}|\epsilon_3| &\le e^{-r\Delta t} {\sum_{k=0}^{N-1}}' |V_k| \cdot \left| \int_{\mathbb{R}\setminus[a,b]} \cos\left(k\pi\frac{y-a}{b-a}\right) f(y|x)dy \right| \\ &\le e^{-r\Delta t} {\sum_{k=0}^{N-1}}' |V_k| \cdot \int_{\mathbb{R}\setminus[a,b]} \left|\cos\left(k\pi\frac{y-a}{b-a}\right)\right| \cdot |f(y|x)|\,dy.\end{aligned}$$

Recall that $f(y|x) \ge 0$ and that $|\cos(t)| \le 1$, we can bound ϵ_3 as follows:

$$|\epsilon_3| \le \epsilon_4 \cdot e^{-r\Delta t} {\sum_{k=0}^{N-1}}' |V_k|\,,$$

where $\epsilon_4 := \int_{\mathbb{R}\setminus[a,b]} f(y|x)dy$, which depends on the size of $[a,b]$. According to Propositions 2.4.1 and 2.4.2, the V_k exhibit at least algebraic convergence. Therefore, the finite summation is bounded, i.e.

$$e^{-r\Delta t}\sum_{k=0}^{N-1}{}' |V_k| \leq Q,$$

for some positive constant Q. It then follows

$$|\epsilon_3| < Q\,|\epsilon_4|$$

□

Thus, two of the three error components are truncation range related. When the truncation range is sufficiently large, the overall error is dominated by ϵ_2.

Equation (2.37) indicates that ϵ_2 depends on both $A_k(x)$ and V_k, the series coefficients of the density and that of the payoff, respectively. We assume that the density is typically smoother than the payoff functions in finance and that the coefficients A_k decay faster than V_k. Consequently, the product of A_k and V_k converges faster than either A_k or V_k, and we can bound this product as follows:

$$\left|\sum_{k=N}^{+\infty} A_k(x)\cdot V_k\right| \leq C\sum_{k=N}^{+\infty}|A_k(x)|\,, \tag{2.41}$$

with C some constant. Error ϵ_2 is thus dominated by the series truncation error of the density function.

Proposition 2.4.3 (Series truncation error of geometrically converging series [13], p. 48)**.** *If a series has geometrical convergence, then the error after truncation of the expansion after $(N+1)$ terms, $E_T(N)$, reads*

$$E_T(N) \sim P^* \exp(-N\nu).$$

Here constant $\nu > 0$ is called the asymptotic rate of convergence of the series, which satisfies

$$\nu = \lim_{n\to\infty}\left(-\log|E_T(n)|/n\right),$$

and P^* denotes a factor which varies less than exponentially with N.

Lemma 2.4.2. *Error ϵ_2 converges exponentially in the case of density functions $g(x) \in \mathbb{C}^\infty([a,b])$ with nonzero derivatives:*

$$|\epsilon_2| < P\exp(-(N-1)\nu), \tag{2.42}$$

where $\nu > 0$ is a constant and P is a term that varies less than exponentially with N.

The proof of this is straightforward, applying Proposition 2.4.3 to (2.41).

Based on Proposition 2.4.2, we can prove the following lemma.

Lemma 2.4.3. *Error ϵ_2 for densities having discontinuous derivatives can be bounded as follows:*

$$|\epsilon_2| < \frac{\bar{P}}{(N-1)^{\beta-1}}, \tag{2.43}$$

where $\bar{P}$ is a constant and $\beta \geq n \geq 1$ (n the algebraic index of convergence of V_k).

The proof of this lemma is straightforward. Note that $\beta \geq n$ because the density function is usually smoother than a payoff function.

Collecting the results (2.36), (2.40), (2.42), and (2.43), we can summarize that, with a properly chosen truncation of the integration range, the overall error converges either exponentially for density functions, with nonzero derivatives, belonging to $\mathbb{C}^{\infty}([a,b] \subset \mathbb{R})$, i.e.,

$$|\epsilon| < |\epsilon_1| + Q\,|\epsilon_4| + Pe^{-(N-1)\nu}, \tag{2.44}$$

or algebraically for density functions with a discontinuity in one of its derivatives, i.e.,

$$|\epsilon| < |\epsilon_1| + Q\,|\epsilon_4| + \frac{\bar{P}}{(N-1)^{\beta-1}}. \tag{2.45}$$

2.5 Numerical Results

In this section, we perform a variety of numerical tests to evaluate the efficiency and accuracy of the COS method. Implementation of the COS formula is straightforward. We focus on the plain vanilla European options and consider different processes for the underlying asset from GBM to the Heston stochastic volatility process and the infinite activity Lévy processes VG and CGMY. In the latter case we choose a value for parameter Y close to 2, representing a distribution with very heavy tails. We will choose long and short maturities in the tests.

The underlying density function for each individual experiment is also recovered with the help of the cosine series based inversion technique presented in Section 2.2. This may help the reader to get some insight into the relationship between the error convergence and the properties of the densities.

We compare our results with the COS method to two of its competitors, the Carr–Madan method [20] and the CONV method (see Chapter 1). However, contrary to the common implementations of these methods we use the Simpson rule for the Fourier integrals in order to achieve fourth order accuracy. The FFT has been used for the Carr–Madan as well as for the CONV method.

By these numerical experiments and comparisons with the other methods, we aim to demonstrate the stability and robustness of the COS method, also under extreme conditions.

It should be noted that parameter N in the experiments to follow denotes, for the COS method, the number of terms in the Fourier-cosine expansion, and it denotes the number of grid points for the other two methods.

All CPU times presented, in milliseconds, are determined after averaging the computing times obtained from 10^4 experiments. The computer used for all experiments has an Intel Pentium 4 CPU, 2.80GHz with cache size 1024 KB; the code is written in MATLAB 7-4.

Remark 2.5.1. *Some experience is helpful when choosing the correct truncation range and damping factor α in the Carr–Madan method. A suitable choice appears to be $\alpha = 0.75$ from [68] for the experiments based on GBM as well as on the Heston model. This is the parameter used in the experiments to follow. However, many α-values have been suggested in the literature for optimal convergence, even $\alpha = 25$ in [63]. Optimal values are determined numerically in [55].*

2.5.1 Truncation range for COS method

To determine the interval of integration $[a, b]$ within the COS method, we propose the following:

$$[a,b] := \left[\xi_1 - L\sqrt{\xi_2 + \sqrt{\xi_4}}, \quad \xi_1 + L\sqrt{\xi_2 + \sqrt{\xi_4}}\right] \quad \text{with } L = 10. \tag{2.46}$$

Here ξ_n denotes the nth cumulant of $\ln(S_T/K)$. The cumulants for the models employed are presented in Section 1.2.1.

Cumulant ξ_4 is included in (2.46), because the density functions of many Lévy processes for short maturity, T, have sharp peaks and fat tails (correctly indicated via ξ_4).

Formula (2.46) is accurate[5] in the range $T = 0.1$ to $T = 10$. It then defines a truncation range which gives a truncation error around 10^{-12}. Larger values of parameter L would require larger N to reach the same level of accuracy.

Remark 2.5.2. *When pricing call options, the method's accuracy exhibits some sensitivity regarding the choice of parameter L in (2.46). A call payoff grows exponentially with the log-stock price which may introduce a significant cancelation error for large values of L. Put options do not suffer from this, as their payoff value is bounded by the value K. For pricing call options, one can therefore either stay with $L \in [7.5, 10]$ or rely on the well-known put-call parity,*

$$v^{call}(\mathbf{x}, t_0) = v^{put}(\mathbf{x}, t_0) + S_0 e^{-qT} - \mathbf{K} e^{-rT}. \tag{2.47}$$

In the experiments to follow, we use (2.47) when pricing calls, which gives a slightly higher accuracy than directly applying (2.28) with (2.46).

2.5.2 Geometric Brownian Motion

The first set of call option experiments is performed under the GBM process with a short time to maturity. Parameters selected for this test are

$$S_0 = 100, \quad r = 0.1, \quad q = 0, \quad T = 0.1, \quad \sigma = 0.25. \tag{2.48}$$

The convergence behavior at three different strike prices, $K = 80, 100$, and 120, is checked.

Figure 2.1 shows that the recovered density function with the small maturity time T does not have fat tails, as is commonly known. This, however, implies that the tails of the characteristic function in the Fourier domain are fat. As a result, the truncation range for the Carr–Madan method in the Fourier domain has to be selected relatively large, requiring a significantly larger value of N compared to the other two methods to achieve the same level of accuracy.

As shown in Figure 2.2, the error convergence of the COS method is exponential (geometric) and superior to that of the fourth order CONV and Carr–Madan

[5] A truncation rule which includes cumulant ξ_6, such as $[a,b] := [\xi_1 - L\sqrt{\xi_2 + \sqrt{\xi_4 + \sqrt{\xi_6}}}, \quad \xi_1 + L\sqrt{\xi_2 + \sqrt{\xi_4 + \sqrt{\xi_6}}}]$, is more accurate for extremely short maturities, like $T = 0.001$. The sixth cumulant is, however, relatively difficult to derive for many models.

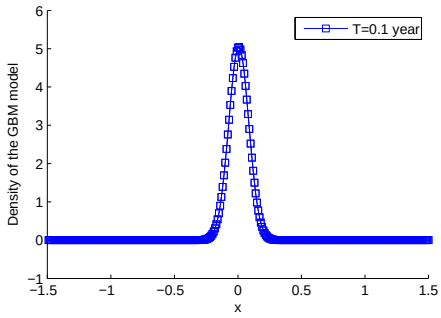


Figure 2.1: Recovered density function of the GBM model involved in the experiments; $K = 100$, with other parameters as in (2.48).

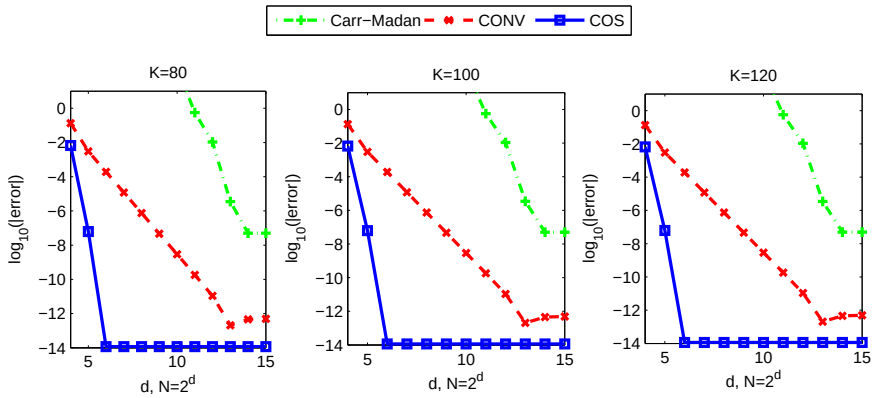


Figure 2.2: COS versus Carr–Madan and CONV in error convergence for pricing European call options under the GBM model.

methods. With $N = 2^6$, the COS results already coincide with the reference values. Further, we observe that the error convergence rate is basically the same for the different strike prices.

In Table 2.2, CPU time and error convergence information, comparing the COS and the Carr–Madan method, are displayed for pricing the options at $K = 80, 100$, and 120. The maximum error of the option values over the three strike prices is presented. The results for these strikes are obtained in one single computation for both methods.

To get the same level of accuracy, the COS method uses significantly less CPU time, which becomes more prominent when the desired accuracy is high. For the Carr–Madan computation we have used a truncation range of size $[0, 100]$ in this latter experiment.[6]

Remark 2.5.3. *In all numerical experiments we observe a linear computational*

[6]To produce the Carr–Madan results from Figure 2.2 with the very small errors, we needed a larger truncation range, i.e., $[0, 1200]$.

complexity for the COS method. By doubling N, performing the computations, and checking the differences *between subsequent timings, we can distinguish the linear complexity from the computational overhead.*

Table 2.2: Error convergence and CPU time comparing the COS and Carr–Madan methods for European calls under GBM, with parameters as in (2.48); $K = 80, 100, 120$; reference val. $= 20.799226309\ldots$, $3.659968453\ldots$, and $0.044577814\ldots$, respectively.

	N	16	32	64	128	256
COS	msec.	0.33	0.38	0.50	0.73	1.30
	max. abs. err.	6.66e-03	7.17e-08	3.91e-14	3.91e-14	3.91e-14
Carr–Madan	msec.	2.45	2.57	2.74	3.18	3.85
	max. abs. err.	2.45e+07	1.76e+06	1.62e+03	1.62e+01	7.95e-02

Cash-or-nothing option

We confirm that the convergence of the COS method does not depend on a discontinuity in the payoff function, provided we have an analytic expression for the coefficients V_k^{cash} by pricing a cash-or-nothing call option here. The underlying process is GBM, so that an analytic solution exists. Parameters selected for this test are

$$S_0 = 100, \quad K = 120, \quad r = 0.05, \quad q = 0, \quad T = 0.1, \quad \sigma = 0.2. \tag{2.49}$$

Table 2.3 presents the exponential convergence of the COS method. Since the payoff is bounded here, we apply the COS formula (2.30) directly.

2.5.3 The Heston model

As a second test we choose the Heston model and price calls with the following parameters:

$$\begin{aligned} S_0 &= 100, \quad K = 100, \quad r = 0, \quad q = 0, \quad \lambda = 1.5768, \quad \eta = 0.5751, \\ \bar{\nu} &= 0.0398, \quad u_0 = 0.0175, \quad \rho = -0.5711. \end{aligned} \tag{2.50}$$

Two maturities, $T = 1$ and $T = 10$, are considered. Since the analytic formula for ξ_4 is involved (it can be obtained using Maple, but it is lengthy), we define the truncation range, instead of (2.46), by

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

Cumulant ξ_2 may become negative for sets of Heston parameters that do not satisfy the Feller condition, i.e., $2\bar{\nu}\lambda > \eta^2$. We therefore use the absolute value of ξ_2.

Figure 2.3 presents the recovered density functions. It shows that $T = 1$ gives rise to a sharper-peaked density than $T = 10$, as expected.

In this test, we compare the COS method with the Carr–Madan method, which is often used for the calibration of the Heston model in industry. The option price reference values are obtained by the Carr–Madan method using $N = 2^{17}$ points, and the truncated Fourier domain is set to $[0, 1200]$ for the experiment with $T = 1$ and to $[0, 500]$ for $T = 10$.

Table 2.3: Error and CPU time for a cash-or-nothing call option with the COS method, with parameters as in (2.49); reference val. $= 0.273306496\ldots$.

N	40	60	80	100	120	140
Error	2.46e-02	1.64e-02	6.35e-04	6.85e-06	2.44e-08	2.79e-11
CPU time (msec.)	0.330	0.334	0.38	0.43	0.49	0.50

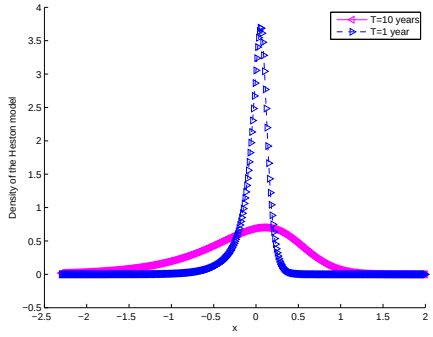


Figure 2.3: Recovered density functions of the Heston experiments, with parameters as in (2.50).

Tables 2.4 and 2.5 illustrate the high efficiency of the COS method compared to the Carr–Madan method.

Table 2.4: Error convergence and CPU times for the COS and Carr–Madan methods for calls under the Heston model with $T = 1$, with parameters as in (2.50); reference val. $= 5.785155450\ldots$.

COS			Carr–Madan		
N	Error	Time (msec.)	N	Error	Time (msec.)
64	−4.92e-03	0.61	256	−2.29e+06	4.70
96	−2.99e-04	0.78	512	2.31e+01	6.94
128	1.94e-05	0.94	1024	−2.61e-01	11.30
160	2.99e-06	1.11	2048	−2.14e-03	20.29
192	−3.17e-07	1.27	4096	3.76e-07	38.54

Note the very different values of N that the two methods require for satisfactory convergence. All CPU times are given in milliseconds. The COS method appears to be approximately a factor 20 faster than the Carr–Madan method for the same level of accuracy. The convergence rate of the COS method is somewhat slower for the short maturity example, as compared to the 10 year maturity. This is due to the fact that the density function for the latter case is smoother, as seen in Figure 2.3. The COS convergence rate for $T = 1$ is, however, still exponential in the Heston model.

Additionally, for a fair comparison, we mimic the calibration situation, in which around 20 strikes are priced simultaneously. We repeat the experiment for $T = 1$

Table 2.5: Error convergence and CPU time for the COS and Carr–Madan methods for calls under the Heston model with $T = 10$, with parameters as in (2.50); reference val. $= 22.318945791\ldots$.

COS			Carr–Madan		
N	Error	Time (msec.)	N	Error	Time (msec.)
32	7.40e-03	0.46	128	−1.99e+06	3.64
64	−5.02e-05	0.62	256	1.36e+05	4.78
96	1.40e-07	0.81	512	3.27e+01	7.08
128	4.92e-10	0.98	1024	−2.61e-01	11.38
160	−1.85e-10	1.36	2048	−2.15e-03	20.93

but now with 21 consecutive strikes, $K = 50, 55, 60, \ldots, 150$; see the results in Table 2.6. The maximum error over all strike prices is presented. With $N = 160$, the COS method can price all options for 21 strikes highly accurately, within 3 milliseconds.

Table 2.6: Error convergence and CPU time for calls under the Heston model by the COS and Carr–Madan method, pricing 21 strikes, with $T = 1$, with parameters as in (2.50).

COS	N	32	64	96	128	160
	CPU time (msec.)	0.85	1.45	2.04	2.64	3.22
	max. abs. err.	1.43e-01	6.75e-03	4.52e-04	2.61e-05	4.40e-06
Carr–Madan	N	512	1024	2048	4096	8192
	CPU time (msec.)	7.44	12.84	20.36	37.69	76.02
	max. error	4.70e+06	6.69e+01	2.61e-01	2.15e-03	2.08e-07

2.5.4 Variance Gamma

As a next example we price call options under the VG process, which belongs to the class of infinite activity Lévy processes. The VG process is usually parameterized with parameters σ, θ, and ν related to C, G, and M in (1.7) through

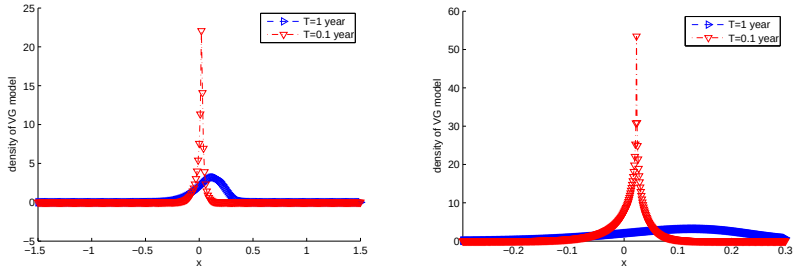
$$C = \frac{1}{\nu}, \quad G = \frac{\theta}{\sigma^2} + \sqrt{\frac{\theta^2}{\sigma^4} + \frac{2}{\nu\sigma^2}}, \quad M = -\frac{\theta}{\sigma^2} + \sqrt{\frac{\theta^2}{\sigma^4} + \frac{2}{\nu\sigma^2}}. \tag{2.51}$$

The parameters selected in the numerical experiments are

$$K = 90,\ S_0 = 100,\ r = 0.1,\ q = 0,\ \sigma = 0.12,\ \theta = -0.14,\ \nu = 0.2,\ L = 10. \tag{2.52}$$

This case has been chosen because a relatively slow convergence was reported for the CONV method for very short maturities in [54]. Here we compare the convergence for $T = 1$ year and for $T = 0.1$ year.

Figure 2.4 presents the difference in shape of the two recovered density functions. For $T = 0.1$, the density is much more peaked. Results are summarized in Table 2.7. Note that for $T = 0.1$ the error convergence of the COS method is algebraic instead of exponential. This is in agreement with the recovered density function in Figure



(a) Whole density function (b) Zoom in

Figure 2.4: Recovered density functions for the VG model and two maturity dates; $K = 90$, with other parameters as in (2.52).

Table 2.7: Convergence of the COS method for a call under the VG model with $K = 90$ and other parameters as in (2.52).

$T = 0.1$; Reference val. $= 10.993703187\ldots$			$T = 1$; Reference val. $= 19.099354724\ldots$		
N	Error	Time (msec.)	N	Error	Time (msec.)
64	−1.66e-03	0.46	32	−6.57e-04	0.35
128	4.35e-04	0.65	64	2.10e-06	0.47
256	4.55e-05	1.03	96	−3.32e-08	0.56
512	−1.13e-06	1.79	128	4.19e-10	0.64
1024	2.52e-08	3.40	160	−1.88e-11	0.75

2.4, which is clearly not in $C^\infty([a, b])$. In the extreme case, we would observe a delta function-like function for $T \to 0$.

We also plot the errors in Figure 2.5, comparing the convergence of the COS method to that of the CONV method.[7] The convergence rate of the COS method for $T = 1$ is significantly faster than that of the CONV method, but for $T = 0.1$ the convergence is comparable.

2.5.5 CGMY process

Finally, we evaluate the method's convergence for calls under the CGMY model. It has been reported in [1, 73] that PIDE methods have difficulty solving the cases for which parameter $Y \in [1, 2]$. Therefore we evaluate the COS method with $Y = 0.5$, $Y = 1.5$, and $Y = 1.98$, respectively. The other parameters are selected as follows:

$$S_0 = 100,\ K = 100,\ r = 0.1,\ q = 0,\ C = 1,\ G = 5,\ M = 5,\ T = 1. \tag{2.53}$$

In Figure 2.6, the recovered density functions for the three cases are plotted. For large values of Y, the tails of the density function are fatter and the center of the distribution shifts.

Reference values for the numerical experiments are computed by the COS method with $N = 2^{14}$, as there are no reference values available for the latter cases. The

[7] The Simpson rule did not improve the convergence rate here.

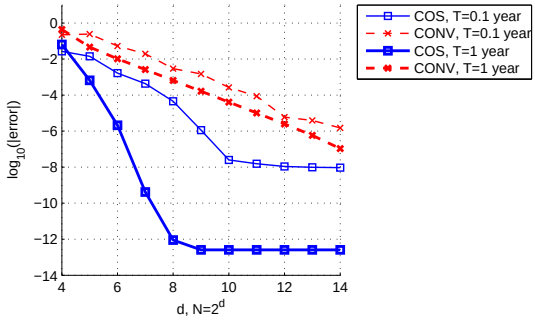
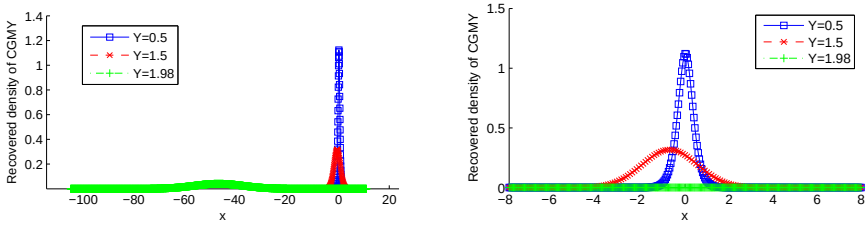


Figure 2.5: Convergence of the COS method for the VG model.



(a) Complete density functions

(b) Zoom of density functions

Figure 2.6: Recovered density functions for the CGMY model with different values of Y; other parameters are as in (2.53).

numerical results are presented in Tables 2.8 and 2.9 for $Y = 0.5$ and $Y = 1.5$, respectively.

Again, the COS method converges exponentially, which is faster than the fourth order convergence of the CONV method. With a relatively small value of N, i.e., $N \leq 100$, the COS results are accurate up to seven digits. The computational time spent is less than 0.1 millisecond. Comparing Tables 2.8 and 2.9, we notice that the convergence rate with $Y = 1.5$ is faster than that of $Y = 0.5$, because density functions from fat-tailed distributions can often be well represented by cosine basis functions. In Table 2.10, for example, with $Y = 1.98$ we need very small values of N for highly accurate call option prices. No other pricing method, to our knowledge, can price options for very large $Y \approx 2$ accurately in a robust way.

2.6 Conclusions and Discussion

In this chapter we have introduced an option pricing method based on Fourier-cosine series expansions, the COS method, for pricing European-style options. The method can be used as long as a characteristic function for the underlying price process is available. The COS method is based on the insight that the series coefficients of many density functions can be accurately retrieved from their characteristic functions. As such, one can decompose a density function into a linear combination of

Table 2.8: Comparison of the COS and CONV methods in accuracy and speed for CGMY with $Y = 0.5$ and other parameters as in (2.53); reference val. $= 19.812948843\ldots$.

COS			CONV		
N	Error	Time (msec.)	N	Error	Time (msec.)
32	1.36e-02	0.61	64	1.53e-02	0.66
48	5.61e-04	0.69	128	5.31e-04	0.94
64	3.32e-05	0.78	256	3.15e-05	1.49
80	2.57e-06	0.89	512	1.62e-06	2.89
96	2.44e-07	0.95	1024	−1.82e-07	4.90
128	3.11e-09	1.11	2048	−2.71e-07	9.64

Table 2.9: Comparison of the COS and CONV methods in accuracy and speed for CGMY with $Y = 1.5$ and other parameters from (2.53); reference val. $= 49.790905469\ldots$.

COS			CONV		
N	Error	Time (msec.)	N	Error	Time (msec.)
8	2.40e-01	0.53	64	1.21e-02	0.70
16	−4.92e-02	0.56	128	7.12e-04	1.13
24	−1.73e-03	0.58	256	4.37e-05	1.79
32	−1.23e-05	0.63	512	2.81e-06	3.13
40	−2.16e-08	0.68	1024	1.49e-07	5.32
48	−3.60e-11	0.72	2048	6.49e-10	9.98

Table 2.10: The COS method for CGMY model with $Y = 1.98$ and other parameters as in (2.53); reference val. $= 99.999905510\ldots$.

N	8	16	24	32	40	48
Msec.	0.52	0.55	0.61	0.63	0.66	0.70
Error	−6.36e-01	2.65e-02	1.00e-04	4.29e-06	3.25e-09	1.18e-11

cosine functions. It is this decomposition that makes the numerical computation of the risk-neutral valuation formula easy and highly efficient.

Derivation of the COS method has been accompanied by an error analysis. In several numerical experiments, the convergence rate of the COS method has shown to be exponential, in accordance with the analysis. When the density function of the underlying process has a discontinuity in one of its derivatives an algebraic convergence is expected and was observed. The computational complexity of the COS method is linear in the number of terms, N, chosen in the Fourier-cosine series expansion. Very fast computing times were reported here for the Heston and the Lévy models. With $N < 150$, all numerical results (except for the VG model with very short maturities) are accurate up to eight digits, in less than 1 millisecond of CPU time. By recovering the density function we can estimate the convergence behavior of our numerical method.

Chapter 3

Pricing Early-Exercise and Barrier Options under Exponential Lévy Processes

3.1 Introduction

In this chapter, the COS method is generalized to pricing Bermudan, American and discrete barrier options under Lévy processes. It exhibits the same convergence behavior as for European options, and the computational complexity is almost linear. This chapter contains essentially the contents of the paper [33].

Within stock option pricing applications, interesting numerical mathematics questions can be found in product pricing and in calibration. Whereas the former topic requires especially robust numerical techniques, the latter also relies on efficiency and speed of computation.

Numerical integration methods, based on a transformation to the Fourier domain (the so-called *transform methods*), are traditionally very efficient, due to the availability of the Fast Fourier Transform (FFT) [20, 61], for the pricing of basic European products, and thus for calibration purposes.

Recently, transform methods have been generalized to solving somewhat more complicated option contracts, like Bermudan, American or barrier options, see, for example, [54, 30, 6, 7, 45, 70, 26, 69]. These exotic options, still with basic features, are used in the financial industry as building blocks for more complicated products. A natural aim for the near future with these transform methods is to calibrate to these exotic products and to price the huge portfolios (at the end of a trading day) very fast.

Next to FFT-based methods, new techniques based on the Fast Gauss or the Hilbert Transform have been introduced for this purpose [15, 16, 36]. In this chapter we will also generalize a transform method to pricing Bermudan, American and discretely-monitored barrier options. It is the method based on Fourier-cosine series expansions, called the COS method, introduced by us in [32], where we showed that it was highly efficient for pricing European options.

The integration-based methods are, for these option contracts, in competition

with the methods that require the solution of discrete partial (integro-) differential equation-based operators (PIDE) [74, 18]. PIDE-based methods are traditionally used since early-exercise and the exotic features can often be interpreted as special payoffs or boundary conditions. They represent the state of the art for pricing options under the local volatility process. Generally speaking, however, the computational process with PIDE is rather expensive, especially for the infinite activity Lévy processes we are interested in, because they give rise to an integral in the PIDE with a weakly singular kernel [2, 42, 73].

We will therefore compare our results with other highly efficient transform methods, i.e., with the Convolution (CONV) method [54] (see Chapter 1), based on the FFT, which is one of the state-of-the-art methods for pricing Bermudan and American options. Its computational complexity for pricing a Bermudan option with M exercise dates is $O((M-1)N\log_2(N))$, where N denotes the number of grid points used for numerical integration. Quadrature rule based techniques are, however, not of the highest efficiency when solving Fourier transformed integrals. As these integrands are highly oscillatory, a relatively fine grid has to be used for satisfactory accuracy with the FFT. The COS method presented here requires a substantially smaller value of N (the number of leading terms in the Fourier-cosine series expansion).

Especially for barrier options, another highly efficient alternative method from [36] is based on the Hilbert transform. Its error convergence is exponential for models with rapidly decaying characteristic functions, also with a computational complexity of $O((M-1)N\log_2 N)$ for a barrier option with M monitoring dates. This method is, however, not applicable for Bermudan options.

This chapter is organized as follows. In Section 3.2 the COS method for pricing Bermudan and barrier options is presented. The handling of the discretely monitored barrier options is discussed in particular in Subsection 3.2.4. Error analysis is performed in Section 3.3. Numerical results are presented in Section 3.4, where we focus on option pricing under exponential Lévy processes, in particular under the CGMY [19] and the Normal Inverse Gaussian [8] processes.

3.2 Pricing Bermudan and Barrier Options

A Bermudan option can be exercised at pre-specified dates before maturity. The holder receives the exercise payoff when he/she exercises the option. Between two consecutive exercise dates the valuation process can be regarded as that for a European option, which can be priced with the help of the risk-neutral valuation formula.

Let t_0 denote the initial time and $\mathcal{T}=\{t_1,\cdots,t_M\}$ be the collection of all exercise dates with $\Delta t := (t_m - t_{m-1})$, $t_0 < t_1 < \cdots < t_M = T$. The pricing formula for a Bermudan option with M exercise dates has been given in (1.16). Here we recall the two main functions: for $m = M, M-1, \ldots, 2$, it holds that

$$\begin{cases} c(x,t_{m-1}) &= e^{-r\Delta t}\int_{\mathbb{R}} v(y,t_m)f(y|x)dy, \\ v(x,t_{m-1}) &= \max\left(g(x,t_{m-1}), c(x,t_{m-1})\right), \end{cases} \tag{3.1}$$

followed by

$$v(x,t_0) = e^{-r\Delta t}\int_{\mathbb{R}} v(y,t_1)f(y|x)dy. \tag{3.2}$$

Here x and y are state variables, defined as the logarithm of the ratio of the asset price S_t over the strike price K,

$$x := \ln(S(t_{m-1})/K) \quad \text{and} \quad y := \ln(S(t_m)/K),$$

$v(x,t)$, $c(x,t)$ and $g(x,t)$ are the option value, the continuation value and the payoff at time t, respectively. Note that for vanilla options, $g(x,t)$ equals $v(x,T)$, with

$$v(x,T) = [\varrho K(e^x - 1)]^+, \quad \varrho = \begin{cases} 1 & \text{for a call,} \\ -1 & \text{for a put.} \end{cases}$$

The probability density function of y given x under a risk-neutral measure is denoted by $f(y|x)$ in (3.2), and r is the (deterministic) risk-neutral interest rate.

