

SOLVING MULTIVARIATE EXPECTATIONS USING DIMENSION-REDUCED FOURIER-COSINE SERIES EXPANSION AND ITS APPLICATION IN FINANCE

MSC THESIS

Delft University of Technology



SOLVING MULTIVARIATE EXPECTATIONS USING DIMENSION-REDUCED FOURIER-COSINE SERIES EXPANSION AND ITS APPLICATION IN FINANCE

MSC THESIS

by

Marnix BRANDS

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on March 22, 2023 at 11:00.

Student number:	4588665	
Project duration:	July 25, 2022 - March 22, 2023	
Thesis committee:	Dr. F. Fang,	TU Delft, supervisor
	Prof.Dr.ir. C. Vuik,	TU Delft
	Prof.Dr. A. Papapantoleon,	TU Delft
	Dr. X. Shen,	FFQuant, supervisor
Institution:	Delft University of Technology, Faculty of Electrical Engineering, Mathematics & Computer Science (EEMCS)	
Master programme:	Applied Mathematics	
Specialisation:	Financial Engineering	

An electronic version of this dissertation is available at
<http://repository.tudelft.nl/>.



ABSTRACT

The computation of multivariate expectations is a common task in various fields related to probability theory. This thesis aims to develop a generic and efficient solver for multivariate expectation problems, with a focus on its application in the field of quantitative finance, specifically for the quantification of Counterparty Credit Risk (CCR).

The proposed COS-CPD method utilizes the COS method to recover the exposure distribution by its Fourier-cosine series expansion, from which measures such as the PFE and EE can be obtained. The key insight is that the corresponding Fourier coefficients are readily available from the characteristic function, which can be solved using numerical integration methods. However, the efficiency of standard quadrature rules is limited to only a few risk factors, as the dimension of integration is determined by the number of risk factors involved.

To address this limitation, the COS-CPD method reduces the dimension of integration of the characteristic function through two steps. Firstly, the joint density function of the risk factors in the characteristic function is replaced by a dimension-reduced Fourier-cosine series expansion, which is obtained through CPD. With CPD, the computational complexity of computing the Fourier coefficient tensor is reduced to a linear growth with respect to the number of dimensions. Secondly, the portfolio is divided into segments that share the same risk factors. These two steps reduce the evaluation of the characteristic function to the calculation of only one- and two-dimensional integrals, which are solved by the Clenshaw-Curtis quadrature rule. As a result, the COS-CPD method is suitable for portfolios with more than three risk factors.

Numerical comparisons of the COS-CPD method and Monte Carlo (MC) method are made for netting-set PFE and EE profiles of multiple derivative portfolios up to five risk factors. For similar accuracy levels, the COS-CPD method greatly outperforms the Monte Carlo method in computation time. This difference increases for larger portfolios, which makes the COS-CPD method a much more efficient alternative for the MC method, especially for large portfolios.

Furthermore, the COS-CPD method is applied in the context of multi-asset option pricing. A six-dimensional basket option is considered, and the results are compared to a recently developed sparse grid method. The comparison shows that the COS-CPD method outperforms the sparse grid method in both accuracy and computation time. Moreover, the COS-CPD method allows the computation of the option value for multiple strike prices simultaneously, with no significant additional computational cost.

Keywords: multivariate expectations, COS method, Monte Carlo simulation, Fourier-cosine series expansion, canonical polyadic decomposition, counterparty credit risk, potential future exposure, expected exposure, multi-asset option pricing.

PREFACE

This thesis marks the end of my academic journey at the TU Delft. It began in 2016 when I started my Bachelor in Applied Mathematics. Initially, my understanding of mathematics was limited to solving simple equations by hand, with the expectation that every problem could be solved with elegant analytic derivations. Through my six years of studies, I have come to realize that mathematics is much more than just solving equations by hand. It is an indispensable tool in today's world and all its practical applications. The knowledge and skills I have gained throughout my study in modeling, analyzing, and programming numerous mathematical problems has formed the foundation for this thesis.

I would like to express my gratitude to my supervisors, Dr. Fang and Dr. Shen, who provided me with the opportunity to carry out this project at FF Quant. I am grateful for their continuous feedback, insightful comments, and suggestions, which helped me improve my research skills and enabled me to complete my thesis successfully.

Next, I would like to thank the other students who worked at FF Quant during my thesis project. There was a pleasant working atmosphere and it was nice to have discussions regarding our respective projects, as well as the breaks and lunch walks that provided moments of relaxation. I am particularly thankful to Gijs and Zhimin, who introduced me to my thesis' topics and were always available to answer the questions I had. This helped me to have a good understanding of the various aspects of the project right from the beginning of my thesis.

Of course, the support of my friends throughout the thesis has been invaluable. It was really nice to have the necessary distraction from the hard work of the thesis by drink a beer or some other activities. I especially want to thank Rogier, whose mental support and companionship were a most welcome distraction from the demands of the thesis. From all the short coffee breaks to watching complete seasons of RuPaul's Drag Race together, it provided me with the energy and motivation I needed to persevere through the challenges of my thesis. Additionally, I deeply appreciated the preparation of dinners for me when I came home late from the office.

Finally, I wouldn't be here today with the completion of this Master's thesis without the continuous support of my parents and brother throughout my study time. I want to thank them for all the valuable life advice, all the taxi rides to Delft, all the vitamins I could catch up on in the weekends, and much more than I can write down in these few lines.

*Marnix Brands
Delft, March 2023*

LIST OF FIGURES

3.1	Fibers of a third-order tensor [56]	30
4.1	Reconstructed Fourier-cosine series of the density function of a standard normal random variable using the CPD algorithm in the scenario $K < m$. The training data points used to construct the CPD approximation are indicated with the black dots. For this example, $K = 10$ and $m = 30$ were used.	40
4.2	Reconstructed Fourier-cosine series of the density function of a standard normal random variable using the CPD algorithm in the scenario $K > m$. The training data points used to construct the CPD approximation are indicated with the black dots. For this example, $K = 60$ and $m = 30$ were used.	41
4.3	Reconstructed Fourier-cosine series of the density function of a standard normal random variable using the CPD algorithm in the scenario $K = m$. The training data points used to construct the CPD approximation are indicated with the black dots. For this example, $K = m = 30$ was used.	41
4.4	L^∞ error of the Fourier-cosine coefficients for the standard normal density, computed using CPD with different sampling frequencies of the training data.	42
4.5	Error in the first Fourier coefficient α_0 using the CPD algorithm for different sampling frequencies. The functions $f(x)$, $x^2 \cdot f(x)$ and $e^x \cdot f(x)$, $\cos(x) \cdot f(x)$ and $\cos(2x) \cdot f(x)$ are considered, with $f(x)$ denoting the standard normal density function. The tolerance level in the expansion interval $[a, b]$ is set to 10^{-12}	43
4.6	Error in the CPD approximation of the Fourier-cosine series for the standard normal distribution. The number of training data points is set to $M = 50$. The error in the Fourier coefficients is defined as $\ \alpha_k - \hat{\alpha}_k\ _\infty$	45
4.7	L^∞ error in the CPD approximation of the Fourier-cosine series for the standard normal density function in the scenario $K = m$. The error is presented for the (non-) training data and the Fourier coefficients.	45
4.8	Error in the CPD approximation of the Fourier-cosine series for the bivariate normal density function. The error is presented for the (non-) training data and the Fourier coefficients.	47
4.9	Impact of the correlation level on the error in the first Fourier coefficient α_0 for different rank. In (a) different levels of (positive) correlation are used, while the difference between positive and negative correlation of the same order is presented in (b). For all tests we took $K = m = 30$	48

4.10	Impact of insufficient iterations used in the CG solver when computing A_1 and A_2 . The error in the first Fourier coefficient is presented in (a), while (b) shows the number of CG iterations needed for an accuracy in the relative residual of 10^{-12} . Parameters are set to $K = m = 30$ and $\rho = 0.3$	49
4.11	Computation time of the CPD algorithm for different rank to construct factor matrices A_1 and A_2	50
4.12	Application of the CPD method on the standard normal density function for dimension 1 up to 6. The error in the first Fourier coefficient for different tolerance levels (a) and the computation time in each dimension (b) are shown. Parameter values used for this test are $K = m = 12$ and $R = 1$	51
4.13	Error in the first Fourier coefficient and computation time using CPD on the six-dimensional normal density. For this test we have $K = m$ and $R = 1$	51
4.14	Error in the first Fourier coefficient of the four-dimensional normal density function for different number of correlated random variables. Parameters are $K = m = 12$ and all correlations are set to 10%.	52
7.1	Correlation between state variables Z_d , Z_f , and Z_X over a period of 15 years.	78
7.2	Recovered PDF and CDF of the MtM value and the exposure of the XCS at time $t = 3.4$ using the COS-CPD method.	79
7.3	The PFE and EE profile of the XCS computed using both the COS-CPD method and a MC benchmark using $5 \cdot 10^6$ simulations. The PFE and EE values are computed for 50 distinct time points.	80
7.4	Recovered CDF of the MtM value of the XCS at time $t = 9.9$ for different COS support $[l, u]$, together with the PFE profile when using different COS support for $t > 9.0$	81
7.5	PFE and EE profile at a netting-set level for a portfolio consisting of 100 derivatives, computed at 50 equidistant points.	82
7.6	The PFE and EE profile of the netting-set level exposure the portfolio consisting of 100 derivatives for the first 3 years, together with the relative error in the resulting PFE and EE values (as a percentage of the MC benchmark).	83
7.7	The PFE and EE profile of two XCS derivatives computed using both the COS-CPD method and a MC benchmark using $5 \cdot 10^6$ simulations. The profile is generated using 20 equidistant time points up to $t = 3.0$	88
7.8	PFE and EE profile at a netting-set level for a portfolio consisting of 100 derivatives, computed at 50 equidistant time points. The MC benchmark was generated using $5 \cdot 10^6$ simulated paths.	89
7.9	The PFE and EE profile of the netting-set level exposure the portfolio consisting of 100 derivatives for the first 3 years, together with the relative error in the resulting PFE and EE values (as a percentage of the MC benchmark).	89
8.1	Recovered payoff density of the six-dimensional basket put option with the COS-CPD method, utilizing $N = 64$ expansion terms in the COS approximation. Regarding the option, the parameters used were $r = 0$, $T = 1$, $K = 100$, $\sigma_1 = (0.4, 0.4, 0.4, 0.4, 0.4, 0.4)$ and $S_i(0) = 100$, $i = 1, \dots, 6$	98

-
- 8.2 Value of the six-dimensional basket put option for the strike vector $\mathbf{K} = (75, 76, \dots, 100)$. The volatility parameters are $\sigma_2 = (0.2, 0.3, 0.4, 0.5, 0.6, 0.7)$ and $N = 64$ COS expansion terms are utilized. 100

LIST OF TABLES

4.1	Order of the Fourier-cosine series truncation error of the bivariate normal density function for different correlation levels, in case $K = 30$ expansion terms per dimension are used.	48
7.1	Maximum and time-averaged relative error in the PFE and EE computation of the XCS using the COS-CPD method for different rank, using a MC benchmark of $5 \cdot 10^6$ paths. The PFE and EE values are computed for 50 distinct time points.	80
7.2	The accuracy and CPU time (in seconds) required to calculate the PFE and EE of netting-set level exposure of the portfolio with 100 derivatives, evaluated at 50 time points up to time $t = 3.0$. The time-averaged error was calculated by averaging the relative errors over the 50 time points.	83
7.3	Comparison of the CPU time (in seconds) needed to compute the PFE and EE profile for a netting-set portfolio with 10, 20 and 50 time steps, respectively. The MC measurements were performed with $5 \cdot 10^5$ paths.	84
7.4	Relative error in the PFE and EE computation of two XCS products up to time $t = 3.0$ with 20 time steps using the COS-CPD method for different rank, using a MC benchmark of $5 \cdot 10^6$ paths.	88
7.5	The accuracy and CPU time (in seconds) required to calculate the PFE and EE of netting-set-level exposure of the portfolio with 100 derivatives evaluated at 50 time points up to time $t = 3.0$. The time-averaged error was calculated by averaging the relative differences over the 50 time points. . .	90
7.6	Comparison of the CPU time (in seconds) needed to compute the PFE and EE profile for a netting-set portfolio with 10, 20 and 50 time steps, respectively. For the MC method, $5 \cdot 10^5$ paths are considered	91
8.1	Comparison in computation time (in seconds) and accuracy of the COS-CPD method with the sparse grid method from [35] in pricing a six-dimensional basket put option with volatility parameters $\sigma_1 = (0.4, 0.4, 0.4, 0.4, 0.4)$. The relative error for both methods is derived by using a Monte Carlo benchmark utilizing 10^9 paths.	99
8.2	Comparison in computation time (in seconds) and accuracy of the COS-CPD method with the sparse grid method from [35] in pricing a six-dimensional basket put option with volatility parameters $\sigma_2 = (0.2, 0.3, 0.4, 0.5, 0.6, 0.7)$. The relative error for both methods is derived by using a Monte Carlo benchmark utilizing 10^9 paths.	99

8.3	Computation time (in seconds) and accuracy of the COS-CPD method to price the six-dimensional basket put option for the strike price vector $\mathbf{K} = (75, 76, \dots, 100)$. The volatility parameters are $\sigma_2 = (0.2, 0.3, 0.4, 0.5, 0.6, 0.7)$ and the relative error is derived by using a Monte Carlo benchmark utilizing 10^8 paths.	99
-----	--	----

LIST OF ABBREVIATIONS

- ALS** Alternating Least Squares 33
- CCR** Counterparty Credit Risk 1
- CDF** Cumulative Distribution Function 26
- CG** Conjugate Gradient 37
- CPD** Canonical Polyadic Decomposition 29
- CVA** Credit Value Adjustment 2
- DVA** Debit Value Adjustment 15
- EE** Expected Exposure 2
- FDMC** Finite Difference Monte Carlo 6
- FRA** Forward-Rate Agreement 17
- FSA-HTF** Fourier Series Approximation via Hidden Tensor Factorization 37
- FX** Foreign Exchange 20
- G1++** Gaussian One Factor 19
- GBM** Geometric Brownian Motion 22
- GPR** Gaussian Process Regression 6
- HOSVD** Higher-Order Singular Value Decomposition 29
- IRS** Interest Rate Swap 18
- LSMC** Least Squares Monte Carlo 5
- MC** Monte Carlo 2
- MC-Tree** Monte Carlo-Tree 5
- MtM** Mark-to-Market 2
- OTC** Over-The-Counter 2
- PDF** Probability Density Function 25
- PFE** Potential Future Exposure 2

- PFS** Payer IRS 18
- ppf** percent-point function 42
- RFS** Receiver IRS 18
- SGBM** Stochastic Grid Bundling Method 5
- SGD** Stochastic Gradient Descent 37
- SPD** symmetric, positive definite 44
- WWR** Wrong-Way Risk 6
- XCS** Cross-Currency Swap 21
- ZCB** Zero-coupon Bond 16

CONTENTS

Abstract	iii
Preface	v
List of Figures	vii
List of Tables	xi
List of Abbreviations	xiii
1 Introduction	1
1.1 Multivariate Expectations	1
1.2 Introduction to Counterparty Credit Risk	2
1.3 Numerical Integration	3
1.4 Literature Review	5
1.4.1 Counterparty Credit Risk.	5
1.4.2 Option Pricing	7
1.5 Thesis Objective and Outline	8
2 Mathematical Framework	11
2.1 Stochastic Calculus	11
2.2 CCR Quantification	13
2.2.1 Exposure Profiles.	13
2.2.2 Credit Value Adjustment (CVA).	14
2.3 Interest Rate Framework	16
2.3.1 Interest Rate Definitions and Products	16
2.3.2 Interest Rate Models	18
2.4 Foreign Exchange Framework.	20
2.4.1 FX Products	21
2.4.2 FX Rate Model	22
2.5 Option Basics	22
2.6 The COS Method	25
3 Canonical Polyadic Decomposition (CPD)	29
3.1 Tensor Definitions and Notation	29
3.2 Introduction to CPD	32
3.3 Our contribution 1: Dimension-reduced Fourier-cosine Series Expansions via CPD	33
3.4 Hidden Tensor Factorization	36

4	Convergence Tests for CPD	39
4.1	Prevent Overfitting	39
4.2	Sampling Frequency Training Data	41
4.3	Sensitivity Analysis	43
4.3.1	One-dimensional Case	44
4.3.2	Two-dimensional Case	46
4.3.3	Multi-dimensional Case	49
5	Our Novel Method: The COS-CPD Method	53
5.1	Solving Multivariate Expectations	54
5.2	Computing Potential Future Exposure	55
5.3	Computing Expected Exposure	60
6	Error Analysis of the COS-CPD Method	63
6.1	Convergence of Fourier-cosine Series	63
6.2	Error Analysis CPD	65
6.2.1	Error Sources	65
6.2.2	Theoretical Error Bounds	65
6.3	Error Analysis COS Method	67
6.3.1	Error Sources	67
6.3.2	Theoretical Error Bounds	68
7	Application of COS-CPD in Counterparty Credit Risk Quantifications	73
7.1	Three-dimensional Case	74
7.1.1	Model	74
7.1.2	Numerical Results	77
7.2	Five-dimensional Case	85
7.2.1	Model	85
7.2.2	Numerical results	86
8	Application of COS-CPD in Multi-dimensional Option Pricing	93
8.1	Six-dimensional Basket Option	93
8.1.1	Model	93
8.1.2	Numerical Results	97
9	Conclusions and Future Research	101
	Bibliography	104
A	Additional Literature Review	113
A.1	Wrong-Way Risk	113
B	Mathematical Proofs	115
B.1	Solution Ornstein-Uhlenbeck Process	115
B.2	Zero-Coupon Bond Formula under the G1++ model	116
B.3	Correlation Structure of State Variables	119

C	CPD as Direct Integration Method	123
D	Clenshaw-Curtis Quadrature	127
E	Portfolios	129
E.1	Portfolio 100 derivatives 3D	130
E.2	Portfolio 50 derivatives 3D	133
E.3	Portfolio 25 derivatives 3D	135
E.4	Two XCS 5D	136
E.5	Portfolio 100 derivatives 5D	137
E.6	Portfolio 50 derivatives 5D	140
E.7	Portfolio 25 derivatives 5D	142

1

INTRODUCTION

1.1. MULTIVARIATE EXPECTATIONS

The computation of multivariate expectations is a common occurrence in various disciplines related to probability theory, including game theory, time-series analysis, data sciences and weather forecasting. Also in the field of quantitative finance, many problems can often be formulated in terms of the expectation of a function that depends on multiple random variables. These random variables then correspond to the stochastic processes found in the financial markets, such as interest rates, exchange rates and stock prices. The calculation of the multivariate expectation involves the evaluation of a multi-dimensional integral. To illustrate this, consider a multivariate random variable $\mathbf{X} = (X_1, \dots, X_N)$ with corresponding joint density function $f(\cdot)$. The multivariate expectation of the target function $g(\cdot)$ under a certain probability measure is defined as

$$\mathbb{E}[g(\mathbf{X})] = \int_{\mathbb{R}^N} g(x_1, \dots, x_N) f(x_1, \dots, x_N) dx_1 \cdots dx_N.$$

Consequently, solving multivariate expectations is equivalent to the task of evaluating multi-dimensional integrals. As analytical solutions often do not exist, numerical integration methods are employed to approximate the multivariate expectation. However, standard numerical integration techniques are optimized for one- or two-dimensional problems, but can struggle with the high-dimensional situation, as they suffer from the so-called "curse of dimensionality", resulting in an exponential increase in the computational costs of the methods. This leads to impracticable computation times and potential memory overflow issues. Therefore, much effort has been put into improving the performance of numerical integration methods in high-dimensional contexts, which will be discussed in more detail in the literature review presented in Section 1.4.

This thesis aims to develop a generic and efficient solver for multivariate expectation problems, with a focus on its application in the field of quantitative finance, specifically for the quantification of [Counterparty Credit Risk \(CCR\)](#). To demonstrate the versatility of the method, a secondary example in the context of multi-asset option pricing will also be presented.

1.2. INTRODUCTION TO COUNTERPARTY CREDIT RISK

Quantifying and managing counterparty credit risk is one of the most challenging tasks nowadays in financial institutions. According to the Basel Committee on Banking Supervision, CCR is defined as the risk that the counterparty to a transaction could default before the final settlement of the transaction's cash flows [1]. When the counterparty defaults, the non-defaulting party suffers possible losses due to the uncompleted payments of contractual cash flows. CCR only plays a role in the **Over-The-Counter (OTC)** derivatives, which are contracts privately negotiated between counterparties. Derivatives traded on the regular exchange are generally not affected by CCR, as the exchange guarantees the cash flows promised by the derivatives to the counterparties.

Prior to the credit crisis in 2007, CCR was not considered to be a significant factor when trading OTC. The larger derivative counterparties were perceived as being too big to fail, and CCR was often overlooked due to their high credit ratings. For many years, it was common practice in the industry to value OTC derivative products without considering CCR, assuming that the likelihood of counterparty default was negligible.

However, during the 2007 financial crisis, numerous corporate collapses occurred, including large financial institutions such as Lehman Brothers. Consequently, market participants became aware that the "too big to fail" mentality was not applicable, and CCR should be taken into account when valuing OTC contracts for any financial entity. After the crisis, new regulations were established with regards to the modelling and quantification of CCR, which financial institutions are obligated to adhere appropriately. To this day, the development of effective CCR models remains a key priority for financial institutions.

The first step in quantifying and managing CCR is the computation of exposure profiles for the OTC derivatives. Exposure can be seen as the maximal amount of money that can be lost if the default of a counterparty occurs. The future exposure of a financial contract cannot be predicted with certainty as it is dependent on the behavior of uncertain risk factors, such as interest rates and exchange rates. Therefore, only a distribution of the **Mark-to-Market (MtM)** price of the portfolio can be obtained for future time points based on stochastic models of the risk factors. Mathematically, exposure is defined as the positive part of this MtM price distribution. This exposure distribution then indicates the probability of which a certain exposure value occurs.

Once the exposure distribution is computed, different exposure measures can be quantified. Two important exposure measures, which are also incorporated in the Basel II regulations, are the **Expected Exposure (EE)** and **Potential Future Exposure (PFE)**. The EE is the expectation of the exposure distribution, while the PFE represents the 97.5 % quantile of the exposure distribution. The Basel III regulations introduce a new measure for CCR called **Credit Value Adjustment (CVA)**. CVA is defined as the difference between the risk-neutral valuation of a financial derivative contract and its valuation that considers the possibility of counterparty default. In other words, CVA is the market value of CCR [2]. The computation of CVA also depends on exposure profiles, i.e. the term-structure of EE, driving the need of the industry for fast and accurate computation methods of these exposure quantities.

The standard method in the industry to compute exposure profiles, if not the only method, is the **Monte Carlo (MC)** simulation. With this method, future values of the

risk factors are simulated according to pre-selected stochastic models. These simulated scenarios are then used to price all the derivatives in the portfolio at a future time point, resulting in an approximation of the total exposure distribution of the portfolio. Once this exposure distribution is computed, CCR measures like PFE or CVA can be quantified. MC is a robust and flexible method, suitable for portfolios consisting of exotic and path-dependent derivatives. However, the convergence rate for the Monte Carlo method is slow and many simulations are required to obtain an accurate estimate of the exposure distribution, making it a time consuming method. Much of the research regarding CCR has therefore been focused on faster and more efficient computation of the exposure profiles than the standard MC method. A literature review on this topic will be presented in Section 1.4.1.

1.3. NUMERICAL INTEGRATION

Since a multivariate expectation is essentially a multi-dimensional integral, it can be solved using general high-dimensional integration techniques. Therefore, a brief introduction of different numerical integration methods is provided in this section.

STANDARD NUMERICAL METHODS

The trapezoidal rule is a simple and widely used numerical integration method. This method approximates the area under a curve by computing the sum of the areas of trapezoids that fit under the curve. While this rule is relatively easy to use, it has certain limitations in terms of accuracy and sensitivity to the shape of the integrated function. Specifically, the trapezoidal rule is a first-order accurate method, which implies that the error in the approximation is proportional to h^2 , where h denotes the distance between two consecutive grid points.

A more accurate integration scheme is the Clenshaw-Curtis method, proposed in [3]. This method employs a sum of Chebyshev polynomials to approximate the integrated function and can provide high accuracy for functions that exhibit significant oscillation or irregular behaviour. Additionally, the Clenshaw-Curtis method converges exponentially to the true value of the integral, which implies that it may require only a small number of function evaluations to achieve a high level of accuracy. The Clenshaw-Curtis quadrature rule also has the advantage of being able to compute the weights in an efficient way using fast Fourier transform (FFT) techniques. This makes the method particularly suitable for problems that involve a large number of function evaluations.

Another class of integration methods is the Gaussian quadrature rules. These quadrature rules use a weighted sum of function values at specific points to approximate the integral. These methods are typically more accurate than the trapezoidal rule and Clenshaw-Curtis method, but can be more difficult to implement. Gaussian quadrature rules can achieve exact integration for polynomials up to a certain degree, and they can also provide highly accurate approximations for other functions. To achieve such accuracy, Gaussian quadrature rules use non-uniformly spaced nodes, which lead to greater accuracy for the integration of functions that are not uniformly distributed across the integration interval.

MONTE CARLO METHODS

Monte Carlo methods are a class of computational algorithms that use random sampling to solve mathematical problems. In the context of numerical integration, the key idea in the Monte Carlo methods is to estimate the value of an integral by generating a large number of random points in the domain of integration and then computing the average of the function values at those points. By the law of large numbers, this will then converge to the true value of the integral when increasing the number of random points.

Monte Carlo methods have some advantages over the standard numerical methods presented before. For example, Monte Carlo methods can be used for high-dimensional integration or complicated domains, where standard quadrature methods may not be practical or accurate. However, the convergence rate of the Monte Carlo methods can be quite slow. Consequently, a lot of random points have to be generated to obtain an accurate approximation, making the method computationally expensive and hence can lead to slow computation times.

Significant efforts have been made to increase the convergence speed of Monte Carlo algorithms through the implementation of efficient sampling techniques and variance reduction methods. In Section 1.4.1, an extensive literature review on the application of Monte Carlo methods within the framework of CCR is presented.

MACHINE LEARNING TECHNIQUES

Machine learning techniques have been increasingly used for numerical integration in recent years. These methods offer a promising alternative to traditional numerical integration methods, particularly for high-dimensional or complex functions. A common approach involves the use of machine learning models to approximate the integrand function, which can then be used to estimate the integral value. Several machine learning techniques have been utilized in the context of numerical integration, such as neural networks, Gaussian processes and support vector machines (SVMs). In Section 1.4.1, an overview of recent literature is provided in which machine learning techniques have been applied in the context of CCR.

SPARSE GRID TECHNIQUES

Sparse grid methods are a class of numerical integration methods that use a sparse set of grid points to approximate integrals. These methods are particularly effective for high-dimensional integration problems, as they can reduce the number of function evaluations required to achieve a given level of accuracy.

This reduction of function evaluations is achieved by partitioning the input space into a set of lower-dimensional grids, and then selectively evaluating the integrand on only a subset of these grids. This selective evaluation strategy allows for more efficient use of computational resources while still accurately approximating the integral.

One of the most commonly used sparse grid algorithms is Smolyak's algorithm [4]. It works by recursively building a set of tensor-product grids of increasing resolution and selectively evaluating the integrand on only a subset of these grids.

Another approach to construct sparse grids is to make use of hierarchical basis functions, such as wavelets or Chebyshev polynomials. These basis functions can be used to construct sparse grids that accurately approximate the integrand with fewer function evaluations.

Sparse grid techniques have been employed in the literature of quantitative finance to solve high-dimensional expectations. Section 1.4.2 presents some literature on the application of sparse grid methods to price multi-asset options.

1.4. LITERATURE REVIEW

As various numerical integration methods have been introduced in the previous section, a literature review is performed how these numerical integration techniques are utilized in the field of quantitative finance to evaluate multivariate expectations. Specifically, the focus of this literature review is on the quantification methods of CCR. However, an additional section on option pricing methods is included, as the solver developed in this thesis will also be applied to price multi-asset options.

1.4.1. COUNTERPARTY CREDIT RISK

MONTE CARLO BASED METHODS

The Monte Carlo approach is the standard approach in the industry to compute exposure profiles and CVA prices. However, the downside is the slow convergence rate, because of which numerous simulations are required and hence is computationally intensive. Therefore, a lot of research is done on improving the MC framework.

The **Least Squares Monte Carlo (LSMC)** method, introduced in [5], is a regression based method often used when dealing with path-dependent or other exotic instruments. The benefit of LSMC is that nested Monte Carlo simulation can be avoided when dealing with path-dependent derivatives, which is a very time consuming method. In [6], LSMC was used to speed up exposure calculations of a portfolio consisting of exotic instruments. The speed of computing the PFE and EE profiles was accelerated in average by a factor of 60 compared to the standard MC framework. In [7] they showed that the MSE of crude Monte Carlo estimators of EPE, CVA, and EEPE can be substantially reduced by solving approximate MSE minimization problems that specify how to achieve an approximately optimal balance between bias squared and variance. However, the method does depend on numerical experiments for model calibration, and worse performance can be expected when estimating the tail of the distribution.

The **Monte Carlo-Tree (MC-Tree)** method combines the MC method with the binomial tree. It was shown that the MC-Tree method is more accurate than LSMC for a single American option at similar computational cost [8]. The method is only tested with options depending on a single asset. Pricing multi-asset options, or other derivatives with multiple risk factors, requires the use of multinomial trees, resulting in greater complexity of the method.

Another Monte Carlo based approach is the so-called **Stochastic Grid Bundling Method (SGBM)**. It was introduced in [9] in the context of pricing multi-dimensional Bermudan options, but later applied to CVA and exposure profiles of options [10] and interest rate derivatives [11], [12]. With SGBM, Monte Carlo paths within the same region of realisation are bundled, assuming these have similar properties. Computation of the MtM prices is then performed locally in these bundles by means of local regression. The method shows a significant variance reduction compared to the LSMC methods, especially when estimating the distribution's tail (for PFE calculations). However, the conver-

gence rate is similar to LSMC methods at a higher computational cost.

In [13], the Monte Carlo simulation is accelerated using Chebyshev polynomials as approximation for the original pricing function. An EPE and PFE profile on a Bermudan Swaption was considered. The calculation via the Chebyshev approximation decreases the computational effort compared to the benchmark full-revaluation, without any loss of accuracy. Furthermore, the incremental computational effort of increasing number of scenarios in the Monte Carlo simulation is negligible [14], [15]. However, only exposure at the level of individual contracts has been taken into account, with consideration given to only one risk factor. Furthermore, the method cannot be utilized for dynamically changing portfolios, as the pricing function of the total portfolio would change accordingly. Moreover, as the number of risk factors increases, the number of Chebyshev coefficients that must be solved at each time point would increase exponentially.

Finite Difference Monte Carlo (FDMC) method uses the scenario generation of the Monte Carlo method and the pricing approach of the finite difference method. The method calculates option prices based on the option pricing partial differential equation, for an entire grid of underlying values. Therefore, this method can easily be used to compute CVA sensitivities, which is done for vanilla and barrier options in [10] and Bermuda options in [16] under Heston's model. The method shows accurate results for single instruments with one or two risk factors. However, applying FDMC on more risk factors becomes impracticable, as the differential equation needs to be discretized over each dimension space corresponding to each risk factor.

Another way of computing CVA sensitivities is the so-called pathwise sensitivity method, described in [17]. The key idea is to interchange the differentiation and expectation operators within the valuation formula. The pathwise sensitivity method can be applied under mild conditions regarding the smoothness of the payoff function and the stochastic processes of the risk factors involved. In [18], a new method is presented to apply the pathwise sensitivity method also to Barrier options with discontinuous payoff. Compared to the finite difference method, there is no extra Monte Carlo simulation required to "shock" the parameters, resulting in a computational advantage.

The literature on CVA calculations also involves the quantification of **Wrong-Way Risk (WWR)**. A literature review on the inclusion of WWR in CVA calculations can be found in Appendix A.1.

MACHINE LEARNING METHODS

Machine learning methods provide new techniques for computing exposure profiles. The Bank for International Settlements (BIS) points out machine learning as one of the main drivers of the ongoing transformation of the financial sector [19]. Machine learning may provide alternatives to costly numerical methods in many areas of the financial industry, but its wide adoption is challenged by requirements on transparency, reliability and accountability.

A popular technique used is the **Gaussian Process Regression (GPR)**, as is done for CVA calculations in [20],[21]. GPR is a supervised learning method, in which regression and probabilistic classification problems can be solved. The prediction is probabilistic (Gaussian), which means that one can compute empirical confidence intervals. Based on these intervals, decisions can be made if one should refit the prediction in some region of interest. The benefit of GPR is that the uncertainty of the predictions are already

quantified in the model. It can also be used to price path-dependent options by fitting the continuation values through GPR, as is done for Bermudan options in [22].

Another popular machine learning method involves the use of neural networks. For example, the standard LSMC method is adapted in [23], in which the least-square regression is replaced with a neural network approximation. Numerous studies have been done on the application of neural networks in the context of exposure and CVA calculation [24]–[29]. As mentioned before, these methods show promising results in terms of fast computation, but the complexity and lack of transparency prevent the use of these techniques in the real world so far.

1.4.2. OPTION PRICING

In this thesis, the solver developed is utilized in the context of pricing multi-asset options. Therefore, a brief overview of the primary option evaluation methods available in the literature is presented.

Under the premise of no-arbitrage, the values of multi-asset options are determined by multivariate expectations under an equivalent martingale measure. Consequently, the Monte Carlo-based and machine learning methods discussed in the previous section, which are specifically designed for computing multivariate expectation problems, can also be employed to estimate the multi-asset option values

A different approach to computing option values involves the use of Fourier techniques. One of the most popular used methods of this type is the COS method, which was introduced by Fang and Oosterlee in [30]. The essence of the method lies in approximating the probability density function, which appears in the expectation, through the use of its Fourier-cosine series expansion. It has been established that the Fourier-cosine series coefficients have a closed-form solution with the characteristic function. This allows the COS method to be applied to a broad class of asset price processes for which the characteristic function is available, including exponential Lévy processes and affine jump diffusion processes [31]. For smooth density functions, COS method is optimal in terms of error convergence and in computational complexity for European options.

The Feynman-Kac theorem provides a connection between the multivariate expectation of the option's payoff value and the solution of the corresponding pricing partial differential equation (PDE). Therefore, another way to compute option values is to solve the corresponding pricing PDE. Techniques to solve PDE's such as finite difference schemes can then be applied to obtain the option's value. The main benefit of the PDE approach is that the option's value can be generated for the entire discretization range of the underlying stock values simultaneously. The PDE approach is commonly employed when only one or two stock processes are involved in the option's value. However, dealing with multi-asset options through the PDE approach can become impracticable, as the discretization must be performed for every underlying stock process, resulting in a high-dimensional grid to evaluate.

The direct numerical integration techniques discussed in Section 1.3 can be used to approximate multivariate expectations and hence also approximate option values. However, the standard quadrature rules are not practical for multi-asset options due to their exponential computational complexity. Recently, researchers have explored the use of adaptive, sparse grids techniques for pricing multi-asset options [32]–[34]. Another re-

cent paper focuses on evaluating multi-asset basket options using adaptive sparse grid techniques [35]. The method described in here will be compared with the solver developed in this thesis to investigate the competitiveness of the method with recently developed methods in the literature.

1.5. THESIS OBJECTIVE AND OUTLINE

The goal of this thesis is to develop a generic and efficient solver for multivariate expectation problems, with a focus on its application in the field of quantitative finance. The solver, which is referred to as the COS-CPD method, comprises two main components: the COS method and a tensor decomposition technique called CPD. The COS method will be used to recover the portfolio distribution at future time points, from which the EE and PFE quantities can be derived.