Equations (3.1), (3.2) can be efficiently evaluated by the COS method in [32], provided that the Fourier-cosine series coefficients of $v(y,t_m)$ are known.

3.2.1 The COS Method for Continuation Values

The continuation value in (3.1) can be obtained using the same idea as in Chapter 2. In the following, we briefly recall the derivation.

Suppose that we have, with $[a,b] \subset \mathbb{R}$,

$$\int_{\mathbb{R}\setminus[a,b]} f(y|x)dy < \text{ TOL}, \tag{3.3}$$

for some given tolerance, TOL, then we can approximate $c(x,t_{m-1})$ in (3.1) by

$$c_1(x,t_{m-1}) = e^{-r\Delta t} \int_a^b v(y,t_m) f(y|x) dy. \tag{3.4}$$

(The intermediate terms, c_i, are used in the error analysis in Section 3.3.) We replace the density function by its Fourier-cosine series expansion on $[a,b]$,

$$f(y|x) = {\sum_{k=0}^{\infty}}' A_k(x) \cos\left(k\pi \frac{y-a}{b-a}\right). \tag{3.5}$$

The series coefficients $\{A_k(x)\}_{k=0}^{\infty}$ are defined by

$$A_k(x) := \frac{2}{b-a} \int_a^b f(y|x) \cos\left(k\pi \frac{y-a}{b-a}\right) dy. \tag{3.6}$$

Interchanging the summation and integration operators yields

$$c_1(x,t_{m-1}) = \frac{1}{2}(b-a)e^{-r\Delta t} {\sum_{k=0}^{\infty}}' A_k(x) V_k(t_m), \tag{3.7}$$

with $V_k(t_m)$ the Fourier-cosine series coefficients of $v(y,t_m)$ on $[a,b]$, i.e.

$$V_k(t_m) := \frac{2}{b-a} \int_a^b v(y,t_m) \cos\left(k\pi \frac{y-a}{b-a}\right) dy. \tag{3.8}$$

Truncating the infinite series gives

$$c_2(x,t_{m-1}) = \frac{1}{2}(b-a)e^{-r\Delta t}\sum_{k=0}^{N-1}{}' A_k(x)V_k(t_m). \tag{3.9}$$

As a third step, we use the relation between $A_k(x)$ and the conditional characteristic function, $\phi(\omega;x)$, defined as

$$\phi(\omega;x) := \int_{\mathbb{R}} f(y|x)e^{i\omega y}dy. \tag{3.10}$$

Coefficients $A_k(x)$ can be written as

$$A_k(x) = \frac{2}{b-a}\mathrm{Re}\left\{e^{-ik\pi\frac{a}{b-a}}\int_a^b e^{i\frac{k\pi}{b-a}y}f(y|x)dy\right\}. \tag{3.11}$$

where $\mathrm{Re}\{\cdot\}$ denotes taking the real part of the input argument. With (3.3), the finite integration in (3.11) can be approximated by

$$\int_a^b e^{i\frac{k\pi}{b-a}y}f(y|x)dy \approx \int_{\mathbb{R}} e^{i\frac{k\pi}{b-a}y}f(y|x)dy =: \phi\left(\frac{k\pi}{b-a};x\right).$$

As a result, $A_k(x)$ can be approximated by $F_k(x)$ with

$$F_k(x) := \frac{2}{b-a}\mathrm{Re}\left\{\phi\left(\frac{k\pi}{b-a};x\right)e^{-ik\pi\frac{a}{b-a}}\right\}. \tag{3.12}$$

Replacing $A_k(x)$ in (3.9) by $F_k(x)$ gives the COS formula for pricing European options for different underlying processes:

$$\hat{c}(x,t_{m-1}) := e^{-r\Delta t}\sum_{k=0}^{N-1}{}' \mathrm{Re}\left\{\phi\left(\frac{k\pi}{b-a};x\right)e^{-ik\pi\frac{a}{b-a}}\right\}V_k(t_m). \tag{3.13}$$

Here the function $\hat{c}(x,t_{m-1})$ represents the approximation of the continuation value $c(x,t_{m-1})$. An error analysis justifying the different approximations for European options was presented in [32] and has been given in Chapter 2.

For exponential Lévy processes, formula (3.13) can be simplified to

$$\hat{c}(x,t_{m-1}) = e^{-r\Delta t}\sum_{k=0}^{N-1}{}' \mathrm{Re}\left\{\varphi_{levy}\left(\frac{k\pi}{b-a}\right)e^{ik\pi\frac{x-a}{b-a}}\right\}V_k(t_m), \tag{3.14}$$

where $\varphi_{levy}(\omega) := \phi_{levy}(\omega;0)$, see [32]. Using this, we can also approximate $v(x,t_0)$ in (3.2) by

$$\hat{v}(x,t_0) = e^{-r\Delta t}\sum_{k=0}^{N-1}{}' \mathrm{Re}\left\{\varphi_{levy}\left(\frac{k\pi}{b-a}\right)e^{ik\pi\frac{x-a}{b-a}}\right\}V_k(t_1), \tag{3.15}$$

provided that the series coefficients, $V_k(t_1)$, are known. We will show that the $V_k(t_m)$, $k=0,1,\cdots,N-1$, can be recovered from $V_j(t_{m+1})$, $j=0,1,\cdots,N-1$.

3.2.2 Pricing Bermudan Options

The idea of pricing Bermudan options is to compute $V_k(t_1)$, the cosine coefficients of the option value at time point t_1, and insert it into (3.15), to obtain the value of the option. The main contribution of this section is the derivation of an induction formula for $V_k(t_1)$.

The integral in the definition of $V_k(t_m)$ in (3.8) can be split into two parts, if we determine the *early-exercise point*, x_m^*, at time t_m, which is the point where the continuation value equals the payoff, i.e., $c(x_m^*, t_m) = g(x_m^*, t_m)$.

Once we have x_m^*, we can split the integral that defines $V_k(t_m)$ into two parts: One on the interval $[a, x_m^*]$ and the other on $(x_m^*, b]$, i.e.

$$V_k(t_m) = \begin{cases} C_k(a, x_m^*, t_m) + G_k(x_m^*, b), & \text{for a call,} \\ G_k(a, x_m^*) + C_k(x_m^*, b, t_m), & \text{for a put,} \end{cases} \tag{3.16}$$

for $m = M-1, M-2, \cdots, 1$, and

$$V_k(t_M) = \begin{cases} G_k(0, b), & \text{for a call} \\ G_k(a, 0), & \text{for a put,} \end{cases} \tag{3.17}$$

whereby

$$G_k(x_1, x_2) := \frac{2}{b-a} \int_{x_1}^{x_2} g(x, t_m) \cos\left(k\pi \frac{x-a}{b-a}\right) dx. \tag{3.18}$$

and

$$C_k(x_1, x_2, t_m) := \frac{2}{b-a} \int_{x_1}^{x_2} c(x, t_m) \cos\left(k\pi \frac{x-a}{b-a}\right) dx. \tag{3.19}$$

Remark 3.2.1 (Newton's Method)**.** *Since the numerical approximation for $c(x, t_m)$, denoted by $\hat{c}(x, t_m)$, in (3.14) is a semi-analytic formula which returns a numerical approximation of $c(x, t_m)$ on the whole support of x, we can easily find the derivatives of $\hat{c}(x, t_m)$ w.r.t. x, and we can therefore employ Newton's method to determine x_m^*. On each time lattice, there is at most one point which satisfies $\hat{c}(x, t_m) - g(x, t_m) = 0$, for the option problems considered here* [1]*. If x_m^* is not in $[a, b]$, it is set equal to the nearest boundary point.*

Note that in this chapter we only consider the pricing of Bermudan put options.

Result 3.2.1. *The $G_k(x_1, x_2)$ in (3.18) can be determined analytically.*

Proof. With $g(x, t_m) \equiv \pm K(1 - e^x)^+$, it follows for a put, with $x_2 \leq 0$, that

$$G_k(x_1, x_2) = \frac{2}{b-a} \int_{x_1}^{x_2} K(1 - e^x) \cos\left(k\pi \frac{x-a}{b-a}\right) dx, \tag{3.20}$$

and for a call, with $x_1 \geq 0$, that

$$G_k(x_1, x_2) = \frac{2}{b-a} \int_{x_1}^{x_2} K(e^x - 1) \cos\left(k\pi \frac{x-a}{b-a}\right) dx, \tag{3.21}$$

[1] Generalizations for more early-exercise points are easily determined.

The fact that $x_m^* \leq 0$, for put options, and $x_m^* \geq 0$, for call options, $\forall t \in \mathcal{T}$, gives

$$G_k(x_1, x_2) = \frac{2}{b-a} \varrho K \left[\chi_k(x_1, x_2) - \psi_k(x_1, x_2)\right], \quad \varrho = \begin{cases} 1 & \text{for a call,} \\ -1 & \text{for a put,} \end{cases} \tag{3.22}$$

with $\chi_k(x_1, x_2)$ and $\psi_k(x_1, x_2)$ defined in (2.22) and (2.23), respectively. □

We now derive the formulas for the Fourier cosine coefficients of the option values, $V_j(t_m)$ with $j = 0, 1, \cdots, N-1$ and $m = 1, 2, \cdots, M$.

At time t_M, these coefficients, $V_j(t_M)$, are exact, see Equation (3.17). At time t_{M-1}, from COS formula (3.14) we obtain approximation $\hat{c}(x, t_{M-1})$, the continuation value at t_{M-1}, which is inserted into (3.19). Interchanging summation and integration gives the following coefficients, $\hat{C}$:

$$\hat{C}_k(x_1, x_2, t_{M-1}) = e^{-r\Delta t} \text{Re} \left\{ \sum_{j=0}^{N-1}{}' \varphi_{levy}\left(\frac{j\pi}{b-a}\right) V_j(t_M) \cdot \mathcal{M}_{k,j}(x_1, x_2) \right\}, \tag{3.23}$$

with the coefficients $\mathcal{M}_{k,j}(x_1, x_2)$ defined as

$$\mathcal{M}_{k,j}(x_1, x_2) := \frac{2}{b-a} \int_{x_1}^{x_2} e^{ij\pi \frac{x-a}{b-a}} \cos\left(k\pi \frac{x-a}{b-a}\right) dx, \tag{3.24}$$

and $i = \sqrt{-1}$ being the imaginary unit.

For time points t_m, $m = M-2, M-3, \cdots, 1$, we can define

$$\hat{C}_k(x_1, x_2, t_m) := e^{-r\Delta t} \text{Re} \left\{ \sum_{j=0}^{N-1}{}' \varphi_{levy}\left(\frac{j\pi}{b-a}\right) \hat{V}_j(t_{m+1}) \cdot \mathcal{M}_{k,j}(x_1, x_2) \right\}, \tag{3.25}$$

which is the result of replacing $V_j(t_{m+1})$ in the definition of $C_k(x_1, x_2, t_m)$ by its numerical approximation $\hat{V}_j(t_{m+1})$.

Replacing C_k in (3.16) by $\hat{C}_k$ gives us the numerical approximation of the Fourier cosine coefficients of the option values at times t_m for $m = 1, 2, \cdots, M-1$. In vector form, it reads

$$\hat{\mathbf{V}}(t_m) = \begin{cases} \hat{\mathbf{C}}(a, x_m^*, t_m) + \mathbf{G}(x_m^*, b), & \text{for a call,} \\ \hat{\mathbf{C}}(x_m^*, b, t_m) + \mathbf{G}(a, x_m^*), & \text{for a put.} \end{cases} \tag{3.26}$$

with

$$\hat{\mathbf{C}}(x_1, x_2, t_m) = \begin{cases} e^{-r\Delta t} \text{Re}\left\{\mathcal{M}(x_1, x_2)\ \Lambda\right\}\ \mathbf{V}(t_M), & m = M-1, \\ e^{-r\Delta t} \text{Re}\left\{\mathcal{M}(x_1, x_2)\ \Lambda\right\}\ \hat{\mathbf{V}}(t_{m+1}), & m = 1, 2, \cdots, M-2. \end{cases} \tag{3.27}$$

where we use bold-faced letters to denote vectors, e.g. $\mathbf{V}(t_M)$ is the vector $(V_0(t_M), V_1(t_M), \cdots, V_{N-1}(t_M))^T$. "$\mathcal{M}\ \Lambda$" denotes a matrix-matrix multiplication with $\mathcal{M}$ being a matrix with elements $\{\mathcal{M}_{k,j}\}_{k,j=0}^{N-1}$ and Λ a diagonal matrix with elements $\left\{\varphi_{levy}\left(\frac{j\pi}{b-a}\right)\right\}_{j=0}^{N-1}$.

This matrix-vector product representation is useful for analyzing the convergence properties of Bermudan option values to their American counterparts (with $M \to \infty$),

in Section 3.4.3. It should, however, not be employed to determine the coefficients, since these matrix-vector product costs $O(N^2)$ operations and is thus expensive.

In Section 3.2.3 we will present an efficient algorithm for the computation of $\hat{\mathbf{V}}(t_m)$, with complexity $O(N\log_2(N))$, based on the FFT algorithm.

We first summarize the algorithm for pricing Bermudan options:

Algorithm 3.2.1 (Pricing Bermudan options with the COS method).

Initialization: For $k = 0, 1, \cdots, N-1$,

- $V_k(t_M) = G_k(0, b)$ for call options; $V_k(t_M) = G_k(a, 0)$ for put options;

Main Loop to Recover $\hat{V}_k(t_m)$: For $m = M-1$ to 1,

- Determine early-exercise point x_m^* by Newton's method;
- Compute $\hat{V}_k(t_m)$ (with the help of the FFT algorithm).

Final step: Reconstruct $\hat{v}(x, t_0)$ by inserting $\hat{V}_k(t_1)$ into (3.15).

Remark 3.2.2 (The Greeks). *To compute the Greeks, one only needs to modify the final step in Algorithm 3.2.1, from t_1 to t_0, as the Greeks can be approximated by*

$$\hat{\Delta} = e^{-r\Delta t}\frac{2}{b-a}\sum_{k=0}^{N-1}{}' Re\left\{\left\{\varphi\left(\frac{k\pi}{b-a}\right)e^{ik\pi\frac{x-a}{b-a}}\frac{ik\pi}{b-a}\right\}\right\}\frac{\hat{V}_k(t_1)}{S_0} \tag{3.28}$$

and

$$\hat{\Gamma} = e^{-r\Delta t}\frac{2}{b-a}\sum_{k=0}^{N-1}{}' Re\left\{\left\{\varphi\left(\frac{k\pi}{b-a}\right)e^{ik\pi\frac{x-a}{b-a}}\left[-\frac{ik\pi}{b-a}+\left(\frac{ik\pi}{b-a}\right)^2\right]\right\}\right\}\frac{\hat{V}_k(t_1)}{S_0^2}. \tag{3.29}$$

3.2.3 Efficient Algorithm

In the following we will develop an FFT-based algorithm for computing the matrix-vector product in (3.27). The main insight is that matrix $\mathcal{M}$ in (3.27) is a sum of a Hankel and a Toeplitz matrix.

Theorem 3.2.1. *$\hat{\mathbf{C}}(x_1, x_2, t_m)$ in (3.27) can be computed in $O(N\log_2(N))$ operations with the help of the Fast Fourier Transform (FFT) algorithm.*

Proof. Replacing $e^{i\alpha} = \cos(\alpha) + isin(\alpha)$ in the definition of $\mathcal{M}_{k,j}(x_1, x_2)$ in (3.24) gives the following representation:

$$\mathcal{M}_{k,j}(x_1, x_2) = -\frac{i}{\pi}\left(\mathcal{M}^c_{k,j}(x_1, x_2) + \mathcal{M}^s_{k,j}(x_1, x_2)\right), \tag{3.30}$$

where

$$\mathcal{M}^c_{k,j} := \begin{cases} \dfrac{(x_2-x_1)\pi i}{(b-a)}, & k = j = 0, \\ \dfrac{\exp\left(i(j+k)\dfrac{(x_2-a)\pi}{b-a}\right) - \exp\left(i(j+k)\dfrac{(x_1-a)\pi}{b-a}\right)}{j+k}, & \text{otherwise} \end{cases} \tag{3.31}$$

and

$$\mathcal{M}^s_{k,j} := \begin{cases} \dfrac{(x_2 - x_1)\pi i}{b-a}, & k = j, \\ \dfrac{\exp\left(i(j-k)\dfrac{(x_2-a)\pi}{b-a}\right) - \exp\left(i(j-k)\dfrac{(x_1-a)\pi}{b-a}\right)}{j-k}, & k \neq j. \end{cases} \tag{3.32}$$

After inserting (3.30) into (3.23) and (3.25), we obtain a matrix-vector product representation for $\hat{\mathbf{C}}(x_1, x_2, t_m)$, i.e.,

$$\hat{\mathbf{C}}(x_1, x_2, t_m) = \frac{e^{-r\Delta t}}{\pi} \mathrm{Im}\left\{(\mathcal{M}_c + \mathcal{M}_s)\mathbf{u}\right\}, \tag{3.33}$$

where $\mathrm{Im}\{\cdot\}$ denotes taking the imaginary part of the input argument, and

$$\mathbf{u} := \{u_j\}_{j=0}^{N-1}, \quad u_j := \varphi\left(\frac{j\pi}{b-a}\right) V_j(t_{m+1}), \; u_0 = \frac{1}{2}\varphi(0)\, V_0(t_{m+1}). \tag{3.34}$$

The matrices

$$\mathcal{M}_c := \{\mathcal{M}^c_{k,j}(x_1, x_2)\}_{k,j=0}^{N-1} \qquad \text{and} \qquad \mathcal{M}_s := \{\mathcal{M}^s_{k,j}(x_1, x_2)\}_{k,j=0}^{N-1}$$

have special structure, so that the FFT algorithm can be employed for the efficient computation of matrix-vector products.

In particular, matrix $\mathcal{M}_c$ is a *Hankel* matrix,

$$\mathcal{M}_c = \begin{bmatrix} m_0 & m_1 & m_2 & \cdots & m_{N-1} \\ m_1 & m_2 & \cdots & \cdots & m_N \\ \vdots & & & & \vdots \\ m_{N-2} & m_{N-1} & \cdots & & m_{2N-3} \\ m_{N-1} & \cdots & & m_{2N-3} & m_{2N-2} \end{bmatrix}_{N\times N} \tag{3.35}$$

and $\mathcal{M}_s$ is a *Toeplitz* matrix,

$$\mathcal{M}_s = \begin{bmatrix} m_0 & m_1 & \cdots & m_{N-2} & m_{N-1} \\ m_{-1} & m_0 & m_1 & \cdots & m_{N-2} \\ \vdots & & \ddots & & \vdots \\ m_{2-N} & \cdots & m_{-1} & m_0 & m_1 \\ m_{1-N} & m_{2-N} & \cdots & m_{-1} & m_0 \end{bmatrix}_{N\times N} \tag{3.36}$$

with

$$m_j := \begin{cases} \dfrac{(x_2 - x_1)\pi}{b-a} i & j = 0 \\ \dfrac{\exp\left(ij\dfrac{(x_2-a)\pi}{b-a}\right) - \exp\left(ij\dfrac{(x_1-a)\pi}{b-a}\right)}{j} & j \neq 0 \end{cases} \tag{3.37}$$

This concludes the proof. □

The matrix-vector product, with these special matrices, can be transformed into a circular convolution. This is well-known for Toeplitz matrices, described in detail, for example, in [2]. The product $\mathcal{M}_s\mathbf{u}$ is equal to the first N elements of $\mathbf{m}_s \circledast \mathbf{u}_s$ with the $2N$-vectors:

$$\mathbf{m}_s = [m_0, m_{-1}, m_{-2}, \cdots, m_{1-N}, 0, m_{N-1}, m_{N-2}, \cdots, m_1]^T,$$

and $\mathbf{u}_s = [u_0, u_1, \cdots, u_{N-1}, 0, \cdots, 0]^T$. For the Hankel matrix this is less known, so we formulate it in the following result:

Result 3.2.2. *The product $\mathcal{M}_c\mathbf{u}$ is equal to the first N elements of $\mathbf{m}_c \circledast \mathbf{u}_c$, in reversed order, with the $2N$-vectors: $\mathbf{m}_c = [m_{2N-1}, m_{2N-2}, \cdots, m_1, m_0]^T$ and $\mathbf{u}_c = [0, \cdots, 0, u_0, u_1, \cdots, u_{N-1}]^T$.*

For the efficient computation of $\mathcal{M}_c\mathbf{u}$, we need to construct the following circulant matrix, $\mathcal{M}_u$,

$$\mathcal{M}_u = \begin{bmatrix} 0 & u_{N-1} & u_{N-2} & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & u_{N-1} & u_{N-2} & \cdots & \cdots & 0 \\ \vdots & & \ddots & & \ddots & & \vdots \\ 0 & \cdots & 0 & u_{N-1} & u_{N-2} & \cdots & u_0 \\ u_0 & 0 & \cdots & 0 & u_{N-1} & \cdots & u_1 \\ u_1 & u_0 & 0 & \cdots & 0 & \cdots & u_2 \\ \vdots & & \ddots & & & \ddots & \vdots \\ u_{N-2} & \cdots & u_0 & 0 & \cdots & 0 & u_{N-1} \\ u_{N-1} & u_{N-2} & \cdots & u_0 & 0 & \cdots & 0 \end{bmatrix}_{(2N)\times(2N)}. \tag{3.38}$$

Straightforward computation shows that the first N elements of the product of $\mathbf{u}_c$ and $\mathbf{m}_c$ equal $\mathcal{M}_c\mathbf{u}$, in reversed order.

A circular convolution of two vectors is equal to the inverse discrete Fourier transform ($\mathcal{D}^{-1}$) of the products of the forward DFTs, $\mathcal{D}$, i.e.,

$$\mathbf{x} \circledast \mathbf{y} = \mathcal{D}^{-1}\{\mathcal{D}(\mathbf{x}) \cdot \mathcal{D}(\mathbf{y})\}.$$

We now summarize the algorithm of computing $\hat{\mathbf{C}}(x_1, x_2, t_m)$ as follows:

Algorithm 3.2.2 (Computation of $\hat{\mathbf{C}}(x_1, x_2, t_m)$).

1. Compute $m_j(x_1, x_2)$ for $j = 0, 1, \cdots, N-1$ using (3.37).
2. Construct $\mathbf{m_s}(x_1, x_2)$ and $\mathbf{m_c}(x_1, x_2)$ using the properties of m_j's.
3. Compute $\mathbf{u}(t_m)$ using (3.34).
4. Construct $\mathbf{u_s}$ by padding N zeros to $\mathbf{u}(t_m)$.
5. $\mathbf{Msu}$ = the first N elements of $\mathcal{D}^{-1}\{\ \mathcal{D}(\mathbf{m_s}) \cdot \mathcal{D}(\mathbf{u_s})\ \}$.
6. $\mathbf{Mcu}$ = reverse$\{$ the first N elements of $\mathcal{D}^{-1}\{\ \mathcal{D}(\mathbf{m_c}) \cdot \mathbf{sgn} \cdot \mathcal{D}(\mathbf{u_s})\ \}\}$.
7. $\hat{\mathbf{C}}(x_1, x_2, t_m) = e^{-r\Delta t}\mathrm{Im}\left\{\mathbf{Msu} + \mathbf{Mcu}\right\}/\pi$.

Note that the operation $\mathcal{D}(\mathbf{u_s})$ is computed only once, and"reverse$\{\mathbf{x}\}$" denotes an $\mathbf{x}$-generated vector, whose elements are the same as those of $\mathbf{x}$ but sorted in reversed order.

Remark 3.2.3 (Efficient computation)**.** *It is worth mentioning that the computation of the exponentials takes significantly more computer clock cycles than additions or multiplications. One can however benefit from some special properties of the m_j's, like $m_{-j} = -\overline{m_j}$ and, for $j \neq 0$,*

$$m_{j+N} = \frac{\exp\left(iN\frac{(x_2-a)\pi}{b-a}\right)\cdot\exp\left(ij\frac{(x_2-a)\pi}{b-a}\right) - \exp\left(iN\frac{(x_1-a)\pi}{b-a}\right)\cdot\exp\left(ij\frac{(x_1-a)\pi}{b-a}\right)}{j+N}.$$

So, in order to construct $\mathbf{m}_s$ and $\mathbf{m}_c$, the factors $\exp\left(ij\frac{(x_2-a)\pi}{b-a}\right)$ and $\exp\left(ij\frac{(x_1-a)\pi}{b-a}\right)$, for $j = 0, 1\cdots, N-1$, should be computed only once.

Also, the DFT of $\mathbf{u}_c$ and of $\mathbf{u}_s$ need not be computed separately, as the shift property of DFTs gives $\mathcal{D}(\mathbf{u}_c) = \mathbf{sgn}\cdot\mathcal{D}(\mathbf{u}_s)$ with $\mathbf{sgn} = [1, -1, 1, -1, \cdots]^T$.

Remark 3.2.4 (Overall Computational Complexity)**.** *Since the computation of $G_k(x_1, x_2)$ is linear in N, the overall complexity of this recovery procedure is dominated by the computation of $\hat{\mathbf{C}}(x_1, x_2, t_m)$, whose complexity is $O(N\log_2 N)$ with the FFT. As a result, the overall computational complexity for pricing a Bermudan option with M exercise dates is $O((M-1)N\log_2 N)$, as the work needed for the final step, from t_1 to t_0, is $O(N)$.*

Remark 3.2.5 (Use of FFT algorithm)**.** *In the main loop of the CONV method from [54] (Chapter 1), the FFT algorithm is required five times, the same as in the COS method presented above, and the length of the CONV input vectors is halved compared to the COS method. Therefore, the CONV method would be approximately twice as fast, if we did not take the method's accuracy into account. However, for models characterized by density functions in $\mathrm{C}^\infty[a, b]$, the COS method exhibits an exponential convergence rate, which is superior to the second order convergence of the CONV method. For the same level of accuracy, the COS method is therefore significantly faster than the CONV method.*

3.2.4 Discretely-Monitored Barrier Options

Discretely-monitored "out" barrier options are options that cease to exist if the asset price hits a certain barrier level, H, at one of the pre-specified observation dates. If $H > S_0$, they are called "up-and-out" options, and "down-and-out" otherwise. The payoff for an up-and-out option reads

$$v(x, T) = (\max(\alpha(S_T - K), 0) - r_b)\mathbf{1}_{\{S_{t_i} < H\}} + r_b, \tag{3.39}$$

where $\alpha = 1$ for a call and $\alpha = -1$ for a put, r_b is a rebate, and $\mathbf{1}_A$ is the indicator function,

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

With the set of observation dates, $\mathcal{T} = \{t_1, \cdots, t_M\}$, $t_1 < \cdots < t_{M-1} < t_M = T$, the price of an up-and-out option, monitored M times, satisfies the following recursive

formula

$$\begin{cases} c(x,t_{m-1}) &= e^{-r(t_m-t_{m-1})}\int_{\mathbb{R}} v(x,t_m)f(y|x)dy, \\ v(x,t_{m-1}) &= \begin{cases} e^{-r(T-t_{m-1})}r_b, & x \geq h, \\ c(x,t_{m-1}), & x < h, \end{cases} \end{cases} \tag{3.40}$$

where $h := \ln(H/K)$ and $m = M, M-1, \cdots, 2$.

Note that the recursive pricing formula (3.40) is very similar to that for the Bermudan options. What makes barrier pricing easier is that the root-finding algorithm is not needed as the barrier points are known in advance. Thus, similar to Bermudan options, discrete barrier options can be priced in two steps:

1. Recovery of series coefficients of the option value at t_1,
2. The COS formula for European options given by (3.15).

Based on the derivation for Bermudan options, we have the following lemma:

Lemma 3.2.1 (Backward Induction for Discrete Barrier Options)**.** *By backward recursion we find the following numerical approximation for discretely monitored barrier options: For* $m = M-1, M-2, \cdots, 1$,

$$\hat{V}_k(t_m) = \hat{C}_k(a,h,t_m) + e^{-r(T-t_{m-1})}r_b\frac{2}{b-a}\psi_k(h,b) \tag{3.41}$$

with $\hat{C}_k(x_1,x_2,t_m)$ *and* $\psi_k(x_1,x_2)$ *given by (3.33) and (2.23), respectively. If* $h < 0$, *we have*

$$V_k(t_M) = \begin{cases} 2r_b\psi_k(h,b)/(b-a) & \text{for a call,} \\ G_k(a,h) + 2r_b\psi_k(h,b)/(b-a) & \text{for a put.} \end{cases} \tag{3.42}$$

For $h \geq 0$, *we find*

$$V_k(t_M) = \begin{cases} G_k(0,h) + 2r_b\psi_k(h,b)/(b-a) & \text{for a call,} \\ G_k(a,0) & \text{for a put.} \end{cases} \tag{3.43}$$

A similar recursion formula for a down-and-out option can be derived easily.

The proof is straightforward, as it goes along the lines of the derivation for Bermudan options in the previous section.

The computation of $\hat{\mathbf{C}}(a,h,t_m)$ via (3.33) is less expensive than for Bermudan options, because h is known in advance, and consequently, $\psi_k(h,b)$ in (2.23), $\mathcal{M}_c$ and $\mathcal{M}_s$ in (3.33) are known before the recursion step. Therefore, the FFT technique is required only three times.

Barrier options with an "in" barrier, or double barrier options, can be priced as easily with the COS method. Alternatively, one could apply the barrier parity and symmetry results on "out" barrier options [71, 39].

We summarize the method by means of the following algorithm:

Algorithm 3.2.3 (Pricing Discrete Barrier Options by the COS Method).

Initialization:

- Compute $V_k(t_M)$ using (3.42) or (3.43) .
- For up-and-out: $x_1 = a$ and $x_2 = h$, and $c = h$ and $d = b$;
  For down-and-out: $x_1 = h$ and $x_2 = b$, and $c = a$ and $d = h$.
- Construct $\mathbf{m_s}(x_1, x_2)$ and $\mathbf{m_c}(x_1, x_2)$ using the properties of m_j's.
- $d_1 = \mathcal{D}\{\mathbf{m_s}(x_1, x_2)\}$, $d_2 = \mathbf{sgn} \cdot \mathcal{D}\{\mathbf{m_c}(x_1, x_2)\}$
- $\mathbf{G} = \frac{2}{b-a} r_b \ \{\psi_k(c, d)\}_{k=0}^{N-1}$.

Main Loop to Recover $\hat{\mathbf{V}}(t_{m-1})$: For $m = M$ to 2,

1. Compute $\mathbf{u}(t_m)$ using Equation (3.34).
2. Construct $\mathbf{u_s}$ by padding N zeros to $\mathbf{u}(t_m)$.
3. **Msu** = the first N elements of $\mathcal{D}^{-1}\{\ d_1 \cdot \mathcal{D}(\mathbf{u_s})\ \}$.
4. **Mcu** = reverse$\{$ the first N elements of $\mathcal{D}^{-1}\{\ d_2 \cdot \mathcal{D}(\mathbf{u_s})\ \}\quad\}$.
5. $\hat{\mathbf{C}}(t_{m-1}) = e^{-r\Delta t}/\pi \mathrm{Im}\,\{\mathbf{Msu} + \mathbf{Mcu}\}$.
6. $\hat{\mathbf{V}}(t_{m-1}) = \hat{\mathbf{C}}(t_{m-1}) + e^{-r(T-t_{m-1})}\mathbf{G}$

Finalization: Compute $\hat{v}(t_0, x)$ according to (3.15); Or Greeks by (3.28) and (3.29).