The replacement of the Monte Carlo method by the COS method to compute PFE profiles has been studied in a previous master thesis carried out at FF Quant [36], which included portfolios with a maximum of three risk factors. When applying the COS method, the characteristic function of the portfolio value has to be evaluated. In [36], this characteristic function was approximated by directly employing the Clenshaw-Curtis quadrature method. However, as the dimension of integration is determined by the number of risk factors, this approach is only feasible when dealing with portfolios that involve up to two or three risk factors. This is because high-dimensional integration becomes impracticable with the Clenshaw-Curtis quadrature rule (or any other standard quadrature method) due to its exponential growth in computational costs.

However, portfolios contain easily more than three risk factors in practice. In order to apply the COS approach on portfolios with more than three risk factors, an alternative method of evaluating the characteristic function has to be developed. Therefore, this thesis proposes an improvement on the COS approach by reducing the dimension of integration for the characteristic function. This is done in two steps. First, the joint density function of the risk factors, appearing in the characteristic function, is replaced by a dimension-reduced Fourier-cosine series expansion, obtained via CPD. In this way, the computational complexity of computing the Fourier coefficient tensor is reduced to a linear growth with respect to the number of dimensions. The application of CPD to decompose the joint density into a Fourier-cosine series expansion can be applied to any multivariate expectation problem, which shows the generic nature of the method. Secondly, we make use of the specific portfolio structure. Although a derivative portfolio can consist of many risk factors, each individual contract comprises two paying legs that depend on only one or two risk factors. By dividing the portfolio into segments that share the same risk factors, the evaluation of the characteristic function can be reduced to the calculation of only one- and two-dimensional integrals, which can be solved numerically again using the Clenshaw-Curtis quadrature rule. Consequently, the characteristic function can be obtained by solving only one- and two-dimensional integrals, making the method applicable to portfolios with more than three risk factors.

The outline of the thesis is as follows: Chapter 2 provides the necessary mathematical foundation for modeling, formulating, and solving the relevant quantities involved in counterparty credit risk and option pricing. After that, Chapter 3 presents an introduction to the concept of Canonical Polyadic Decomposition (CPD) and explains how

this decomposition technique can be used in the context of Fourier-cosine series expansions to decrease the computational complexity of approximating the Fourier coefficients. A supervised learning algorithm is presented to obtain the CPD factor matrices, from which the Fourier coefficient tensor can be constructed. This algorithm is numerically analysed in Chapter 4 to obtain optimal parameter settings that will be used later when applying the CPD algorithm in the context of CCR quantification and option pricing. The methodology of the COS-CPD method to solve multivariate expectations is described in Chapter 5. Additionally, the methodology is presented in detail for the generation of PFE and EE values in the context of CCR quantification. Subsequently, a theoretical error analysis of the COS-CPD method is outlined in Chapter 6, in which each error source is identified and assigned a theoretical upper bound. In Chapter 7, the COS-CPD method is applied to quantify counterparty credit risk for portfolios of three and five risk factors, with numerical results compared to the Monte Carlo method. To demonstrate the versatility of the COS-CPD method, Chapter 8 presents the application of the COS-CPD method into the context of multi-asset option pricing, as the evaluation of a six-dimensional basket option is considered. The numerical results are compared in both accuracy and computation time with a recently introduced sparse grid method for solving multi-asset basket options to show the competitiveness of the method. Chapter 9 provides a summary of all the findings, draws a conclusion on the method, and suggests future research directions.

2

MATHEMATICAL FRAMEWORK

In this chapter, all the relevant mathematical tools are presented to formulate, model and solve the CRR quantification and option pricing problems.

2.1. STOCHASTIC CALCULUS

This section contains definitions and theorems from the field of stochastic calculus. The notation is based on [37], [38]. With these results, the stochastic models of the risk factors can be defined and analysed.

Definition 2.1.1 (Filtration). *Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. A filtration on $(\Omega, \mathcal{F}, \mathbb{P})$ is a family of sub σ -fields $\{\mathcal{F}_t, t \geq 0\}$ of \mathcal{F} indexed by $t \in [0, \infty)$, such that $\mathcal{F}_s \subset \mathcal{F}_t$ for every $s \leq t \leq \infty$.*

Definition 2.1.2 (Adapted process). *A process $X = \{X_t, t \geq 0\}$ is said to be adapted to a filtration $\{\mathcal{F}_t, t \geq 0\}$ if for all $t \geq 0$, X_t is \mathcal{F}_t measurable.*

Definition 2.1.3 (Martingale). *Let $M = \{M_t, t \geq 0\}$ be a process defined on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ equipped with a filtration $\{\mathcal{F}_t, t \geq 0\}$. Then M is said to be a martingale if*

1. M is an adapted process.
2. For all $t \geq 0$, M_t is integrable.
3. M satisfies the martingale property, which reads

$$\mathbb{E}[M_t | \mathcal{F}_s] = M_s, \quad \forall 0 \leq s < t.$$

Definition 2.1.4 (Semimartingale). *A stochastic process $X = \{X_t, t \geq 0\}$ is called a semimartingale if it can be decomposed as follows:*

$$X = X_0 + M + A,$$

where the random variable X_0 is finite and \mathcal{F}_0 -measurable, the stochastic process M is a local martingale and the stochastic process A has finite variation.

Definition 2.1.5 (Brownian Motion). *A real-valued process $\{W(t), t \geq 0\}$ is called a Brownian motion if*

1. *Starting at 0: $W(0) = 0$.*
2. *Normally distributed increments: For all $0 \leq s < t$, $W(t) - W(s) \simeq N(0, t - s)$.*
3. *Independent increments: For $0 \leq t_0 < t_1 < \dots < t_n$, the random variables $Y_i := W(t_i) - W(t_{i-1})$, $i = 1, \dots, n$ are independent.*
4. *Continuous trajectories: The map $t \mapsto W(t)$ is continuous.*

Definition 2.1.6 (Itô Integral). *For any square-integrable adapted process $g(t)$ with continuous sample paths, the Itô integral is given by:*

$$I(T) = \int_0^T g(t) dW(t) := \lim_{m \rightarrow \infty} I_m(T), \quad \text{in } L^2.$$

Here, $I_m(T) = \int_0^T g_m(t) dW(t)$ for some elementary process $g_m(t) = \sum_{j=0}^{n-1} \eta_j \mathbf{1}_{[t_j, t_{j+1})}$, satisfying:

$$\lim_{m \rightarrow \infty} \mathbb{E} \left[\int_0^T (g_m(t) - g(t))^2 dt \right] = 0,$$

where η_j is \mathcal{F}_{t_j} measurable for all $j = 0, 1, \dots, n-1$ and square-integrable.

Theorem 2.1.1 (Itô Isometry). *For any stochastic process $g(t)$, satisfying the usual regularity conditions, the following equality holds,*

$$\mathbb{E} \left[\left(\int_0^T g(t) dW(t) \right)^2 \right] = \int_0^T \mathbb{E}[g^2(t)] dt.$$

Theorem 2.1.2 (Itô's Formula). *Let $f \in C^2(\mathbb{R})$ and consider a continuous semimartingale X with decomposition $X = M + A$. Then, the stochastic process $(f(X_t))_{t \geq 0}$ is also a semimartingale and holds*

$$f(X_t) = f(X_0) + \int_0^t \frac{\partial f}{\partial x}(X_u) dX_u + \frac{1}{2} \int_0^t \frac{\partial^2 f}{\partial x^2}(X_u) d[X]_u,$$

with $[X]$ denotes the quadratic variation of the process $(X_t)_{t \geq 0}$.

Itô's formula is often expressed in differential form:

$$df(X_t) = \frac{\partial f}{\partial x}(X_t) dX_t + \frac{1}{2} \frac{\partial^2 f}{\partial x^2}(X_t) d[X]_t.$$

Definition 2.1.7 (Girsanov's Theorem). *Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. In addition, let $(W_t)_{t \geq 0}$ be a Brownian motion defined on $(\Omega, \mathcal{F}, \mathbb{P})$ with natural filtration $\mathcal{F}_t = \sigma(W(s), 0 \leq s \leq t)$. Let $(L_t)_{t \geq 0}$ be a (local) \mathbb{P} -martingale, and define the process $(Z_t)_{t \geq 0}$ via the stochastic exponential,*

$$Z_t := \mathcal{E}(L_t) = e^{L_t - \frac{1}{2}[L]_t}.$$

Then, Z_t is a \mathbb{P} -martingale, and a new probability measure \mathbb{Q} , equivalent with \mathbb{P} , can be defined via the Radon–Nikodym derivative such that

$$\frac{d\mathbb{Q}}{d\mathbb{P}} \Big|_{\mathcal{F}_t} = Z_t.$$

Furthermore, the Brownian motion W_t under the new measure \mathbb{Q} is given by

$$\tilde{W}_t = W_t - [W, L]_t,$$

where $[W, L]$ denotes the quadratic covariance between the processes W_t and L_t .

2.2. CCR QUANTIFICATION

In this section, we present the mathematical framework for quantifying CCR. Most of the notation used in this framework is derived from [37], [39]. First, the concept of exposure is defined and two exposure related measures are formulated. After that, we introduce the concept of CVA and derive a formula for calculating the CVA of a portfolio.

2.2.1. EXPOSURE PROFILES

In order to present the definition of exposure for a derivative portfolio, it is important to clarify what is meant by the "value" of a portfolio. In this thesis, the mark-to-market (MtM) value will be used, which is formulated in the next definition.

Definition 2.2.1 (Mark-to-market). *Mark-to-market (MtM) is a method of measuring the fair value of financial products based on current market conditions. In context of CCR, the MtM with respect to a particular counterparty defines what could be potentially lost today if default occurs.*

Definition 2.2.2 (Exposure for single derivative). *The exposure E_t at time t for a single derivative contract is defined as,*

$$E_t = E(X_t) = \max(V(X_t), 0), \quad 0 \leq t \leq T,$$

where X_t indicate the risk factors involved and $V(X_t)$ the MtM value of the derivative contract at time t .

In practice, two counterparties often trade several derivative contracts with one another, which are grouped together in one or more portfolios. In this case, CCR can be mitigated using a technique called *netting*.

Definition 2.2.3 (Netting agreement). *A netting agreement is a legally binding contract that allows aggregation, and therefore possible offsetting, of multiple transactions with one counterparty.*

Definition 2.2.4 (Exposure on counterparty level). *Let $V = \sum_{i=1}^N V_i$ be a portfolio consisting of N derivative contracts traded with one counterparty. The exposure of the portfolio in the case of no netting is given by*

$$E_t^c = \max(V_1(X_t), 0) + \dots + \max(V_N(X_t), 0), \quad 0 \leq t \leq T.$$

If netting is allowed, the exposure of the netting set, denoted with E_t^n , is of the following form,

$$E_t^n = \max\left(\sum_{i=1}^N V_i(X_t), 0\right), \quad 0 \leq t \leq T.$$

Since $\max(X + Y, 0) \leq \max(X, 0) + \max(Y, 0)$, netting always reduces the exposure of the portfolio and therefore also the CCR.

Once the exposure distribution at a future time point is known, different measures to quantify CCR can be extracted. One of these measures often used in the industry is the expected exposure, which also plays a central role in the context of CVA.

Definition 2.2.5 (Expected Exposure). *The expected (positive) exposure (EE) at time t is defined as the expectation of the exposure profile E_t ,*

$$EE_t = \mathbb{E}[E_t] = \mathbb{E}[E(X_t)], \quad 0 \leq t \leq T.$$

For risk management purposes, the potential future exposure is computed. This measure can be viewed as the worst exposure given a certain confidence interval. Mathematically, the PFE is found by taking a specific quantile of the exposure distribution.

Definition 2.2.6 (Potential Future Exposure). *The potential future exposure (PFE) at time t is defined as the α -quantile of the exposure distribution E_t ,*

$$PFE_{\alpha,t} = \inf\{x \in \mathbb{R} : \alpha \leq F_{E_t}(x)\},$$

where $F_{E_t}(x) := \mathbb{P}(E_t \leq x)$ is the cumulative distribution function of the exposure E_t under real-world measure \mathbb{P} . For PFE calculations, the quantile $\alpha = 97.5\%$ is commonly used.

2.2.2. CREDIT VALUE ADJUSTMENT (CVA)

CVA is by definition the difference between the risk-free portfolio value and the true portfolio value that takes into account the possibility of a counterparty's default [2]. In other words, CVA is the market value of counterparty credit risk,

$$CVA = V(\text{default free}) - V(\text{credit risky}).$$

There are a variety of different models for CVA. If only the credit risk of one counterparty is taken into account, the model is referred to as *unilateral* CVA. However, if the credit risk of both counterparties is considered, a *bilateral* CVA model is utilized.

Before presenting the mathematical formula for unilateral CVA, the definition of the recovery rate is presented.

Definition 2.2.7 (Recovery rate). *The recovery rate R_c is the percentage amount that a creditor receives in relation to claims on a defaulted counterparty.*

For a constant recovery rate R_c , the formula for the unilateral CVA is as follows,

$$CVA(t) = (1 - R_c) \mathbb{E}^{\mathbb{Q}} [D(t, T) \mathbb{1}_{t_D \leq T} \max(V_{t_D}, 0) | \mathcal{F}_t],$$

where R_c denote the (constant) recovery rate, $D(t, T)$ the stochastic discount factor defined in Definition 2.3.2, t_D the time of default, $\mathbb{1}$ the indicator function and V_{t_D} the MtM value of the portfolio at time of default. A derivation of this formula can be found in [40]. Using the tower property for expectations we obtain

$$\begin{aligned} CVA(t) &= (1 - R_c) \mathbb{E}^{\mathbb{Q}} \left[\mathbb{E}^{\mathbb{Q}} \left[D(t, T) \mathbb{1}_{t_D \leq T} \max(V_{t_D}, 0) \middle| \mathcal{F}_{t_D} \right] \middle| \mathcal{F}_t \right] \\ &= (1 - R_c) \mathbb{E}^{\mathbb{Q}} \left[\mathbb{1}_{t_D \leq T} \mathbb{E}^{\mathbb{Q}} \left[D(t, T) \max(V_{t_D}, 0) \middle| \mathcal{F}_{t_D} \right] \middle| \mathcal{F}_t \right]. \end{aligned}$$

Note that the expectation of the exposure is conditional on the counterparty default occurring at time t_D . When assuming independence between exposure and the default time t_D , CVA can be approximated with

$$\begin{aligned} CVA(t) &= (1 - R_c) \int_t^T \mathbb{E}^{\mathbb{Q}} \left[D(t, T) \max(V(t_D, T), 0) \middle| t_D = z \right] f_{t_D}(z) dz \\ &= (1 - R_c) \int_t^T \mathbb{E}^{\mathbb{Q}} \left[D(t, T) \max(V(t_D, T), 0) \middle| t_D = z \right] dF_{t_D}(z) \\ &\approx (1 - R_c) \sum_{k=1}^m EE_{T_k} q(T_{k-1}, T_k), \end{aligned} \tag{2.1}$$

where EE_{T_k} denotes the (discounted) expected exposure as defined in 2.2.5 and $q(T_{k-1}, T_k) := F_{t_D}(T_k) - F_{t_D}(T_{k-1})$ the probability of default between time T_{k-1} and T_k . The assumption of independence between exposure and default time may lead to an underestimation of the risk, resulting in a potential significant loss. The phenomenon which is not modelled in this scenario is commonly referred to as *wrong-way risk*, which is presented in the next definition.

Definition 2.2.8 (Wrong-way risk). *In the context of CVA, wrong-way risk occurs whenever there is an adverse co-dependence between the exposure to a counterparty and the risk that the counterparty will default, while right-way risk occurs whenever there is a favourable co-dependence.*

Wrong-way risk removes the assumption of independence between the expected exposure and the probability of default. Hence, the model presented in 2.1 cannot be used when wrong-way risk has to be incorporated in the model. Appendix A provides a literature review on various techniques that can be utilized to account for wrong-way risk in CVA models.

Bilateral CVA models add an additional term called **Debit Value Adjustment (DVA)**, arising from the credit risk of the reporting institution. In other words, it represents the counterparty credit risk from the point of view of the "seller". DVA is a very controversial topic due to its potential to increase the accounting value of a derivatives portfolio, while the creditworthiness of the institution is decreasing. Therefore, the Basel III framework does not include DVA as part of the CVA capital charge [41].

2.3. INTEREST RATE FRAMEWORK

This section is devoted to the mathematical formulation of several interest rate related quantities used in the market. Additionally, it introduces two popular interest rate derivatives and provides a valuation formula for each. The final part of this section elaborates on the Gaussian one factor model, which is used throughout this thesis to model the short rate. Properties of this model are analyzed, and an expression for the **Zero-coupon Bond (ZCB)** price under this model is derived.

2.3.1. INTEREST RATE DEFINITIONS AND PRODUCTS

The definitions presented in this section are taken from [42]. We start with the concept of the money-savings account or bank account. This is considered as a risk-free investment, which accrues according to a continuously compounded interest rate. This rate is also known as the instantaneous spot rate, or briefly as short rate.

Definition 2.3.1 (Bank account). *The value of a bank account is defined as $B(t)$ at time $t \geq 0$. We assume $B(0) = 1$ and that the bank account evolves according to the following differential equation:*

$$dB(t) = r(t)B(t)dt, \quad B(0) = 1,$$

where $r(t)$ is the instantaneous spot rate. As a consequence,

$$B(t) = e^{\int_0^t r(s) ds}. \quad (2.2)$$

The above definition shows that investing a unit amount at time 0 in the bank account yields the value in 2.2 at time t .

Definition 2.3.2 (Stochastic discount factor). *The (stochastic) discount factor $D(t, T)$ between two time instants t and T is the amount at time t that is “equivalent” to one unit of currency payable at time T , and is given by*

$$D(t, T) = \frac{B(t)}{B(T)} = e^{-\int_t^T r(s) ds}.$$

Definition 2.3.3 (Zero-coupon bond). *A T -maturity zero-coupon bond (pure discount bond) is a contract that guarantees its holder the payment of one unit of currency at time T , with no intermediate payments. The contract value at time $t < T$ is denoted by $P(t, T)$. Clearly, $P(T, T) = 1$ for all T . The zero-coupon bond can be viewed as the expectation of the stochastic discount factor $D(t, T)$,*

$$P(t, T) = \mathbb{E}^{\mathbb{Q}} [D(t, T) | \mathcal{F}_t] = \mathbb{E}^{\mathbb{Q}} \left[e^{-\int_t^T r(s) ds} | \mathcal{F}_t \right],$$

where the superscript \mathbb{Q} denotes that the expectation is taken under the risk-neutral probability measure.

An alternative to continuous compounding is simple compounding, which applies when the interest accrues proportionally to the time of the investment. Simple compounding is used to define interbank rates like the LIBOR. For an amount M_t invested at

time t with simply-compounded interest rate $r(t, T)$, the value of the investment at time T is then as follows:

$$M_T = (1 + r(t, T))\tau(t, T)M_t,$$

where $\tau(t, T)$ denotes the time difference between t and T .

A special case is when $M_t = P(t, T)$. The interest rate for which $M_T = 1$ is called the simply-compounded spot interest rate, and is presented in the next definition.

Definition 2.3.4 (Simply-compounded spot interest rate). *The simply-compounded spot interest rate prevailing at time t for the maturity T is denoted by $L(t, T)$ and is the constant rate at which an investment has to be made to produce an amount of one unit of currency at maturity, starting from $P(t, T)$ units of currency at time t , when accruing occurs proportionally to the investment time. In formulas:*

$$L(t, T) = \frac{1 - P(t, T)}{\tau(t, T)P(t, T)},$$

where $P(t, T)$ is the ZCB at time t with maturity T , and $\tau(t, T)$ is the year fraction according to the day-count convention, usually Act365.

Another type of interest rates are the so-called forward rates. Forward rates are interest rates that can be locked in today for an investment in a future time period. To define the forward rate, three specific time instants have to be established, which include the current time t at which the rate is being considered, the expiry time T , and the maturity S , where $t \leq T \leq S$. Forward rates are typically agreed upon through a [Forward-Rate Agreement \(FRA\)](#). This contract gives the holder an interest rate payment for the period between T and S . At the maturity S , a fixed payment based on a fixed rate K is exchanged against a floating payment based on the spot rate $L(T, S)$ resetting in T and with maturity S . Basically, this contract allows one to lock-in the interest rate between times T and S at a desired value K , with the rates in the contract that are simply compounded. The payoff of a FRA contract at time S is therefore

$$\begin{aligned} V^{FRA}(t, T, S, \tau(T, S), N, K) &= N\tau(T, S)(K - L(T, S)) \\ &= N[P(t, S)\tau(T, S)K - P(t, T) + P(t, S)], \end{aligned} \quad (2.3)$$

where $\tau(t, T)$ is the year fraction, $P(t, T)$ is the ZCB at time t with maturity T and N is the notional amount of the transaction. In general, forward contracts are traded without any initial cash flow between the two parties at time t . This is achieved by selecting a fixed rate K that ensures that the value of the FRA contract is zero at time t . It is clear from 2.3 that there exists a unique value for K that renders the FRA a fair contract at time t . This value for K is called the simply-compounded forward interest rate and is defined below.

Definition 2.3.5 (Simply-compounded forward interest rate). *The simply-compounded forward interest rate prevailing at time t for the expiry $T > t$ and maturity $S > T$ is denoted by $F(t; T, S)$ and is defined by*

$$F(t; T, S) = \frac{1}{\tau(T, S)} \left(\frac{P(t, T)}{P(t, S)} - 1 \right).$$

It is the value of the fixed rate K in a prototypical FRA with expiry T and maturity S that renders the FRA a fair contract at time t .

The forward rate $F(t; T, S)$ may thus be viewed as an estimate of the future spot rate $L(T, S)$ based on the market conditions at time t . If the maturity time S gets infinitely close to expiry time T , the instantaneous forward rate is obtained.

Definition 2.3.6 (Instantaneous forward interest rate). *The instantaneous forward interest rate prevailing at time t for the maturity $T > t$ is denoted by $f(t, T)$ and is defined as*

$$f(t, T) = \lim_{S \rightarrow T^+} F(t, T, S) = -\frac{\partial \ln P(t, T)}{\partial T}.$$

Intuitively, the instantaneous forward interest rate $f(t, T)$ is a forward interest rate at time t whose maturity is very close to its expiry T , i.e. $f(t, T) \approx F(t; T, T + \Delta T)$ with ΔT small.

A popular interest rate derivative is the **Interest Rate Swap (IRS)**, which can be considered as a generalization of the FRA. An IRS is a derivative exchanging two different legs, often a fixed and a floating leg. During the contract, both legs pay coupons at pre-arranged dates $\mathcal{T} := \{T_{\alpha+1}, \dots, T_{\beta}\}$. Throughout this thesis, we will assume that the payments for the fixed and floating leg are at the same dates. However, the payment dates for fixed and floating leg can differ¹. For the the fixed leg, the payment is based on a fixed interest rate K , whereas the floating leg pays a coupon depending on the forward rate. At time T_{i-1} , the interest rate is reset for the coupon payment in the period $[T_{i-1}, T_i]$. In case the fixed leg is paid and the floating leg is received, the IRS is called a **Payer IRS (PFS)**, while in the other case we have a **Receiver IRS (RFS)**. The MtM of an IRS is the (expected) discounted payoff, which can be found by discounting all the cash flows of the fixed and floating leg,

$$\begin{aligned} V^{IRS}(t, \mathcal{T}, \tau, N, K) &= \bar{\eta} \sum_{i=\alpha+1}^{\beta} V^{FRA}(t, T_{i-1}, T_i, \tau_i, N, K) \\ &= \bar{\eta} N \sum_{i=\alpha+1}^{\beta} \tau_i P(t, T_i) (K - F(t; T_{i-1}, T_i)) \\ &= \bar{\eta} \left[-NP(t, T_{\alpha}) + NP(t, T_{\beta}) + N \sum_{i=\alpha+1}^{\beta} \tau_i KP(t, T_i) \right], \end{aligned} \quad (2.4)$$

with $\bar{\eta} = 1$ for a RFS and $\bar{\eta} = -1$ for a PFS. Furthermore, N denotes the notional amount, K the fixed rate, \mathcal{T} the set of payment dates and $\tau := \{\tau_{\alpha+1}, \dots, \tau_{\beta}\}$ the set of year fractions.

2.3.2. INTEREST RATE MODELS

One of the first steps in the quantification of CCR is to select an appropriate interest rate model. There is a great variety of (stochastic) short rate models available. One-factor short rate models describe interest rate movements with only one stochastic factor that determines the future evolution of the interest rate. These models are popular for their simplicity and, in certain cases, the availability of a closed-form solution. A well-known

¹A typical IRS in the market has a fixed leg with annual payments and a floating leg with quarterly or semiannual payments.

one-factor model for the short rate was introduced by Vasicek in [43]. Vasicek's model was the first one to capture mean reversion, an essential characteristic of the interest rate process. However, under Vasicek's model, it is theoretically possible for the interest rate to become negative, which was considered as an unrealistic scenario for a long time. This drawback was fixed in the Cox–Ingersoll–Ross (CIR) model [44]. Another popular model used in the industry is the Hull–White (HW) model introduced in [45]. This model is also an extension of Vasicek's model, as the constant drift and volatility parameters are now defined as a deterministic function of time. Because of this feature, the HW model is capable of calibrating the model on the entire term structure observed in the market, while this is not possible for the (simpler) Vasicek model.

In this thesis, the **Gaussian One Factor (G1++)** model will be used to model the short rate. The G1++ model is equivalent to the Hull–White model, but has a more convenient expression for the corresponding ZCB formula. The derivation of the G1++ model below is based on [46].

In the G1++ model, the short rate dynamics under the risk-neutral measure are of the form

$$r(t) = x(t) + \beta(t),$$

where the shifted short rate $x(t)$ follows an Ornstein–Uhlenbeck process:

$$dx(t) = -ax(t)dt + \sigma dW(t), \quad x(t_0) = 0,$$

where a is a positive constant, σ the volatility and $W(t)$ is a standard Brownian motion. The function $\beta(t)$ is deterministic and is found by the exact fitting of the model to the term structure observed in the market. The solution $x(t)$ of the Ornstein–Uhlenbeck process is known and given by

$$x(t) = x(t_0)e^{-a(t-t_0)} + \sigma \int_{t_0}^t e^{-a(t-s)} dW(s).$$

A derivation for this solution can be found in Appendix B.1. Since $x(t_0) = 0$, we can see that $x(t)$ is normally distributed with mean and variance

$$\begin{aligned} \mathbb{E}[x(t)|\mathcal{F}_{t_0}] &= x(t_0)e^{-a(t-t_0)} = 0, \\ \text{Var}[x(t)|\mathcal{F}_{t_0}] &= \frac{\sigma^2}{2a} (1 - e^{-2a(t-t_0)}). \end{aligned}$$

As $\beta(t)$ is a deterministic function, $r(t)$ is also normally distributed, with conditional mean and variance

$$\begin{aligned} \mathbb{E}[r(t)|\mathcal{F}_s] &= \mathbb{E}[x(t) + \beta(t)|\mathcal{F}_s] = \mathbb{E}[x(t)|\mathcal{F}_s] + \beta(t) = x(s)e^{-a(t-s)} + \beta(t), \\ \text{Var}[r(t)|\mathcal{F}_s] &= \text{Var}[x(t) + \beta(t)|\mathcal{F}_s] = \text{Var}[x(t)|\mathcal{F}_s] = \frac{\sigma^2}{2a} (1 - e^{-2a(t-s)}). \end{aligned}$$

From Definition 2.3.3, the value of a ZCB under the risk-neutral measure \mathbb{Q} is

$$P(t, T) = \mathbb{E}^{\mathbb{Q}} \left[e^{-\int_t^T r(s) ds} \mid \mathcal{F}_t \right].$$

For the G1++ model, this ZCB formula can be expressed as

$$P(t, T) = A(t, T)e^{-B(t, T)x(t)}, \quad (2.5)$$

where

$$A(t, T) = \frac{P^M(0, T)}{P^M(0, t)} e^{\frac{1}{2}[V(t, T) - V(0, T) + V(0, t)]},$$

$$B(t, T) = \frac{1}{a} [1 - e^{-a(T-t)}].$$

$V(t, T)$ is the variance of $\int_t^T x(s)ds$ conditional on \mathcal{F}_t given by

$$V(t, T) = \frac{\sigma^2}{a^2} \left(T - t - 2 \frac{1 - e^{-a(T-t)}}{a} + \frac{1 - e^{-2a(T-t)}}{2a} \right).$$

A proof for this ZCB expression is given in Appendix B.2. A crucial observation is that the ZCB formula for the G1++ model only depends on the shifted short rate process $x(t)$, together with the market discount curve P^M . The ZCB pricing formula in 2.5 will play a central role within the simulation framework, as it can be used to derive all other rates. This allows for the valuation of different derivatives to be expressed in terms of zero-coupon bonds. For example, the value of an IRS presented in 2.4 is written as a linear combination of different zero-coupon bonds.

2.4. FOREIGN EXCHANGE FRAMEWORK

Foreign Exchange (FX) derivatives are another type of liquid financial products traded in the OTC market. In 2021, around 17% of the total value of outstanding derivatives on the OCT market was related to FX type of products [47]. FX derivatives are characterised by payments in two currencies, often referred to as the domestic and foreign currency. The conversion between two currencies is done using the FX rate, which will be denoted with X .

Definition 2.4.1 (Foreign Exchange rate). *The foreign exchange (FX) rate is a rate at which one (domestic) currency will be exchanged for another (foreign) currency and affects trade and the movement of money between countries.*

The convention in literature is to set the FX rate $X(t)$ as the amount of domestic currency needed to buy one unit of foreign currency at time t . Using this relation, an amount N^f in foreign currency can be exchanged with an amount of $N^d = N^f \cdot X$ in domestic currency.

To compute the value of a FX product, all the cash flows of the contract must be expressed into the same valuation currency. Throughout this thesis, the domestic currency will be used as the valuation currency, implying that the cash flows from the traded legs in the foreign currency will be converted to the domestic currency.

2.4.1. FX PRODUCTS

One of the most traded FX products is the so-called FX forward. This contract is an agreement to exchange a certain amount of the domestic currency N^d for an amount of foreign currency N^f at a fixed exchange rate \tilde{X}_T on a specified future time point T . The value of the FX contract will be expressed in the domestic currency, requiring the conversion of the cash flow of the foreign currency to the domestic currency using the exchange rate $X(t)$. Consequently, the value of a FX forward at time $t \leq T$, in which the domestic notional is paid and foreign notional is received, has the following form,

$$\begin{aligned} V_{FX}(t) &= N^f P^f(t, T) X(t) - N^d P^d(t, T) \\ &= \frac{N^d}{\tilde{X}_T} P^f(t, T) X(t) - N^d P^d(t, T) \\ &= N^d \left[P^f(t, T) \frac{X(t)}{\tilde{X}_T} - P^d(t, T) \right], \end{aligned}$$

where N^d is the notional amount in domestic currency, P^d and P^f the ZCB value in the domestic and foreign currency and $X(t)$ the exchange rate at time t .

Similar to the interest forward rate in the FRA contract discussed before, the exchange rate \tilde{X}_T is often chosen such that the value of the contract is fair at initial time t . The exchange rate for which the value is fair is called the FX forward rate.

Definition 2.4.2 (FX forward rate). *The FX forward rate prevailing at time t for the maturity $T > t$ is denoted by $X_F(t; T)$ and is defined by*

$$X_F(t; T) = \frac{P^f(t, T)}{P^d(t, T)} X(t).$$

Here P^d, P^f are the zero-coupon bonds in the domestic and foreign currency respectively. $X(t)$ is the exchange rate at time t defined as the amount of units in domestic currency for one unit of foreign currency. X_F is the value of the FX exchange rate in a FX forward agreement with maturity T that renders the FX forward a fair contract at time t .

Another multi-currency derivative is the **Cross-Currency Swap (XCS)**. The XCS is similar to the IRS derivative defined in 2.4. However, the payments' legs now consist of different currencies. There are three types of cash flows possible in a XCS, namely the fixed-for-fixed, fixed-for-floating and floating-for-floating. These indicate the fixed or floating types of the traded legs. In the fixed-for-floating XCS, one receives a fixed amount of one currency and pays the floating amount of the other currency. For example, the value of a fixed-for-floating XCS, in which one receives the fixed foreign currency and pays the

domestic floating currency, can be valued as

$$\begin{aligned}
 V_{XCS}(t) &= \sum_{i=\alpha+1}^{\beta} \left[N^f X(t) K \tau_i P^f(t, T_i) - N^d F^d(t, T_{i-1}, T_i) \tau_i P^d(t, T_i) \right] \\
 &= -N^d \sum_{i=\alpha+1}^{\beta} F^d(t, T_{i-1}, T_i) \tau_i P^d(t, T_i) + N^f X(t) K \sum_{i=\alpha+1}^{\beta} \tau_i P^f(t, T_i) \\
 &= N^d \left[P^d(t, T_\beta) - P^d(t, T_\alpha) \right] + N^f X(t) K \sum_{i=\alpha+1}^{\beta} \tau_i P^f(t, T_i),
 \end{aligned}$$

where N^d, N^f are the notional amounts in the domestic and foreign currency, P^d and P^f the ZCB value in the domestic and foreign currency, τ_i the time fraction between time between time T_i and T_{i+1} , K the fixed rate and $X(t)$ the exchange rate at time t .

2.4.2. FX RATE MODEL

The exchange rate X is not a deterministic value, but continuously changes based on the value of both currencies. Therefore, the exchange rate is not known at future time points, and has to be modelled according to a suitable stochastic process. In this section, the **Geometric Brownian Motion (GBM)** model is described, which will be used as the FX rate model throughout the thesis.

The dynamics of the GBM model are given by

$$dX(t) = \mu X(t) dt + \sigma X(t) dW^{\mathbb{P}}(t),$$

where μ is the drift parameter, σ the volatility and \mathbb{P} indicates the real-world measure. A closed-form solution of the GBM model can be found by applying Itô's formula from Theorem 2.1.2 with $f(x) = \log(x)$, resulting in

$$\log(X(t)) = \log(X(0)) + \left(\mu - \frac{1}{2} \sigma^2 \right) t + \sigma W^{\mathbb{P}}(t),$$

$$X(t) = X(0) e^{(\mu - \frac{1}{2} \sigma^2) t + \sigma W^{\mathbb{P}}(t)}.$$

From the properties of the Brownian motion, we find that $X(t)$ is log-normally distributed, i.e.,

$$\log(X(t)) \sim N \left(\log(X(0)) + \left(\mu - \frac{1}{2} \sigma^2 \right) t, \sigma^2 t \right).$$

2.5. OPTION BASICS

This section covers the basic concepts that are used in option pricing, which will be used in Chapter 8 when the COS-CPD method is applied in the context of option pricing. The definitions presented are modified from [37] and [48].

The holder of an option contract has the right to trade in an underlying asset in the future at a predetermined price. As the name already indicates, the contract is characterised by the *optionality* of the contract. If the underlying assets does not perform in a favourable way, the holder can choose to not trade in the asset.

Options trading represents a significant portion of overall trading activity in many financial markets. The European call and put options are the most basic option which can be traded, and their definitions are presented here.

Definition 2.5.1 (European Call Option). *A European call option gives its holder the right (but not the obligation) to purchase from the writer a prescribed asset S , at a prescribed time in the future T , for a prescribed price K .*

Definition 2.5.2 (European Put Option). *A European put option gives its holder the right (but not the obligation) to sell the writer a prescribed asset S , at a prescribed time in the future T , for a prescribed price K .*

The prescribed time in future is called the *maturity* or *exercise time* T . The prescribed price is often referred to as the *strike price* K .