3.3 Error Analysis

In this section, we analyze the rate of convergence as well as the stability of the COS method.

3.3.1 Convergence for European Options

We define ϵ as

$$\epsilon\left(x; N, [a, b]\right) := c(x) - \hat{c}\left(x; N, [a, b]\right). \tag{3.44}$$

An upper bound for this local error with respect to the truncation range as well as the convergence rate of ϵ in dependence on N, the number of leading terms in the Fourier cosine series, have been derived in Section 2.4. Here we briefly recall the main conclusions.

The COS formula for European options was derived in three steps in Section 3.2.1. Thus, error ϵ is decomposed in three components:

1. The integration range truncation error:

$$\epsilon_1\left(x; [a, b]\right) := c(x) - c_1(x; [a, b]) = \int_{\mathbb{R}\setminus[a,b]} v(y) f(y|x) dy. \tag{3.45}$$

2. The series truncation error on $[a, b]$:

$$\epsilon_2\left(x; N, [a,b]\right) := c_1(x; [a,b]) - c_2(x; N, [a,b]) = \frac{1}{2}(b-a)e^{-r\Delta t}\sum_{k=N}^{\infty} A_k(x)\cdot V_k. \tag{3.46}$$

3. The error related to approximating $A_k(x)$ by $F_k(x)$ in (3.12):

$$\begin{aligned}\epsilon_3\left(x; N, [a,b]\right) &:= \hat{c}\left(x; N, [a,b]\right) - c_2\left(x; N, [a,b]\right)\\ &= e^{-r\Delta t}\sum_{k=0}^{N-1}{}' \operatorname{Re}\left\{\int_{\mathbb{R}\setminus[a,b]} e^{ik\pi\frac{y-a}{b-a}} f(y|x)dy\right\} V_k.\end{aligned} \tag{3.47}$$

Our focus here is on Bermudan puts, for which the option value, $v(y)$, is bounded on $[a, b]$. We then have

$$\epsilon_1\left(x; [a,b]\right) \sim O\left(\int_{\mathbb{R}\setminus[a,b]} f(y|x)\right) \sim O(\text{TOL}),$$

according to (3.3). To study the impact of x on ϵ_1, we use the property $f(y|x) = f(y-x)$, which holds for Lévy processes. After a change of variables on (3.45), we find

$$\left|\epsilon_1\left([a,b]\right)\right| = \left|\int_{\mathbb{R}\setminus[a-x,b-x]} v(x+z)f(z)dz\right| \sim O\left(\int_{\mathbb{R}\setminus[a-x,b-x]} f(z)dz\right). \tag{3.48}$$

So, when $[a, b]$ is centered around x, or $\min(|a|, |b|) >> x$, the influence of x on ϵ_1 can be ignored, and ϵ_1 only depends on the size of the truncation range: Larger intervals $[b-a]$ result in smaller values of ϵ_1. Numerical experiments supporting this are presented in Figure 3.1. The definition of a proper truncation range is given in (3.71), which is almost the same as the one for European options defined in Chapter 2 but with a different interval center.

The second error component, ϵ_2, converges *exponentially* for probability density functions of class $C^\infty([a, b])$, given a value of x [13, 32], i.e.,

$$|\epsilon_2| < P \cdot \exp(-(N-1)\nu), \tag{3.49}$$

where $\nu > 0$ is a constant and P is a term which varies less than exponentially with N. When the probability density function has a discontinuous derivative, the Fourier-cosine expansion converges *algebraically*, i.e.

$$|\epsilon_2| < \frac{\bar{P}}{(N-1)^{\beta-1}}, \tag{3.50}$$

where $\bar{P}$ is a constant and $\beta \geq n \geq 1$ (and n is the algebraic index of convergence of the series coefficients).

For Lévy processes, a non-zero x corresponds to a shift, $f(z := y - x)$, and is thus not related to the smoothness of $f(z)$. As a result, the convergence speed, ν in (3.49) or β in (3.50), does not depend on x.

The third error component, ϵ_3, consists of the integration range related truncation error [32], and can be bounded by

$$|\epsilon_3| < |\epsilon_1| + Q \left| \int_{\mathbb{R}\setminus[a,b]} f(y|x)dy \right|, \tag{3.51}$$

where Q is some constant independent of N. Applying a change of variables as for ϵ_1, it is clear that also here the choice of x has no impact on ϵ_3, if $\min(|a|, |b|) >> x$ or if $[a, b]$ is centered around x.

Collecting the three error components and applying the triangle inequality, we can bound the local error, ϵ, as follows:

$$\begin{aligned} |\epsilon(x; N, [a, b])| &\leq |c - c_1| + |c_1 - c_2| + |c_2 - \hat{c}| \\ &\leq \bar{Q} \cdot \left| \int_{\mathbb{R}\setminus[a,b]} f(y|x)dy \right| + |\epsilon_2(x; N)|, \end{aligned} \tag{3.52}$$

with $\bar{Q}$ some constant not depending on $[a, b]$ and N. With integration interval $[a, b]$ chosen sufficiently wide, the series truncation error, $\epsilon_2(N)$, dominates the overall error, which implies that for smooth density functions, ϵ converges exponentially; otherwise it goes algebraically.

3.3.2 Error Propagation in the Backward Recursion

In this section we study the error in the Fourier coefficients,

$$\varepsilon(k, t_m) := V_k(t_m) - \hat{V}_k(t_m), \tag{3.53}$$

and its propagation in the backward recursion, which is directly related to the error in the Bermudan option values. We focus on *put options* here and assume that the error resulting from applying Newton's method is not significant, i.e., the early-exercise points are determined exactly.

For ease of presentation, we analyze the case that the underlying density function is infinitely differentiable. Similar analysis can be done for other cases.

Theorem 3.3.1. *With $[a, b] \subset \mathbb{R}$ sufficiently large and a probability density function in $C^\infty([a, b])$, error $\varepsilon(k, t_m)$ converges exponentially in N.*

Proof. The proof is obtained by an induction argument. At time t_{M-1}, we compare (3.26) and (3.16), and find

$$\begin{aligned} \varepsilon(k, t_{M-1}) &= C_k(x^*_{M-1}, b, t_{M-1}) - \hat{C}_k(x^*_{M-1}, b, t_{M-1}), \\ &= \int_{x^*_{M-1}}^{b} (c(x, t_{M-1}) - \hat{c}(x, t_{M-1})) \cos\left(k\pi \frac{x-a}{b-a}\right) dx. \end{aligned} \tag{3.54}$$

Since $V_k(t_M)$ is exact, $\hat{c}(x, t_{M-1})$ resulting from the COS formula only consists of local error $\epsilon(x; N, [a, b])$. So,

$$\varepsilon(k, t_{M-1}) = \int_{x^*_{M-1}}^{b} \epsilon(x; N, [a, b]) \cos\left(k\pi \frac{x-a}{b-a}\right) dx. \tag{3.55}$$

This equation can be seen as an inner product of two square-integrable functions. With the Cauchy-Schwarz inequality, we bound error $\varepsilon(k, t_{M-1})$ as follows:

$$|\varepsilon(k, t_{M-1})| \leq \sqrt{\int_{x^*_{M-1}}^{b} \epsilon^2 \left(x; N, [a,b]\right) dx} \cdot \sqrt{\int_{x^*_{M-1}}^{b} \cos^2\left(k\pi \frac{x-a}{b-a}\right) dx}. \tag{3.56}$$

We assume that the integration interval $[a, b]$ is chosen sufficiently large, so that the local error, ϵ, is dominated by the series truncation error ϵ_2. Based on the analysis in Section 3.3.1, it then follows that, for density functions belonging to $\mathrm{C}^\infty([a, b])$, error $\epsilon(x; N)$ converges exponentially w.r.t. N, i.e.,

$$|\epsilon(x; N, [a,b])| \leq P(x, N) \exp\left(-(N-1)\nu\right),$$

where $\nu > 0$ is a constant not depending on N and x, and $P(x, N) > 0$ is a function which varies less than exponentially in N. With

$$p(N) := \max_{x \in [a,b]} P(x, N), \tag{3.57}$$

it then holds that

$$\int_{x^*_{M-1}}^{b} \epsilon^2 \left(x; N, [a,b]\right) dx \leq (b - x^*_{M-1}) \cdot \left(p(N) \cdot \exp(-(N-1)\nu)\right)^2.$$

Since $\cos^2(\alpha) \leq 1$, we have

$$\int_{x^*_{M-1}}^{b} \cos^2\left(k\pi \frac{x-a}{b-a}\right) dx \leq (b - x^*_{M-1}).$$

After inserting these parts, Equation (3.56) can be written as:

$$|\varepsilon(k, t_{M-1})| \leq (b - x^*_{M-1}) \cdot p(N) \cdot \exp(-(N-1)\nu), \tag{3.58}$$

for $k = 0, 1, \cdots, N-1$.

This indicates that the convergence behaviour of $\varepsilon(k, t_{M-1})$ in N is as the local error for pricing European options. Written in vector form, with

$$\boldsymbol{\varepsilon}(t_{M-1}) := \left(\varepsilon(0, t_{M-1}), \varepsilon(1, t_{M-1}), \cdots, \varepsilon(N-1, t_{M-1})\right)^T,$$

it follows that

$$|\boldsymbol{\varepsilon}(t_{M-1})|_\infty \leq (b - x^*_{M-1}) \cdot p(N) \cdot \exp(-(N-1)\nu). \tag{3.59}$$

As a second step, we prove that if the theorem holds for time t_{m+1}, i.e.

$$|\boldsymbol{\varepsilon}(t_{m+1})|_\infty \leq p(N) \exp\left(-(N-1)\nu\right), \tag{3.60}$$

with $p(N)$ as in (3.57), then it follows that

$$|\boldsymbol{\varepsilon}(t_m)|_\infty \sim O\left(\exp(-(N-1)\nu)\right),$$

for $m = M-2, M-3, \cdots, 1$.

At time t_m, the definition of error $\varepsilon(k,t_m)$ gives

$$\varepsilon(k,t_m) = \int_{x_m^*}^{b} \left(c(x,t_m) - \bar{c}(x,t_m)\right) \cos\left(k\pi\frac{x-a}{b-a}\right) dx, \tag{3.61}$$

where $\bar{c}(x,t_m)$ is obtained by inserting $\hat{V}_k(t_{m+1})$ into the COS formula. So,

$$\begin{aligned}\bar{c}(x,t_m) &= e^{-r\Delta t}{\sum_{j=0}}' \mathrm{Re}\left\{\varphi\left(\frac{j\pi}{b-a}\right)e^{ij\pi\frac{x-a}{b-a}}\right\}\left(V_j(t_{m+1}) - \varepsilon(j,t_{m+1})\right)\\ &= \hat{c}(x,t_m) - e^{-r\Delta t}{\sum_{j=0}}' \mathrm{Re}\left\{\varphi\left(\frac{j\pi}{b-a}\right)e^{ij\pi\frac{x-a}{b-a}}\right\}\varepsilon(j,t_{m+1}).\end{aligned}$$

Inserted in Equation (3.61), we find that $\varepsilon(k,t_m)$ consists of two parts: One related to the local error, as for the European options, and a second related to $\varepsilon(k,t_{m+1})$, i.e.

$$\varepsilon(k,t_m) = \int_{x_m^*}^{b} \left(\epsilon(x;N,[a,b]) + \bar{\epsilon}(x,t_{m+1})\right) \cos\left(k\pi\frac{x-a}{b-a}\right) dx, \tag{3.62}$$

where

$$\bar{\epsilon}(x,t_{m+1}) := e^{-r\Delta t}{\sum_{j=0}}' \mathrm{Re}\left\{\varphi\left(\frac{j\pi}{b-a}\right)e^{ij\pi\frac{x-a}{b-a}}\right\}\varepsilon(j,t_{m+1}). \tag{3.63}$$

Interchanging summation in (3.63) with integration in (3.62) gives the matrix-vector product form for the errors:

$$\boldsymbol{\varepsilon}(t_m) = \boldsymbol{\varepsilon}_1(t_m) + e^{-r\Delta t}\mathrm{Re}\left\{\mathcal{M}(x_m^*,b)\,\Lambda\right\}\boldsymbol{\varepsilon}(t_{m+1}), \tag{3.64}$$

where the matrices $\mathcal{M}$ and Λ are the same as in (3.27). Error $\boldsymbol{\varepsilon}_1$ is an N-vector with as k-th element,

$$\int_{x_m^*}^{b} \epsilon(x;N,[a,b]) \cos\left(k\pi\frac{x-a}{b-a}\right) dx,\ k = 0,1,\cdots,N-1.$$

Equation (3.64) explains how $\boldsymbol{\varepsilon}_1(t_m)$ and $\boldsymbol{\varepsilon}(t_{m+1})$ evolve in the backward recursion. To bound $\boldsymbol{\varepsilon}_1$, we can repeat the steps from (3.55) to (3.59), to find:

$$|\boldsymbol{\varepsilon}_1(t_m)|_\infty \leq (b - x_m^*)\cdot p(N)\cdot \exp\left(-(N-1)\nu\right). \tag{3.65}$$

For the term $e^{-r\Delta t}\mathrm{Re}\left\{\mathcal{M}(x_m^*,b)\,\Lambda\right\}\boldsymbol{\varepsilon}(t_{m+1})$, whose k-th element reads

$$\int_{x_m^*}^{b} \bar{\epsilon}(x,t_{m+1}) \cos\left(k\pi\frac{x-a}{b-a}\right) dx,$$

we start with the definition of $\bar{\epsilon}(x,t_{m+1})$ in (3.63) and apply the Cauchy-Schwarz inequality, as follows

$$\left(e^{r\Delta t}\bar{\epsilon}(x,t_{m+1})\right)^2 \leq {\sum_{j=0}}' \left(\mathrm{Re}\left\{\varphi\left(\frac{j\pi}{b-a}\right)e^{ij\pi\frac{x-a}{b-a}}\right\}\right)^2 {\sum_{j=0}}' \varepsilon^2(j,t_{m+1}). \tag{3.66}$$

From $A_j(x)$ and $F_j(x)$, as defined in (3.11) and (3.12), respectively, we have

$$\mathrm{Re}\left\{\varphi\left(\frac{j\pi}{b-a}\right)e^{ij\pi\frac{x-a}{b-a}}\right\} = \frac{1}{2}(b-a)\left[A_j(x) + \int_{\mathbb{R}\setminus[a,b]} f(y|x)\cos\left(j\pi\frac{y-a}{b-a}\right)dy\right].$$

Since $f(y|x) \in \mathbb{R}^+$, it is clear that $f(y|x)\cos(\alpha) \leq f(y|x)$, and thus,

$$\left(\mathrm{Re}\left\{\varphi\left(\frac{j\pi}{b-a}\right)e^{ij\pi\frac{x-a}{b-a}}\right\}\right)^2 \leq \frac{1}{4}(b-a)^2\left[A_j^2(x) + 2A_j(x)\int_{\mathbb{R}\setminus[a,b]} f(y|x)dy + \left(\int_{\mathbb{R}\setminus[a,b]} f(y|x)dy\right)^2\right]. \tag{3.67}$$

Assuming now that the interval of integration is set sufficiently large, so that the related truncation error can be neglected, and including the leading term of (3.67) into (3.66), one finds

$$\left(e^{r\Delta t}\bar{\epsilon}(x,t_{m+1})\right)^2 \leq \frac{1}{4}(b-a)^2\sum_{j=0}^{N-1}{}' A_j^2(x)\sum_{j=0}^{N-1}{}' \varepsilon^2(j,t_{m+1}).$$

For density functions belonging to $\mathrm{C}^\infty([a,b])$, the series coefficients $A_j(x)$ converge exponentially in j, see [13], so that $\sum_{j=0}^{N-1}{}' A_j^2(x)$ represents the sum of a geometric series and is therefore bounded. Define

$$W := \max_{x\in[a,b]} \sum_{j=0}^{N-1}{}' A_j^2(x).$$

With Assumption (3.60) one obtains:

$$|\bar{\epsilon}(x,t_{m+1})| \leq \frac{1}{2}(b-a)e^{-r\Delta t}\sqrt{NW}p(N)e^{-(N-1)\nu}. \tag{3.68}$$

Application of the Cauchy-Schwarz inequality results in:

$$\left|\int_{x_m^*}^{b} \bar{\epsilon}(x,t_{m+1})\cos\left(k\pi\frac{x-a}{b-a}\right)dx\right| \leq (b-x_m^*)\bar{p}(N)e^{-(N-1)\nu}, \tag{3.69}$$

or, in vector form:

$$e^{-r\Delta t}\left|\mathrm{Re}\left\{\mathcal{M}(x_m^*,b)\,\Lambda\right\}\boldsymbol{\epsilon}(t_{m+1})\right|_\infty \leq (b-x_m^*)\bar{p}(N)e^{-(N-1)\nu}, \tag{3.70}$$

where $\bar{p}(N) := \frac{1}{2}(b-a)e^{-r\Delta t}\sqrt{NW}p(N)$. Inserting (3.65) and (3.70) in (3.64) completes the proof. □

Summarizing, when the local error evolves through time, via the backward recursion, the method's convergence rate does not change. This is an indication for the method's stability.

Similarly, we can prove that if the local error converges algebraically, so does $\varepsilon(k,t_m)$.

Remark 3.3.1. *The choice of integration range, $[a, b]$, is quite important. An interval which is chosen too small will lead to a significant integration-range truncation error, whereas an interval which is set very large would require a large value for N to achieve a certain level of accuracy, as determined in (3.65) and (3.70).*

3.3.3 Choice of Truncation Range

We use the same definition of the truncation range as given in Chapter 2 (Section 2.5.1) but now we center the interval at $x_0 := \ln(S_0/K)$, i.e.

$$[a, b] := \left[(\xi_1 + x_0) - L\sqrt{\xi_2 + \sqrt{\xi_4}}, \quad (\xi_1 + x_0) + L\sqrt{\xi_2 + \sqrt{\xi_4}}\right], \tag{3.71}$$

with $L \in [8, 12]$ depending on the *user-defined tolerance level*, TOL, as given in (3.3) and $\xi_1, \ldots, \xi_4$ being the cumulants of the underlying process as given in Section 1.2.

Here, we analyze the relation between TOL and L in (3.71) via numerical experiments, aiming to determine one value of L for different exponential Lévy asset price processes. We present the observed error for different values of L in Figure 3.1. With N large, e.g. $N = 2^{14}$, the series truncation error is negligible and the integration range error, which has a direct relation to the user-defined TOL, dominates. The results in Figure 3.1 can therefore be used as a guidance for setting parameter L, given a tolerance TOL. Again, BS denotes the Black-Scholes model (Geometric Brownian Motion), VG stands for the Variance Gamma model [57], CGMY denotes the model from [19], NIG is short for the Normal Inverse Gaussian Lévy process [8], Merton denotes the jump-diffusion model developed in [59], and Kou is the jump-diffusion model from [48]. We see in Figure 3.1 that the integration range error decreases exponentially with L. The use of $L = 8$ seems appropriate for all the Lévy processes considered. This value is used in all numerical experiments to follow. Via experiments, we also found that formula (3.71), together with a proper choice of L, defines an appropriate truncation range for any maturity time longer than 0.1 years. For even shorter maturities, one can use a larger value of L.

3.4 Numerical Results

We will show the method's impressive convergence by pricing Bermudan, American and discretely-monitored barrier options. In the following, we present numerical results for the BS, CGMY and NIG models. Extensive tests (not given here) have demonstrated that the COS method also shows excellent performance for other Lévy processes. The characteristic functions as well as the cumulants for several exponential Lévy asset price processes have been give in Section 1.2.

The same computer as for Chapter 1 is used for the numerical experiments below.

In order to observe the exponential error convergence, we define a ratio,

$$\text{ratio} = \frac{\ln\left(\left|err(2^{d+1})\right|\right)}{\ln\left(\left|err(2^{d})\right|\right)}, \quad d \in \mathbb{Z}^+, \tag{3.72}$$

where $err(2^d)$ denotes the error between reference solution and approximation obtained with $N = 2^d$. If $err(N) = C_1 \exp(-P_1 N)$ with C_1 and P_1 not depending

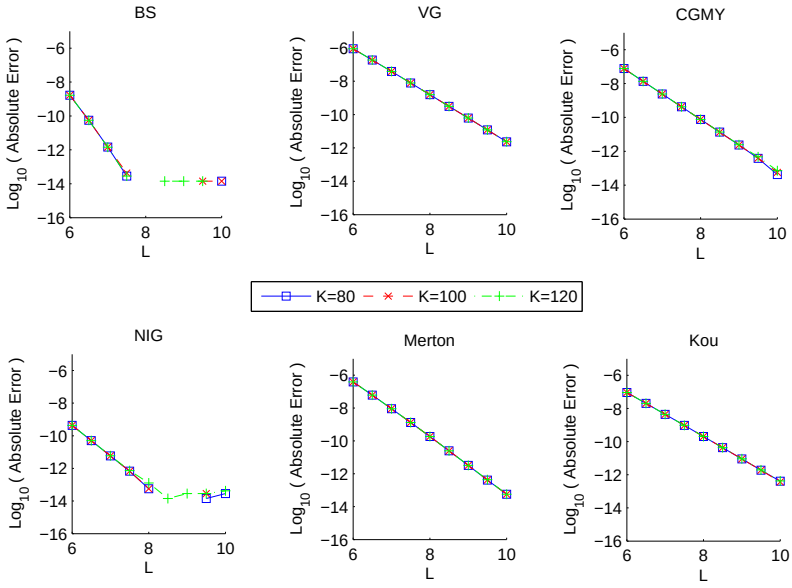


Figure 3.1: L versus the logarithm of the absolute errors for pricing calls by the COS method with $N = 2^{14}$, $T = 1$ year and three different strike prices.

on N, this ratio should be equal to 2; If the error convergence is algebraic, i.e. $err(N) = C_2 N^{-P_2}$ with C_2 and P_2 not depending on N, this ratio should equal $(d+1)/d$.

Next to the series and the integration range truncation error, another error for Bermudan options is related to the stopping criterion of the root-searching algorithm, i.e., Newton's method. With an initial guess $x^*_{m+1} = x^*_m$, $m = M-2, \ldots, 2$ (and $x^*_{M-1} = 0$), this error becomes sufficiently small, of $O(10^{-7})$ in 4 Newton iterations and of $O(10^{-10})$ in 5 iterations. In the experiments to follow, we use 5 iterations.

3.4.1 Bermudan and American Options

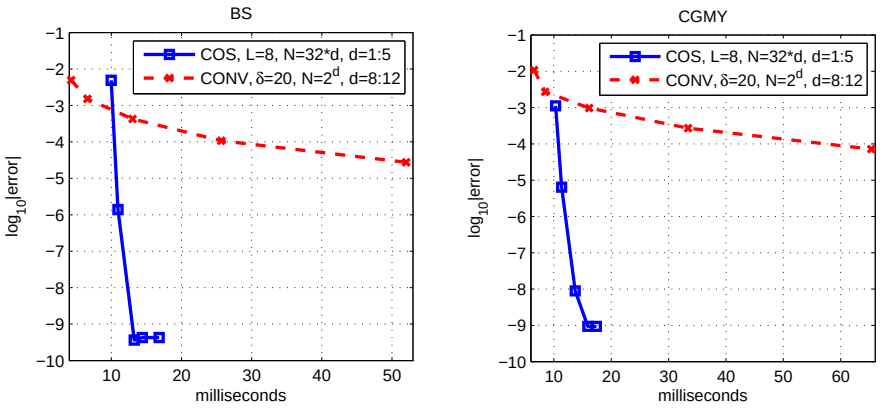
Here we price Bermudan put options with 10 exercise dates. Test parameters for two test cases are given in Table 3.1. These parameters are related to the characteristic functions presented in Table 1.1 and the cumulants from Table 1.2.

Table 3.1: Test parameters for pricing Bermudan options

Test No.	Model	S_0	K	T	r	σ	Other Parameters
1	BS	100	110	1	0.1	0.2	–
2	CGMY	100	80	1	0.1	0	$C = 1, G = 5, M = 5, Y = 1.5$

The CPU times are reported in milli-seconds, and all reference values are obtained by another method, i.e., by the CONV method from [54] (see also in Chapter 1),

setting $N = 2^{20}$.



(a) BS (Test No. 1)

(b) CGMY with $Y = 1.5$ (Test No. 2)

Figure 3.2: Error versus CPU time for pricing Bermudan put options under (a) BS and (b) CGMY model, comparing the COS and the CONV method.

The first test is for the classical BS model with as the reference value 10.479520123. In Figure 3.2a it is shown that a highly accurate solution is obtained in less than 20 milli-seconds with exponential convergence (the log-error plot displays a straight line). Compared to the quadrature rule based CONV method, which exhibits a second-order convergence, we see a significant improvement in the CPU time.

As the second test, we consider a Lévy process of infinite activity, i.e., the CGMY model with $Y > 1$ (Test 2 in Table 3.1). For this set of CGMY parameters it is now well-known that PIDE-based methods have convergence difficulties [2, 73]. The reference value is found to be $28.829781986\ldots$. The performance of the COS method for this test, shown in Figure 3.2b is highly efficient. Again, in less than 20 milli-seconds, the solution is accurate to 9 digits, compared to the reference value. Also here, we observe the exponential error convergence of the COS method.

Remark 3.4.1 (VG and Algebraic Convergence)**.** *In [32] and chapter 2 it was shown that for certain sets of parameters the Variance Gamma (VG) process gives rise to a probability density function which is not in* $C^\infty(\mathbb{R})$*, and thus option pricing under VG with these parameter sets exhibits only an algebraic convergence. This is observed for contracts with* $T < \nu$*, where* ν *denotes the variance of the VG model, see the characteristic function in Table 1.1.*

When dealing with Bermudan options this also implies that we will encounter algebraic convergence when the time between two exercise dates, $\Delta t < \nu$*.*

As mentioned in Chapter 1, the prices of American options can be obtained by applying Richardson extrapolation on the prices of a few Bermudan options with small M. Let $v(M)$ denote the value of a Bermudan option with M early exercise dates. We will use the following 4-point Richardson extrapolation scheme,

$$v_{AM}(l) = \frac{1}{21}\left(64v(2^{l+3}) - 56v(2^{l+2}) + 14v(2^{l+1}) - v(2^l)\right), \tag{3.73}$$

where $v_{AM}(l)$ denotes the approximated value of the American option.

Now we price an American option using (3.73) with the 4-point Richardson extrapolation on Bermudan puts and vary the number of exercise dates. The parameters, presented in Table 3.2, are taken from [1] and the reference value given was $V(0) = 0.112152$. We deal with the pure Lévy CGMY jump model ($\sigma = 0$) and no dividend payment ($q = 0$) here.

Table 3.2: Parameters for American put options under the CGMY model

Test No.	S_0	K	T	r	Other Parameters
3	1	1	1	0.1	$C = 1, G = 5, M = 5, Y = 0.5$

We compare the results of the COS method with those obtained by the CONV method using the same extrapolation. For the COS method, $N = 512$ and the number of Newton iterations is 5; For the CONV method $N = 4096$ to reach a very similar accuracy. The accuracy of the American prices then mainly depends on parameter d in the extrapolation (3.73). Results are summarized in Table 3.3. We can see that large values of d give highly accurate results. The COS method in combination with Richardson extrapolation gives, however, a very satisfactory accuracy within 75 milli-seconds.

Table 3.3: Errors and CPU times for pricing American puts under CGMY model, Test No. 3

l in Eq. (3.73)	COS		CONV	
	error	time (milli-sec.)	error	time (milli-sec.)
0	4.41e-05	71.41	4.37e-05	134.4
1	7.69e-06	109.2	7.01e-06	198.0
2	9.23e-07	219.3	1.05e-06	336.7
3	3.04e-07	438.9	1.29e-07	610.9

3.4.2 Barrier Options

Now we price monthly-monitored ($M = 12$) up-and-out call and put options, (UOC) and (UOP), down-and-out call and put options, (DOC) and (DOP), by the COS method. The test parameters are in Table 3.4, again related to the characteristic functions in Table 1.1. We solve the same problems as in [36] with the barrier level, $H = 120$ for the up-and-out and $H = 80$ for the down-and-out options.

The numerical results under the CGMY model (Test 4) are presented in Table 3.5. The CPU times are again measured in milli-seconds, and the reference values are obtained by the CONV method [54], with $N = 2^{15}$. Note that "ratio", as presented in the table, is different from the commonly used ratio defining the rate of convergence. In (3.72), it is the ratio of the *logarithm* of two consecutive errors. This ratio should be equal to two in the case of exponential convergence.

Table 3.4: Test parameters for pricing barrier options

Test No.	Model	S_0	K	T	r	q	Other Parameters
4	CGMY	100	100	1	0.05	0.02	$C = 4, G = 50, M = 60, Y = 0.7$
5	NIG	100	100	1	0.05	0.02	$\alpha = 15, \beta = -5, \delta = 0.5$

As expected, the COS method is more efficient for discrete barrier options than for Bermudan options, because the barrier levels are known in advance.

Exponential error convergence is observed, as the ratios (3.72) are around 2, in less than 5 milli-seconds with the results accurate up to 8 decimal places.

Table 3.5: Errors and CPU times for pricing monthly-monitored barrier options under the CGMY model (Test No. 4)

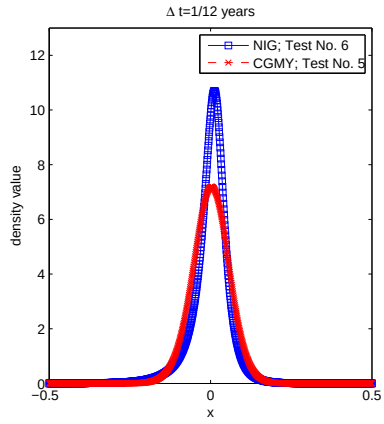
Option Type	Ref. Val.	N	time (milli-sec.)	error	ratio
DOP	2.339381026	2^4	2.8	2.23e-1	–
		2^5	2.7	1.98e-2	2.6
		2^6	3.4	3.23e-4	2.0
		2^7	4.6	7.20e-9	2.3
DOC	9.155070561	2^4	2.7	5.06e-2	–
		2^5	2.9	5.67e-3	1.7
		2^6	3.3	1.99e-4	1.6
		2^7	4.7	5.55e-9	2.2
UOP	6.195603554	2^4	3.0	5.58e-2	–
		2^5	2.9	8.98e-3	1.6
		2^6	3.6	1.96e-4	1.8
		2^7	4.8	2.23e-8	2.1
UOC	1.814827593	2^4	2.8	3.38e-1	–
		2^5	2.8	1.24e-2	4.0
		2^6	3.5	3.45e-6	2.9
		2^7	4.7	1.93e-8	1.4

Next, we focus on the NIG model (Test 5) and repeat the barrier option tests in Table 3.6. To reach the same level of accuracy as for CGMY, we need a slightly larger value of N under the NIG model. This is because the NIG density function is more peaked with the parameters from Table 3.4, as shown in Figure 3.3a. Consequently, one typically requires some more terms in the series to reconstruct the density function from its Fourier-cosine series expansion. Nevertheless, the performance of the COS method is still excellent: In less than ten milli-seconds, the accuracy is up to the 7-th decimal place.