American-style options are another popular type of traded options. American options provide the holder with the flexibility to exercise the option at any time prior to the exercise time T , while European-style options can only be exercised at maturity. This added flexibility of American-style options often makes them more expensive than European-style options, as there is a greater possibility of the option being exercised. Within this thesis, only European-style options will be considered.

At maturity time T , the holder of the option can either exercise or can choose to not trade in the asset. Therefore, the payoff can be represented as is given in the next definition.

Definition 2.5.3 (Payoff European Call/Put Option). *The payoffs of a European call and put option at maturity T are given by*

$$H_{call}(T, S) := \max(S(T) - K, 0)$$

$$H_{put}(T, S) := \max(K - S(T), 0),$$

where $S(T)$ denotes the value of the underlying asset at maturity and K the strike price.

Options can exhibit a more complicated payoff function, which are commonly referred to as *exotic* options, due to the presence of exotic features in their payoff structure. Such options may incorporate path-dependent characteristics, whereby the final payoff depends on the movement of the stock price throughout the option's lifetime. Additionally, exotic options may involve multiple underlying assets. These options are exclusively traded in the OTC market, since they are customized by financial institutions to meet the specific requirements of their clients. Popular examples of exotic options are

- Barrier options: These options are dependent on whether or not the underlying asset price crosses a pre-defined barrier level during the option's lifetime.
- Asian options: These options have a payoff based on the average price of the underlying asset over a certain period of time.
- Lookback options: These options have a payoff based on the maximum or minimum price of the underlying asset over a certain period of time.

- Basket options: These options have a payoff based on multiple underlying assets.

In the field quantitative finance, a lot of research is devoted to find the fair value of the option before expiry time $t < T$. The word "fair" refers to the sense that it is acceptable by both the buyer and the seller of the option. One of the key principles on which option valuation theory rests is the no-arbitrage principle. The concept of arbitrage is presented in the following definition.

Definition 2.5.4 (Arbitrage). *An investment strategy is called an arbitrage if the value process V of the strategy satisfies the following two properties:*

- $\mathbb{P}(V_T > (1 + r)V_0) > 0$
- $\mathbb{P}(V_T \geq (1 + r)V_0) = 1$,

where r denotes the risk-free interest rate in a money-savings account and T the maturity time.

In other words, the no-arbitrage principle assures that there is never an opportunity to make a risk-free profit that gives a greater return than that provided by the interest from a bank deposit.

One of the most important results in option value theory is the derivation of the partial differential equation (PDE) by Fischer Black and Myron Scholes to price European options, published in 1973 [49]. Assuming that the asset price process S is modelled according to a geometric Brownian motion, the value of an European option can be represented as the solution of the following PDE,

$$\frac{\partial V}{\partial t} + rS \frac{\partial V}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rV = 0. \quad (2.6)$$

The derivation of the Black-Scholes PDE involves several steps, including the creation of a *replicating portfolio*. The concept of this replicating portfolio is to create a portfolio of the underlying asset and a risk-free asset that replicates the payoff of the option. In other words, by adjusting the amount of the underlying asset and the risk-free asset in the portfolio, one can create a combination of these two assets that has the same value as the option at any point in time. By the no-arbitrage principle, two portfolios with identical cash flows must have the same value. Therefore, the value of the option must be equal to the value of the replicating portfolio at any point in time. Using the replication strategy, Black and Scholes were able to derive the pricing PDE for European options.

An alternative approach to derive the Black-Scholes PDE is by applying the theory of martingales. This approach involves the construction of a *risk-neutral* or *equivalent martingale* probability measure, which is a probability measure under which the discounted asset price process is a martingale. The First Fundamental Theorem of Asset Pricing (FTAP I) ensures the existence of an equivalent martingale measure in case the no-arbitrage principle holds. By using this measure, the expected value of the option at maturity can be discounted to its value at time $t < T$,

$$\begin{aligned} V_t &= \mathbb{E}^{\mathbb{Q}} [e^{-r(T-t)} V_T] \\ &= \mathbb{E}^{\mathbb{Q}} [e^{-r(T-t)} H(S, T)], \end{aligned} \quad (2.7)$$

where $H(T, S)$ denotes the payoff function of the option. The measure \mathbb{Q} indicates that the expectation is taken under the risk-neutral measure.

The value of the option can be derived either by solving the pricing PDE presented in 2.6, or by evaluating the discounted expected payoff obtained from 2.7. By the no-arbitrage principle, these approaches should then lead to the same value. The equivalence of both approaches is proved in the famous Feynman-Kac theorem, which connects the discounted expectation as the closed-form solution to the Black-Scholes PDE.

2.6. THE COS METHOD

The COS method was introduced in the context of option pricing by Fang and Oosterlee in [30]. However, the essence of the method is to recover the unknown **Probability Density Function (PDF)** from its Fourier-cosine series expansion, with the key insight that the series coefficients are almost readily available from the characteristic function, which is very often easier to derive than the density function itself. Therefore, the COS method can also be used in the context of CCR to compute the exposure density function of a derivative portfolio. Once this exposure density function is computed, CRR measures such as the PFE and EE can be derived from it.

The COS method belongs to the family of Fourier-based methods. These methods are all based on the fact that the density function and its corresponding characteristic function, $f(x)$ and $\phi(\omega)$, are an example of a Fourier pair,

$$\phi(\omega) = \int_{\mathbb{R}} e^{i\omega x} f(x) dx \quad (2.8)$$

$$f(x) = \frac{1}{2\pi} \int_{\mathbb{R}} e^{-i\omega x} \phi(\omega) d\omega \quad (2.9)$$

As already mentioned, the COS method recovers the density function by computing its Fourier-cosine series expansion. Any real function has a cosine expansion when it is finitely supported. Furthermore, the Fourier-cosine series expansion gives an optimal approximation of functions with a finite support [50]. Therefore, the derivation starts with a truncation of the infinite integration range in 2.9. The existence of a Fourier transform implies that the integrands in 2.9 have to decay to zero at $\pm\infty$, so we can truncate the integration range in a proper way without losing accuracy. Suppose this truncation domain is given by $[a, b] \in \mathbb{R}$. The Fourier-cosine series expansion on $[a, b]$ is then of the form

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

with the coefficients given by

$$A_k = \frac{2}{b-a} \int_a^b f(x) \cdot \cos\left(k\pi \frac{x-a}{b-a}\right) dx. \quad (2.10)$$

The symbol \sum' indicates that the first term in the summation is weighted by one-half. The key part of the COS method lies in the connection between the Fourier-cosine coefficients A_k and the characteristic function. Since the density function is truncated to

the finite interval $[a, b]$, the corresponding characteristic function has to be computed on this finite interval as well. The density only has a small loss of accuracy on $[a, b]$, so that the truncated integral approximates the infinite counterpart in 2.8 very well, i.e.,

$$\hat{\phi}(\omega) := \int_a^b e^{i\omega x} f(x) dx \approx \int_{\mathbb{R}} e^{i\omega x} f(x) dy = \phi(\omega) \quad (2.11)$$

Comparing 2.11 with the coefficients A_k in 2.10, we obtain

$$A_k \equiv \frac{2}{b-a} \operatorname{Re} \left\{ \hat{\phi} \left(\frac{k\pi}{b-a} \right) \cdot \exp \left(-i \frac{k a \pi}{b-a} \right) \right\}, \quad (2.12)$$

with $\operatorname{Re}\{\cdot\}$ denotes the real part of the argument. From 2.11 we have $\hat{\phi}(\omega) \approx \phi(\omega)$, resulting in $A_k \approx F_k$ with

$$F_k \equiv \frac{2}{b-a} \operatorname{Re} \left\{ \phi \left(\frac{k\pi}{b-a} \right) \cdot \exp \left(-i \frac{k a \pi}{b-a} \right) \right\}, \quad (2.13)$$

where $\phi(\omega)$ denotes the characteristic function on the infinite domain. Therefore, the COS method provides the following formula for approximating the density function,

$$\hat{f}(x) \approx \sum_{k=0}^{\infty} F_k \cos \left(k\pi \frac{x-a}{b-a} \right),$$

and truncate the summation, so that

$$\hat{f}(x) \approx \sum_{k=0}^{N-1} F_k \cos \left(k\pi \frac{x-a}{b-a} \right). \quad (2.14)$$

Since the Fourier-cosine series expansions of so-called entire functions² exhibit an exponential convergence [50], we can expect that the formula in 2.14 gives a highly accurate approximation, already for small value for N , to density functions that have no singularities on $[a, b]$.

The **Cumulative Distribution Function (CDF)** can also be found with the COS method. The COS formula for the CDF can be directly derived from the PDF expression in 2.14. To do this, the definition of the CDF is used, which is given as the integral of the probability density function,

$$F(x) = \int_{-\infty}^x f(t) dt$$

Similar to the PDF calculation, the infinite interval is truncated to $[a, x]$. Then, the COS formula for the PDF can be inserted to obtain

$$\begin{aligned} F(x) &\approx \int_a^x f(t) dt \\ &\approx \int_a^x \sum_{k=0}^{\infty} F_k \cos \left(k\pi \frac{t-a}{b-a} \right) dt \end{aligned}$$

²Entire functions are functions without any singularities anywhere in the complex plane, except at ∞ .

Since the cosine basis functions are bounded, Fubini's theorem allows to interchange integration and summation,

$$\hat{F}(x) = \sum_{k=0}^{\infty} F_k \int_a^x \cos\left(k\pi \frac{t-a}{b-a}\right) dt$$

When integrating the cosine basis functions, the value $k = 0$ has to be treated separately, as the basis function takes the form $\cos(0 \cdot \pi \frac{x-a}{b-a}) = 1$ in this case. Doing so, we find

$$\int_a^x \cos\left(k\pi \frac{t-a}{b-a}\right) dt = \begin{cases} (x-a) & \text{if } k=0, \\ \frac{b-a}{k\pi} \sin\left(k\pi \frac{x-a}{b-a}\right) & \text{if } k \neq 0. \end{cases}$$

Therefore, the COS formula to recover the CDF is given by

$$\hat{F}(x) = \frac{F_0}{2}(x-a) + \sum_{k=1}^{\infty} F_k \frac{b-a}{k\pi} \sin\left(k\pi \frac{x-a}{b-a}\right).$$

Again, the series summation is truncated to end up with

$$\hat{F}(x) = \frac{F_0}{2}(x-a) + \sum_{k=1}^{N-1} F_k \frac{b-a}{k\pi} \sin\left(k\pi \frac{x-a}{b-a}\right) \quad (2.15)$$

From this, it can be seen that the series coefficients F_k can be used to recover both the PDF and CDF. Therefore, the goal within the COS method is to obtain the coefficients F_k in a fast and accurate way, which essentially boils down to the computation of the characteristic function. In the context of exposure distributions, it turns out that both the CDF and PDF will be of importance when computing the PFE and EE. Once the coefficients F_k are computed, the exposure CDF will be used to find the PFE, while the EE can be obtained in an efficient way using the exposure PDF. Chapter 5 will focus on the computation of the characteristic function to obtain the coefficients F_k , from which the exposure distribution can be recovered using 2.14 and 2.15.

3

CANONICAL POLYADIC DECOMPOSITION (CPD)

This chapter introduces the concept of [Canonical Polyadic Decomposition \(CPD\)](#) and how it can be applied to the computation of Fourier-cosine series expansions. As a tensor decomposition technique, CPD aims to reduce the computational complexity that arises from calculations involving multi-dimensional tensors. The CPD process represents a high-dimensional tensor as a product of smaller and simpler factors, making it more efficient to perform calculations. As a result, CPD is effectively used for dimension reduction in multi-dimensional problems. Along CPD, there are many other decomposition techniques in literature, such as the [Higher-Order Singular Value Decomposition \(HOSVD\)](#). The technique of tensor decomposition has been successfully applied in various fields, including as signal processing, statistics and machine learning [51]–[53].

This chapter begins with a basic introduction to the field of tensors, followed by a section on the methodology of CPD. Finally, the application of CPD in the context of Fourier-cosine series expansions is presented to demonstrate its ability to reduce the computational complexity of constructing high-dimensional Fourier-cosine series.

3.1. TENSOR DEFINITIONS AND NOTATION

This section covers some fundamental definitions and concepts in tensor calculus, which will be of importance when the CPD technique is explained in the next section. The content in this section is based on [54] and [55].

A tensor is a multi-dimensional array. The number of dimensions (ways) of a tensor is its *order*, often denoted by N . Each dimension (way) is called a *mode*. The n th-mode of a tensor is often referred to as mode- n . For example, a scalar can be described as a zero-order tensor, a vector is a first-order tensor, and a matrix is a second-order tensor. Tensors of order three or higher are called higher-order tensors. An N th-order tensor \mathcal{X} needs N sets of indices $\{i_n\}$, $n = 1, \dots, N$ to indicate one specific element. Each index i_n runs from 1 to I_n , addressing the mode- n of \mathcal{X} . As a convention, a N th-order tensor can

be explicitly denoted as $\mathcal{X} = \mathbb{R}^{I_1 \times \dots \times I_N}$, where I_n for $n = 1, \dots, N$ directly indicates the number of elements in the n -th mode.

Another concept that has to be introduced is one of the tensor *fibers*. Fibers can be seen as the higher-order analogue of matrix rows and columns. A fiber is defined by fixing every index but one. When extracted from the tensor, fibers are always assumed to be oriented as column vectors. For example, a matrix column is a mode-1 fiber and a matrix row is a mode-2 fiber. Third-order tensors have column, row, and tube fibers, as is illustrated in Figure 3.1.

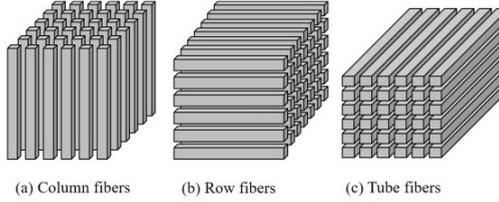


Figure 3.1: Fibers of a third-order tensor [56]

Higher-order tensor calculations can be quite involved and abstract. Therefore, tensors are often transformed into a matrix form, on which well-known matrix operations can be applied. Reordering the elements of a N th-order tensor array into a matrix is called *unfolding* or *matricization*. For instance, a $2 \times 3 \times 4$ tensor can be arranged as a 6×4 , 2×12 or 3×8 matrix. There exists different ways of unfolding a tensor in a matrix, but for the applications in this thesis only the *mode- n unfolding* is relevant to discuss. A more general treatment of unfolding techniques can be found in [57]. The mode- n unfolding of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ is denoted by $\mathcal{X}_{(n)}$ and arranges the mode- n fibers to be the columns of the resulting matrix. Since the unfolding technique can be quite abstract when explaining it for a general N th-order tensor, the idea of unfolding will be illustrated with an example of a third-order tensor. Let $\mathcal{X} \in \mathbb{R}^{4 \times 3 \times 2}$ be a third-order tensor with elements given by $\mathcal{X}[i, j, k] = x_{ijk}$. The mode-1 unfolding is found by taking the column fibers (see Figure 3.1a) as the columns of the resulting matrix, which results in a matrix of size 4×6 ,

$$\mathcal{X}_{(1)} = \begin{bmatrix} x_{111} & x_{121} & x_{131} & x_{112} & x_{122} & x_{132} \\ x_{211} & x_{221} & x_{231} & x_{212} & x_{222} & x_{232} \\ x_{311} & x_{321} & x_{331} & x_{312} & x_{322} & x_{332} \\ x_{411} & x_{421} & x_{431} & x_{412} & x_{422} & x_{432} \end{bmatrix}$$

Similarly, the mode-2 unfolding is constructed by taking the row fibers (see Figure 3.1b) as the columns of the resulting matrix, which gives a matrix of size 3×8 ,

$$\mathcal{X}_{(2)} = \begin{bmatrix} x_{111} & x_{211} & x_{311} & x_{411} & x_{112} & x_{212} & x_{312} & x_{412} \\ x_{121} & x_{221} & x_{321} & x_{421} & x_{122} & x_{222} & x_{322} & x_{422} \\ x_{131} & x_{231} & x_{331} & x_{431} & x_{132} & x_{232} & x_{332} & x_{432} \end{bmatrix}$$

Finally, the mode-3 unfolding is found by taking the tube fibers (see Figure 3.1c) as the columns of the resulting matrix, obtaining a matrix of size 2×12 ,

$$\mathcal{X}_{(3)} = \begin{bmatrix} x_{111} & x_{211} & x_{311} & x_{411} & x_{121} & x_{221} & x_{321} & x_{421} & x_{131} & x_{231} & x_{331} & x_{431} \\ x_{112} & x_{212} & x_{312} & x_{412} & x_{122} & x_{222} & x_{322} & x_{422} & x_{132} & x_{232} & x_{332} & x_{432} \end{bmatrix}$$

The ordering of the columns for the mode- n unfolding used in literature can be different. However, the specific permutation of columns is not important so long as it is consistent across related calculations.

The process of unfolding transforms a tensor into a matrix form, allowing for the use of well-known (and less abstract) matrix operations. Several matrix products will be of importance in the sections that follow, and their definitions will be introduced here.