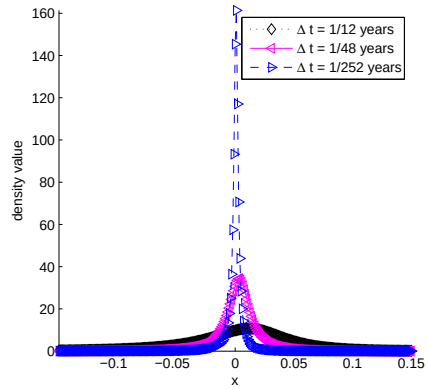
Note that, the smaller the value of Δt, the larger the value of N needs to be chosen to reach the same level of accuracy. This is because many Lévy processes have highly peaked density functions for a very small Δt. An example is presented in Figure 3.3b, where the recovered density functions of the NIG model for monthly-, weekly- and daily-monitored barrier options are plotted. We can see that for $\Delta t = 1/252$

Table 3.6: Errors and CPU times for pricing monthly-monitored barrier options under the NIG model (Test No. 5)

Option Type	Ref. Val.	N	time (milli-sec.)	error	ratio
DOP	2.139931117	2^6	3.1	4.25e-2	–
		2^7	3.7	1.28e-3	2.1
		2^8	5.4	4.65e-5	1.5
		2^9	8.4	1.39e-7	1.6
		2^{10}	14.7	1.38e-12	1.7
DOC	8.983106036	2^6	3.1	1.26e-2	–
		2^7	3.7	1.09e-3	1.6
		2^8	5.3	3.99e-5	1.5
		2^9	8.3	9.47e-8	1.6
		2^{10}	14.8	5.61e-13	1.7
UOP	5.995341168	2^6	3.4	4.84e-3	–
		2^7	3.7	1.14e-3	1.3
		2^8	5.3	7.50e-5	1.4
		2^9	8.3	1.52e-7	1.7
		2^{10}	14.7	1.24e-12	1.7
UOC	2.277861597	2^6	3.1	3.83e-2	–
		2^7	3.7	1.10e-3	2.1
		2^8	5.5	8.67e-5	1.4
		2^9	8.6	7.98e-8	1.7
		2^{10}	15.1	7.38e-13	1.7



(a) NIG and CGMY with $\Delta t = \frac{1}{12}$



(b) NIG with $\Delta t = \frac{1}{12}, \frac{1}{48}$, and $\frac{1}{252}$

Figure 3.3: The recovered density functions for (a) the NIG and the CGMY models and *monthly-monitored* barrier options and (b) the NIG model for *monthly-, weekly- and daily-monitored* barrier options.

the density is highly peaked, compared to $\Delta t = 1/12$. Nevertheless, as long as the density function is in $C^\infty(\mathbb{R})$, the error convergence rate is exponential.

We now price *daily-monitored* DOP and DOC options under the NIG model with the parameters from Test 5 in Table 3.4. The reference values are taken from [36]. The results with the COS method are summarized in Table 3.7. We observe that, as expected, the convergence rate of the COS method is exponential, but the values of N are somewhat larger than in the previous numerical experiments. The almost linear computational complexity of the method can clearly be observed from this table.

Table 3.7: Errors and CPU times for pricing daily-monitored ($M = 252$) barrier options under the NIG model (Test 5).

Option Type	Ref. Val.	N	time (milli-sec.)	error	ratio
DOP	1.88148753	2^9	130	1.25e-2	–
		2^{10}	230	2.20e-3	1.4
		2^{11}	460	1.32e-4	1.5
		2^{12}	1170	1.98e-6	1.5
		2^{13}	2560	4.70e-8	1.3
DOC	8.96705248	2^9	140	3.67e-4	–
		2^{10}	230	9.18e-5	1.2
		2^{11}	460	3.14e-5	1.1
		2^{12}	950	2.00e-6	1.3
		2^{13}	2430	5.73e-9	1.4

For results accurate up to the 4th digit, the COS method needs about 0.2 seconds for the daily-monitored DOP as well as for the DOC.

Remark 3.4.2 (Comparison to Hilbert Transform Method)**.** *The complexity of the COS method is $O((M-1)N\log_2(N))$, as the length of the induction loop (in which the FFT is employed) is $M-1$, and the final step uses N operations. Additionally, its error convergence is exponential for models with density function in the class $C^\infty([a,b])$. By considering both complexity and error convergence, the COS method is as efficient as the Hilbert transform method in [36]. The experiments above show that the COS method is as fast in terms of CPU time (although we have a slower CPU and the code is written in Matlab). The Hilbert transform based method, however, cannot be used to price Bermudan options, as the information of the early-exercise points is not known in advance. Moreover, the COS method uses more-or-less the same CPU time for different types of barrier options, which is not the case in [36].*

3.4.3 Extreme Tests

From Bermudan to American options

In this section, we discuss the behavior of the error if M, the number of early-exercise dates, goes to infinity. We also check how the Bermudan option prices converge to their American option counterparts.

American options can, in this framework, be priced basically by two approaches. One can either price a Bermudan option with very many exercise dates, or employ extrapolation methods. Whereas the latter approach is much more practical, in terms of CPU time, and has been used in Section 3.4.1, the former approach is interesting from a stability point-of-view. Here we therefore consider the pricing of Bermudan options with many exercise dates, for reasons of stability. It is interesting to consider the limit case, and check whether the method presented is still applicable.

The series truncation error, ϵ_2, may be problematic at first sight, for $\Delta t \to 0$. For small time intervals the transitional probability density function tends to become highly peaked. However, by letting the number of terms in the Fourier-cosine expansion increase, for $\Delta t \to 0$, the method can deal with such highly peaked functions, as long as they are in $\mathrm{C}^{\infty}[a,b]$. Moreover, the size of the integration range is, by means of the cumulants involved in (3.71), automatically adapted to the shape of the function.

Density recovery: influence of adapted truncation range

From the discussion above it is clear that the recovery of the probability density functions of the Lévy processes, for $\Delta t \to 0$, from their Fourier cosine series expansion, is crucial. Figure 3.4 shows the importance of the proper adaptive choice of the integration range, by means of the cumulants in (3.71). For several values of Δt, with even $\Delta t = 10^{-7}$, the density recovery with a fixed and an adapted integration range are compared, for GBM as well as for NIG. It is clear that the adaptive choice of integration range is superior for the recovery. Whereas, we see that for GBM the density can be recovered on the adapted integration range without significant difficulties, when $\Delta t \to 0$, it is less trivial for the NIG process. For this latter process, the recovery gets difficult for the smallest time interval, even with the interval adaptation.

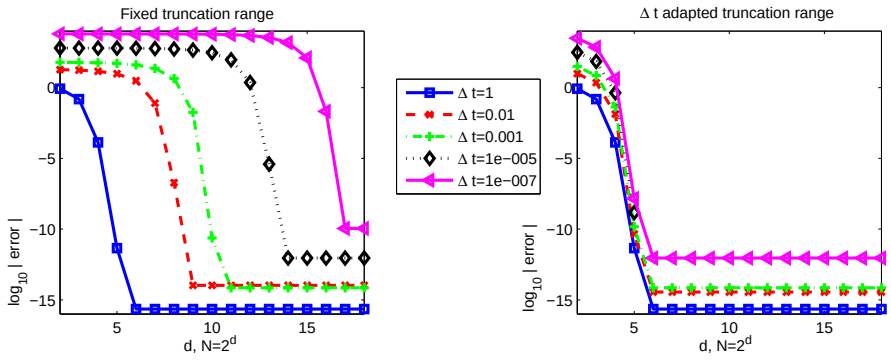
Stability of the method

Based on Theorem 3.3.1 and its proof in Section 3.3.2, we saw an exponential convergence in N. Inequality (3.70) indicates that the proportionality constant in the convergence estimate may grow with the number of exercise dates, M, so that with N fixed the error may increase substantially for increasing M.

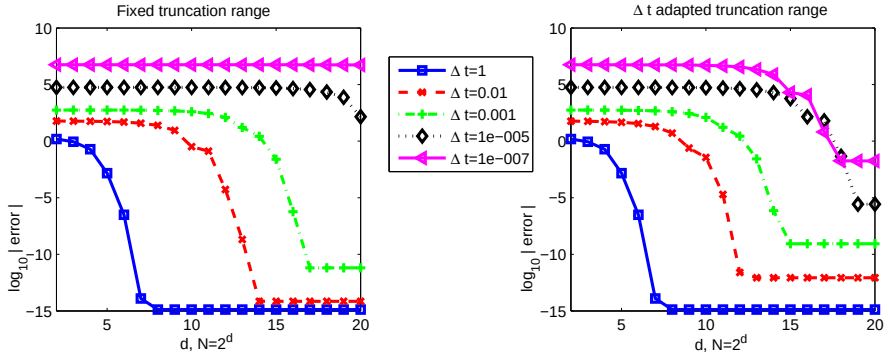
However, the error convergence is as fast as exponential in N, so that one can achieve very high accuracy by slightly increasing N.

We show here, by means of some numerical experiments that the resulting error in $\hat{\mathbf{V}}(t_1)$ is bounded. In Figure 3.5, we present the convergence for Bermudan call options under GBM with a varying number of exercise dates with respect to the number of terms in the cosine series, N. Shown is the logarithm of the error, $\log_{10}|\mathrm{err}|$, versus N. The dividend rate is set to zero, so that there are no early-exercise opportunities, i.e. $x_m^* = b$ for $m = 0, 1, \cdots, M-1$, and thus, the Bermudan call options have the same values as their European counterparts. The other parameters are as in Table 3.1. The truncation range is defined according to the description in Section 3.3.3.

We see a convergence of extreme accuracy, because the reference values are also obtained by the COS method for European options, and therefore the error related to the truncation range cannot be observed.

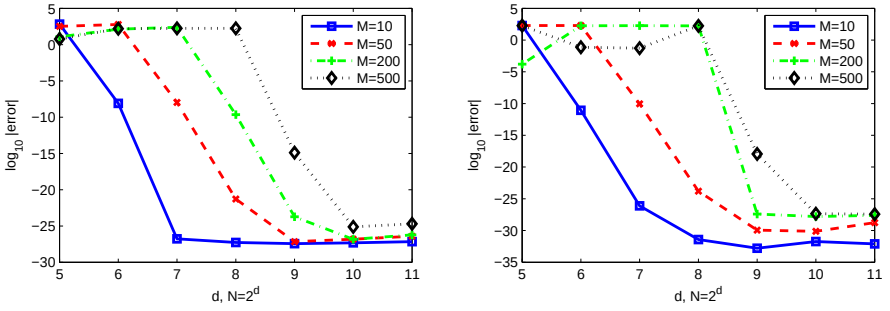


(a) GBM



(b) NIG

Figure 3.4: The recovered density functions for $\Delta t \to 0$; (a) the GBM model with fixed and adapted integration range, (b) the NIG model with fixed and adapted ranges.



(a) $T = 1$ year (b) $T = 0.1$ year

Figure 3.5: Error convergence for increasing M and N.

For two maturities, $T = 1$ year and short maturity $T = 0.1$ year, relatively large values of M do not show any significant impact on the error convergence with respect

to N. Merely, the start of convergence shifts to larger values of N as M increases, which confirms the intuition that higher values for N can compensate for the higher peakedness of the density function. With N small, however, the error presented remains bounded.

3.5 Conclusions and Discussion

In this chapter, we have generalized the COS option pricing method, based on Fourier-cosine expansions, to Bermudan and discretely-monitored barrier options. The method can be used whenever the characteristic function of the underlying price process is available (i.e., for regular affine diffusion processes and, in particular, for exponential Lévy processes).

The main insights are that the COS formula for European options from [32] can be used for pricing Bermudan and barrier options, if the series coefficients of the option values at the first early-exercise (or monitoring) date are known. These coefficients can be recursively recovered from those of the payoff function. The computational complexity is $O((M-1)N \log_2 N)$, for Bermudan and barrier options with M exercise, or monitoring, dates. The COS method exhibits an exponential convergence in N for density functions in $\mathrm{C}^\infty[a,b]$ and an impressive computational speed. With a small value of N, it typically produces highly accurate results. For example, with $N = 128$, results are accurate up to the 8th digit in less than 20 milli-seconds for 10-times exercisable Bermudan options and less than 10 milli-seconds for monthly-monitored barrier options.

However, the smaller the time interval between two consecutive dates, the more peaked the underlying density function, and thus larger values of N are required for a similar accuracy. For problems with small time intervals, like daily-monitored barrier options, the COS method shows a similar performance as the Hilbert transform based method [36].

Compared to the CONV method [54], which is one of the fast methods for Bermudan options, the COS method converges significantly faster to the same level of accuracy. Pricing American options can be done by a Richardson extrapolation method on Bermudan options with a varying number of exercise dates.

Chapter 4

Recovering Survival Probability and Pricing CDSs

4.1 Introduction

This chapter contains essentially the contents of the paper [31].

Credit default swaps (CDSs), the basic building block of the credit risk market, offer investors the opportunity to either buy or sell default protection on a reference entity. The protection buyer pays a premium periodically for the possibility to get compensation if there is a credit event on the reference entity until maturity or the default time, which ever is first. If there is a credit event the protection seller covers the losses by returning the par value. The premium payments are based on the CDS spread.

The spread of CDSs depends on the default probability of the underlying reference entity and it is possible to back out the market view of default probabilities for individual names from quoted market prices. It is therefore essential to be able to use advanced models in credit default modeling.

Lately Lévy models have attracted attention in the field of credit risk, see e.g. [17], [18] and [58]. In [18] CDSs are priced using the structural approach with a Variance Gamma model driving the firm value. To calculate the default probabilities they derive the partial integro-differential equation (PIDE) satisfied by the barrier option price and solve the equation by adapting a numerical scheme developed by [42] for pricing American options. If the driving Lévy process in the firm value model only has negative jumps, i.e., it is a single-sided or spectrally negative Lévy process, then the default probabilities can be found by numerically performing a double Fourier inversion as shown in [58]. To price path dependent options on assets driven by jump diffusions with exponentially distributed Poisson jumps with the use of fluctuation identities from the theory of Lévy processes has been worked out in [53] and [50, 51]. The resulting price formulas are of relatively simple explicit form when written as functions of the Laplace variable.

To price vanilla as well as exotic options on Credit Default Swaps can easily be set up in a structural model. In [47], the single-sided firm value models were used to generate dynamic CDS spreads by mapping the firm value paths to CDS spreads.

In this way a Monte Carlo engine to price options on CDSs could be set up to price (exotic) options on CDSs. Having a method to generate dynamic CDS spreads available, it is also possible to value so called Constant Maturity CDSs, where the spread is reset periodically to the market spread of a CDS with constant maturity tenor, as described in [46].

Here we take a structural approach towards the modeling of credit risk, following the same methodology as [11], which defines the credit event to be the first time the value of the reference entity is below a predefined lower barrier representing the total debt of the firm. In contrast to [11], which used a Geometric Brownian motion to drive the firm value, we set up a firm value model driven by an exponential Lévy process. In particular, we study the model developed by [19] (CGMY), and the Normal Inverse Gaussian (NIG) processes.

We will show that default probabilities can be efficiently recovered from the Fourier-cosine series expansion of the underlying density, following the path of the COS method for European options in [32] and that for Bermudan options and discretely monitored barrier options in [33]. We can price a single CDS within fractions of a second and several CDSs in less than half a second with a high accuracy. Switching from one underlying model to another is furthermore as easy as switching from one characteristic function to another. This enables us to calibrate the Lévy models to market quotes of CDS prices easily and efficiently.

The chapter is organized as follows. In Section 4.2 we present the mechanics and valuation of CDSs and introduce the Lévy firm value model. The COS method for survival probabilities is described in Section 4.3. In Section 4.4, the relevant values of the parameters in the COS method are determined and discussed in detail. A calibration study and some numerical examples are presented in Section 4.5. The paper ends with conclusions in Section 4.6.

4.2 Lévy Default Model and Valuation of CDSs

In this chapter we follow the approach taken by [11] and model the default event of a firm as the first time the firm value crosses a low barrier.

Let us denote the risk neutral value of a firm at time t as v_t, and assume that under an admissible pricing measure, $\mathbb{Q}$, it follows an exponential Lévy process, i.e.

$$v_t = v_0 \exp(X_t), \quad t \geq 0,$$

with X_t being a Lévy process, which has independent and stationary increments and is stochastically continuous. We will assume that $\mathbb{Q}$ is the mean-correcting Martingale measure, such that $X_t = \mu t + Y_t$, where $\mu = r - \log(\varphi_Y(-i))$, $i^2 = -1$, with $\varphi_Y(\cdot)$ being the characteristic function of Y_t and r being the risk-free interest rate (assumed to be constant).

Of our particular interest here are the cases when Y_t is either the CGMY or the NIG process.

In the calibration study, we add a diffusion part to the NIG model and we denote this extended model NIG-BM. By doing this both the CGMY and the NIG-BM model are then having four parameters. More details on the diffusion part are discussed later on. As a result, the dynamics of the NIG-BM model are driven by four parameters:

$[\sigma, \alpha, \beta, \delta]$, where σ is the volatility of the diffusion, and the characteristic function has been given in Table 1.1 of Section 1.2.1.

In what follows, we use φ_{levy} to denote the characteristic functions of Lévy processes.

4.2.1 Lévy Default Model

For a given recovery rate, R, default occurs the first time the firm's value is below the "reference value" RV_0. In particular, the time of default is defined as

$$\tau_{def} := \inf\{t \geq 0 : v_t \leq RV_0\}.$$

If we focus on

$$X_s = \ln(v_s/v_0),$$

then the risk-neutral survival probability in the time period $(0, t]$, $\mathrm{P}_{surv}(t) = \mathrm{P}_{\mathbb{Q}}(\tau_{def} > t)$, satisfies

$$\begin{aligned} \mathrm{P}_{surv}(t) &= \mathrm{P}_{\mathbb{Q}}\left(X_s > \ln R, \text{for all } 0 \leq s \leq t\right) \\ &= \mathrm{P}_{\mathbb{Q}}\left(\min_{0\leq s\leq t} X_s > \ln R\right) \\ &= \mathbb{E}_{\mathbb{Q}}\left[\mathbf{1}\left(\min_{0\leq s\leq t} X_s > \ln R\right)\right] \end{aligned} \tag{4.1}$$

where the indicator function $\mathbf{1}(A)$ equals 1 if the event A is true and zero otherwise, and the subindex $\mathbb{Q}$ refers to the fact that we are working in a risk-neutral setting (under the mean-correcting Martingale measure). Eq. (4.1) is nothing but the price of a *Binary Down-and-Out Barrier* (BDOB) option without discounting. This is a key observation that we will exploit in the remainder of this chapter.

Different methods can be applied to find the default probabilities. For single-sided Lévy models, where the firm value only has negative jumps, the default probabilities can be calculated using the Wiener-Hopf factorization and a double Fourier inversion as shown in [58]. In case of VG, the default probabilities can be calculated by solving a PIDE as described in [18].

In this chapter we use the COS method to compute the survival probabilities and thus the CDS spreads under Lévy models. It is called the COS method and is based on the Fourier cosine-series expansion of the underlying density ([32]).

4.2.2 Valuation of Credit Default Swaps

Given a time period, say, $(0, \tau]$, we assume that there are only a finite number of observing dates, $\mathcal{T} := \{\tau_0, \tau_1, \tau_2, \cdots, \tau_M\}$ with $\tau_m := m\Delta\tau$ $(m = 0, 1, \cdots, M)$ and $\Delta\tau := \tau/M$ (and therefore $\tau_0 = 0$ and $\tau = \tau_M$), on which the firm value is monitored, such that

$$\mathrm{P}_{surv}(\tau) = \mathbb{E}_{\mathbb{Q}}\left[\mathbf{1}\Big(X_{\tau_1} \in [\ln R, \infty)\Big) \cdot \mathbf{1}\Big(X_{\tau_2} \in [\ln R, \infty)\Big) \cdots \mathbf{1}\Big(X_{\tau_M} \in [\ln R, \infty)\Big)\right]. \tag{4.2}$$

This coincides with the pricing formula for discrete digital options without discounting.

Equation (4.2) can be re-written as a recursive formula as follows. The joint probability density of $X_{\tau_1}, X_{\tau_2}, \cdots, X_{\tau_M}$ follows

$$\begin{aligned}
& f_{X_{\tau_1},X_{\tau_2},\cdots,X_{\tau_M}}(x_{\tau_1}, x_{\tau_2}, \cdots, x_{\tau_M}) \\
= \; & f_{X_{\tau_M}|X_{\tau_1},X_{\tau_2},\cdots,X_{\tau_{M-1}}}(x_{\tau_M}|x_{\tau_1}, x_{\tau_2}, \cdots, x_{\tau_{M-1}}) \\
& \cdot f_{X_{\tau_{M-1}}|X_{\tau_1},X_{\tau_2},\cdots,X_{\tau_{M-2}}}(x_{\tau_{M-1}}|x_{\tau_1}, x_{\tau_2}, \cdots, x_{\tau_{M-2}}) \\
& \cdot \cdots \cdot f_{X_{\tau_1}|X_{\tau_0}}(x_{\tau_1}|x_{\tau_0}).
\end{aligned}$$

The Markov property simplifies the above to

$$\begin{aligned}
& f_{X_{\tau_1},X_{\tau_2},\cdots,X_{\tau_M}}(x_{\tau_1}, x_{\tau_2}, \cdots, x_{\tau_M}) \\
= \; & f_{X_{\tau_M}|X_{\tau_{M-1}}}(x_{\tau_M}|x_{\tau_{M-1}}) \cdot f_{X_{\tau_{M-1}}|X_{\tau_{M-2}}}(x_{\tau_{M-1}}|x_{\tau_{M-2}}) \\
& \cdot \cdots \cdot f_{X_{\tau_1}|X_{\tau_0}}(x_{\tau_1}|x_{\tau_0}) \\
= \; & \Pi_{m=1}^{M} f_{X_{\tau_m}|X_{\tau_{m-1}}}(x_{\tau_m}|x_{\tau_{m-1}}),
\end{aligned}$$

where $f_{X_{\tau_m}|X_{\tau_{m-1}}}(\cdot|\cdot)$ denotes the conditional probability density of X_{τ_m} given $X_{\tau_{m-1}}$. Inserting it into (4.2), one yields

$$\begin{aligned}
\mathrm{P}_{surv}(\tau) &= \int_{\mathbb{R}^M} \Pi_{m=1}^{M} \mathbf{1}\Big(x_{\tau_m} \in [\ln R, \infty)\Big) f_{X_{\tau_m}|X_{\tau_{m-1}}}(x_{\tau_m}|x_{\tau_{m-1}}) d\mathbb{R}^M \\
&= \int_{\ln R}^{+\infty} \cdots \int_{\ln R}^{+\infty} \cdots \int_{\ln R}^{+\infty} f_{X_{\tau_M}|X_{\tau_{M-1}}}(x_{\tau_M}|x_{\tau_{M-1}}) dx_{\tau_M} \cdots \\
&\qquad \cdot f_{X_{\tau_m}|X_{\tau_{m-1}}}(x_{\tau_m}|x_{\tau_{m-1}}) dx_{\tau_m} \cdots \\
&\quad \cdot f_{X_{\tau_1}|X_{\tau_0}}(x_{\tau_1}|x_{\tau_0}) dx_{\tau_1}.
\end{aligned}$$

Define

$$p(x, \tau_M) := 1.$$

We then have the following recursive relation:

$$\begin{cases} p(x, \tau_m) & := \int_{\ln R}^{\infty} f_{X_{\tau_{m+1}}|X_{\tau_m}}(y|x) p(y, \tau_{m+1}) \, dy, \quad m = M-1, \cdots, 2, 1, 0, \\ \mathrm{P}_{surv}(\tau) & := p(x_{\tau_0}, \tau_0), \end{cases} \tag{4.3}$$

where $x_{\tau_0} = \ln(v_0/v_0) = 0$ and $\tau_0 = 0$. p can be interpreted as value of the virtual digital option without discounting.

This is the starting point of the numerical method derived in the next section.

Let us denote by T the maturity of a CDS. The *fair spread*, C, of a CDS at the initialization date is the spread that equalizes the present value of the premium leg and the present value of the protection leg, i.e.

$$C = \frac{(1-R)\left(\int_0^T \exp(-r(s)s) \mathrm{dP}_{def}(s)\right)}{\int_0^T \exp(-r(s)s) \mathrm{P}_{surv}(s) \mathrm{d}s}, \tag{4.4}$$

where $r(t)$ is the risk-free discount rate over the time period $(0, t]$, and $\mathrm{P}_{def}(t)$ and $\mathrm{P}_{surv}(t)$ are the *probability of default* and the *probability of survival*, respectively, in the time period $(0, t]$. Note that in case of a default event the protection buyer is

receiving $(1 - R)$ for every insured currency unit, R being the recovery. Eq. (4.4) indicates that the price of a CDS depends on the survival probability (and of course on the default probability) of the firm.

It is in principle possible to model a stochastic recovery rate, but here we assume it to be constant.

When the interest rate is assumed to be constant in $[0, T]$, (4.4) can be simplified, via integration-by-parts, to

$$C = (1 - R)\left(\frac{1 - e^{-rT}\mathrm{P}_{surv}(T)}{\int_0^T e^{-rs}\mathrm{P}_{surv}(s)\mathrm{d}s} - r\right), \tag{4.5}$$

where $\mathrm{P}_{surv}(t) = 1 - \mathrm{P}_{def}(t)$ is used, see also [18, 58].

The price of a CDS is based on a series of survival probabilities on different time intervals. To see this, we discretize (4.5) using the composite trapezoidal rule, i.e.

$$C_{trap} := (1 - R)\left(\frac{1 - e^{-rT}\mathrm{P}_{surv}(T)}{\sum_{j=0}^{J} w_j e^{-rt_j}\mathrm{P}_{surv}(t_j)\Delta t} - r\right) = C + \epsilon_c, \tag{4.6}$$

where ϵ_c denotes the discretization error of the numerical integration rule (discussed later), and $w_j = \frac{1}{2}$ for $j = 0$ and $j = J$ and $w_j = 1$ otherwise. Eq. (4.6) suggests that a CDS price depends on a sequence of survival probabilities defined on the sequence of time intervals $(0, t_0], (0, t_1], \ldots, (0, t_J]$, with $t_j := j\Delta t$ and $\Delta t := T/J$. We will subsequently show that these survival probabilities can be approximated simultaneously in (almost) linear complexity.

4.3 Recovering Survival Probability with the COS Method

Let us start from the assumption that the underlying probability density $f_{X_t|X_s}(y|x)$ with $0 \leq s \leq t$ satisfies

$$\int_{\mathbb{R}\setminus[a,b]} f_{X_t|X_s}(y|x)dy < \mathrm{TOL},$$

where TOL is some predefined tolerance for numerical errors, e.g. 10^{-14}, and the interval $[a, b] \in \mathbb{R}$ is called the truncation range. The idea then sets in that any smooth function (such Lévy densities in many cases) defined on a finite interval can be accurately recovered from its Fourier-cosine series.

Recall that the Fourier-cosine series coefficients of the density $f_{X_t|X_s}(y|x)$ of a Lévy process are related to the characteristic function as follows:

$$f_{X_t|X_s}(y|x) = \frac{2}{b-a}{\sum}'_{n=0}^{N-1} \mathrm{Re}\left\{\varphi_{levy}\left(\frac{n\pi}{b-a}, t-s\right)e^{in\pi\frac{x-a}{b-a}}\right\}\cos\left(n\pi\frac{y-a}{b-a}\right) + \epsilon_f, \tag{4.7}$$

for $x, y \in [a, b] \subset \mathbb{R}$, and where $\sum'$, again, denotes that the first term in the summation is halved.

Efficiency of a numerical algorithm is best measured by the error convergence rate. Fast error convergence guarantees that one can always achieve satisfactory accuracy by slightly increasing the number of computations.

Detailed error analysis, along with various experiments, has been given in [32]. It shows that error ϵ_f in (4.7) consists of an integration-range related truncation error and a series truncation error. When the truncation range, $[a, b]$, is sufficiently large, the series truncation error dominates. Since a Fourier series exhibits an exponential convergence for smooth functions (in $\mathbb{C}^\infty[a, b]$ with nonzero derivatives), ϵ_f converges exponentially for processes governed by smooth probability densities (or, equivalently, for processes whose characteristic functions have rapidly decaying tails).

As for the truncation interval $[a, b]$, we use the same definition as given in (3.3) of Chapter 3, i.e.,

$$[a, b] := \left[\xi_1 - L\sqrt{\xi_2 + \sqrt{\xi_4}}, \quad \xi_1 + L\sqrt{\xi_2 + \sqrt{\xi_4}}\right], \tag{4.8}$$

where ξ_j is the jth cumulant of X_t. This range is found to be sufficiently large to make the truncation error of order 10^{-14} as was illustrated in Chapter 3. As such, with the above truncation interval and for a small N (e.g. less than 2^{10}), Eq. (4.7) produces a highly accurate approximation (with an error of order $O(10^{-7})$) for probability densities that are in $\mathbb{C}^\infty[a, b]$ and with nonzero derivatives.

As will be shown in the following sections, a density of an asset value defined on a very short time interval, such as one week (1/48 year), is of our main interest here. In this case, some stochastic processes yield static distributions with sharp peaks in the probability density functions. A typical example is the VG model. In [32], it has been shown that for a peaked VG density, the convergence of Fourier-cosine series (and thus the COS method) is no longer exponentially but geometrically, i.e. the error is of order $O(N^{-d})$ with $d \in \mathbb{R}^+$. For other models like CGMY and NIG-BM, it is shown in Section 3.4.3 that the COS method still displays a fast convergence, even when the time interval, Δt, decreases to 10^{-7}. To get a first impression of the performance of the COS method for peaked densities, in Figure 4.1 we compare the recovered VG density of one week with the analytic solution. The graphs indicate that recovered densities match very well with the closed-form expressions. Note that the error can be further reduced with larger values of N, which is due to the regular convergence of Fourier-cosine series for continuous functions. Here N is set to 2^{14}.

4.3.1 The COS Formula of Survival Probability

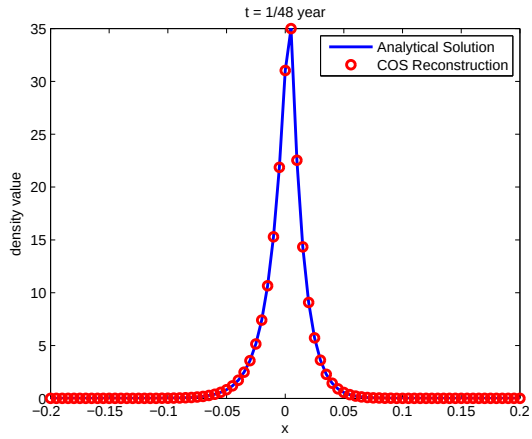
With the same rule for the range of integration as in (4.8), Equation (4.3) can be rewritten as

$$p(x, \tau_m) = \int_{\ln R}^{b} f_{X_{\tau_{m+1}}|X_{\tau_m}}(y|x)p(y, \tau_{m+1})dy + \epsilon_p, \tag{4.9}$$

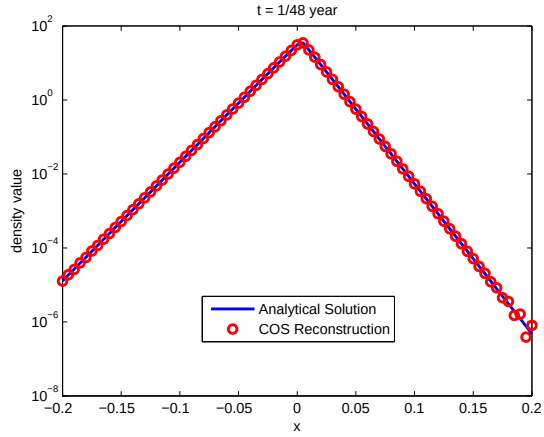
where error ϵ_p, due to the size of the integration interval, is negligible (as $p(y, \tau_{m+1}) > 0$ and $\int_{\mathbb{R}} p(y, \tau_{m+1})dy = 1$, error ϵ_p is of the same order as $\int_{x\in\mathbb{R}\setminus[a,b]} f_{X_{\tau_{m+1}}|X_{\tau_m}}(y|x)dy$).