Definition 3.1.1 (Outer Product). *Let $\mathbf{u} \in \mathbb{R}^m$, $\mathbf{v} \in \mathbb{R}^n$ be two vectors. Their outer product is denoted with $\mathbf{u} \circ \mathbf{v} \in \mathbb{R}^{m \times n}$, and the resulting matrix can be obtained by multiplying each element of \mathbf{u} by each element of \mathbf{v} ,*

$$\begin{aligned} \mathbf{u} \circ \mathbf{v} &= \begin{bmatrix} u_1 v_1 & u_1 v_2 & \cdots & u_1 v_n \\ u_2 v_1 & u_2 v_2 & \cdots & u_2 v_n \\ \vdots & \vdots & \ddots & \vdots \\ u_m v_1 & u_m v_2 & \cdots & u_m v_n \end{bmatrix} \\ &= [\mathbf{u} \cdot v_1 \quad \mathbf{u} \cdot v_2 \quad \cdots \quad \mathbf{u} \cdot v_n] \end{aligned}$$

Definition 3.1.2 (Kronecker Product). *The Kronecker product of matrices $A \in \mathbb{R}^{I \times J}$ and $B \in \mathbb{R}^{K \times L}$ is denoted with $A \otimes B \in \mathbb{R}^{IK \times JL}$, and is defined by*

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1J}B \\ a_{21}B & a_{22}B & \cdots & a_{2J}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}B & a_{I2}B & \cdots & a_{IJ}B \end{bmatrix}$$

Definition 3.1.3 (Khatri-Rao Product). *The Khatri-Rao product can be viewed as the column-wise Kronecker product. Given the matrices $A \in \mathbb{R}^{I \times K}$ and $B \in \mathbb{R}^{J \times K}$, the Khatri-Rao product, denoted with $A \odot B \in \mathbb{R}^{IJ \times K}$, is defined by*

$$A \odot B = [a_1 \otimes b_1 \quad a_2 \otimes b_2 \quad \cdots \quad a_K \otimes b_K]$$

Definition 3.1.4 (Hadamard Product). *The Hadamard product is the element-wise matrix product. Therefore, given the matrices $A, B \in \mathbb{R}^{I \times J}$, the Hadamard product $A \circledast B \in \mathbb{R}^{I \times J}$ produces the matrix*

$$A \circledast B = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} & \cdots & a_{1J}b_{1J} \\ a_{21}b_{21} & a_{22}b_{22} & \cdots & a_{2J}b_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}b_{I1} & a_{I2}b_{I2} & \cdots & a_{IJ}b_{IJ} \end{bmatrix}$$

The final definition presented in this section refers to the Frobenius norm of a tensor, which is similar to the matrix Frobenius norm. The Frobenius norm frequently appears in optimization problems regarding tensor decomposition, as the objective function is often minimized in terms of the Frobenius norm. This will also be the case in the CPD method described in the next section, as can be seen in 3.3.

Definition 3.1.5 (Frobenius Norm). *Given a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$, its Frobenius norm, often abbreviated with F -norm, is defined as the square root of the sum of the squares of all its elements:*

$$\|\mathcal{X}\|_F = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \dots \sum_{i_N=1}^{I_N} x_{i_1 i_2 \dots i_N}^2}$$

3.2. INTRODUCTION TO CPD

This section provides the general idea of the CPD technique. These results will be used in the subsequent sections, where CPD is applied in the context of Fourier-cosine series. Within the CPD approach, rank-one tensors play a central role. A N th-order tensor \mathcal{X} is called a rank-one tensor if it can be written as an outer product of N vectors, i.e.,

$$\mathcal{X} = a_1 \circ a_2 \circ \dots \circ a_N.$$

With CPD, a N th-order tensor is factorised into a sum of component rank-one tensors,

$$\mathcal{X} = [\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_N]_R = \sum_{r=1}^R a_r^1 \circ a_r^2 \circ \dots \circ a_r^N, \quad (3.1)$$

where $\mathbf{A}_n = [a_1^n, \dots, a_R^n] \in \mathbb{R}^{I_n \times R}$ for $n = 1, \dots, N$. The rank R is defined as the minimum number of components needed to synthesize \mathcal{X} . This minimal rank is often called the canonical rank. Every tensor admits a CPD of finite rank, and is unique under mild conditions [51]. However, determining this canonical rank is a NP-hard problem, implying that there is no algorithm to find its value [58]. In practice, a lower rank \tilde{R} is used to approximate the original tensor,

$$\mathcal{X} \approx \sum_{r=1}^{\tilde{R}} a_r^1 \circ a_r^2 \circ \dots \circ a_r^N.$$

These lower-rank CPD models are widely applied to break the curse of dimensionality. The expression in 3.1 can also be written element-wise as follows,

$$\mathcal{X}[i_1, \dots, i_N] = \sum_{r=1}^R \prod_{n=1}^N \mathbf{A}_n[i_n, r] = \sum_{r=1}^R \prod_{n=1}^N a_r^n[i_n], \quad (3.2)$$

which will be useful in the context of the Fourier-cosine series in the subsequent sections.

To obtain a CPD of a given tensor, the factor matrices $\{\mathbf{A}_n\}_{n=1}^N$ have to be solved. This can be done by minimizing the least square error between the original tensor and the

CPD approximation,

$$\min_{\{\mathbf{A}_n\}_{n=1}^N} \left\| \mathcal{X} - \llbracket \mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_N \rrbracket_R \right\|_F^2 = \min_{\{\mathbf{A}_n\}_{n=1}^N} \left\| \mathcal{X} - \sum_{r=1}^R \mathbf{a}_r^1 \circ \mathbf{a}_r^2 \circ \dots \circ \mathbf{a}_r^N \right\|_F^2, \quad (3.3)$$

where F denotes the Frobenius norm as defined in 3.1.5. Initially, this minimization problem seems sophisticated, as the objective function includes N matrices to solve for. However, this optimization problem consists of a multilinear form, and a very popular method to solve such problems is called **Alternating Least Squares (ALS)**. The key idea of ALS is to update the variables cyclically, while keeping all but one fixed. To illustrate this idea, we first combine the technique of tensor unfolding and Khatri-Rao product to construct a connection between tensor modes and factor matrices [51],

$$\mathcal{X}_{(n)} = \mathbf{A}_n (\odot_{i \neq n} \mathbf{A}_i)^T, \quad n = 1, \dots, N,$$

where

$$(\odot_{i \neq n} \mathbf{A}_i) = \mathbf{A}_N \odot \dots \odot \mathbf{A}_{n+1} \odot \mathbf{A}_{n-1} \odot \dots \odot \mathbf{A}_1.$$

Minimizing the Frobenius norm of these N tensor modes is equivalent to minimizing 3.3 and therefore we can consider the following minimization problems,

$$\min_{\{\mathbf{A}_n\}_{n=1}^N} \left\| \mathcal{X}_{(n)} - \mathbf{A}_n (\odot_{i \neq n} \mathbf{A}_i)^T \right\|_F^2, \quad n = 1, \dots, N.$$

Note that the problem above is multilinear in the factor matrices $\{\mathbf{A}_n\}_{n=1}^N$. However, we can apply the ALS algorithm by updating the factor matrices iteratively. Fixing all factor matrices except \mathbf{A}_n , the problem becomes (conditionally) linear in \mathbf{A}_n , so that we may update

$$\mathbf{A}_n \leftarrow \operatorname{argmin}_{\mathbf{A}_n} \left\| \mathcal{X}_{(n)} - \mathbf{A}_n (\odot_{i \neq n} \mathbf{A}_i)^T \right\|_F^2.$$

In this way, the factor matrices are updated one by one until the stopping criterion is satisfied.

3.3. OUR CONTRIBUTION 1: DIMENSION-REDUCED FOURIER-COSINE SERIES EXPANSIONS VIA CPD

Having introduced the concept of CPD, this section describes how it can be applied to reduce the dimensionality of Fourier-cosine series expansions for multi-variate functions. Therefore, some general theory on Fourier-cosine series will be presented first.

For a one-dimensional function $f(x)$, its Fourier-cosine expansion in the interval $[a, b]$ is given by

$$f(x) = \sum_{k=0}^{\infty} \alpha_k \cos\left(k\pi \frac{x-a}{b-a}\right),$$

with the coefficients α_k being defined by the following one-dimensional integrals,

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

The prime at the sum, Σ' , indicates the division by two of the first term in the summation. In practice, the Fourier-cosine expansion is truncated, resulting in an approximation of the function, i.e.

$$f(x) \approx \sum'_{k=0}^{K-1} \alpha_k \cos\left(k\pi \frac{x-a}{b-a}\right).$$

In this one-dimensional framework, the Fourier coefficients can be stored as a first-order tensor (or vector) $\alpha = [\alpha_0, \dots, \alpha_{K-1}]$.

Fourier-cosine expansions can also be used to approximate multivariate functions. In the two-dimensional case, the Fourier-cosine expansion of the function $f(x_1, x_2)$ on $[a_1, b_1] \times [a_2, b_2]$ has the form,

$$f(x_1, x_2) \approx \sum'_{k_1=0}^{K-1} \sum'_{k_2=0}^{K-1} \alpha_{k_1, k_2} \cos\left(k_1\pi \frac{x_1 - a_1}{b_1 - a_1}\right) \cos\left(k_2\pi \frac{x_2 - a_2}{b_2 - a_2}\right)$$

with the coefficients now being defined by the two-dimensional integrals given by

$$\alpha_{k_1, k_2} = \frac{2}{b_1 - a_1} \frac{2}{b_2 - a_2} \int_{a_1}^{b_1} \int_{a_2}^{b_2} f(x_1, x_2) \cos\left(k_1\pi \frac{x_1 - a_1}{b_1 - a_1}\right) \cos\left(k_2\pi \frac{x_2 - a_2}{b_2 - a_2}\right) dx_1 dx_2.$$

In this case, the Fourier coefficients can be stored as a second-order tensor (or matrix)

$$\alpha = \begin{bmatrix} \alpha_{0,0} & \alpha_{0,1} & \dots & \alpha_{0,K-1} \\ \alpha_{1,0} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \alpha_{K-1,0} & \dots & \dots & \alpha_{K-1,K-1} \end{bmatrix}$$

In general, a N -dimensional Fourier-cosine expansion on the domain $[a_1, b_1] \times \dots \times [a_N, b_N]$ is given by

$$f(x_1, \dots, x_N) \approx \sum'_{k_1=0}^{K-1} \dots \sum'_{k_N=0}^{K-1} \alpha_{\mathbf{k}} \prod_{n=1}^N \cos\left(k_n\pi \frac{x_n - a_n}{b_n - a_n}\right), \quad (3.4)$$

where $\mathbf{k} = (k_1, \dots, k_N) \in \mathbb{N}^N$ is a multi-index. The coefficients $\alpha_{\mathbf{k}}$ are defined by the following N -dimensional integrals,

$$\alpha_{\mathbf{k}} = \prod_{n=1}^N \left(\frac{2}{b_n - a_n} \right) \int_{a_1}^{b_1} \dots \int_{a_N}^{b_N} f(x_1, \dots, x_N) \prod_{n=1}^N \cos\left(k_n\pi \frac{x_n - a_n}{b_n - a_n}\right) dx_1 \dots dx_N.$$

The Fourier coefficients can now be represented as a N th-order tensor, with $\alpha \in \mathbb{R}^{K \times K \times \dots \times K}$. These derivations show that the number of Fourier coefficients scales exponentially with

$\mathcal{O}(K^N)$. Furthermore, an increase in dimensionality also adds an additional computational cost of the coefficients itself, as the number of dimensions of integration grows.

To avoid the exponential growth in the computational complexity for increasing N , the CPD technique can be used to approximate the Fourier coefficient tensor with a sum of rank-1 tensors,

$$\alpha \approx \llbracket \mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_N \rrbracket_R = \sum_{r=1}^R a_r^1 \circ a_r^2 \circ \dots \circ a_r^N.$$

Consequently, the CPD approximation is obtained by computing the N factor matrices $\{\mathbf{A}_n\}_{n=1}^N$ of sizes $K \times R$. Therefore, the computational complexity of computing the Fourier coefficients drops from $\mathcal{O}(K^N)$ to $\mathcal{O}(NKR)$, which makes this approach suitable for computing multi-dimensional Fourier-cosine series. The methodology of computing the factor matrices will be presented in Section 3.4.

Once the factor matrices are computed, the Fourier coefficient tensor can be written as in Equation 3.2 to obtain

$$\alpha[k_1, \dots, k_N] = \sum_{r=1}^R \prod_{n=1}^N a_r^n[k_n].$$

This expression can now be inserted into 3.4,

$$f(x_1, \dots, x_N) \approx \sum_{k_1=0}^{K-1} \dots \sum_{k_N=0}^{K-1} \sum_{r=1}^R \prod_{n=1}^N a_r^n[k_n] \cos\left(k_n \pi \frac{x_n - a_n}{b_n - a_n}\right). \quad (3.5)$$

To simplify the notation, we define $\mathbf{v}_n[k_n] := \cos\left(k_n \pi \frac{x_n - a_n}{b_n - a_n}\right)$. Substituting this into 3.5, we have

$$\begin{aligned} f(x_1, \dots, x_N) &\approx \sum_{k_1=0}^{K-1} \dots \sum_{k_N=0}^{K-1} \sum_{r=1}^R \prod_{n=1}^N a_r^n[k_n] \mathbf{v}_n[k_n] \\ &= \sum_{r=1}^R \sum_{k_2, \dots, k_N=0}^{K-1} \prod_{n=2}^N a_r^n[k_n] \mathbf{v}_n[k_n] \underbrace{\sum_{k_1=0}^{K-1} a_r^1[k_1] \mathbf{v}_1[k_1]}_{f_{1,r}(x_1)}. \end{aligned}$$

By carrying out the computations in this manner, the expression for $f(\cdot)$ is greatly simplified to

$$f(x_1, \dots, x_N) \approx \sum_{r=1}^R f_{1,r}(x_1) \dots f_{N,r}(x_N) \quad (3.6)$$

Each univariate function in 3.6 can be expressed as

$$f_{n,r} = \sum_{k_n=0}^{K-1} a_r^n[k_n] \mathbf{v}_n[k_n] = \mathbf{v}_n^T \mathbf{A}_n[:, r], \quad (3.7)$$

showing that each component $f_{n,r}$ represents an univariate Fourier-cosine series. Consequently, the Fourier-cosine series found with CPD can be expressed as a product of univariate Fourier-cosine series. This is different from the standard Fourier-series expansion, which results in N nested summations over all the expansion terms. The absence of these nested summations in the CPD expression will have a great computational benefit when evaluating expectations of multi-variate functions, such as the characteristic function in the context of exposure and option price computations. This will be made more clear in the corresponding chapters. Combining 3.6 and 3.7, the output of the constructed Fourier-cosine series for a given input vector (x_1, \dots, x_N) can be written in terms of the Hadamard product,

$$\hat{y} = (\mathbf{v}_1^T \mathbf{A}_1 \otimes \dots \otimes \mathbf{v}_N^T \mathbf{A}_N) \mathbf{1} = (\otimes_{n=1}^N \mathbf{v}_n^T \mathbf{A}_n) \mathbf{1}. \quad (3.8)$$

3.4. HIDDEN TENSOR FACTORIZATION

This section outlines the procedure for deriving the factor matrices $\{\mathbf{A}_n\}_{n=1}^N$ to construct the CPD model of the Fourier coefficient tensor, which has been developed in our working paper at FF Quant [59] and [60].

Normally, the CPD technique is applied to the full tensor on which the decomposition has to be performed. However, this approach is not feasible for the Fourier coefficient tensor, as it requires to still compute all the Fourier coefficients first, which takes $\mathcal{O}(K^N)$ computations to solve the N -dimensional integrals. This will become impracticable for large values of N . Therefore, we want a method to construct the factor matrices \mathbf{A}_n without computing the full tensor first. Kargas and Sidiropoulos presented a way of avoiding the computation of the full tensor in [61]. The idea is to transform the problem into a regression model, in which the distance between the exact function values and the truncated Fourier-cosine series values is minimized,

$$\min \frac{1}{M} \sum_{m=1}^M L(y_m - f(\mathbf{x}_m)),$$

where $L(\cdot)$ denotes the loss function, y_m the exact function value in the point \mathbf{x}_m and $f(\mathbf{x}_m)$ the approximated function value from the truncated Fourier-cosine series. Furthermore, M represents the total number of sampling points used to train the function $f(\cdot)$. With this approach, the Fourier coefficients are fitted directly on the training data. The key now is to replace the original Fourier coefficient tensor by the CPD representation in Equation 3.8, where we use the squared error as the loss function to obtain,

$$\min_{\{\mathbf{A}_n\}_{n=1}^N} \frac{1}{M} \sum_{m=1}^M (y_m - (\otimes_{n=1}^N (\mathbf{V}_n[:, m]^T \mathbf{A}_n)) \mathbf{1})^2,$$

where $\mathbf{V}_n \in \mathbb{R}^{K \times M}$ with $\mathbf{V}_n[k, m] = \cos\left(k\pi \frac{x_n^m - a_n}{b_n - a_n}\right)$. The value x_n^m denotes the value of the n th coordinate in sampling point \mathbf{x}_m . The ALS algorithm can now be exploited to solve the factor matrices iteratively. Fixing all factor matrices except for \mathbf{A}_n we have,

$$\min_{\mathbf{A}_n} \frac{1}{M} \sum_{m=1}^M (y_m - \mathbf{V}_n[:, m]^T \mathbf{A}_n \mathbf{Q}_n[:, m])^2, \quad (3.9)$$

where $\mathbf{Q}_n \in \mathbb{R}^{R \times M}$ with

$$\mathbf{Q}_n = (\otimes_{i \neq n} (\mathbf{A}_i^T \mathbf{V}_i)). \quad (3.10)$$

Solving this minimization problem generates the CPD model of the Fourier-cosine series without pre-calculating the full coefficient tensor. Therefore, this approach is referred to as the [Fourier Series Approximation via Hidden Tensor Factorization \(FSA-HTF\)](#), as it bypasses the population of the full coefficient tensor. The minimization problem in 3.9 is conceptually similar to a linear least squares problem, which belongs to the family of convex optimization problems. Therefore, the (global) minimum is found by setting the gradient equal to zero,

$$\frac{2}{M} \sum_{m=1}^M (y_m - \mathbf{V}_n[:, m]^T \mathbf{A}_n \mathbf{Q}_n[:, m]) \cdot -\mathbf{V}_n[:, m] \mathbf{Q}_n^T[:, m] = 0.$$

This leads to the following system of linear equations that needs to be solved to obtain the factor matrices $\{\mathbf{A}_n\}_{n=1}^N$,

$$\frac{1}{M} \sum_{m=1}^M (\mathbf{V}_n[:, m]^T \mathbf{A}_n \mathbf{Q}_n[:, m]) \mathbf{V}_n[:, m] \mathbf{Q}_n^T[:, m] = \frac{1}{M} \sum_{m=1}^M y_m \mathbf{V}_n[:, m] \mathbf{Q}_n^T[:, m]. \quad (3.11)$$

Note that 3.11 is not a standard linear system of the form $Ax = b$, which has to be solved for the unknown vector x . Instead, the system can be seen as a matrix equation, in which the unknown matrix \mathbf{A}_n has to be solved. A previous thesis done at FF Quant showed how to transform 3.11 into a equation of the form $Ax = b$ by vectorizing the factor matrix \mathbf{A}_n [60]. Furthermore, it was shown that the [Conjugate Gradient \(CG\)](#) method converges faster and is more accurate compared to the [Stochastic Gradient Descent \(SGD\)](#) method, which is normally a popular method for solving optimization problems in machine learning. Therefore, the CG method will also be used throughout this thesis to solve 3.11. The pseudo code for computing the factor matrices $\{\mathbf{A}_n\}_{n=1}^N$ using the FSA-HTF approach is provided in Algorithm 1. This code will be employed in the next chapter, when an extensive numerical analysis is performed on the dimension-reduced Fourier-cosine series via CPD.