As a second step, we replace the conditional density in (4.9) by (4.7), so that

$$p(x, \tau_m) = {\sum}'_{n=0}^{N-1} \phi_n(x) \cdot P_n(\tau_{m+1}) + \epsilon_{cos}, \tag{4.10}$$



(a) Normal scale



(b) Log scale

Figure 4.1: Recovered VG density vs. closed-form solution; $t = 1/48$, $r = 0.04$, $\sigma = 0.12$, $\theta = -0.14$ and $\nu = 0.02$

where ϵ_p is included in ϵ_{cos}. For $m = 0, 1, \ldots, M-1$, we have:

$$P_n(\tau_{m+1}) := \frac{2}{b-a} \int_{\ln R}^{b} \cos\left(n\pi \frac{y-a}{b-a}\right) p(y, \tau_{m+1})\, dy, \tag{4.11}$$

and

$$\phi_n(x) := \mathrm{Re}\left\{\varphi_{levy}\left(\frac{n\pi}{b-a}, \Delta\tau\right) e^{in\pi\frac{x-a}{b-a}}\right\}. \tag{4.12}$$

Eq. (4.10) is in essence the COS formula for discrete barrier options (without discounting). By ϵ_{cos} we denote the error in the COS formula, in which ϵ_f in (4.7) is the main error contribution.

Finally, the COS formula for the survival probability reads

$$\begin{cases} \mathrm{P}_{surv}(\tau) &= p(x=0,\tau_0). \\ p(x,\tau_0) &= \sum\nolimits'_{n=0}^{N-1} \phi_n(x)\cdot P_n(\tau_1). \end{cases} \tag{4.13}$$

Equation (4.13) suggests that, to get the survival probability, one needs $\{P_n(\tau_1)\}_{n=0}^{N-1}$, the cosine coefficients of $p(x,\tau_1)$, that only depend on $\{P_n(\tau_2)\}_{n=0}^{N-1}$ in Equations (4.10) and (4.11), and so forth.

In what follows, we will demonstrate that $\{P_n(\tau_m)\}_{n=0}^{N-1}$ can be recovered from $\{P_n(\tau_{m+1})\}_{n=0}^{N-1}$ in almost linear computational complexity, and that $\{P_n(\tau_1)\}_{n=0}^{N-1}$ can therefore be recursively recovered from $\{P_n(\tau_M)\}_{n=0}^{N-1}$, the cosine coefficients of $p(x,\tau_M)$.

4.3.2 Backward Induction

Starting from the definition of $P_n(\tau_m)$ in (4.11), we replace $p(y,\tau_m)$ by (4.10) and insert (4.12) to obtain

$$P_n(\tau_m) = \sum\nolimits'_{k=0}^{N-1} \mathrm{Re}\left\{\varphi_{levy}\left(\frac{k\pi}{b-a},\Delta\tau\right)\cdot\omega_{n,k}\right\} P_k(\tau_{m+1}), \tag{4.14}$$

where $n=0,1,\cdots,N-1$, and

$$\omega_{n,k} := \frac{2}{b-a}\int_{\ln R}^{b} e^{ik\pi\frac{y-a}{b-a}}\cos\left(n\pi\frac{y-a}{b-a}\right)dy.$$

In matrix-vector-product form, (4.14) becomes

$$\mathbf{P}(\tau_m) = \mathrm{Re}\left\{\Omega\ \Lambda\right\}\ \mathbf{P}(\tau_{m+1}), \tag{4.15}$$

where we use bold-faced letters to denote vectors, e.g. $\mathbf{P}(\tau_m)$ is the vector $(P_0(\tau_m),P_1(\tau_m),\ldots,P_{N-1}(\tau_m))^T$. "$\Omega\ \Lambda$" denotes a matrix-matrix multiplication with Ω being the matrix with elements $\{\omega_{n,k}\}_{n,k=0}^{N-1}$ and Λ a diagonal matrix filled by $\left\{\varphi_{levy}\left(\frac{k\pi}{b-a},\Delta\tau\right)\right\}_{k=0}^{N-1}$. By applying (4.15) recursively, i.e. backwards in time, we obtain the induction formula for $\mathbf{P}(\tau_1)$:

$$\mathbf{P}(\tau_1) = \left(\mathrm{Re}\left\{\Omega\ \Lambda\right\}\right)^{M-1}\mathbf{P}(\tau_M) \tag{4.16}$$

with $\mathbf{P}(\tau_M)$ admitting an analytic solution since, for $n=0,1,\cdots,N-1$,

$$P_n(\tau_M) := \frac{2}{b-a}\int_{\ln R}^{b}\cos\left(n\pi\frac{y-a}{b-a}\right)dy.$$

Straight-forward computation of (4.16) is time-consuming. However, fortunately an efficient valuation technique exists.

From their definition, we know that the Fourier-cosine coefficients of real-valued functions are also real-valued, so that we can expand (4.16) into a recursive matrix-vector-product. For example, if there are 3 monitoring dates, we need to compute

$$\mathbf{P}(\tau_1) = \mathrm{Re}\left\{\Omega\ \left[\Lambda\quad \mathrm{Re}\left\{\Omega\ \left[\Lambda\quad \mathrm{Re}\left\{\Omega\ \left[\Lambda\ \mathbf{P}(\tau_3)\right]\right\}\right]\right\}\right].$$

Note that "Λ times vector $\mathbf{P}(\tau_3)$" can be transformed into an element-wise multiplication of two vectors as Λ is a diagonal matrix. Similarly, for M monitoring dates, we have

$$\mathbf{P}(\tau_1) = \mathrm{Re}\left\{\Omega\ \left[\Lambda \quad \cdots \quad \mathrm{Re}\left\{\Omega\ \left[\Lambda \quad \mathrm{Re}\left\{\Omega\ \left[\Lambda\ \mathbf{P}(\tau_M)\right]\right\}\right]\right\}\right]\right\}. \tag{4.17}$$

Furthermore, matrix Ω has a special structure, see [33]:

$$\Omega = H + T,$$

where

$$H = \begin{bmatrix} w_0 & w_1 & w_2 & \cdots & w_{N-1} \\ w_1 & w_2 & \cdots & \cdots & w_N \\ \vdots & & & & \vdots \\ w_{N-2} & w_{N-1} & \cdots & & w_{2N-3} \\ w_{N-1} & \cdots & & w_{2N-3} & w_{2N-2} \end{bmatrix}_{N\times N}, \tag{4.18}$$

is a Hankel matrix, and T is a Toeplitz matrix:

$$T = \begin{bmatrix} w_0 & w_1 & \cdots & w_{N-2} & w_{N-1} \\ w_{-1} & w_0 & w_1 & \cdots & w_{N-2} \\ \vdots & & \ddots & & \vdots \\ w_{2-N} & \cdots & w_{-1} & w_0 & w_1 \\ w_{1-N} & w_{2-N} & \cdots & w_{-1} & w_0 \end{bmatrix}_{N\times N}, \tag{4.19}$$

with

$$w_j := \begin{cases} \dfrac{(b-\ln R)}{b-a} & j = 0, \\ -\dfrac{i}{\pi}\cdot\dfrac{\exp(ij\pi) - \exp\left(ij\dfrac{(\ln R - a)\pi}{b-a}\right)}{j} & j \neq 0. \end{cases} \tag{4.20}$$

It is well-known that matrix-vector products with the matrix being either a Hankel or a Toeplitz matrix can be transformed into a *circular convolution* of two vectors. Therefore, the FFT algorithm can be applied, and thus the recursive matrix-vector products in (4.17) can be computed in $O((M-1)N\log_2(N))$ operations.

Remark 4.3.1. *Computational effort can be saved further if we compute several survival probabilities simultaneously, in one computation. For example, suppose that we have two time intervals $(0,t_1] \subset (0,t_2]$. We can then define the time partitioning on $(0,t_2]$ in such a way that t_1 is exactly on the grid, e.g., $t_1 = \lambda t_2/M$ with λ being a positive integer less than M. We now find that*

$$\mathbf{P}(\tau_1;t_1) \equiv \mathbf{P}(\tau_{M-\lambda+1};t_2).$$

Thus, the Fourier-cosine coefficients of the survival probability on $(0,t_1]$ can be recovered at no cost during the computation of $\mathrm{P}_{surv}(t_2)$.

Remark 4.3.2. *As shown in (4.14), to apply this method for more general processes, the underlying density $f_{X_{\tau_{m+1}}|X_{\tau_m}}(x_{\tau_{m+1}}|x_{\tau_m})$ should have a characteristic function which satisfies $\phi(\omega|x_{\tau_m}) = \phi(\omega|0)e^{i\omega x_{\tau_m}}$, i.e. x_{τ_m} should be easily separated from the kernel $\Omega\Lambda$, so that the FFT algorithm can be employed for fast computation. This is the case for the Lévy processes considered in this article.*

4.4 Choice of Parameters and Error Analysis

To use (4.13) for CDS spreads, we need to determine three relevant parameters: N, the number of terms of the cosine series expansion; M, the number of monitoring dates and J, the number of quadrature points to discretize the integrals in (4.4). These parameter values are based on the following insights.

4.4.1 Local Error Convergence

The choice of N is comparatively simple, as N is only related to the error convergence of the COS reconstruction of the underlying probability density, which is directly related to the convergence rate of its Fourier-cosine series expansion. A detailed error analysis of ϵ_{cos} has been given in Section 2.4 already and analysis of the error propagation in the recursive induction in Section 3.3.

For example, as Figure 4.2 suggests, the convergence speed of the absolute errors is exponential, and with $N = 2^{10}$ the errors are substantially smaller than 1 basis point.

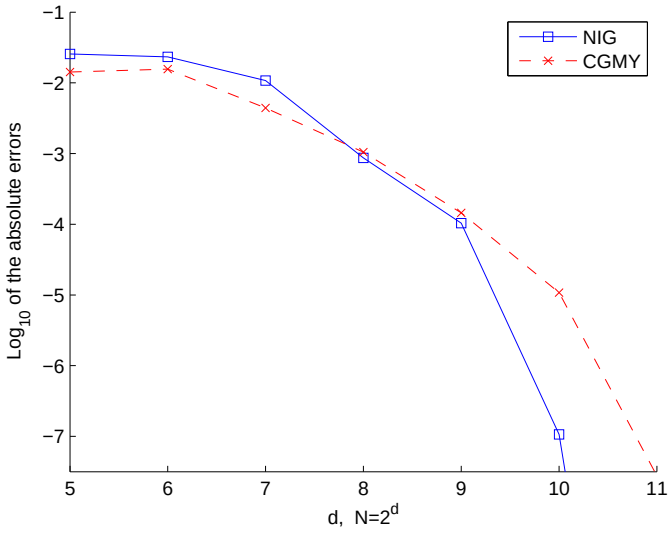


Figure 4.2: Convergence of $\mathrm{P}_{surv}(\Delta\tau = 1/48)$ w.r.t. the number of terms in the cosine series expansion (N) for NIG-BM and CGMY; Parameters are given in Table 4.1.

4.4.2 Number of Monitoring Dates and Integration Points

From a practical point of view, all CDSs should be monitored daily, i.e. the mesh size in time, $\Delta\tau$, has to be $1/252$ years if there are 252 working days per year. This is equivalent to setting $M = 2520$ for a CDS which matures in 10 years. Such values for M make the overall computation expensive for calibration purposes.

However, one can employ larger values of $\Delta\tau$ at the cost of some accuracy. In [49] the convergence of the price of discrete barrier options with m monitoring dates to the price of the equivalent continuous barrier option was discussed within the Black-Scholes model. A proof of convergence under Lévy processes is not available, however, via various numerical experiments under these processes, we have observed a regular convergence pattern in the prices of discrete barrier options w.r.t. the number of monitoring dates. We observed that the survival probabilities computed with a coarser time step converge to those with $\Delta\tau = 1/252$.

Experiments give some evidence that this convergence is found for extreme parameter settings as well. An example can be found in the Series 8 iTraxx quotes for any component company (during the credit crunch of early 2008). In Table 4.1 the calibrated parameters for "ABN AMRO" on February 20, 2008 under NIG-BM and CGMY are given. The corresponding market quotes for "ABN AMRO" on February 20, 2008 are given in Table 4.2.

Table 4.1: Calibrated parameters for "ABN AMRO Bank" on February 20, 2008

Model	R	T	r	σ	Other Parameters
NIG-BM	0.4	1	0.04	0.206	$\alpha = 3.043, \quad \beta = -2.38, \quad \delta = 0.044$
CGMY	0.4	1	0.04	0	$C = 0.038, \quad G = 0.60, \quad M = 11.10, \quad Y = 1.32$

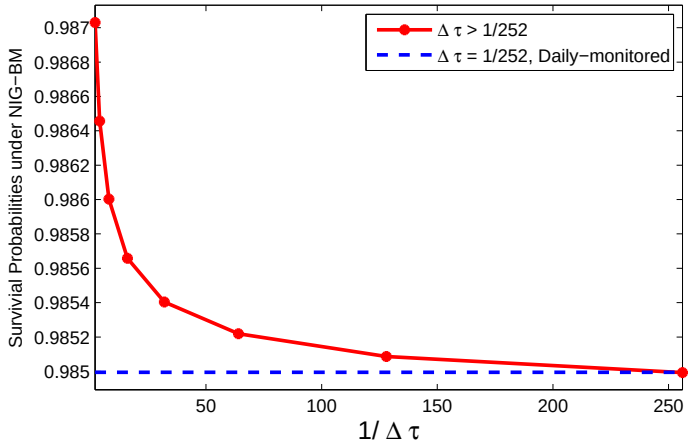
Table 4.2: CDS spreads for "ABN AMRO Bank" on February 20, 2008

Maturity (years)	1	3	5	7	10
CDS spread (bps)	88.4	109.1	126.2	128.8	121.0

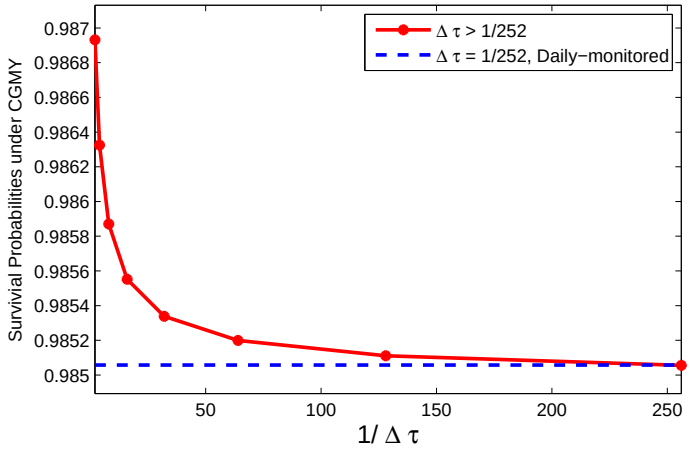
The convergence of the survival probabilities for the example in Table 4.1 is displayed in Figure 4.3. One can see that the difference between the weekly-monitored survival probabilities ($\Delta\tau = 1/48$) and the daily-monitored versions ($\Delta\tau = 1/252$) is at most 2 basis points. This difference is smaller when the parameter values are less extreme. Therefore, we use $\Delta\tau = 1/48$, or $M = 48T$, in the calibration to follow.

To get an idea on how fast P_{surv} converges w.r.t. M, we need to examine the structure of the matrices Ω and Λ in (4.17). According to its definition, Ω is a constant matrix that does not depend on the underlying model, nor on its parameters, whereas the diagonal matrix Λ does. The elements on the diagonal of Λ are defined as $\varphi_{levy}(k\pi/(b-a), \Delta\tau)$, so that the convergence rate of P_{surv} w.r.t. M solely depends on how fast the characteristic function decays.

For the number of points used in the trapezoidal rule (J) in (4.6), we find that the computed CDS spreads are not very sensitive to the size of J. Therefore, we use, in the calibration to follow, $J = M/4$, which gives only small differences (less than 0.1 basis point as shown in Figures 4.4 and 4.5) to the results computed with $J = M$ integration points. The second order convergence of the trapezoidal rule is confirmed in the righthand side pictures of the Figures 4.4 and 4.5.



(a) NIG-BM



(b) CGMY

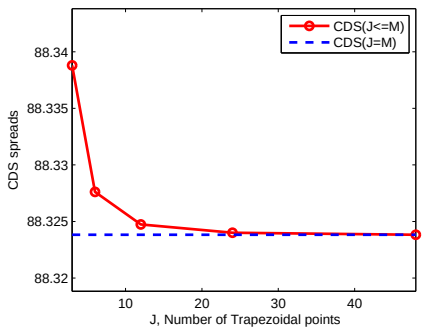
Figure 4.3: Convergence of the 1-year survival probability w.r.t. $\Delta\tau$ with parameters given in Table 4.1.

4.5 Calibration

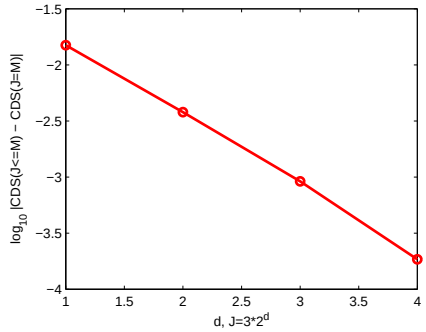
We investigate the performance of the proposed numerical scheme to calculate the CDS spreads by calibrating the NIG-BM and the CGMY models to a set of CDS prices.

4.5.1 Calibration Setting

The data sets are the weekly quotes from iTraxx Series 7 (S7) and 8 (S8). We have chosen to calibrate the models to spreads for CDSs with maturity 1, 3, 5, 7, and 10

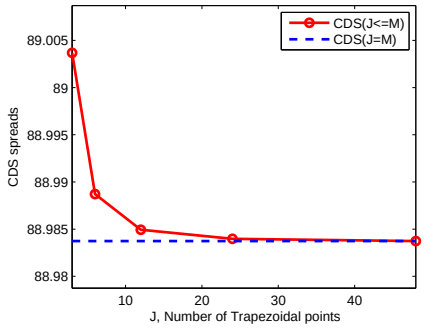


(a) Convergence of CDS spreads w.r.t. J

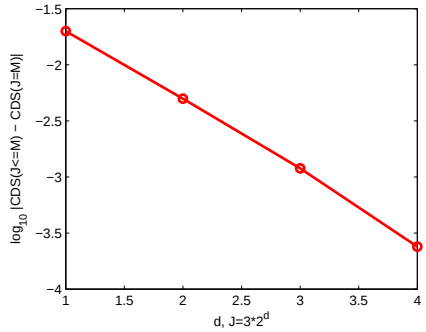


(b) Second order convergence of the trapezoidal rule

Figure 4.4: Convergence of the CDS spreads w.r.t. the number of points used in the trapezoidal rule (J) under NIG, with parameters given in Table 4.1.



(a) Convergence of CDS spreads w.r.t. J



(b) Second order convergence of the trapezoidal rule

Figure 4.5: Convergence of the CDS spreads w.r.t. the number of points used in the trapezoidal rule (J) under CGMY, with parameters given in Table 4.1.

years. This study makes use of 106 firms that are common to both series. For the chosen trading dates, no market quotes are missing for any firm, i.e., we have quotes for all maturities for all firms on each of the trading dates.

As the risk-free discount rate we have used the averaged EURIBOR swap rates.

We deal with the well-known ill-posedness of the inverse problem in the calibration framework (see, for example, [24]) by defining the objective function to be the root mean square error (RMSE) plus a regularization term, i.e.

$$F_{obj} = \text{RMSE} + \gamma \cdot ||\mathbf{X}_2 - \mathbf{X}_1||_2,$$

where

$$\text{RMSE} = \sqrt{\sum_{\text{CDS}} \frac{(\text{market CDS spread} - \text{model CDS spread})^2}{\text{number of CDSs on each day}}},$$

$||\cdot||_2$ denotes the L_2–norm, and $\mathbf{X}_2$ and $\mathbf{X}_1$ are the parameter vectors of two data sets. This kind of objective function also gives parameter consistency over time.

By the weighting factor γ, the regularization term influences the difference between two measures on two consecutive dates. This parameter can also be defined as a vector of the same length as $\mathbf{X}_2$ and $\mathbf{X}_1$, if the sensitivities of the CDS values to the component parameters differ significantly in magnitude. In that case, "$\cdot$" denotes the inner product of two vectors. With the objective function above, we aim to define a satisfactory measure, which fits the market data well and is – more or less – time-invariant.

Note that the choice of γ in the objective function has a significant impact on the quality of the calibration fit. If the weighting on the regularization term is too high, the RMSE increases; If the weighting is too small, the parameter values are not stable over time. We use $\gamma = [2, 0.5, 0.5, 0.5]$ for the NIG-BM model, corresponding to the set $[\sigma, \alpha, \beta, \delta]$. More weight is then assigned to σ because the initial calibration revealed that the CDSs are more sensitive to σ than to the other three parameters. As for the CGMY models, we employ $\gamma = [2, 0.5, 0.5, 2]$ corresponding to the set $[C, G, M, Y]$. More weighting is assigned to C and Y because with these two one has a significant influence on the shape of the densities, compared to G and M.

4.5.2 Calibration Results

Our first observation is that both the NIG-BM and the CGMY models give rise to a very good fit to the market data. A summary of the RMSE results for all the 106 companies that are present in both S7 and S8 of iTraxx is presented in Table 4.3.

Table 4.3: Summary of calibration results (in basis points) of all 106 firms in iTraxx

RMSE	NIG-BM in S7	CGMY in S7	NIG-BM in S8	CGMY in S8
Average (bp.)	0.89	0.79	1.65	1.54
Min. (bp.)	0.22	0.29	0.27	0.46
Max. (bp.)	2.29	1.97	4.27	3.52

From this we can see that the average RMSE of both Lévy models for the Series 7 data are less than 1.0 basis points, and for the Series 8 they are less than 2.0 basis points. Detailed information about the RMSE for each individual company is summarized in the tables in Appendix A.

Because the S7 and S8 CDS spreads data starts from March 2007 and ends in March 2008, i.e., including part of the credit crunch period, the CDS spreads all increase in time. Furthermore, the strong fluctuations in the CDS curves are an indication for the increasing volatility in the credit market.

A typical example from our calibration results is given in Figure 4.6, where we plotted the NIG-BM and CGMY results for ABN AMRO CDS spreads. Note that nearly 80% of all the companies have a very similar CDS evolution. The highly satisfactory match of the computed CDSs to the market CDSs of both Lévy models can be seen.

Even in the extreme case, where the CDS spreads have very high values, the data are still fitted very well, see for example Figure 4.7, where the NIG-BM and the

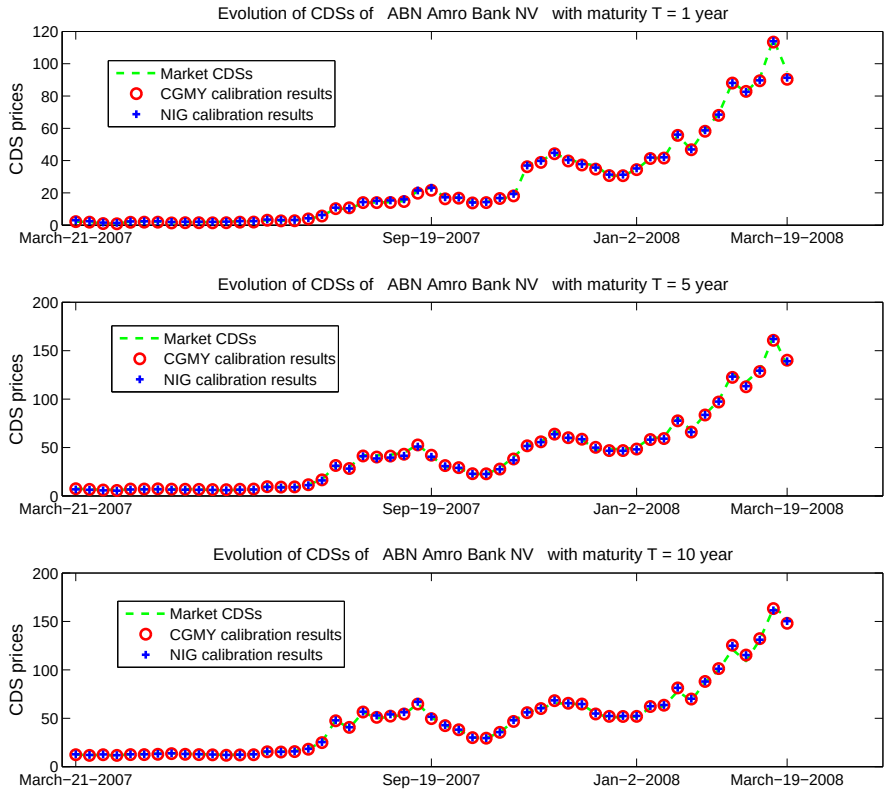


Figure 4.6: Calibration fit to ABN AMRO Bank CDS spreads for the CGMY and the NIG-BM models. Market CDS spreads '- -', CGMY CDS spreads 'o', and NIG-BM CDS spreads '+'.

CGMY model fit the DSG International PLC CDS spreads, even though the CDSs went to nearly 500 basis points.

In Figure 4.8, the evolution of the parameters of the CGMY and NIG-BM densities for ABN AMRO are plotted. We note that the parameters are in reasonable range and that they evolve quite "smoothly" over time. The jump in the parameters around September 19, 2007, reflects the jump in the CDS spreads in this period (cf. Figure 4.6). Recall from Table 4.3 that the average RMSE results for both models are small. It is also worth noting that the value of the σ parameter in the NIG-BM model is in the range of 0.1 to 0.2, indicating that the Brownian Motion contributes to the overall behavior of the model. In Figure 4.9, the evolution of the NIG-BM density is given. Here we can see that the density is more peaked in the beginning of Series 7 and then flattens out in the end of Series 8.

The CPU times for computing 1-, 3-, 5-, 7- and 10-year CDSs are summarized in Table 4.4. The calibration routine was implemented in Matlab and the computer is

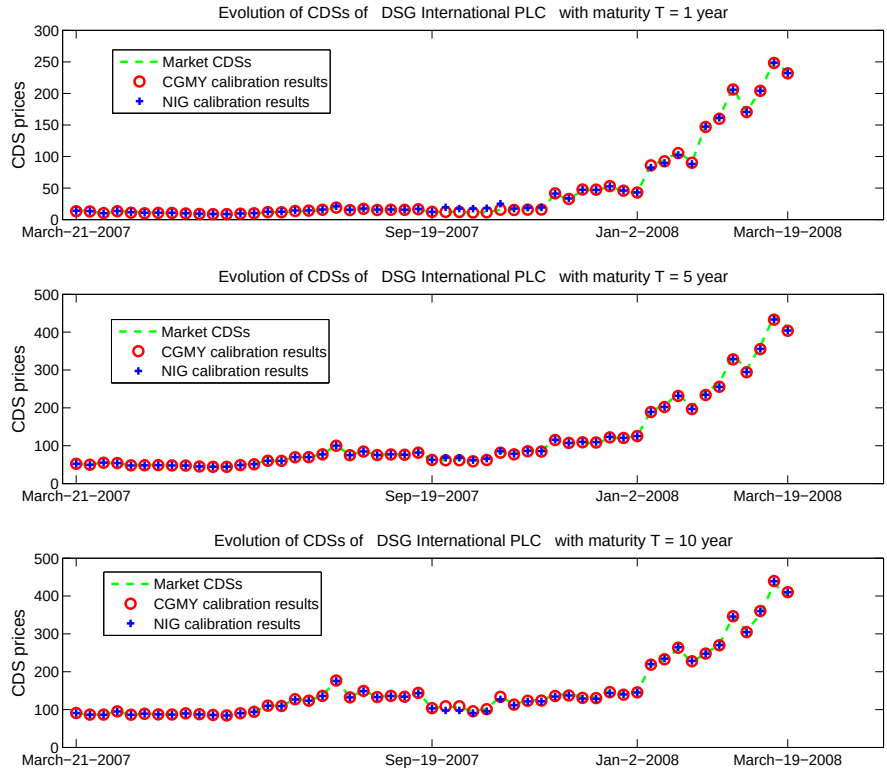


Figure 4.7: Calibration fit to DSG International PLC CDS spreads for the CGMY and the NIG-BM models. Market CDS spreads '- -', CGMY CDS spreads 'o', and NIG-BM CDS spreads '+'.

the same one as used for Chapter 1. In less than 0.5 seconds, the 5 CDSs for one company are computed, independent of the specific type of the underlying process. Although we use $N = 2^{11}$ in the calibration, a value of $N = 2^{10}$ is usually sufficient for the NIG-BM model, see for example Table 4.4 (where "max. abs. err. in bp." denotes "maximum absolute error in basis point").

It is worth mentioning that the calibrated Y values in CGMY often approach their upper limit, $Y = 2$. Whereas this gives rise to significant convergence difficulties for various numerical methods, it is not a problem for the COS method. In fact, since larger values of Y decrease the densities' peakedness, the COS method converges slightly faster, compared to lower values of Y, as explained in [32].

From a numerical point of view, we would like to point out that for smooth density functions we need fewer terms in the Fourier-cosine expansion than for highly-peaked functions. As a result, the number of the cosine series terms (N) can be kept relatively small.

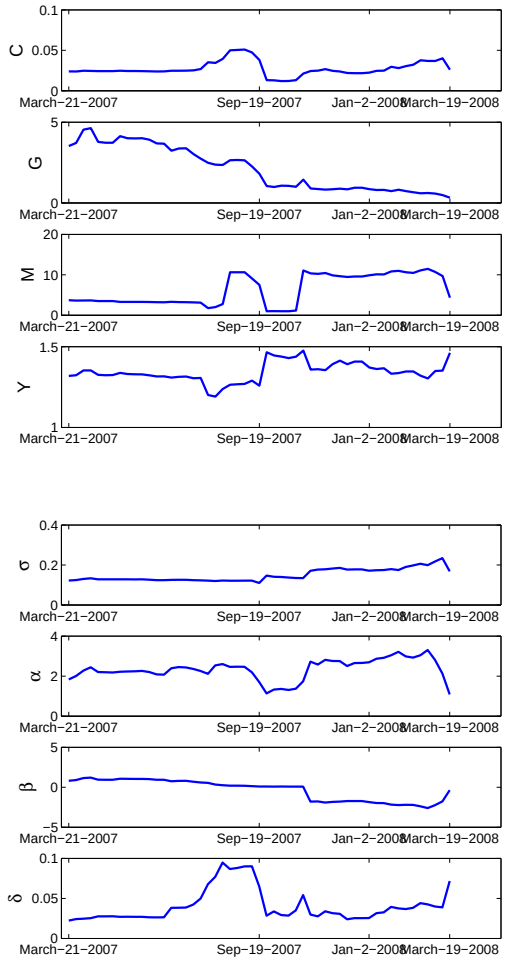


Figure 4.8: Evolution of the parameters of the CGMY (up) and NIG-BM (down) densities, respectively, for ABN AMRO.

Table 4.4: CPU times in computing 1-, 3-, 5-, 7- and 10-year CDSs with the COS method; Parameters are given in Table 4.1 and reference values are obtained by $N = 2^{13}$.

	N	2^9	2^{10}	2^{11}
NIG-BM	CPU times (sec.)	0.121	0.218	0.418
	max. abs. err. in bp.	0.28	7.93e-03	7.32e-06
CGMY	CPU times (sec.)	0.122	0.220	0.423
	max. abs. err. in bp.	6.89	1.07	2.94e-02

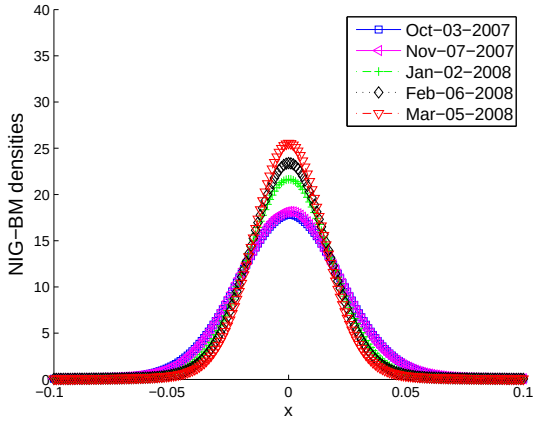


Figure 4.9: Evolution of the weekly NIG-BM density for ABN AMRO.

4.5.3 Default Probability Term Structure

It is interesting to see what the default probability term structure generated by the models look like. In Figure 4.10, we give an example of the term structure for DSG International PLC under the NIG-BM model. As expected the evolution of the default probability term structure resembles closely the evolution of the CDS spreads in Figure 4.7, that is, the CDS spreads increase over time, which is reflected in higher default probabilities. The CGMY model gives a similar default probability term structure.

4.6 Conclusion

In this chapter we employed the efficient and flexible COS method for calculating survival/default probabilities for pricing single name Credit Default Swaps.

We take a structural approach where the firm's value is modeled by an exponential Lévy process, focusing on two well known Lévy models: the NIG-BM (a NIG model extended with a Brownian Motion) model and the CGMY model.

The main idea is to relate the credit default spreads to a series of survival/default probabilities with different maturities, and to exploit the relationship between these survival probabilities and the price of Binary Down-and-Out Barrier options. To rapidly evaluate these option prices, and thus, the survival probabilities, we generalized the option pricing method based on the Fourier-cosine series expansion of the underlying density. In less than half a second, the 1-, 3-, 5, 7- and 10-year default probabilities were computed with very satisfactory accuracy. We also checked the convergence of survival probabilities w.r.t. the number of monitoring dates.

The method's potential was demonstrated via calibration of the NIG-BM and the CGMY Lévy models to the quotes of the constituents of the iTraxx Series 7 and Series 8. Both models give very good fits to the market quotes. The average Root Mean Square Error is less than 1.0 basis point for both Lévy models with respect to the Series 7 data, and is less than 2.0 basis points with respect to the Series 8

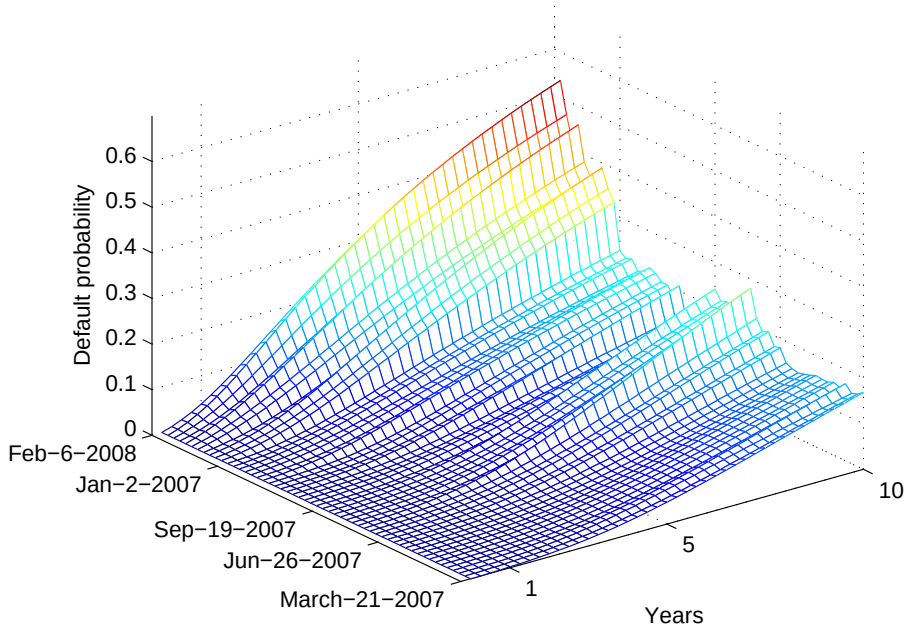


Figure 4.10: Default probability term structure of DSG International PLC under the NIG-BM model given by calibrating the model to the CDS weekly quotes from March 21, 2007 to February 6, 2008.

data. We have presented in the paper the evolution of the CDS market quotes and the related model prices over the year covered by the two iTraxx series. What can be seen is that the models and the method manage to reproduce the market prices of CDSs even at those times when there are dramatic changes in the prices. The evolution of the model parameters, resulting from the calibration, shows reasonable behavior, staying quite stable over time unless there are large changes in the market CDS spreads. The default probability term structures extracted from the market quotes mirror the change of the market CDS spreads over time.

From a numerical point-of-view, we saw that in many experiments we could use a smaller number of cosine series terms for the NIG-BM model than for the CGMY model as the former has a smoother density due to the diffusion part, making the COS method converge faster.

Chapter 5

Pricing Bermudan and Barrier Options under the Heston Stochastic Volatility

5.1 Introduction

This chapter contains essentially the contents of the paper [34].

In Mathematical Finance, stochastic volatility models have been developed to capture the volatility smiles and skews present in market quotes. Within this class, the Heston stochastic volatility model [41], in which the variance of (the logarithm of) the stock price is modeled by a square-root process, has become popular in industrial practice. The pricing of European options is particularly efficient.

Many exotic financial products include some form of path dependency. Monte Carlo simulation methods are often used for the valuation of such products in practice. As a result, the recent numerical advances in the context of Heston's model were obtained mainly for Monte Carlo simulation methods [14, 5]. However, it is well-known that the development of efficient simulation methods for pricing problems with early exercise features, as they are encountered for example when pricing Bermudan or discretely-monitored barrier options, is not a trivial task.

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

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

  The variance in the Heston model is governed by a non-central chi-square distribution. For some combinations of the relevant parameters, the density of the variance grows drastically in the left-side tail, i.e. the density values tend to infinitely large numbers as the variance approaches zero. Truncation of the integration range for the variance may then easily introduce significant truncation

errors.

- *The integration kernel is not known explicitly:*

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

- *Quadratic computational complexity:*

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

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

The contributions of this chapter are the following. We determine parameters sets for which the near-singular behavior matters, and tackle the problem by a transformation from the variance domain to the *log-variance domain*. Secondly, to solve the two-dimensional problem in a robust and efficient manner, we combine the Fourier cosine expansion with quadrature rules.

The chapter is organized as follows. In Section 5.2, we describe the Heston asset dynamics. We focus on the issue of the left-side tail of the variance density. In Section 5.3, the discrete pricing formula for Bermudan options is derived and an efficient recursive algorithm is developed. Minor differences when pricing discrete-barrier options are highlighted in Section 5.4. In Section 5.5 the error convergence and the error propagation are analyzed. Various numerical experiments are presented in Section 5.6, and conclusions are drawn in Section 5.7.

5.2 Heston Model Details

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

5.2.1 Heston Model Basics

The Heston stochastic volatility model has been briefly described in Chapter 1. Here we recall the dynamics: the logarithm of the stock price (*log-stock*), x_t, and the variance, ν_t, are governed by the following stochastic differential equations (SDEs) [41]:

$$dx_t = \left(\mu - \frac{1}{2}\nu_t\right) dt + \rho\sqrt{\nu_t}dW_{1,t} + \sqrt{1-\rho^2}\sqrt{\nu_t}dW_{2,t} \tag{5.1}$$

$$d\nu_t = \lambda\left(\bar{\nu} - \nu_t\right) dt + \eta\sqrt{\nu_t}dW_{1,t}, \tag{5.2}$$

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

The square-root process defined in (5.2) precludes negative values for ν_t, and if ν_t reaches zero it can subsequently become positive. The Feller condition, $2\lambda\bar{\nu} \geq \eta^2$, guarantees that ν_t stays positive; otherwise, it may reach zero. As indicated in [35, 25], with

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

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

$$p_\nu\left(\nu_t|\nu_s\right) = \zeta e^{-\zeta(\nu_s e^{-\lambda(t-s)}+\nu_t)}\left(\frac{\nu_t}{\nu_s e^{-\lambda(t-s)}}\right)^{\frac{q}{2}} I_q\left(2\zeta e^{-\frac{1}{2}\lambda(t-s)}\sqrt{\nu_s\nu_t}\right), \tag{5.3}$$

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

The Feller condition is thus equivalent to "$q \geq 0$". This is difficult to satisfy in practice. It has, for example, been reported [5] that one often finds $2\lambda\bar{v} << \eta^2$ from market data, in which case the cumulative distribution of the variance shows a *near-singular behavior* near the origin, or, in other words, the left tail of the variance density grows extremely fast in value.

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

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

The exact simulation method, developed in Broadie and Kaya [14], provides, next to an exact formula to sample the log-stock price, insight in the distribution for stochastic volatility models. Integration of (5.1) and (5.2) yields [14]:

$$x_t - x_s = \mu(t-s) - \frac{1}{2}\int_s^t \nu_\tau d\tau + \rho\int_s^t \sqrt{\nu_\tau}dW_{1,\tau} + \sqrt{1-\rho^2}\int_s^t \sqrt{\nu_\tau}dW_{2,\tau}, \tag{5.4}$$

$$\nu_t - \nu_s = \lambda\bar{\nu}(t-s) - \lambda\int_s^t \nu_\tau d\tau + \eta\int_s^t \sqrt{\nu_\tau}dW_{1,\tau}. \tag{5.5}$$

Equation (5.5) can be rewritten as an equation for $\int_s^t \sqrt{\nu_\tau}dW_{1,\tau}$, which, substituted in (5.4), gives the following exact formula for x_t:

$$\begin{aligned} x_t - x_s = \mu(t-s) + \frac{\rho}{\eta}\left(\nu_t - \nu_s - \lambda\bar{\nu}(t-s)\right) + \left(\frac{\lambda\rho}{\eta} - \frac{1}{2}\right)\int_s^t \nu_\tau d\tau \\ + \sqrt{1-\rho^2}\int_s^t \sqrt{\nu_\tau}dW_{2,\tau}. \end{aligned} \tag{5.6}$$

Equation (5.6) can be used to sample x_t, once the values of the variance, ν_t, and the time-integrated variance, $\int_s^t \nu_\tau d\tau$, are available. The variance is then sampled from (an approximation of) the non-central chi-square distribution, and the time-integrated variance is sampled from a distribution which is recovered from the ChF, $\Phi(u;\nu_t,\nu_s)$, for which a closed-form expression is available:

$$\begin{aligned}\Phi(\upsilon;\nu_t,\nu_s) &:= \mathbb{E}\left[\exp\left(i\upsilon\int_s^t \nu_\tau d\tau\right)\middle|\,\nu_t,\nu_s\right]\\ &= \frac{I_q\left[\sqrt{\nu_t\nu_s}\frac{4\gamma(\upsilon)e^{-\frac{1}{2}\gamma(\upsilon)(t-s)}}{\eta^2(1-e^{-\gamma(\upsilon)(t-s)})}\right]}{I_q\left[\sqrt{\nu_t\nu_s}\frac{4\lambda e^{-\frac{1}{2}\lambda(t-s)}}{\eta^2(1-e^{-\lambda(t-s)})}\right]}\cdot\frac{\gamma(\upsilon)e^{-\frac{1}{2}(\gamma(\upsilon)-\lambda)(t-s)}(1-e^{-\lambda(t-s)})}{\lambda(1-e^{-\gamma(\upsilon)(t-s)})}\cdot\\ &\quad\exp\left(\frac{\nu_s+\nu_t}{\eta^2}\left[\frac{\lambda(1+e^{-\lambda(t-s)})}{1-e^{-\lambda(t-s)}}-\frac{\gamma(\upsilon)(1+e^{-\gamma(\upsilon)(t-s)})}{1-e^{-\gamma(\upsilon)(t-s)}}\right]\right),\end{aligned}\tag{5.7}$$

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

$$\gamma(\upsilon):=\sqrt{\lambda^2-2i\eta^2\upsilon}.\tag{5.8}$$

In [14] the cumulative distribution function of the time-integrated variance is recovered numerically from the expression,

$$Pr\left(\int_s^t \nu_\tau d\tau\le x\right)=\frac{2}{\pi}\int_0^\infty\frac{\sin(ux)}{u}\mathrm{Re}\left\{\Phi(u)\right\}du,\tag{5.9}$$

($\mathrm{Re}\{\cdot\}$ denoting the real part of the expression in brackets) by means of the composite Trapezoidal rule. This un-biased simulation method requires a significant amount of CPU time [5], mainly because of the numerical inversion step.

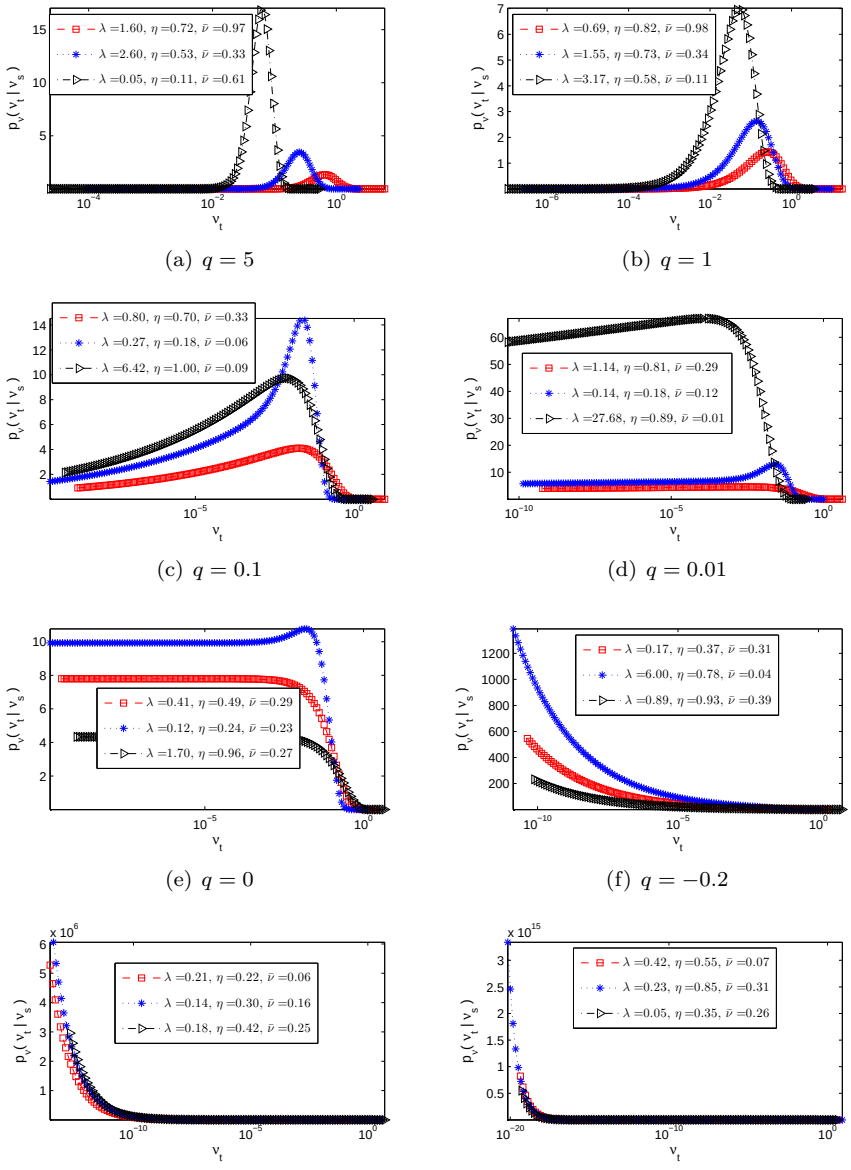
Remark 5.2.1 (Fast inverse Fourier transform by a Fourier cosine expansion)**.** *Application of the composite Trapezoidal rule for (5.9) is time-consuming, because the ChF is highly oscillatory, which implies that large values for N are required for accuracy.*

5.2.2 The Left-Side Tail

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

Result 5.2.1 (The left-side tail)**.** *Although each of the three parameters, λ, $\bar{\nu}$ and η, in (5.2) plays a unique role in the tuning of the shape and the magnitude of the variance density, the decay rate at the left tail can be well characterized by values of q, whose definition interval is $[-1,\infty)$. Based on the non-negativeness of λ, $\bar{\nu}$ and η, the near-singular problem occurs when $q\in[-1,0]$, which is directly related to the Feller condition.*

The experiments that support this insight are set up as follows: The values of $\bar{\nu}$ and η are drawn randomly from $[0, 1]$ (we consider interval $[0, 1]$ reasonable for both $\bar{\nu}$ and η), and λ is given by $(1 + q)\eta^2/(2\bar{\nu})$. The experimental results are displayed in Figure 5.1.



(a) $q = 5$ (b) $q = 1$

(c) $q = 0.1$ (d) $q = 0.01$

(e) $q = 0$ (f) $q = -0.2$

(g) $q = -0.5$ (h) $q = -0.8$

Figure 5.1: Decay rate in the left-side tail of the variance density, as q approaches -1 from above.

As shown in Figure 5.1, the value of q determines the decay rate in the left tail

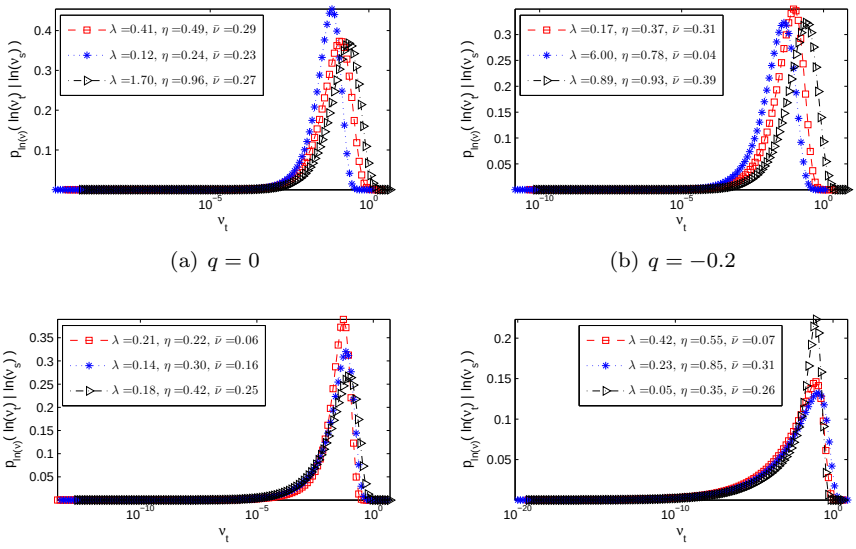
of the variance density function, whereas the right-side tail always decays to zero rapidly. For $q >> 0$ the density values tend towards zero in both tails. For q smaller and approaching 0, the decay of the left-side tail slows down. Near $q = 0$, the left tail stays almost constant. For $q \in [-1,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

In a recent paper, [5], several challenging test cases, based on different values of λ,η and $\bar{\nu}$, were illustrated. For all those test cases we find $q \approx -0.96$, which indeed is an indication of difficult tests, see Figure 5.1.

The fact that q determines the decay rate of the densities' left tail can be understood if we take a closer look at Equation (5.3) for the variance density function. When q changes sign, both functions, $(\cdot)^{q/2}$ and $I_q(\cdot)$, change shape around the origin, i.e., from monotonically increasing they become monotonically decreasing.

5.2.3 Transformation to Log-Variance Process

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



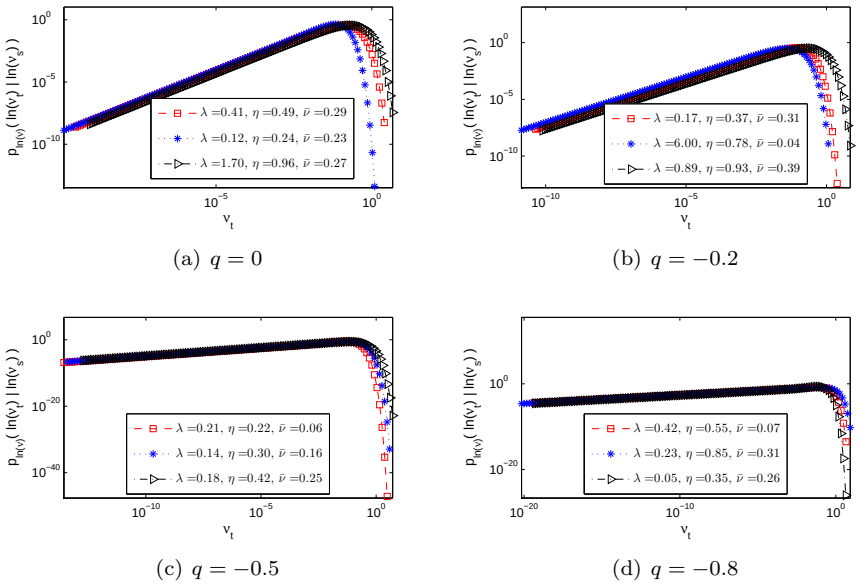
(a) $q = 0$ (b) $q = -0.2$

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

Figure 5.2: Decay rate of the left tail of the *log-variance density* as q approaches -1 from above

By the change of variables, the density of the log-variance process, based on (5.3), reads:

$$p_{\ln(\nu)}\left(\sigma_t|\sigma_s\right) = \zeta e^{-\zeta(e^{\sigma_s}e^{-\lambda(t-s)}+e^{\sigma_t})}\left(\frac{e^{\sigma_t}}{e^{\sigma_s}e^{-\lambda(t-s)}}\right)^{\frac{q}{2}} e^{\sigma_t} I_q\left(2\zeta e^{-\frac{1}{2}\lambda(t-s)}\sqrt{e^{\sigma_s}e^{\sigma_t}}\right), \tag{5.10}$$



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

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

Figure 5.3: Decay rate *in log-scale* of the left tail of the *log-variance density* as q approaches -1 from above

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

With the change of variables, a term e^{σ_t} appears, which, for $q \in [-1, 0]$, compensates the $(\cdot)^{\frac{q}{2}}$–term, so that it converges towards zero as $\sigma_t \to -\infty$. It is shown in Figure 5.2 that the densities of the log-variance process for different parameter sets are *more symmetric* than those from Figure 5.1; It is also illustrated in log-scale, in Figure 5.3, that for $q \in [-1, 0]$ the left tails of the densities do not increase significantly in value anymore. Instead, these tails decay to zero rapidly as $\sigma_t \to -\infty$, although the decay rate decreases as q approaches -1. In Figures 5.2 and 5.3 we have only shown the problematic cases from Figure 5.1; the left-side tails of the cases with $q > 0$ also decay very well for the log-variance process, of course.

Remark 5.2.2 (Truncation range)**.** *Before applying any numerical method, we need to define a proper truncation range for the log-variance density. For this, information about the* center *of the density as well as the decay of the left and right tails is required.*

Instead of giving a rule-of-thumb for this truncation range, as in [32, 54], we propose to use Newton's method to determine the interval boundaries, according to a pre-defined error tolerance, TOL. In accordance with this tolerance, the stopping criteria of the Newton's method reads $p_{\ln(\nu)}(x|\sigma_0; T) < TOL$ *for* $x \in \mathbb{R}\backslash[a_\nu, b_\nu]$.

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

help of Maple:

$$\frac{dp_{\ln(\nu)}(\sigma_t|\sigma_s)}{d\sigma_t} = -\left[(-\zeta e^{\sigma_t} - q - 1)\, I_q\left(2\sqrt{\zeta e^{\sigma_t} u}\right) - I_{q+1}\left(2\sqrt{\zeta e^{\sigma_t} u}\right)\right] \cdot \zeta e^{-u-\zeta e^{\sigma_t}+\sigma_t} \cdot \left(\frac{\zeta e^{\sigma_t}}{u}\right)^{q/2}, \tag{5.11}$$

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

A proper initial guess *for interval boundaries is also required. We estimate the center of the truncation range by the logarithm of the mean value of the variance, see e.g. [5],*

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

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

$$[a_\nu^0, b_\nu^0] = \left[\ln(\mathbb{E}(\nu_t)) - \frac{5}{1+q},\ \ln(\mathbb{E}(\nu_t)) + \frac{2}{1+q}\right]. \tag{5.12}$$

5.2.4 Joint Distribution of Log-stock and Log-variance

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

The SDEs in (5.1), (5.2) indicate that the variance at a future time is independent from the log-stock value at the current time, i.e. $p_\nu(\nu_t|\nu_s, x_s) = p_\nu(\nu_t|\nu_s)$. As a result, we have

$$p_{x,\nu}(x_t, \nu_t|x_s, \nu_s) = p_{x|\nu}(x_t|\nu_t, x_s, \nu_s) \cdot p_\nu(\nu_t|\nu_s), \tag{5.13}$$

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

$$p_{x,\ln(\nu)}(x_t, \sigma_t|x_s, \sigma_s) = p_{x|\ln(\nu)}(x_t|\sigma_t, x_s, \sigma_s) \cdot p_{\ln(\nu)}(\sigma_t|\sigma_s), \tag{5.14}$$

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

The probability density of the log-variance, $p_{\ln(\nu)}(\sigma_t|\sigma_s)$, is already given in (5.10) and therefore we need $p_{x|\ln(\nu)}(x_t|\sigma_t, x_s, \sigma_s)$. Although there is no closed-form expression for $p_{x|\ln(\nu)}$, one can easily derive its conditional characteristic function, $\varphi(\omega; x_s, \sigma_t, \sigma_s)$, based on (5.6):

$$\begin{aligned}\varphi(\omega; x_s, \sigma_t, \sigma_s) &:= \mathbb{E}_s\left[\exp\left(i\omega x_t|\sigma_t\right)\right] \\ &= \exp\left(i\omega\left[x_s + \mu(t-s) + \frac{\rho}{\eta}\left(e^{\sigma_t} - e^{\sigma_s} - \lambda\bar{\nu}(t-s)\right)\right]\right) \cdot \\ &\quad \Phi\left(\omega\left(\frac{\lambda\rho}{\eta} - \frac{1}{2}\right) + \frac{1}{2}i\omega^2(1-\rho^2); e^{\sigma_t}, e^{\sigma_s}\right),\end{aligned} \tag{5.15}$$

where $\Phi(u;\nu_t,\nu_s)$ is the ChF of the time-integrated variance as given in (5.7).

5.3 The Pricing Method for Bermudan Options

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

5.3.1 The Pricing Equations

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

$$v(x_s,\sigma_s,s)=e^{-r(t-s)}\mathbb{E}_s^{\mathbb{Q}}\left[v(x_t,\sigma_t,t)\right]. \tag{5.16}$$

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

The Markov property enables us to price a Bermudan option between two consecutive early-exercise dates by the risk-neutral valuation formula (5.16). This value is then called the *continuation value*. The arbitrage-free price of the Bermudan option on any early-exercise date is the maximum of the continuation value and the exercise payoff.

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

$$v(x_{t_m},\sigma_{t_m},t_m)=\begin{cases} g(x_{t_m},t_m) & \text{for } m=M;\\ \max\left[c(x_{t_m},\sigma_{t_m},t_m),g(x_{t_m},t_m)\right] & \text{for } m=1,2,\cdots,M-1;\\ c(x_{t_m},\sigma_{t_m},t_m) & \text{for } m=0,\end{cases} \tag{5.17}$$

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

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

$$c(x_m,\sigma_m,t_m)=e^{-r\Delta t}\mathbb{E}_{t_m}^{\mathbb{Q}}\left[v(x_{m+1},\sigma_{m+1},t_{m+1})\right], \tag{5.18}$$

which can be written as:

$$\begin{aligned} c(x_m,\sigma_m,t_m)&=e^{-r\Delta t}\cdot \\ &\int_{\mathbb{R}}\int_{\mathbb{R}} v(x_{m+1},\sigma_{m+1},t_{m+1})p_{x,\ln(\nu)}\left(x_{m+1},\sigma_{m+1}\mid x_m,\sigma_m\right)d\sigma_{m+1}dx_{m+1}. \end{aligned} \tag{5.19}$$

With (5.14) we get:

$$c(x_m,\sigma_m,t_m) = e^{-r\Delta t}\cdot \int_{\mathbb{R}}\left[\int_{\mathbb{R}} v(x_{m+1},\sigma_{m+1},t_{m+1}) p_{x|\ln(\nu)}\left(x_{m+1}|\,\sigma_{m+1},x_m,\sigma_m\right) dx_{m+1}\right] p_{\ln(\nu)}\left(\sigma_{m+1}|\sigma_m\right) d\sigma_{m+1}. \tag{5.20}$$

Equations (5.17) and (5.20) define the problem we would like to solve numerically. The inner integral in (5.20) equals the pricing formula for European options defined between t_m and t_{m+1}, provided the variance value at the future time point is known.

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

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

5.3.2 Density Recovery by Fourier Cosine Expansions

We will apply the COS method to approximate the unknown conditional probability density, $p_{x|\ln(\nu)}$ in (5.20).

We first recall how to recover the density function $p_{x|\ln(\nu)}$ in (5.20) by the COS method.

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

$$\int_a^b p_{x|\ln(\nu)}(x_{m+1}|\sigma_{m+1},x_m,\sigma_m)dy \;\leq \mathrm{TOL}_x, \tag{5.21}$$

for some pre-defined error tolerance TOL_x. In Chapter 2 this interval was defined as

$$[a,b] := [\xi_1 - 12\sqrt{|\xi_2|},\xi_1 + 12\sqrt{|\xi_2|}], \tag{5.22}$$

where ξ_n denotes the n-th cumulant of the log-stock process. With an integration interval $[a,b]$ satisfying (5.22), we recover the probability density by its Fourier cosine series expansion:

$$p_{x|\ln(\nu)}(x_{m+1}|\sigma_{m+1},x_m,\sigma_m) = {\sum_{n=0}}'^{\infty} P_n(\sigma_{m+1},x_m,\sigma_m)\cos\left(n\pi\frac{x_{m+1}-a}{b-a}\right). \tag{5.23}$$

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

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

By the expansion in (5.23), one *separates* x_{m+1} from x_m. This type of variable separation is not restricted to Fourier cosine series expansions, but in this case the Fourier expansion is advantageous as the series coefficients have a direct relation to the characteristic function and are therefore known, i.e.

$$P_n(\sigma_{m+1},x_m,\sigma_m) \approx \frac{2}{b-a}\mathrm{Re}\left\{\varphi\left(\frac{n\pi}{b-a};x_m,\sigma_{m+1},\sigma_m\right)\; e^{-in\pi\frac{a}{b-a}}\right\}, \tag{5.24}$$

with $\varphi(\theta; x, \sigma_{m+1}, \sigma_m)$ given by (5.15).

The error in this approximation is related to TOL_x, as analyzed in Section 2.4, and Equation (5.24) approximates the P_n with machine accuracy if $[a, b]$ is sufficiently wide. Subsequently, we truncate the series summation in (5.23).

From Fourier theory, we know that cosine series of functions belonging to $\mathbb{C}^\infty([a, b] \subset \mathbb{R})$, with non-zero derivatives, converge exponentially with respect to the number of terms in the series, so that the series can be truncated without loosing accuracy. By replacing P_n in (5.23) by (5.24) and truncating the series by N terms, one obtains a *semi-analytic formula* which accurately approximates the probability density:

$$p_{x|\ln(\nu)}(x_{m+1}|\sigma_{m+1}, x_m, \sigma_m) = \frac{2}{b-a} \cdot \sum_{n=0}^{\prime N-1} \text{Re}\left\{\varphi\left(\frac{n\pi}{b-a}; 0, \sigma_{m+1}, \sigma_m\right) e^{in\pi\frac{x_m-a}{b-a}}\right\} \cos\left(n\pi\frac{x_{m+1}-a}{b-a}\right) + \epsilon_{cos}. \tag{5.25}$$

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

The error of this approximation, ϵ_{cos}, decreases exponentially with respect to N, provided that the truncation range is set sufficiently wide (proof is given in Chapter 2).