Algorithm 1 FSA-HTF

Input: $K, R, M, \mathbf{a}, \mathbf{b}, \max_{CG}, \max_{ALS}, \varepsilon_{CG}, \varepsilon_{ALS}$
 Generate training grid \mathbf{X} on $[\mathbf{a}, \mathbf{b}]$ with M points
 Compute exact function values \mathbf{y} on the grid \mathbf{X}
 Initialize $\{\mathbf{A}_n \in \mathbb{R}^{K \times R}\}_{n=1}^N$
 Generate $\{\mathbf{V}_n \in \mathbb{R}^{K \times M}\}_{n=1}^N$
while $it_{ALS} < \max_{ALS}$ and error in factor matrices $> \varepsilon_{ALS}$ **do**
 for $n = 1$ **to** N **do**
 Compute \mathbf{Q}_n via (3.10)
 while $it_{CG} < \max_{CG}$ and CG error $> \varepsilon_{CG}$ **do**
 Compute \mathbf{A}_n with CG via (3.11)
 Update CG error
 end while
 end for
 Update error in factor matrices
end while
return $\{\mathbf{A}_n\}_{n=1}^N$

4

CONVERGENCE TESTS FOR CPD

In this chapter, an extensive numerical analysis is conducted on the application of Algorithm 1 derived in the previous chapter to construct Fourier-cosine series via CPD. The goal is to get an insight in the various parameters involved in the algorithm and to quantify their effect on accuracy when altered. This information will be used in future chapters when the CPD algorithm is applied in the context of exposure calculation and option pricing.

The first section examines the impact of overfitting the model, which is a common issue in supervised learning methods. A correlation is established between the number of expansion terms in the Fourier-cosine series and the number of training data points required to achieve optimal fitting with the CPD algorithm. Subsequently, the effect of the sampling frequency of the training data on accuracy levels and different types of functions is analyzed.

The remainder of this chapter focuses on a sensitivity analysis of the CPD algorithm to evaluate the impact on the accuracy of the reconstructed Fourier-cosine series when different parameter settings are used. The analysis is divided into examples in the one-, two- and multi-dimensional case, each of which applies the CPD algorithm to the (standard) normal density function.

4.1. PREVENT OVERFITTING

The CPD algorithm described in Chapter 3 is a supervised learning technique for approximating the Fourier coefficient tensor. In the context of supervised learning, the model is built by minimizing the misfit between the training data and the model output, which results in a minimization problem. For the coefficient tensor, this minimization problem can be expressed as

$$\min_{\alpha_{\mathbf{k}}} \frac{1}{M} \sum_{m=1}^M \left(y_m - \left[\sum_{k_1=0}^{K-1} \cdots \sum_{k_N=0}^{K-1} \alpha_{\mathbf{k}} \prod_{n=1}^N \cos \left(k_n \pi \frac{x_n^m - a}{b-a} \right) \right] \right)^2, \quad (4.1)$$

where $\mathbf{k} = (k_1, \dots, k_N)$ is a multi-index and x_n^m the value of the n th coordinate in sampling point \mathbf{x}_m . The objective is to minimize the sum of squares of the total misfits between the training data values y_m assigned to the sample \mathbf{x}_m , and the values of the Fourier-cosine series expansion evaluated at the same sample \mathbf{x}_m . The minimization problem in 4.1 involves K^N Fourier coefficients for which the minimization problem has to be solved. Therefore, the number K^N can be viewed as the degree of "flexibility" the model has in minimizing the misfit with the training data. The number of training data points is denoted with M . For the remainder of the analysis that follows, we will assume that the number of training points is equal for each dimension. Hence, the total number of training points can be written as $M = m^N$, with m being the number of training points per dimension. The value M represents the level of "constraint" within the model, as the model tries to match these training data points when solving the minimization problem. Three parameter settings can now be selected in the CPD algorithm when solving the Fourier coefficient tensor:

4

- **$K < m$.** In this case, there is not enough flexibility in the model to match all the training data points. The CPD model can only minimize the distance between the obtained Fourier-cosine series and the training data points. An example of this scenario is shown in Figure 4.1. The output of the CPD model is derived using Equation 3.8. However, it should be noted that the CPD model for this scenario is still applicable to the overall function, making it a suitable approximation for the non-training data.

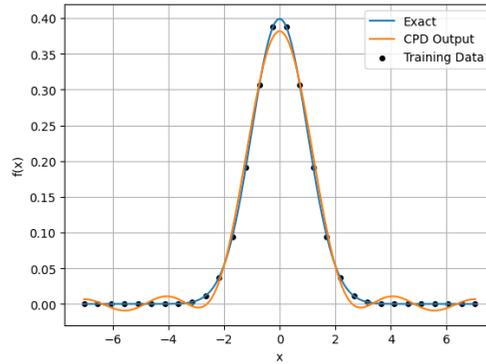


Figure 4.1: Reconstructed Fourier-cosine series of the density function of a standard normal random variable using the CPD algorithm in the scenario $K < m$. The training data points used to construct the CPD approximation are indicated with the black dots. For this example, $K = 10$ and $m = 30$ were used.

- **$K > m$.** In this scenario, the CPD model possesses sufficient flexibility to accurately fit all of the training data. However, there is extra flexibility left to produce weird behaviour between the training data points, as can be seen in Figure 4.2. The obtained Fourier-cosine series does not represent the global function anymore, but only matches in the training data points. This phenomenon is known as *overfitting*, and is an important aspect to be aware of when using supervised learning methods.

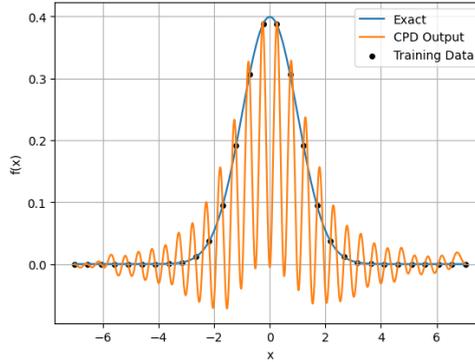


Figure 4.2: Reconstructed Fourier-cosine series of the density function of a standard normal random variable using the CPD algorithm in the scenario $K > m$. The training data points used to construct the CPD approximation are indicated with the black dots. For this example, $K = 60$ and $m = 30$ were used.

- **$K = m$.** The degree of flexibility in the model corresponds to the degree of constraints, resulting in a fit that aligns with all training data points and provides an approximation of the global function. This is also illustrated in Figure 4.3.

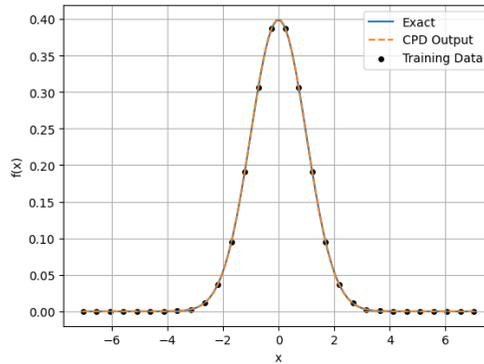


Figure 4.3: Reconstructed Fourier-cosine series of the density function of a standard normal random variable using the CPD algorithm in the scenario $K = m$. The training data points used to construct the CPD approximation are indicated with the black dots. For this example, $K = m = 30$ was used.

Based on these observations, it is concluded that the optimal parameter choice in the CPD algorithm is to take $K = m$. In that case, the approximated Fourier-cosine series can match all the training data points, while still providing an approximation for the global function on the whole expansion interval.

4.2. SAMPLING FREQUENCY TRAINING DATA

Given that the choice of $K = m$ is an optimal parameter setting in the CPD algorithm, the next analysis will focus on the number of training data points m required to obtain an

accurate fit for the recovered Fourier-cosine series. The number of training data points is influenced by the interval width in which the function is approximated and the function's characteristics. The interval width is equivalent to the Fourier expansion interval $[a, b]$. Since we focus on density functions, the interval width can be chosen based on the **percent-point function (ppf)** (also called quantile function) of the corresponding distribution. By using the ppf, a desired tolerance level can be used to control the error that originates from the truncation of the infinite domain on which the density function is defined. As the tolerance level decreases, the interval width will increase.

The sampling frequency is the reciprocal of the sampling distance, which will be denoted with Δm . The impact of different tolerance levels on the sampling frequency to recover the Fourier-cosine series of the standard normal density function is shown in Figure 4.4. The error $\|\alpha_k - \hat{\alpha}_k\|_\infty$ is defined as the maximum difference between the exact and approximated Fourier-cosine coefficients. The exact coefficients can be derived from Equation 2.13 in the COS method, since the characteristic function of the normal distribution has an analytical expression. As expected, an increase in the sampling frequency leads to a higher accuracy of the computed Fourier coefficients, until the tolerance level becomes the dominant error. After that, an increase in the sampling frequency will not improve the accuracy of the coefficients. Therefore, it is important to choose the sampling frequency based on the selected tolerance level in the expansion interval $[a, b]$.

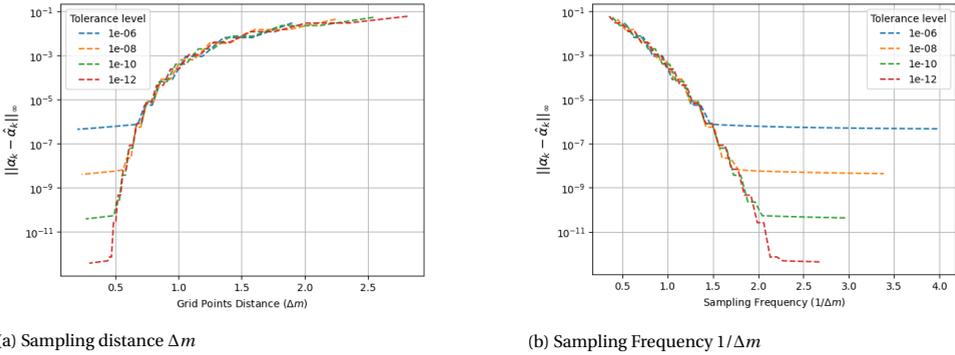


Figure 4.4: L^∞ error of the Fourier-cosine coefficients for the standard normal density, computed using CPD with different sampling frequencies of the training data.

In addition to the interval width, the function's characteristics can also affect the required sampling frequency to achieve accurate results. The convergence speed of the Fourier-cosine series depends on the smoothness of the function, which will be further discussed in a theoretical error analysis in Chapter 6.

As our focus is on evaluating expectations, only functions of the form $g(x)f(x)$ are considered, in which $g(x)$ is referred to as the target function and $f(x)$ is the density function. For this test, the density function $f(x)$ is taken as the standard normal density function, with a tolerance level of 10^{-12} used for the expansion interval $[a, b]$. The results for several functions of the form $g(x)f(x)$ are shown in Figure 4.5. The error $|\alpha_0 - \hat{\alpha}_0|$ is calculated as the difference between the first Fourier coefficient of the CPD approxi-

mation and the exact solution, as an analytical solution can be found for α_0 using the properties of the expectation,

$$\mathbb{E}[g(X)] := \int_{-\infty}^{\infty} g(x)f(x) dx \approx \int_a^b g(x)f(x) dx := \frac{b-a}{2}\alpha_0.$$

Therefore, the expectation of the target function $g(x)$ is calculated to determine the exact value of the Fourier coefficient α_0 . It should be noted that not all functions attain an error of the prescribed tolerance level of 10^{-12} . This is because the tolerance level of the truncated interval $[a, b]$ is based only on the pdf of the density function and does not account for the target function. However, the target function can still attain large values outside the truncated interval $[a, b]$. For example, the evaluation of e^x at the endpoint b gives $e^{7.034} = \mathcal{O}(10^3)$. Consequently, the product $e^x \cdot f(x)$ is of order $\mathcal{O}(10^3 \cdot 10^{-12}) = \mathcal{O}(10^{-9})$ at the endpoint b . Therefore, the CPD approximation can only achieve an accuracy of $\mathcal{O}(10^{-9})$, as observed in Figure 4.5.

For the remainder of this thesis, the CPD algorithm will be applied only to the density function itself. However, when using CPD as a direct integration method to evaluate multivariate expectations, the magnitude of the target function outside the truncated interval $[a, b]$ has to be taken into account. An explanation on how to use the CPD algorithm as a direct integration method is provided in appendix C.

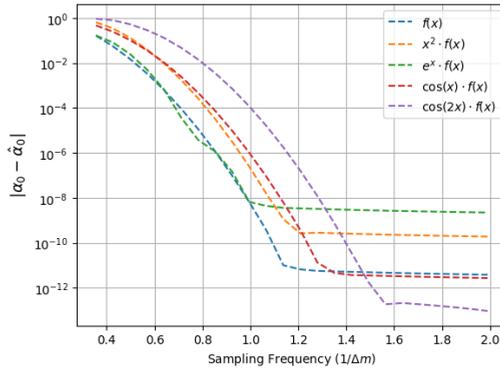


Figure 4.5: Error in the first Fourier coefficient α_0 using the CPD algorithm for different sampling frequencies. The functions $f(x)$, $x^2 \cdot f(x)$ and $e^x \cdot f(x)$, $\cos(x) \cdot f(x)$ and $\cos(2x) \cdot f(x)$ are considered, with $f(x)$ denoting the standard normal density function. The tolerance level in the expansion interval $[a, b]$ is set to 10^{-12} .

4.3. SENSITIVITY ANALYSIS

This section focuses on a sensitivity analysis of the CPD algorithm to evaluate the impact on the accuracy of the constructed Fourier-cosine series when different parameter settings are used. The analysis is structured into examples featuring one, two, and multiple dimensions, each of which implements the CPD algorithm on the normal density function.

4.3.1. ONE-DIMENSIONAL CASE

In the one-dimensional scenario, the Fourier coefficient tensor simplifies to a rank-1 tensor, represented as $\alpha = [\alpha_0, \dots, \alpha_{K-1}]$. As the CPD method decomposes a tensor into a sum of R rank-1 components, the Fourier coefficient tensor is already synthesized for $R = 1$. Furthermore, only one factor matrix \mathbf{A}_1 needs to be computed, which makes it unnecessary to perform the ALS part of the algorithm for updating the factor matrices. As the rank and the ALS updating in the one-dimensional case do not play a role, the accuracy for different number of training data points M and expansion terms K can be examined.

For this sensitivity analysis, the number of training points is fixed at $M = m = 50$ and the value of K varies from 5 to 120. The interval for the Fourier expansion, $[a, b] = [-7.034, 7.034]$, is selected using the ppf of the standard normal distribution with a tolerance level of 10^{-12} . In the CG part of the algorithm for solving \mathbf{A}_1 , the tolerance level is set at $\varepsilon_{CG} = 10^{-12}$, and the maximum number of CG iterations is $\max_{CG} = K \cdot R = K$. This value is based on the theoretical convergence of the CG method. As the (vectorized) system in 3.11 solves for a factor matrix of size $K \cdot R$, the CG method converges in no more than $K \cdot R$ iterations in case of **symmetric, positive definite (SPD)** matrices. Although the SPD criteria are not met, it still provides an indication for the maximum value of CG iterations. This concept will be further discussed in the two-dimensional scenario when the rank in the CPD algorithm plays a role.

The accuracy of the approximated Fourier-cosine series can be determined once \mathbf{A}_1 is computed. By using Equation 3.8, the output of the CPD model can be generated for any data point $x \in [a, b]$ and compared with the exact function value of the normal density function at that point. The accuracy of the model is then established using both the output of the model and the accuracy of the Fourier coefficients themselves. The error $\|\alpha_k - \hat{\alpha}_k\|_\infty$ is computed by comparing the approximated coefficients with the exact values, which can be derived using the COS method, since the characteristic function of the normal distribution is known in closed form. The results of this sensitivity analysis are depicted in Figure 4.6, which distinguishes between the accuracy for training and non-training data. The accuracy of the output for the non-training data is fully determined by the computed Fourier coefficients and hence these two error plots show the same behaviour. For $K < m$, all three errors converge exponentially with increasing K . In this scenario, the accuracy of the fitted function is limited by the series truncation error at level K . However, as K increases, the series truncation error decreases and the function fits better to the data. The minimum error attainable for the Fourier coefficients and non-training data is of order $\mathcal{O}(10^{-12})$, which is due to the interval truncation error of the Fourier-cosine series expansion on $[a, b]$. When $K > m$, the reconstructed Fourier-cosine series no longer represents the global function due to overfitting, as explained in Section 4.1. As a result, the error in the Fourier coefficients and non-training data increases significantly, while the fitted function reaches machine precision for the training data, as it is able to match all the training points when solving the minimization problem.