Remark 5.3.1 (Recover a CDF)**.** *The COS method can also be used to recover a cumulative probability distribution, $F(x)$: We simply insert the COS reconstruction of the density $f(t)$ into the definition integral of the cumulative probability after truncating the integration range, i.e.*

$$\begin{aligned} F(x) &= \int_{-\infty}^{x} f(t)dt \approx \int_{a}^{x} f(t)dt \\ &\approx \int_{a}^{x} \sum_{n=0}^{\prime N-1} \frac{2}{b-a} Re\left\{\varphi\left(\frac{n\pi}{b-a}\right)\right\} \cos\left(n\pi\frac{t-a}{b-a}\right) dt \\ &= \sum_{n=0}^{\prime N-1} \frac{2}{b-a} Re\left\{\varphi\left(\frac{n\pi}{b-a}\right)\right\} \int_{b}^{x} \cos\left(n\pi\frac{t-a}{b-a}\right) dt \\ &= \sum_{n=0}^{\prime N-1} \frac{2}{b-a} Re\left\{\varphi\left(\frac{n\pi}{b-a}\right)\right\} \psi_n(b, x), \end{aligned}$$

where $\psi_n(l, u)$ is given in (2.23).

5.3.3 Discrete Fourier-based Pricing Formula

Equation (5.17) shows that the option price at time t_0 is a continuation value, which, as indicated by (5.20), depends on the continuation values at the times $t_1, t_2, \cdots, t_M$. The option price at time t_0 can be recovered by recursion, backwards in time. This is the same approach as in Chapter 3, but here the integration is more involved, because of the two-dimensional kernel.

Quadrature Rule in Log-variance Dimension

Using the initial values defined in (5.12) and (5.22), we obtain the truncation range $[a_\nu, b_\nu]$ by Newton's method.

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

$$c_1(x_m, \sigma_m, t_m) := e^{-r\Delta t} \cdot \int_{a_\nu}^{b_\nu} \Big[\int_a^b v(x_{m+1}, \sigma_{m+1}, t_{m+1}) p_{x|\ln(\nu)}\left(x_{m+1}|\ \sigma_{m+1}, x_m, \sigma_m\right) dx_{m+1} \Big] p_{\ln(\nu)}\left(\sigma_{m+1}|\sigma_m\right) d\sigma_{m+1}. \tag{5.26}$$

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

There are two ways to discretize the outer integral w.r.t. σ_{m+1}, i.e by interpolation-based quadrature rules or by a spectral series reconstruction of the interpolant (as in the COS method). In the latter case since the ChF of $p_{\ln(\nu)}$ is not known, one would have to use a numerical method to retrieve the series coefficients for a series reconstruction, which would add additional computational costs. However, since $p_{\ln(\nu)}$ itself is known analytically, we apply a J-point quadrature integration rule (like Gauss-Legendre quadrature, composite Trapezoidal rule, etc.) to the outer integral, which gives

$$c_2(x_m, \sigma_m, t_m) := e^{-r\Delta t} \sum_{j=0}^{J-1} w_j \cdot p_{\ln(\nu)}(\varsigma_j|\sigma_m) \cdot \Big[\int_a^b v(x_{m+1}, \varsigma_j, t_{m+1}) p_{x|\ln(\nu)}\left(x_{m+1}|\ \varsigma_j, x_m, \sigma_m\right) dx_{m+1} \Big]. \tag{5.27}$$

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

Remark 5.3.2 (Which quadrature rule to use?)**.** *There are merits and demerits to using high-order quadrature rules, like the Gauss-Legendre quadrature rule, and to low-order equidistant rules, like the composite Trapezoidal rule. The advantage of the former is an exponential error convergence rate for integration of smooth functions, as is the case for $p_{\ln(\nu)}$, whereas the latter has only polynomial error convergence. However, the computational complexity of the method can be greatly reduced by the Trapezoidal rule, due to a special matrix structure which results after discretization on an equidistant grid. We will come back to this issue.*

COS Reconstruction in Log-stock Dimension

In the next step, we replace $p_{x|\ln(\nu)}$, which is not known, by the COS approximation (5.25), and interchange the summation over n with the integration over x_{m+1} to obtain:

$$c_3(x_m, \sigma_m, t_m) := e^{-r\Delta t} \sum_{j=0}^{J-1} w_j {\sum}'_{n=0}^{N-1} V_{n,j}(t_{m+1}) \mathrm{Re}\left\{ \tilde{\varphi}\left(\frac{n\pi}{b-a}, \varsigma_j, \sigma_m\right) e^{in\pi \frac{x_m - a}{b-a}} \right\}, \tag{5.28}$$

with

$$V_{n,j}\left(t_{m+1}\right) := \frac{2}{b-a}\int_a^b v(x_{m+1},\varsigma_j,t_{m+1})\cos\left(n\pi\frac{x_{m+1}-a}{b-a}\right)dx_{m+1}, \tag{5.29}$$

and

$$\tilde{\varphi}(\omega,\sigma_{m+1},\sigma_m) := p_{\ln(\nu)}(\sigma_{m+1}|\sigma_m)\cdot\varphi\left(\omega;0,e^{\sigma_{m+1}},e^{\sigma_m}\right). \tag{5.30}$$

The kernel function $\tilde{\varphi}$ will be the only input which characterizes the Heston model. By combining the lengthy formulas of (5.10) and (5.15), the Bessel function present in $p_{\ln(\nu)}$ cancels with the Bessel function in the denominator of φ, leaving one Bessel-term, $I_q\left(e^{\frac{1}{2}(\sigma_{m+1}+\sigma_m)}\cdot 2\kappa(\upsilon)e^{-\frac{1}{2}\gamma(\upsilon)\Delta t}\right)$ with $\gamma(\upsilon)$ given by (5.8),

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

Coefficients $V_{n,j}\left(t_{m+1}\right)$ defined in (5.29) can be interpreted as the Fourier cosine series coefficients of the option value at time t_{m+1}. Expression $c_3(x_m,\sigma_m,t_m)$ in (5.28) thus becomes a scaled inner product of the Fourier cosine series coefficients of the option price and of the underlying density.

Finally, we interchange the summations in (5.28) which yields the discrete formula for the continuation value:

$$c_3(x_m,\sigma_m,t_m) = e^{-r\Delta t}\mathrm{Re}\left\{\sum\nolimits_{n=0}^{\prime N-1}\beta_n(\sigma_m,t_m)e^{in\pi\frac{x_m-a}{b-a}}\right\}, \tag{5.31}$$

where

$$\beta_n(\sigma_m,t_m) := \sum_{j=0}^{J-1} w_j V_{n,j}(t_{m+1})\tilde{\varphi}\left(\frac{n\pi}{b-a},\varsigma_j,\sigma_m\right). \tag{5.32}$$

Equation (5.31) expresses the continuation value at time t_m as a series expansion. The series coefficients, which depend only on the value of the variance (and not on the log-stock value) at time t_{m+1}, are (scaled) inner products of the cosine series coefficients of the option price at time t_{m+1} and the variance-dependent characteristic function $\tilde{\varphi}$.

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

$$c_3(x_m,\varsigma_p,t_m) = e^{-r\Delta t}\mathrm{Re}\left\{\sum\nolimits_{n=0}^{\prime N-1}\beta_n(\varsigma_p,t_m)\exp\left(in\pi\frac{x_m-a}{b-a}\right)\right\}, \tag{5.33}$$

with

$$\beta_n(\varsigma_p,t_m) := \sum_{j=0}^{J-1} w_j V_{n,j}(t_{m+1})\tilde{\varphi}\left(\frac{n\pi}{b-a},\varsigma_j,\varsigma_p\right). \tag{5.34}$$

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

One of the advantages of this *spectral dimension* is that Expression (5.31) is known for any value of $x_m \in \mathbb{R}$, not just for discrete values. So, one can determine the *early-exercise points* rapidly, by solving

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

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

When the early-exercise points, $x^*(\sigma_m, t_m)$, have been determined, Procedure (5.17) can be used to compute the Bermudan option price. More specifically:

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

$$\hat{v}(x_m, \sigma_m, t_m) = \begin{cases} g(x_m) & \text{for } x \in [a, x^*(\sigma_m, t_m)] \\ c_3(x_m, \sigma_m, m) & \text{for } x \in (x^*(\sigma_m, t_m), b] \end{cases} \tag{5.35}$$

  for a put option, and

$$\hat{v}(x_m, \sigma_m, t_m) = \begin{cases} c_3(x_m, \sigma_m, m) & \text{for } x \in [a, x^*(\sigma_m, t_m)] \\ g(x_m) & \text{for } x \in (x^*(\sigma_m, t_m), b] \end{cases} \tag{5.36}$$

  for a call option.

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

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

With the procedure above and Expression (5.31), we can compute recursively $\hat{v}(x_0, \sigma_0, t_0)$ from $\hat{v}(x_M, \sigma_M, t_M)$, backwards in time.

However, a more efficient technique exists. Instead of reconstructing $\hat{v}$ for each time point, we can recover the cosine series coefficients *using backward recursion*, and only at time t_0 we apply (5.31) to reconstruct $\hat{v}$.

5.3.4 Backward Recursion

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

We first discuss the final time point, t_M. Since the option price at the maturity date equals the payoff (which does not depend on time), one can derive an analytic expression for $V_{n,j}(t_M)$ using (5.29):

$$V_{n,j}(t_M) = \begin{cases} G_n(0, b), & \text{for call options} \\ G_n(a, 0), & \text{for put options,} \end{cases} \tag{5.37}$$

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

$$G_n(l, u) := \frac{2}{b-a} \int_l^u g(y) \cos\left(n\pi \frac{y-a}{b-a}\right) dy, \tag{5.38}$$

with

$$g(y) = [\alpha K (e^y - 1)]^+, \quad \alpha = \begin{cases} 1, & \text{for a call option} \\ -1, & \text{for a put option.} \end{cases} \tag{5.39}$$

The analytic solution of $G_n(l,u)$ has already been given in (3.20).

Subsequently, we continue with time point t_{M-1}. By inserting $V_{n,j}(t_M)$ into (5.34), we obtain $\beta_n(\varsigma_p, t_{M-1})$ for $p = 0, 1, \cdots, J-1$. With (5.33) one finds an analytic formula, $c_3(x_{M-1}, \varsigma_p, t_{M-1})$, for the continuation value at time t_{M-1}. By Newton's method, we then solve $c_3(y, \varsigma_p, t_{M-1}) - g(y) = 0$ to determine the location of the early-exercise point, $y \equiv x^*(\varsigma_p, t_{M-1})$.

With early-exercise point, $x^*(\varsigma_p, t_{M-1})$, known and $\hat{v}(x_{M-1}, \varsigma_p, t_{M-1})$ as in (5.35) or (5.36), we split the integral in (5.29) in two parts (for $p = 0, 1, \cdots, J-1$.):

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

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

$$\hat{C}_{k,p}(l,u,t_{M-1}) := \frac{2}{b-a} \int_l^u c_3(y, \varsigma_p, t_{M-1}) \cos\left(k\pi \frac{y-a}{b-a}\right) dy. \tag{5.40}$$

For the exact cosine coefficient of the continuation value, $C_{k,p}$, we should have used c from (5.40), instead of the COS approximation c_3 from (5.33).

After replacing c_3 in (5.40) by the COS approximation, interchanging summation and integration, we obtain

$$\hat{C}_{k,p}(l,u,t_{M-1}) = e^{-r\Delta t} \operatorname{Re}\left\{ {\sum_{n=0}}' {}^{N-1} \mathcal{M}_{k,n}(l,u) \beta_n(\varsigma_p, t_{M-1}) \right\}, \tag{5.41}$$

with

$$\mathcal{M}_{k,n}(l,u) := \int_l^u \exp\left(in\pi \frac{y-a}{b-a}\right) \cos\left(k\pi \frac{y-a}{b-a}\right) dy. \tag{5.42}$$

Expression (5.42) can be obtained analytically, as given in Chapter 3 and Eq. (3.30).

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

$$\hat{\mathrm{C}}(l,u,t_{M-1}) = e^{-r\Delta t} \operatorname{Re}\left\{ \mathcal{M}(l,u) \mathrm{B}'(t_{M-1}) \right\}, \tag{5.43}$$

where B' indicates that the first row of matrix B is multiplied by one-half.

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

$$\mathrm{B}(t_{M-1}) = [\beta_0(t_{M-1}), \beta_1(t_{M-1}), \cdots, \beta_{J-1}(t_{M-1})]. \tag{5.44}$$

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

$$\beta_p(t_{M-1}) = [\mathrm{V}(t_M) \cdot \tilde{\varphi}(\varsigma_p)]\, \mathbf{w}, \tag{5.45}$$

where $\mathbf{w}$ is a column vector (length J) with the quadrature weights and the (time-invariant) matrix $\tilde{\varphi}(\varsigma_p)$ is an $N \times J$ matrix with as elements $\tilde{\varphi}\left(\frac{n\pi}{b-a}, \varsigma_j, \varsigma_p\right)$, as defined in (5.30). The operator "$\cdot$" in (5.45) denotes an element-wise matrix-matrix product.

From [33] we know that matrix $\mathcal{M}(l,u)$ can be written as the sum of a Hankel matrix, $\mathcal{M}_c(l,u)$, and a Toeplitz matrix, $\mathcal{M}_s(l,u)$. Because matrix-vector products with Hankel and Toeplitz matrices can be transformed into circular convolutions of two vectors, the FFT algorithm can be applied to achieve the $O(N \log_2(N))$ complexity in log-stock space. Details have been given in Section 3.2.3.

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

$$\begin{cases} \hat{\mathrm{V}}(t_m) := \begin{cases} \hat{\mathrm{C}}(x^*(\varsigma_p, t_m), b, t_m) + \mathrm{G}(a, x^*(\varsigma_p, t_m)) & \text{for a put} \\ \hat{\mathrm{C}}(a, x^*(\varsigma_p, t_m), t_m) + \mathrm{G}(x^*(\varsigma_p, t_m), b) & \text{for a call} \end{cases} \\ \hat{\beta}_j(t_{m-1}) := \left[\hat{\mathrm{V}}(t_m) \cdot \tilde{\varphi}(\varsigma_j)\right] \mathbf{w} \\ \hat{\mathrm{B}}(t_{m-1}) := \left[\hat{\beta}_0(t_{m-1}), \hat{\beta}_1(t_{m-1}), \cdots, \hat{\beta}_{J-1}(t_{m-1})\right] \\ \hat{\mathrm{C}}(l,u,t_{m-1}) := e^{-r\Delta t}\, \mathrm{Re}\left\{\mathcal{M}(l,u)\hat{\mathrm{B}}'(t_{m-1})\right\} \end{cases} \tag{5.46}$$

We continue the procedure until $\hat{\mathrm{V}}(t_1)$ is recovered, which is then inserted into (5.34) and (5.31) to get a *grid of option prices*, $\hat{v}(x_0, \varsigma_j, t_0)$, for $j = 0, 1, \cdots, J-1$.

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

We summarize the backward recursion algorithm below.

Algorithm 5.3.1 (Pricing Bermudan options under Heston's model).

Initialization:

- Find a_ν and b_ν by Newton's method;
- Calculate $\mathrm{V}(t_M)$ with the analytic formula;
- Prepare matrix $\tilde{\varphi}(\varsigma_j)$ for $j = 0, 1, \cdots, J-1$.

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

- Determine early-exercise point by Newton's method;
- Calculate the first row and column of $\mathrm{M_s}$ and $\mathrm{M_c}$.
- For $j = 0, 1, \cdots, J-1$, calculate $\hat{\beta}_j(t_m) = \left[\hat{\mathrm{V}}(t_m) \cdot \tilde{\varphi}(\varsigma_j)\right] \mathbf{w}$.
- Multiply the first element of $\hat{\beta}_j(t_m)$ by one-half.
- Compute the column vectors of $\hat{\mathrm{C}}(t_m)$, $e^{-r\Delta t}\ \mathrm{Re}\left\{\mathcal{M}\hat{\beta'}_j(t_{m-1})\right\}$, using the FFT algorithm;
- Recover $\hat{\mathrm{V}}(t_m)$ by (5.35) or (5.36).

Final step: Calculate $\hat{v}(x, \varsigma_j, t_0)$ by inserting $\hat{\mathrm{V}}(t_1)$ into (5.34) and (5.31). Use spline interpolation to get $\hat{v}(x, \sigma_0, t_0)$.

Remark 5.3.3 (Multiple values of S_0). *Due to the use of the spectrally-oriented discretization in the log-stock dimension, the cosine coefficients of $\hat{\mathrm{V}}(t_m)$ do not depend on the initial value of asset prices. Only in the final step, one needs to insert an initial value, S_0, into (5.34) and (5.31) to get the option price. If necessary, the method could thus price multiple options that only vary in the value of S_0 simultaneously, with almost no additional cost.*

Remark 5.3.4 (Scaled Bessel function). *Special attention should be given to the calculation of $\tilde{\varphi}(\omega, \sigma_{m+1}, \sigma_m)$. First of all, it involves a modified Bessel function of the first kind, which increases dramatically in value when $q \to -1$ and/or $\omega \to \infty$. The* scaled Bessel function *should be used instead. A robust package has been developed in [3, 4] with algorithms to compute $I_d^*(z) := \exp\left(-|Re\{z\}|\right) I_d(z)$ with a complex-valued argument, z, and a real-valued order, d. As MATLAB (which we used here) incorporates this package for the MATLAB Bessel function, we replace $I_q(\cdot)$ by $e^{|Re\{\cdot\}|} I_q^*(\cdot)$ during the computations.*

Remark 5.3.5 (Computation of Bessel function). *The computation of the modified Bessel function costs significantly more (approximately a factor of* 1000*) CPU time than a simple multiplication, because the main part of the Bessel function algorithm is based on iterations. If the computation of the Bessel function costs $\mathcal{A}$ times the number of operations needed for a multiplication, a matrix based on $\tilde{\varphi}\left(\frac{k\pi}{b-a}, \varsigma_q, \varsigma_j\right)$ would require $O(NJ^2\mathcal{A})$ operations to compute all matrix elements.*

If one employs equidistant quadrature rules for the log-variance dimension, then for a given value of k, the input argument of the Bessel function is a function of the

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

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

The computations in the main loop of the algorithm are of order $O(MN \log_2(N) J^2)$, dominated by the calculation of matrix $\hat{\mathrm{B}}(t_{m-1})$. Since the computation of *vector* $\hat{\beta}_j(t_{m-1})$ costs $O(NJ)$ operations, the calculation of matrix $\hat{\mathrm{B}}(t_{m-1})$ is of $O(NJ^2)$ complexity.

The direct computation of the matrix-matrix product in (5.43) would cost $O(N^2 J)$ operations. The computational complexity of (5.43) is, however, $O(N \log_2(N) J)$, due to the special structure of matrix $\mathcal{M}(l, u)$ and the use of the FFT algorithm.

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

Remark 5.3.6 (Computation of Bermudan call)**.** *In [32], it has been reported that larger errors for call than for put prices can be observed, because call option prices are not bounded. It has been recommended to use the put-call parity for the valuation of call options. This also holds for Bermudan options, where the put-call parity can be applied to get the continuation values between two adjacent dates.*

5.4 Discrete Barrier Options

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

For an "out" barrier put option with M monitoring dates, the pricing formula reads for $m = 0, 1, \cdots, M-1$:

$$v(x_m, \sigma_m, t_m) = \begin{cases} \text{Rebate } r_b, & \text{when knocked out,} \\ c(x_m, \sigma_m, t_m), & \text{otherwise,} \end{cases} \tag{5.47}$$

and:

$$v(x_M, \sigma_M, t_M) = \begin{cases} \text{Rebate } r_b, & \text{when knocked out,} \\ g(x_M), & \text{otherwise,} \end{cases} \tag{5.48}$$

where the continuation value is governed by (5.20), as for Bermudan options.

The option price at the maturity date, t_M, equals the payoff if the option is not knocked out (or knocked in), otherwise the option price equals the rebate. Following

(5.29), the Fourier cosine coefficients of $v(x_m, \sigma_m, t_M)$, i.e., $V_{n,j}(t_M)$, satisfy

$$V_{n,j}(t_M) = \frac{2}{b-a}\int_{[a,l]\cup[u,b]} r_b \cos\left(n\pi\frac{y-a}{b-a}\right)dy + \frac{2}{b-a}\int_l^u g(y)\cos\left(n\pi\frac{y-a}{b-a}\right)dy$$
$$= \frac{2r_b}{b-a}\left(\psi_n(a,l)+\psi_n(u,b)\right) + G_n(l,u), \tag{5.49}$$

with $g(y)$ as defined in (5.39), l and u denote lower and upper barrier levels, respectively [1], and the G_n-terms are the cosine coefficients of the payoff function $g(y)$, as given in (3.20).

At t_{M-1} the barrier levels split the integral in (5.35) or (5.36) into several parts:

$$\hat{V}_{k,p}(t_{M-1}) = \frac{2r_b}{b-a}\left(\psi_k(a,l)+\psi_k(u,b)\right) + \frac{2}{b-a}\int_l^u c_3(y,\varsigma_p,t_{M-1})\cos\left(k\pi\frac{y-a}{b-a}\right)dy$$
$$= \frac{2r_b}{b-a}\left(\psi_k(a,l)+\psi_k(u,b)\right) + \hat{C}_{k,p}(l,u,t_{M-1}). \tag{5.50}$$

where $\hat{C}_{k,p}$ are the cosine coefficients of the continuation value as given in (5.40).

We can repeat the derivation from before: We replace c_3 in (5.40) by the COS approximation and interchange the summation and the integration, which gives:

$$\hat{\mathrm{C}}(l,u,t_{M-1}) = e^{-r\Delta t}\,\mathrm{Re}\left\{\mathcal{M}(l,u)\mathrm{B}'(t_{M-1})\right\}, \tag{5.51}$$

where, as before, the first row of matrix B is multiplied by one-half, and $\mathrm{B}(t_{M-1})$ is obtained as in (5.44) and (5.45). Matrix $\mathcal{M}(l,u)$ is an $N \times N$ matrix, which is *time-invariant* as l and u are a-priori known barrier levels. As a result, this matrix $\mathcal{M}(l,u)$ (only two columns and two rows needed for the circular convolution) can be *pre-computed*. Compared to Algorithm 5.3.1 the main difference is that the computation of this matrix is not in the main recursion loop.

Following the same procedure, we move backwards in time and find the equations that connect $\hat{\mathrm{V}}(t_{m-1})$ with $\hat{\mathrm{V}}(t_m)$, for $m = M-1, M-2, \cdots, 2$. Having $\hat{\mathrm{V}}(t_1)$ approximated, we insert it in (5.34) and (5.31) to obtain the option price $\hat{v}(x_0,\sigma_0,t_0)$.

5.5 Error Analysis

Using the same approach as in the error analysis sections of Chapter 2 and 3, we study here the convergence of the local error at each time lattice, as well as the propagation of the error from one time lattice to the next.

5.5.1 Local Error

We first analyze the convergence of the *local* error

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

We depart from (5.20) and denote the inner integral by $\vartheta(x_m,\sigma_{m+1},\sigma_m)$, which actually satisfies a risk-neutral valuation formula and thus defines the continuation

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

value at time t_m given σ_{m+1} and σ_m. For analysis purposes, we introduce an intermediate approximation, after the truncation of the integration range of the outer-integral by $[a_\nu, b_\nu]$:

$$c_0(x_m, \sigma_m, t_m) := e^{-r\Delta t} \int_{a_\nu}^{b_\nu} p_{\ln(\nu)}\left(\sigma_{m+1}|\sigma_m\right) \vartheta(x_m, \sigma_{m+1}, \sigma_m) d\sigma_{m+1}. \tag{5.52}$$

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

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

It then follows that

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

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

$$e^{-r\Delta t} \int_{\mathbb{R}\setminus[a_\nu, b_\nu]} p_{\ln(\nu)}\left(\sigma_{m+1}|\sigma_m\right) d\sigma_{m+1} \leq \text{TOL} \cdot \delta_1(q). \tag{5.53}$$

TOL in (5.53) appears because the size $[a_\nu, b_\nu]$ ensures that $p_{\ln(\nu)}(\sigma_{m+1}|\sigma_m) < \text{TOL}$ for $\sigma_{m+1} \in \mathbb{R}\setminus[a_\nu, b_\nu]$. Collecting the information gives:

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

Another intermediate quantity is obtained by replacing $p_{x|\ln(\nu)}$ in (5.52) with the approximation by the Fourier cosine series expansion, i.e.

$$\bar{c}(x_m, \sigma_m, t_m) = e^{-r\Delta t} \int_{a_\nu}^{b_\nu} p_{\ln(\nu)}\left(\sigma_{m+1}|\sigma_m\right) \tilde{\vartheta}(x_m, \sigma_{m+1}, \sigma_m) d\sigma_{m+1}, \tag{5.54}$$

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

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

The error analysis in [32] shows that the error due to the COS approximation,

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

$\forall x_m \in [a, b], \forall \sigma_{m+1}, \sigma_m \in \mathbb{R}$, converges *exponentially* in N for very smooth densities when the integration range $[a + x_m, b + x_m]$ is sufficiently wide. As such, we have

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

The approximation c_3 defined in (5.31) can now be obtained by applying a quadrature rule to the integral of (5.54). Suppose that the (absolute) error from the quadrature rule is $\epsilon_Q(J)$. With the triangle inequality, it then follows that $\forall\, x_m \in [a,b]$ and $\forall\, \sigma_m, \sigma_{m+1} \in [a_\nu, b_\nu]$:

$$\begin{aligned}\epsilon(x_m,\sigma_m,t_m) = |c-c_3| &\leq |c-c_0| + |c_0-\overline{c}| + |\overline{c}-c_3| \\ &\leq \text{TOL}\cdot\delta_0\cdot\delta_1(q) + \epsilon_{cos}(N,a,b) + \epsilon_Q(J) := \epsilon_{loc}. \end{aligned} \tag{5.55}$$

The local error thus consists of three parts:

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

One can observe the numerical convergence of the local error with respect to parameter J by setting N sufficiently large and TOL sufficiently small. This is included in Section 5.6.

5.5.2 Error Propagation during Recursion

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

Starting at t_M, $\text{V}(t_M)$ is exact since the option price at t_M is known analytically. At time t_{M-1}, an error, $\varepsilon_{k,p}(t_{M-1})$, exists because we replaced c by c_3 to determine $V_{k,p}(t_{M-1})$. Based on (5.40), we get

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

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

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

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

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

With (5.55) for all σ_m and y, we obtain

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

In the matrix max-norm, this reads as

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

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

$$\left|\left|\hat{\mathrm{V}}(t_{m+1}) - \mathrm{V}(t_{m+1})\right|\right|_{\max} \sim O\left(\epsilon_{loc}\right), \tag{5.56}$$

then it will also hold for time t_m:

The final equation in (5.46) is equivalent to:

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

where $\hat{c}_3(x_m, \sigma_m, t_m)$ is based on the same definition as $c_3(x_m, \sigma_m, t_m)$ in (5.28), except that $V_{n,j}(t_{m+1})$ is replaced by $\hat{V}_{n,j}(t_{m+1})$. As such, it holds that

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

To analyze this error term, we decompose $\tilde{\varphi}$ using (5.30) and replace the $\mathrm{Re}\{\cdot\}$-term by P_n, defined in (5.24), which gives

$$c_3(x_m, \sigma_m, t_m) - \hat{c}_3(x_m, \sigma_m, t_m) = e^{-r\Delta t}\sum_{j=0}^{J-1} w_j p_{\ln(\nu)}(\varsigma_j|\sigma_m)\Theta(\varsigma_j, x_m, \sigma_m), \tag{5.57}$$

where

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

From (5.22), we know that

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

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

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

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

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

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

It then follows that

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

With (5.56), we can write $\varepsilon_{n,j}(t_{m+1}) \leq \sqrt{\delta_4}\epsilon_{loc}$ for some positive number δ_4, and find that

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

Returning to Eq. (5.57) and employing the Cauchy-Schwarz inequality, gives us

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

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

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

$$\begin{aligned} &\left| \hat{V}_{k,q}(x_m, \sigma_m, t_m) - V_{k,q}(x_m, \sigma_m, t_m) \right| = \left| \hat{C}_{k,q}(a, x^*(\varsigma_q, t_m), t_m) - C_{k,q}(a, x^*(\varsigma_q, t_m), t_m) \right| \\ &\leq \frac{2}{b-a} \sqrt{\int_{x^*(\varsigma_p, t_{M-1})}^{b} (c(y, \varsigma_q, t_m) - c_3(y, \varsigma_q, t_m))^2 dy} \sqrt{\int_{x^*(\varsigma_p, t_{M-1})}^{b} \cos^2\left(k\pi \frac{y-a}{b-a}\right) dy} \\ &\leq 2e^{-r\Delta t} \sqrt{\delta_3\delta_4\delta_5} \cdot \sqrt{JN} \cdot \epsilon_{loc}. \end{aligned} \tag{5.58}$$

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

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

The speed of convergence will, however, decrease when the number of monitoring dates increases, due to the increasing weighting term in (5.58). Larger values for N and J are required in that case. We will examine this via numerical experiments in the next sections.

5.6 Numerical Results

In this section, we first confirm, by numerical experiments, in Subsection 5.6.1 the error convergence analysis from Section 5.5 by pricing *discrete barrier options* for which we set $l = a$ and $u = b$. This should give us the prices of European options with the barrier option pricing algorithm, and therefore we can generate reference values by the European version of the COS method from [32]. Since only a limited number of reference values are found in the literature, we use this special case to study the error convergence.

Subsequently, we price two Bermudan-style options with several early-exercise dates in Subsection 5.6.2. Their values should resemble American reference options that we use for comparison.

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

5.6.1 Error Analysis Experiment

We check the error convergence analysis from Section 5.5 by pricing discrete barrier options for which we set $l = a$ and $u = b$. This gives us European option prices, so we compute highly accurate reference values (accurate up to the 8-th decimal place) by the European option pricing method from [32].

Three tests are extracted from [5], one relatively easy case, with $q > 0$, and two significantly more difficult cases for which $q \in [-1, 0]$:

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

Numerical methods for early-exercise or barrier options are usually either based on finite differences for PDEs [44] or on tree-based methods [72, 9]. Results with these techniques using the parameter sets that give rise to significant pricing difficulties for early-exercise options under Heston's dynamics (i.e. Feller condition not satisfied) have however not yet been published.

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

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

and we do not consider dividend payment here.

First of all, we compare the error convergence in J for the Heston pricing methods with the composite Trapezoidal rule (upper part of Table 5.1) with results obtained by the Gauss-Legendre quadrature rule (lower part of Table 5.1). We prescribe the pre-defined truncation error tolerances, TOL, in log-variance dimension as 10^{-4}, 10^{-6} and 10^{-8}, respectively. The number of monitoring dates is set to 12 and for N we choose $N = 2^7$.

The results in Table 5.1 demonstrate that when N and J are sufficiently large (like $N = J = 2^7$), the truncation error, governed by "TOL", dominates the overall error; For small values of TOL (like TOL$\leq 10^{-6}$) and N is fixed, a very fast error convergence in J is obtained (and the computational complexity is quadratic in J). As pointed out earlier, with the composite Trapezoidal rule, the calculation of $\tilde{\phi}$ in the initialization phase requires less CPU time than with the non-equidistant Gauss-Legendre rule. From the experiments of Test No. 1, we can conclude that for $q > 0$ both methods give highly accurate results within a fraction of a second.

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

Table 5.1: Convergence in J for Test No.1 ($q = 0.6$) with $N = 2^7, M = 12$ and the European option reference value is 7.5789038982.

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

small values of J. Therefore, we only illustrate the results obtained by the Gauss-Legendre rule in log-variance dimension in Table 5.2.

Table 5.2: Convergence in J as $q \to -1$; Fourier cosine expansion plus Gauss-Legendre rule, $N = 2^8, M = 12$, TOL$= 10^{-7}$, European reference values are 6.2710582179 (Test No. 2) and 13.0842710701 (Test No.3).

$(J = 2^d)$	Test No. 2 ($q = -0.84$)				Test No. 3 ($q = -0.96$)			
	time(sec)				time(sec)			
d	total	Init.	Loop	error	total	Init.	Loop	error
6	3.03	2.85	0.18	5.63	3.11	2.93	0.18	-2.27e+1
7	13.3	12.78	0.56	6.89 10^{-3}	12.1	11.55	0.53	-8.51 10^{-2}
8	56.4	52.32	4.07	-2.12 10^{-5}	55.7	51.74	4.00	-1.60 10^{-3}

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

The results presented in Table 5.2 indicate that, as q approaches -1, the initialization step dominates the overall computational time, in particular the expensive computation of the Bessel function. The computations in the main loop of the pricing algorithm cost less than 8 percent of the total time. So, if we can find a proxy for the Bessel function which can be computed in a cheap way (like the moment matching

based functions in [5]), the overall computation time could be significantly reduced. We leave this for further research.

Next, we examine the error convergence in N, keeping the number of points in log-variance direction, J, fixed. The results are presented in Table 5.3. One can observe that the error convergence is faster than quadratic (the linear increments in CPU time are not shown).

Table 5.3: Convergence in N; COS + Gauss-Legendre, $M = 12$, TOL$= 10^{-7}$, $J = 2^7$ for Test No.1 and $J = 2^8$ for Test No.2.

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

We also check the propagation of the error through time. For this, we fixed N and J and measured the error convergence for increasing values of M (presented in Table 5.4). We employ somewhat different values for J here to indicate that it does not need to be a power of 2. The results confirm that the local error grows only very slowly for $q > 0$ and somewhat faster for $q \in [-1, 0]$. The overall error can be further reduced by setting larger values for J and/or N. Doubling parameter M corresponds to doubling of CPU time in the main loop, which is in accordance with the error analysis.

Table 5.4: Error propagation in M; COS + Gauss-Legendre, TOL$= 10^{-7}$; $N = 2^7, J = 100$ for Test No.1, and $N = 2^8, J = 300$ for Test No.2.

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

5.6.2 Bermudan Options

We will now consider Bermudan options, and use Algorithm 5.3.1 to price them. With increasing values for the number of exercise dates, M, the prices of Bermudan options converge towards the equivalent American options. The M time lattices can be viewed as a discretization in time.

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

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

Two parameter test sets are used here. One is chosen in the PDE-based finite differences literature, for example in [44], with $q > 0$; and the second is with $q \in [-1, 0]$, inspired by results with a tree-based method in [72]. The reference value for the first test case is available and accurate up to the 6th digit, see [44]. For the latter test Bermudan reference values are not available. So we provide our results that may serve as a reference test for future computations by other pricing methods.

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

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

which gives $q > 0$. So, a very accurate and efficient pricing performance is expected from our method.

Results are presented in Table 5.5, where CPU time is measured for five different values of S_0 computed simultaneously. The convergence of the Bermudan options to the American option reference values is clearly visible [2].

Table 5.5: Errors of Test No. 4 ($q = 0.98$); COS + Gauss-Legendre, $N = 2^7, J = 2^7$ and TOL= 10^{-7}

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

A negative correlation coefficient, ρ, is often observed in market data. A test example for a Bermudan put with this parameter and $q \in [-1, 0]$ was given in [72], where the parameters were set as:

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

However, reference values were not available in the paper, so that we provide our results as a reference in Table 5.6.

5.7 Conclusions

In this chapter, we have focused on pricing Bermudan and discretely-monitored barrier options under Heston's stochastic volatility model with a Fourier-based method.

[2] Although it is not our main concern in this paper, one can obtain American option prices much more rapidly by extrapolating prices of Bermudan options with small values of M. Details are given in [54] and Chapter 3.

Table 5.6: Results of Test No. 5 ($q = -0.47$); COS + Gauss-Legendre; $N = 2^8, J = 2^8$ and TOL= 10^{-7}.

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

The near-singular problem in the left-side tail of the Heston variance density has been dealt with by a change of variables to the log-variance domain. An efficient discrete pricing formula is derived by applying a Fourier series expansion technique to the log-stock dimension and a quadrature rule to the log-variance dimension. By means of an error analysis we have determined the various sources for the errors, which are verified by numerical experiments.

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

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

Chapter 6

Conclusions and Outlook

6.1 Conclusions

In this dissertation an option pricing method based on Fourier-cosine series expansions, the COS method, has been presented. The COS method can be applied as long as the characteristic function for the underlying asset price process is available. It is based on the insight that the series coefficients of many probability density functions can be accurately and easily retrieved from their characteristic functions. As such, one can decompose a probability density function into a linear combination of cosine functions. It is this decomposition that makes the numerical computation of the risk-neutral valuation formula highly efficient.

The COS method is efficient not only for recovering density and for pricing European options, but also for pricing a number of different financial products under various asset dynamics, such as Bermudan, American and discretely monitored barrier options as well as credit default swaps under exponential Lévy processes, and options with early-exercise features under the (two-dimensional) Heston stochastic volatility model.

Derivation of the COS method in this thesis has been accompanied by an error analysis. The error convergence is exponential in N, the number of leading terms in Fourier-cosine series expansion, for probability density functions in $\mathrm{C}^{\infty}[a,b]$. This convergence rate has also been confirmed by various numerical experiments. If the density function of the underlying process has a discontinuity in one of its derivatives, an algebraic error convergence is expected and has been observed.

The computational complexity of the COS method is linear in N for European options and almost-linear in N for Bermudan and discretely monitored barrier options. In the latter case, the complexity is $O(M-1)N\log_2(N)$ with M being the number of early-exercise/monitoring dates. One should keep in mind that, for a very small time interval between two consecutive exercise dates, the underlying density function becomes a highly peaked function. Therefore, to get a satisfactory accuracy, a larger value for N is required for cases whereby the time interval is smaller or the number of early-exercise/monitoring dates is bigger. For problems with small time intervals, like daily-monitored barrier options, the COS method shows a similar performance as the Hilbert transform based method by Feng and Linetsky. Compared to the CONV method from Chapter 1, however, the COS method converges significantly faster to

reach the same level of accuracy.

In the calculation of survival/default probabilities for pricing single name Credit Default Swaps, the credit default spreads could be related to a series of survival/default probabilities with different maturities. These survival probabilities can be viewed as prices of binary down-and-out barrier options, without discounting, for which the COS method for the discretely monitored barrier options can be efficiently employed. The method's potential has been demonstrated via the calibration of different exponential Lévy jump models to the quotes of the constituents of the iTraxx Series 7 and Series 8.

The valuation of Bermudan and discrete barrier options under Heston's stochastic volatility model gives rise to a two-dimensional option pricing problem. We have focused on the case for which the problem parameters do not satisfy the Feller condition so that zero variance could be reached in the asset price process. This situation is sometimes called *the near-singular problem* of the left-side tail of the variance. A change of variables to the log-variance domain enables us to price the early-exercise options for all parameter settings including the cases the Feller condition is not satisfied. The derived discrete pricing formula is in essence a combination of the COS formula in the log-asset dimension and a quadrature rule in the log-variance dimension. If the Feller condition is satisfied, the solution method returns highly accurate option prices for early-exercise options within a fraction of a second. For nontrivial parameter settings, the overall computational time is dominated by the computation of a Bessel function (that appears in the joint characteristic function of log-asset and log-variance) in the initialization of the option pricing method. The method's fast error convergence and robustness have also been analyzed and verified via various experiments.

6.2 Outlook

Application of the COS method can be generalized for more asset price processes and/or for more option types.

In the Heston case, we expect an additional significant speed-up when the time-consuming evaluation of the Bessel function, which appears in the characteristic function, can be replaced by a suitable proxy function that can be calculated in much less operations. Promising attempts in this direction from the literature of Monte Carlo simulation methods for Heston dynamics could be helpful here.

Alternative asset price processes other than Lévy processes appear in several option pricing settings. We can think of energy derivative pricing and the use of mean-reverting (non-Lévy) processes. In these cases the COS method for options with early-exercise features has to be adapted to reach the highest efficiency. The same is true for hybrid dynamics, an example of which is the generalization of the Heston stochastic volatility dynamics to, for example, similar processes with a stochastic interest rate. It is a challenge to develop efficient Fourier-cosine based pricing methods for early-exercise options under such processes.

Regarding more option types, we have thoughts in several directions. The generalization to swing options and other *control* problems is already on its way. The resulting methods are based on the COS method for Bermudan options enhanced by the possibility of multiple exercise opportunities within one contract.

Generalization to multi-asset option pricing problems is in principle also possible. One needs to find a relation between the multi-dimensional Fourier-cosine series coefficients and the characteristic function or the joint moment-generating function. Initial ideas in this direction already exist.

Finally, in the field of risk management, the COS method can also be used for the recovery of portfolio loss distributions within and beyond the Vasicek framework, and for the computation of risk measures like value-at-risk (VaR) and expected shortfall (ES) as an alternative for Monte Carlo simulation or sampling-based methods.

These are very interesting issues left for future research in the field of "Computational Finance".

Appendix A

RMSE Results of All Companies of iTraxx

Table A.1: RMSE in basis points of the computed CDSs to market CDSs (part 1)

company name	RMSE in S7		RMSE in S8	
	NIG-BM	CGMY	NIG-BM	CGMY
ABN AMRO Bank NV	0.61	0.55	1.33	1.60
Aegon NV	0.92	0.81	2.34	2.18
Allianz SE	0.76	0.71	0.89	1.13
Assicurazione Generali SPA	0.51	0.43	0.77	0.84
Aviva PLC	0.58	0.51	1.94	2.08
AXA	0.79	0.65	1.35	1.49
Banca Monte dei Paschi di Siena SPA	0.65	0.71	1.62	1.82
Banco Bilbao Vizcaya Argentaria SA	0.60	0.46	1.49	1.46
Banco Espirito Santo SA	0.70	0.66	1.09	1.23
Banco Santander Central Hispano SA	0.70	0.64	1.17	1.34
Barclays Bank PLC	0.48	0.46	1.22	1.32
BNP Paribas	0.38	0.30	1.04	1.07
Capitalia SPA	1.29	1.35	1.44	2.07
Commerzbank AG	0.84	0.62	1.06	1.13
Deutsche Bank AG	0.89	0.77	1.21	1.30
Hannover Rueckversicherung AG	0.61	0.62	1.59	1.61
Intesa Sanpaolo	0.40	0.44	0.27	0.46
Muenchener Rueckversicherung AG	0.59	0.71	0.99	0.95
Swiss Reinsurance Company	1.00	0.87	2.70	2.38
Unicredito Italiano SPA	0.57	0.53	1.33	1.63
Bayerische Motorenwerke AG	0.54	0.55	0.95	1.30
Compagnie Financiere Michelin	0.83	0.97	3.67	2.02
Continental AG	1.63	1.10	1.44	1.54
DaimlerChrysler AG	0.76	0.75	1.10	1.12
GKN Holdings PLC	2.19	1.50	2.53	2.17
Peugeot SA	0.74	0.64	1.72	2.28
Renault	0.80	0.64	2.73	2.56
Valeo	1.32	1.01	1.67	1.43
Volkswagen AG	0.78	0.82	2.86	1.67
Accor	0.83	0.89	2.57	2.07
Aktiebolaget Electrolux	0.69	0.35	1.79	2.20
Altadis SA	1.83	1.45	1.60	1.51
British American Tobacco PLC	0.67	0.29	1.55	1.60
Cadbury Schweppes PLC	0.62	0.55	1.60	1.59

Table A.2: RMSE in basis points of the computed CDSs to market CDSs (part 2)

company name	RMSE in S7		RMSE in S8	
	NIG-BM	CGMY	NIG-BM	CGMY
Carrefour	0.71	0.68	1.35	0.90
Compass Group PLC	0.78	0.66	1.55	0.99
Deutsche Lufthansa AG	0.89	0.68	1.61	1.83
Diageo PLC	0.32	0.32	1.49	1.16
DSG International PLC	0.93	0.72	4.27	3.33
Gallaher Group PLC	0.41	0.52	1.07	0.57
Groupe Auchan	0.44	0.46	1.51	0.90
Experian Finance PLC	0.71	0.52	1.82	1.70
Henkel KGaA	0.74	0.73	1.57	1.32
Kingfisher PLC	1.17	0.70	3.58	3.30
Koninklijke Philips Electronics NV	0.81	0.69	1.40	1.47
LVMH Moet Henessy Louis Vuitton	0.62	0.72	1.47	1.17
Marks and Spencer	1.15	0.89	1.76	2.22
Metro AG	0.52	0.60	1.10	1.18
PPR	1.48	0.78	1.92	1.97
Safeway Ltd	1.09	0.94	1.72	1.49
Sodexho Alliance	0.46	0.42	1.19	1.32
Svenska Cellulosa Aktiebolaget SCA	0.52	0.40	1.69	1.73
Tate & Lyle PLC	0.66	0.94	0.94	1.01
Tesco PLC	0.33	0.44	1.28	0.77
Unilever NV	0.52	0.48	1.50	0.51
Centrica PLC	0.80	1.15	2.14	1.23
Edison SPA	0.40	0.58	0.88	0.74
Enel SPA	0.76	0.75	2.96	3.00
Energie Baden-Wuerttemberg AG	0.46	0.52	1.60	0.99
Fortum Oyj	0.59	0.63	1.70	0.98
Gas Natural SDG SA	0.32	0.45	1.87	1.54
GAZ de France	0.25	0.41	1.43	1.00
Iberdrola SA	0.71	1.97	1.66	1.54
National Grid PLC	1.25	0.99	1.69	1.10
Repsol YPF SA	1.06	1.00	1.36	1.30
RWE AG	0.22	0.41	0.89	0.60
SUEZ	0.46	0.52	0.94	0.76
Union Fenosa SA	0.59	0.64	2.11	1.85
United Utilities PLC	0.32	0.90	0.97	0.88
Vattenfall Aktiebolag	0.36	0.52	1.25	0.80

Table A.3: RMSE in basis points of the computed CDSs to market CDSs (part 3)

company name	RMSE in S7		RMSE in S8	
	NIG-BM	CGMY	NIG-BM	CGMY
Veolia Environnement	1.55	0.97	1.19	1.19
Adecco SA	0.63	0.31	1.71	2.06
Akzo Nobel NV	0.67	0.55	1.67	1.28
Arcelor Finance	0.66	0.38	1.71	1.96
Bayer AG	0.57	1.02	1.42	1.25
Ciba Specialty Chemicals Holding Inc.	1.71	0.74	1.96	2.26
Compagnie de Saint-Gobain	0.77	0.51	2.14	1.68
European Aeronautic Defence and Space Company EADS NV	0.65	0.45	1.42	1.03
Glencore International AG	2.29	1.17	2.55	3.52
Imperial Chemical Industries PLC	0.94	0.54	0.74	0.70
Koninklijke DSM NV	0.97	0.37	1.45	1.51
Lafarge	1.40	1.04	1.77	1.89
Linde AG	0.94	0.91	1.13	1.40
Sanofi-Aventis	0.45	0.45	1.16	0.83
Siemens AG	0.74	1.40	1.21	1.05
Solvay	0.48	1.27	0.95	1.04
ThyssenKrupp AG	1.53	0.84	2.23	2.08
UPM-Kymmene Oyj	1.31	1.06	3.06	2.99
VINCI	0.76	0.49	1.28	1.21
Bertelsmann AG	1.33	1.90	1.36	1.10
British Telecommunications PLC	1.56	1.20	1.62	1.65
Deutsche Telekom AG	1.16	1.24	2.19	1.51
France Telecom	1.31	1.77	1.48	1.38
Hellenic Telecommunications Organisation SA	1.44	1.05	3.54	2.79
Koninklijke KPN NV	2.29	0.64	1.90	3.17
Pearson PLC	1.80	0.81	1.68	1.43
Reuters Group PLC	0.87	1.15	1.12	1.04
STMicroelectronics NV	0.80	0.87	0.97	1.79
Telecom Italia SPA	1.82	1.58	2.55	2.46
Telefonica SA	1.20	0.96	1.59	1.59
Telekom Austria Aktiengesellschaft	0.99	0.77	1.59	1.27
Telenor ASA	0.57	0.90	0.85	0.93
TeliaSonera Aktiebolag	1.05	0.70	1.61	1.23
Vivendi	2.02	1.58	2.37	2.64
Vodafone Group PLC	1.27	1.94	2.02	1.69
Wolters Kluwer NV	1.51	1.43	1.45	1.19

Curriculum vitae

- 2009.10.01 – present, quantitative risk analyst in Modeling and Research Department, Rabobank International, Utrecht, the Netherlands; PhD research in spare time until 2010.06.30.
- 2009.01.01 – 2009.09.30, quantitative analyst in Derivatives & Hedging group, Aegon Asset Management, Den Haag, the Netherlands; PhD research in spare time.
- 2006.06.15 – 2008.12.31, PhD student in Numerical Analysis Group, Applied Mathematics, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, the Netherlands; CICAT scholarship.
  Subject: *The COS Method: An Efficient Fourier Method for Pricing Financial Derivatives.*
  Adviser: Prof.dr.ir. C.W. Oosterlee.
- 2004.03.01 – 2006.06.01, Msc in Computational Engineering, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany; Siemens scholarship.
  Master thesis: Pricing Bermudan and American Options Using the FFT Method.
  Advisor: Prof.dr.ir. C.W. Oosterlee from TU Delft (Netherlands) and Prof. Ulrich Rüde from university Erlangen-Nürnberg (Germany).
- 2001.09.01 – 2004.02.28, Msc in Mechanical Engineering, Shanghai Jiaotong university, Shanghai, China.
- 1998.09.01 – 2001.07.30, Secondary Bachelor degree in Finance, Shanghai Jiaotong university, Shanghai, China.
- 1997.09.01 – 2001.07.30, Bachelor in Mechanical Engineering, Shanghai Jiaotong university, Shanghai, China.
- 1979.10.01, born in Shitanjing town, Shizuishan city, Ningxia province, P.R.China.

List of publications

- Journal papers:
  - F. Fang and C. W. Oosterlee. A Fourier-based valuation method of Bermudan and Barrier options under Heston's model. submitted, 2010.
  - F. Fang, H. Jönsson, W. Schoutens, and C. W. Oosterlee. Fast valuation and calibration of credit default swaps under Lévy dynamics. *J. Comp. Finance*, 14(2):1–30, Winter 2010/11
  - F. Fang and C. W. Oosterlee. Pricing early-exercise and discrete barrier options by Fourier-cosine series expansions. *Numerische Mathematik*, 114(1):27–62, 2009.
  - F. Fang and C. W. Oosterlee. A novel option pricing method based on Fourier-cosine series expansions. *SIAM J. Sci. Comput.*, 31(2):826–848, 2008.
  - R. Lord, F. Fang, F. Bervoets, and C.W. Oosterlee. A fast and accurate FFT-based method for pricing early-exercise options under Lévy processes. *SIAM J. Sci. Comput.*, 30:1678–1705, 2008.
- Proceedings:
  - F. Fang and C.W. Oosterlee. Pricing options under stochastic volatility with Fourier Cosine expansions. Progress in Industrial Mathematics at ECMI 2008, Mathematics in Industry, 15(3): 833-838, Springer Berlin 2010.
  - R. Lord, F. Fang, F. Bervoets and C.W. Oosterlee. 'The CONV method for pricing options'. Proc. Appl. Math. Mech. ICIAM2007, 7(1): 1024003-1024004,2008.
  - C.W. Oosterlee., M. Muskulus, K. in't Hout, J. Bierkens, A.P.C. Ploeg, J. in't Panhuis, F. Fang and B. Janssens. The ING problem: a problem from the financial industry. Proc. 58th Europ. Study Group Math. with Industry Utrecht, 91-115, 2007.
  - R. Lord, F. Fang, F. Bervoets and C.W. Oosterlee. A fast method for pricing early-exercise options with the FFT. Computational Science - ICCS 2007, Lect. Notes in Comp. Science, 4488, 415-422, Springer Berlin, 2007.

Attended Conferences

- Third Conference on Numerical Methods in Finance, 15-17 April 2009, Paris, France. (On invitation)
  http://cermics.enpc.fr/cnf.htm
- Quantitative Methods in Finance, 17-20 December 2008, Sydney, Australia.
  http://www.qfrc.uts.edu.au/qmf/2008/index.html
- 2nd International Conference on Numerical Methods for Finance, 4-6 June 2008, Dublin, Ireland.
  http://www.numericalmethodsforfinance.org/
- Workshop Numerics of Finance, 5-6 November 2007, Commerzbank-Hochhaus, Frankfurt a.M., Germany.
  http://www.iwr.uni-heidelberg.de/groups/techsim/websites/Numerics_of_Finance/
- ICIAM 07: 6th International Congress on Industrial and Applied Mathematics, 16-20 July 2007, Zurich, Switzerland.
  http://www.iciam07.ch/poster
- ICCS 2007: Advancing Science and Society through Computation, 27-30 May 2007, Beijing, China.
  http://www.iccs-meeting.org/iccs2007/
- The SIAM Workshop on Combinatorial Scientific Computing (CSC07), 19-23 February 2007, Costa Mesa, California, USA. (On invitation)
  http://www.siam.org/meetings/cse07/
- Workshop on Financial Modeling with Jump Processes, 6-8 September 2006, Ecole Polytechnique, Palaiseau, France.
  http://www.fiquam.polytechnique.fr/AMAMEF/

Bibliography

[1] A. Almendral and C. W. Oosterlee. Accurate evaluation of European and American options under the CGMY process. *SIAM J. Sci. Comput.*, 29:93–117, 2007.

[2] A. Almendral and C. W. Oosterlee. On American options under the Variance Gamma process. *Appl. Math. Finance*, 14(2):131–152, 2007.

[3] D. E. Amos. A subroutine package for Bessel functions of a complex argument and nonnegative order. *Sandia National Laboratory Report*, 1985. SAND85-1018.

[4] D. E. Amos. A portable package for Bessel functions of a complex argument and nonnegative order. *Trans. Math. Software*, 1986.

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

[6] A. D. Andricopoulos, M. Widdicks, P. W. Duck, and D. P. Newton. Universal option valuation using quadrature methods. *J. Fin. Economics*, 67:447–471, 2003.

[7] A. D. Andricopoulos, M. Widdicks, P. W. Duck, and D. P. Newton. Extending quadrature methods to value multi-asset and complex path dependent options. *J. Fin. Economics*, 83(2):471–500, 2007.

[8] O. E. Barndorff-Nielsen. Normal inverse Gaussian distributions and stochastic volatility modelling. *Scand. J. Statist.*, 24(1-13), 1997.

[9] N. Beliaeva and S. Nawalkha. A simple approach to pricing American options under the Heston stochastic volatility model. 2010. working paper.

[10] C. M. Bender and S. A. Orszag. *Advanced Mathematical Methods for Scientists and Engineers*. McGraw–Hill, New York, 1978.

[11] F. Black and J. Cox. Valuing corporate securities: some effects on bond indenture provisions. *The Journal of Finance*, 31:351–367, 1976.

[12] S. I. Boyarchenko and S. Z. Levendorskiĭ. Non-Gaussian Merton-Black-Scholes theory. *Advanced Series on Statist. Science & Appl. Probability.*, 9, 2002.

[13] J. P. Boyd. *Chebyshev & Fourier Spectral Methods*. Springer-Verlag, Berlin, 1989.

[14] M. Broadie and O. Kaya. Exact simulation of stochastic volatility and other affine jump diffusion processes. 2004. Columbia University.

[15] M. Broadie and Y. Yamamoto. Application of the fast Gauss transform to option pricing. *Management Sci.*, 49:1071–1008, 2003.

[16] M. Broadie and Y. Yamamoto. A double-exponential fast Gauss transform for pricing discrete path-dependent options. *Operations Research*, 53(5):764–779, 2005.

[17] J. Cariboni. *Credit Derivatives Pricing under Lévy Models.* PhD Thesis, K.U.Leuven, Leuven, Belgium., 2007.

[18] J. Cariboni and W. Schoutens. Pricing credit default swaps under Lévy models. *J. Comp. finance*, 10(4):71–91, 2008.

[19] P. P. Carr, H. Geman, D. B. Madan, and M. Yor. The fine structure of asset returns: An empirical investigation. *J. Business*, 75:305–332, 2002.

[20] P. P. Carr and D. B. Madan. Option valuation using the fast Fourier transform. *J. Comp. Finance*, 2:61–73, 1999.

[21] C-C Chang, S-L Chung, and R. C. Stapleton. Richardson extrapolation technique for pricing American-style options. *J. Futures Markets*, 27(8):791–817, 2007.

[22] K. Chourdakis. Option pricing using the fractional FFT. *J. Comp. Finance*, 8(2):1–18, 2004.

[23] R. Cont and P. Tankov. *Financial Modelling with Jump Processes.* Chapman and Hall, Boca Raton, FL, 2004.

[24] R. Cont and P. Tankov. Retrieving Lévy processes from option prices: Regularization of an ill-posed inverse problem. *SIAM Journal on Optimization*, 45(1):1–25, 2006.

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

[26] M. A. H. Dempster and S. S. G. Hong. Spread option valuation and the Fast Fourier Transform. *Techn. Rep. WP 26/2000, the Judge Inst. Manag. Studies, Univ. Cambridge*, 2000.

[27] D. Duffie, D. Filipovic, and W. Schachermayer. Affine processes and applications in finance. *Ann. of Appl. Probab.*, 13(3):984–1053, 2003.

[28] D. Duffie, J. Pan, and K. J. Singleton. Transform analysis and asset pricing for affine jump-diffusions. *Econometrica*, 68:1343–1376, 2000.

[29] G. A. Evans and J. R. Webster. A comparison of some methods for the evaluation of highly oscillatory integrals. *J. of Comp. Applied Math.*, 112:55–69, 1999.

[30] A. Eydeland. A fast algorithm for computing integrals in function spaces: financial applications. *Computational Economics*, 7(4):277–285, 1994.

[31] F. Fang, H. Jönsson, C. W. Oosterlee, and W. Schoutens. Fast valuation and calibration of credit default swaps under Lévy dynamics. *J. Comp. Finance*, 14(2):1–30, Winter 2010/11.

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

[33] F. Fang and C. W. Oosterlee. Pricing early-exercise and discrete barrier options by Fourier-cosine series expansions. *Numerische Mathematik*, 114(1):27–62, 2009.

[34] F. Fang and C. W. Oosterlee. A Fourier-based valuation method of Bermudan and Barrier options under Heston's model. 2010. submitted.

[35] W. Feller. Two singular diffusion problems. *Annals of Mathematics*, 54:173–182.

[36] L. Feng and V. Linetsky. Pricing discretely monitored barrier options and defaultable bonds in Lévy process models: a fast Heston transform approach. *Math. Finance*, 18(3):337–384, 2008.

[37] R. Geske and H. Johnson. The American put valued analytically. *J. of Finance*, 39:1511–1542, 1984.

[38] J. Gil-Palaez. Note on the inversion theorem. *Biometrika*, 38:481–482, 1951.

[39] E. G. Haug. Barrier put-call transformations. Working paper Tempus Financial Engineering.

[40] E. G. Haug. *The complete guide to option pricing formulas*. McGraw-Hill, 1998.

[41] S. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Rev. Financ. Studies*, 6:327–343, 1993.

[42] A. Hirsa and D. B. Madan. Pricing American options under Variance Gamma. *J. Comp. Finance*, 7, 2004.

[43] S. Howison. A matched asymptotic expansions approach to continuity corrections for discretely sampled options. part 2: Bermudan options. *Appl. Math. Finance*, 14(1):91–104, 2007.

[44] 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.

[45] K. Jackson, S. Jaimungal, and V. Surkov. Option pricing with regime switching Lévy processes using Fourier space time-stepping. *Proc. 4th IASTED Intern. Conf. Financial Engin. Applic.*, pages 92–97, 2007.

[46] H. Jönsson and W. Schoutens. Pricing constant maturity credit default swaps under jump dynamics.

[47] H. Jönsson and W. Schoutens. Single name credit default swaptions meet single sided jump models. *Review of Derivatives Research*, 11(1-2):153–169, 2009.

[48] S. G. Kou. A jump diffusion model for option pricing. *Management Science*, 48(8):1086–1101, 2002.

[49] S. G. Kou. On pricing of discrete barrier options. *Statistica Sinica*, 13:955–964, 2003.

[50] S. G. Kou and H. Wang. First passage times of a jump diffusion process. *Advances in Applied Probability*, 35:504–531, 2003.

[51] S. G. Kou and H. Wang. Options pricing under a double exponential jump-diffusion model. *Management Science*, 50:1178–1192, 2004.

[52] A. Lewis. A simple option formula for general jump-diffusion and other exponential Lévy processes. SSRN working paper,`http//ssrn.com/abstract=282110`, 2001.

[53] A. Lipton. Assets with jumps. *Risk*, pages 149–153, September 2002.

[54] R. Lord, F. Fang, F. Bervoets, and C.W. Oosterlee. A fast and accurate FFT-based method for pricing early-exercise options under Lévy processes. *SIAM J. Sci. Comput.*, 30:1678–1705, 2008.

[55] R. Lord and C. Kahl. Optimal Fourier inversion in semi-analytical option pricing. *J. Comp. Finance*, 10:1–30, 2007.

[56] R. Lord and Ch. Kahl. Complex logarithms in Heston-like models. *Mathematical Finance*, 2010. To appear.

[57] D. B. Madan, P. R. Carr, and E. C. Chang. The Variance Gamma process and option pricing. *European Finance Review*, 2:79–105, 1998.

[58] D. B. Madan and W. Schoutens. Break on through to the single side. *The Journal of Credit Risk*, 4(3):3–20, 2007.

[59] R. Merton. Option pricing when underlying stock returns are discontinuous. *J. Financial Economics*, 3:125–144, 1976.

[60] M. Mori and M. Sugihara. The double-exponential transformation in numerical analysis. *J. Comput. Appl. Math.*, 127:287–296, 2001.

[61] C. O'Sullivan. Path dependent option pricing under Lévy processes. EFA 2005 Moscow Meetings Paper, `http://ssrn.com/abstract=673424`, 2005.

[62] R. Piessens and F. Poleunis. A numerical method for the integration of oscillatory functions. *BIT*, 11:317–327, 1971.

[63] S. Raible. *Lévy processes in finance: Theory, numerics and empirical facts*. PhD Thesis, Inst. für Math. Stochastik, Albert-Ludwigs-Univ. Freiburg, 2000.

[64] E. Reiner. Convolution methods for path-dependent options. Financial Math. workshop, IPAM UCLA, Jan. 2001.

[65] Scheldon M. Ross. *Introduction to Probability Models*. Academic Press; 6 edition, 1997.

[66] K-I. Sato. Basic results on Lévy processes. *Theory and Appliccations*, pages 3–37, 2001.

[67] W. Schoutens. *Lévy processes in finance: Pricing financial derivatives.* Wiley, 2003.

[68] W. Schoutens, E. Simons, and J. Tistaert. A perfect calibration! now what? *Wilmott Magazine*, pages 66–78, March 2004.

[69] K. J. Singleton and L. Umantsev. Pricing coupon-bond options and swaptions in affine term structure models. *Math. Finance*, 12(4):427–446, 2002.

[70] V. Surkov. Parallel option pricing with Fourier space time-stepping method on graphics processing units. *Parallel Computing*, Feb. 2010. To appear.

[71] N. Taleb. *Dynamic Hedging.* John Wiley & Sons, New York, 2002.

[72] M. Vellekoop and H. Nieuwenhuis. A tree-based method to price American options in the Heston model. *J. Comp. Finance*, 13(1), Fall 2009.

[73] I. Wang, J. W. Wan, and P. Forsyth. Robust numerical valuation of European and American options under the CGMY process. *J. Comp. Finance*, 10(4):31–70, 2007.

[74] P. Wilmott. *Derivatives: The theory and practice of financial engineering.* Wiley Frontiers in Finance Series, 1998.

[75] Y. Yamamoto. Double-exponential fast Gauss transform algorithms for pricing discrete lookback options. *Publ. Res. Inst. Math. Sci.*, 41:989–1006, 2005.