As previously discussed in Section 4.1, the optimal choice for the parameters K and m in the algorithm is to set $K = m$. To evaluate the convergence rate of the Fourier coefficients in this situation, the value of $K = m$ is increased from 5 to 35. Figure 4.7 illustrates

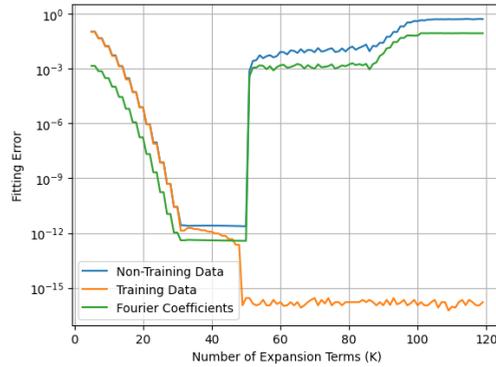


Figure 4.6: Error in the CPD approximation of the Fourier-cosine series for the standard normal distribution. The number of training data points is set to $M = 50$. The error in the Fourier coefficients is defined as $\|\alpha_k - \hat{\alpha}_k\|_\infty$

the convergence rate of the Fourier coefficients and the error in the (non-) training data for $K = m$. In this scenario, the computed Fourier-cosine series can exactly match all of the training data points, which results in a horizontal line close to machine precision for these points. The error in the non-training data and Fourier coefficients is entirely determined by the series truncation error at level K . As K increases, the series truncation error decreases, which leads to a error decrease in the Fourier coefficients and non-training data. After 30 expansion terms, the interval truncation error on $[a, b]$ becomes the dominant source of error. This indicates that a further increase in the number of expansion terms will not decrease the error anymore.

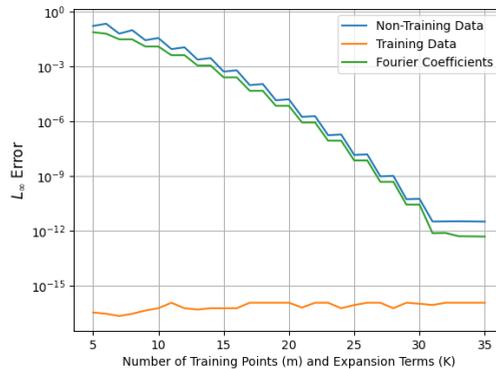


Figure 4.7: L^∞ error in the CPD approximation of the Fourier-cosine series for the standard normal density function in the scenario $K = m$. The error is presented for the (non-) training data and the Fourier coefficients.

4.3.2. TWO-DIMENSIONAL CASE

In the one-dimensional analysis, the Fourier coefficient tensor is always synthesized when we set $R = 1$ in the CPD algorithm. This changes when considering the two-dimensional case, where the Fourier coefficient tensor can be represented as a $K \times K$ matrix. The CPD algorithm will approximate this tensor by a sum of R rank-1 components. The analysis will be performed using the standard bivariate normal density function, which allows for the introduction of correlation between the random variables. As a result, a differentiation will be made between the independent and correlated scenarios.

INDEPENDENT VARIABLES

In case the two random variables are independent, the density function can be written as a product of the marginals,

$$f(x, y) = f_1(x)f_2(y),$$

where $f_1(x)$ and $f_2(y)$ representing two univariate normal density functions. Using this relation in the Fourier-cosine series, we obtain

$$\begin{aligned} & \sum_{k_1=0}^{K-1} \sum_{k_2=0}^{K-1} \alpha_{k_1, k_2} \cos\left(k_1\pi \frac{x-a_1}{b_1-a_1}\right) \cos\left(k_2\pi \frac{y-a_2}{b_2-a_2}\right) \\ &= \left[\sum_{k_1=0}^{K-1} \alpha_{k_1}^x \cos\left(k_1\pi \frac{x-a_1}{b_1-a_1}\right) \right] \cdot \left[\sum_{k_2=0}^{K-1} \alpha_{k_2}^y \cos\left(k_2\pi \frac{y-a_2}{b_2-a_2}\right) \right] \\ &= \sum_{k_1=0}^{K-1} \sum_{k_2=0}^{K-1} \alpha_{k_1}^x \alpha_{k_2}^y \cos\left(k_1\pi \frac{x-a_1}{b_1-a_1}\right) \cos\left(k_2\pi \frac{y-a_2}{b_2-a_2}\right). \end{aligned} \quad (4.2)$$

From Equation 4.2, it can be deduced that the Fourier coefficient tensor α can be expressed as the outer product of the Fourier coefficient vectors of the marginal distributions, i.e. $\alpha = \alpha^x \circ \alpha^y$. As a result, the Fourier coefficient tensor is a rank-1 tensor, and can be synthesized using only one rank-1 tensor in the CPD algorithm. Therefore, the rank in the CPD algorithm is set to $R = 1$ for an independent density function. Similar to the one-dimensional case, the convergence of the Fourier coefficients is evaluated for different parameter settings. Furthermore, the error in the reconstructed Fourier-cosine series for both training and non-training data is analyzed. The ALS part of the algorithm must also be used, as two factor matrices \mathbf{A}_1 and \mathbf{A}_2 have to be computed. The number of training points is set to $m = 30$ per dimension, which results in a total of $M = m^2 = 900$ training points. The number of expansion terms per dimension K ranges from 5 to 60, and the Fourier expansion interval is defined by utilizing a tolerance level of 10^{-12} , so that $[a, b]^2 = [-7.034, 7.034]^2$. The parameters in the CG part are the same as in the one-dimensional case, but a new stopping criterion for the ALS updating must be introduced. Once both factor matrices are computed using CG, the relative improvement in the first Fourier coefficient is taken as the stopping criterion.

$$\left| \frac{\alpha_0^{i+1} - \alpha_0^i}{\alpha_0^i} \right| < \varepsilon_{ALS}$$

For this analysis, the tolerance level in the ALS part is set to $\varepsilon_{ALS} = 10^{-12}$, with a maximum number of ALS iterations set to $\max_{ALS} = 5$. The convergence of the two-dimensional Fourier-cosines series is illustrated in Figure 4.8. The results are similar to the one-dimensional analysis, which is expected in the case of independent random variables, as the density can be written as a product of the one-dimensional marginals. Hence, the resulting Fourier-cosine series can be written as a product of the univariate Fourier-cosine series, which was already shown in Equation 4.2. Therefore, the convergence rate of the CPD model for the bivariate density function will be of the same order as the convergence rate for the univariate density functions.

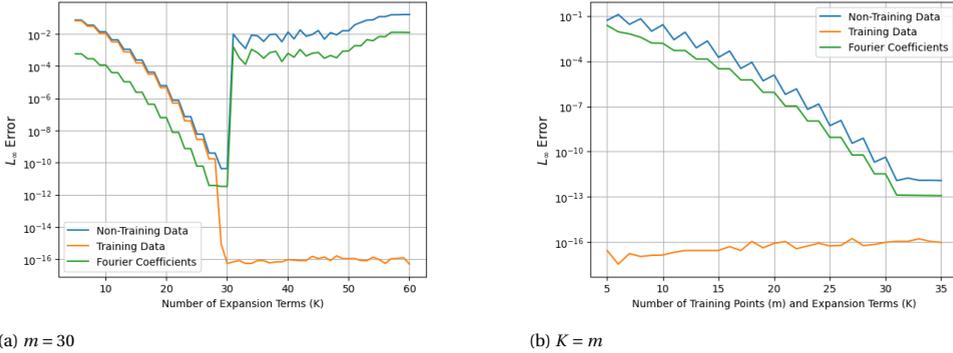


Figure 4.8: Error in the CPD approximation of the Fourier-cosine series for the bivariate normal density function. The error is presented for the (non-) training data and the Fourier coefficients.

CORRELATED VARIABLES

In the event of a correlation between the two random variables, Equation 4.2 does not longer hold, and the Fourier coefficient tensor cannot be synthesized using only a single rank-1 tensor. Therefore, the effect of different ranks used in the CPD algorithm for the approximation of the Fourier coefficient tensor can now be assessed. A sensitivity analysis is performed for the bivariate normal density function with varying levels of correlation between the two random variables. To study the impact of the rank choice on the performance of the CPD algorithm, the Fourier coefficient tensor will be approximated with a rank ranging from $R = 1$ to $R = 30$. We set $K = m = 30$ in each dimension, which makes a total of $K^2 = m^2 = 900$ expansion terms and training data points. The tolerance level in the integration truncation range $[a, b]^2$ is set at 10^{-12} based on the ppf of the normal distribution. The stopping criteria in the CG solver has to be selected carefully. Similar to the one-dimensional case, the CG solver will have a tolerance level in the relative residual of 10^{-12} . However, the maximum number of CG iterations will depend on the rank R used in the CPD. The CG method is applied to the vectorized form of Equation 3.11 to solve for the factor matrices $\mathbf{A}_1, \mathbf{A}_2 \in \mathbb{R}^{K \times R}$, which results in a solution vector of size $K \cdot R$. Therefore, a larger system must be solved within the CG algorithm for higher ranks. This leads to an increase in the number of CG iterations required to reach the desired accuracy. Since the CG method needs at most $K \cdot R$ iterations in case of SPD matrices, this will be used as an indication for the maximum number of CG iterations in

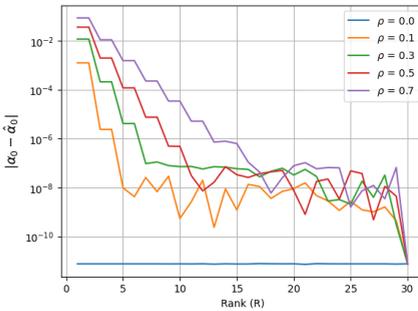
the CPD algorithm. The ALS criteria are the same as in the independent case. Furthermore, the difference $|\alpha_0 - \hat{\alpha}_0|$ will be used to quantify the error, as the analytical solution for the first Fourier coefficient α_0 can be easily obtained.

The outcomes of this analysis are depicted in Figure 4.9a. When $\rho = 0$, the results align with those of the independent scenario. This is expected, as zero correlation implies independence in case of the normal distribution. In the presence of correlation, the Fourier coefficients exhibit initial exponential convergence as the rank increases, but eventually oscillate around a certain accuracy level. These equilibrium levels are close to the corresponding series truncation errors, as this error source increases when correlation is introduced. Table 4.1 shows the order of the series truncation error for the different correlation levels in case $K = 30$ expansion terms per dimension are utilized. These results are found by computing the exact Fourier coefficients with the two-dimensional COS method introduced in [62].

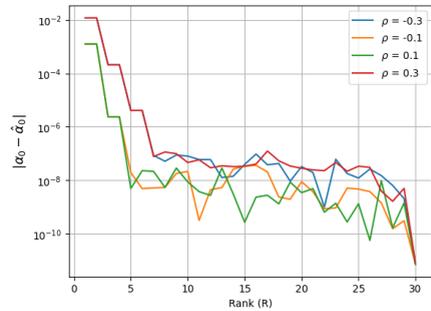
Table 4.1: Order of the Fourier-cosine series truncation error of the bivariate normal density function for different correlation levels, in case $K = 30$ expansion terms per dimension are used.

$\rho = 0$	$\rho = 0.1$	$\rho = 0.3$	$\rho = 0.5$	$\rho = 0.7$
$\mathcal{O}(10^{-16})$	$\mathcal{O}(10^{-12})$	$\mathcal{O}(10^{-11})$	$\mathcal{O}(10^{-9})$	$\mathcal{O}(10^{-7})$

From this, we can see that the CPD approximation of the Fourier coefficients in Figure 4.9a is converging up or close to the accuracy level of the series truncation error, which then becomes the dominant error source. Consequently, the error due to the lower-rank approximation in the CPD shows a stable exponential convergence in this experiment. Additionally, if the correlation level increases, a higher rank in the CPD algorithm is required to obtain the best possible approximation. Figure 4.9b illustrates that the direction of the correlation does not affect the convergence rate in the CPD approximation.



(a) Different correlation levels.

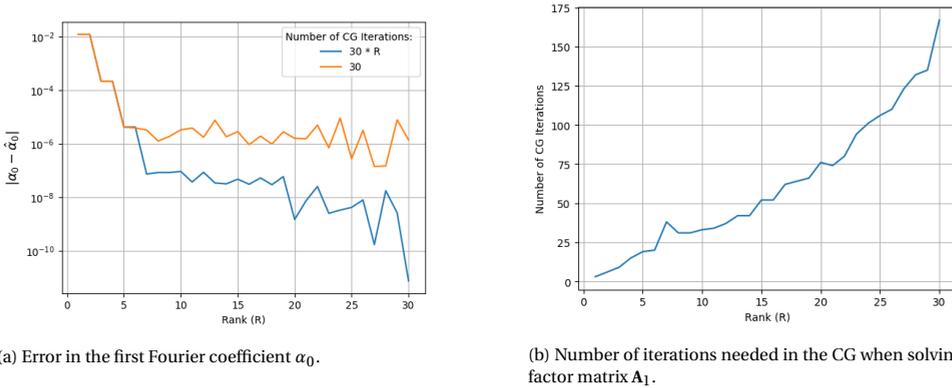


(b) Positive and negative correlation.

Figure 4.9: Impact of the correlation level on the error in the first Fourier coefficient α_0 for different rank. In (a) different levels of (positive) correlation are used, while the difference between positive and negative correlation of the same order is presented in (b). For all tests we took $K = m = 30$.

To highlight the importance of selecting a sufficient number of CG iterations, a comparative test is performed to assess the effect of the maximum number of CG iterations.

A fixed number of CG iterations for all ranks is compared with the dynamically changing number $K \cdot R$, which was already used in the previous analysis. The fixed number of CG iterations is set to 30. Figure 4.10 shows the need for a dynamic number of CG iterations over a fixed number. It can be seen from 4.10b that the number of CG iterations required to achieve the desired accuracy increases along with the rank, which is what we expected, as a larger system of equations has to be solved in the CG algorithm. For a rank of 7 or higher, more than 30 iterations are needed to accurately solve for the factor matrix \mathbf{A}_1 . When using a fixed number of 30 CG iterations, the desired accuracy cannot be obtained and the final error in the Fourier coefficients will be enlarged, as can be seen in 4.10a.



(a) Error in the first Fourier coefficient α_0 .

(b) Number of iterations needed in the CG when solving factor matrix \mathbf{A}_1 .

Figure 4.10: Impact of insufficient iterations used in the CG solver when computing \mathbf{A}_1 and \mathbf{A}_2 . The error in the first Fourier coefficient is presented in (a), while (b) shows the number of CG iterations needed for an accuracy in the relative residual of 10^{-12} . Parameters are set to $K = m = 30$ and $\rho = 0.3$.

Since $\mathbf{A}_1, \mathbf{A}_2 \in \mathbb{R}^{K \times R}$, the computational complexity of computing the Fourier coefficient tensor with CPD is reduced from $\mathcal{O}(K^2)$ to $\mathcal{O}(2KR)$. Hence, choosing a higher rank will increase the computational complexity. Consequently, the total computation time for the construction of \mathbf{A}_1 and \mathbf{A}_2 will increase with R , as shown in Figure 4.11.

4.3.3. MULTI-DIMENSIONAL CASE

The final part of the sensitivity analysis focuses on the application of the CPD algorithm to construct multivariate Fourier-cosine series up to six variables. Again, this will be performed using the multivariate normal density function with a maximum of six random variables involved. In the subsequent chapters on exposure computation and option pricing, we will encounter these multivariate normal distributions, which makes it beneficial to already gain insights in the behavior of the CPD algorithm for these multi-dimensional scenarios.

For multi-dimensional problems, two crucial factors often play a role when numerical methods are employed; the memory storage and the computation time. In the one- and two-dimensional tests, it is possible to choose a sufficient number of training data points without encountering memory overflow or excessive computation time. However, this changes when we deal with more dimensions. The total number of training

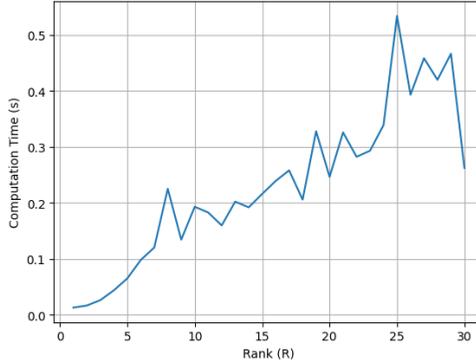


Figure 4.11: Computation time of the CPD algorithm for different rank to construct factor matrices \mathbf{A}_1 and \mathbf{A}_2 .

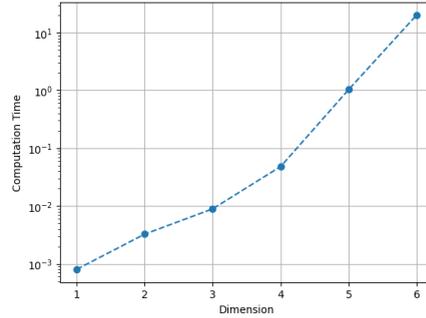
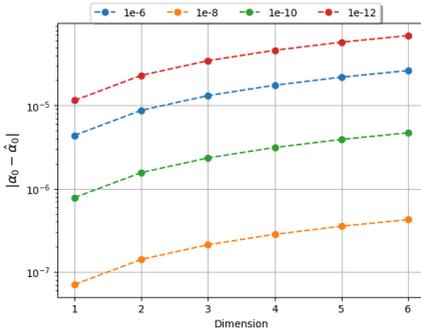
4

points, given by $M = m^N$ (with N representing the dimension of the problem), grows exponentially. This leads to a substantial impact on the memory storage and computation time required in the CPD algorithm. To manage all dimensions, we choose $K = m = 12$ per dimension for the next analysis. Consequently, a suitable tolerance level must be selected in the expansion interval $[a, b]^N$ to achieve optimal convergence with this sampling frequency. To determine the best tolerance level, a test is conducted where the tolerance level is varied from 10^{-6} to 10^{-12} .

The independent case is considered first and therefore, we set $R = 1$. The stopping criteria in the CG and ALS parts are the same as for the two-dimensional analysis. The convergence results for different dimensions, along with the computation time needed to construct the Fourier-cosine series with CPD, are shown in Figure 4.12. The convergence of the first Fourier coefficient is consistent across dimensions, although there is a slight increase due to the accumulation of the truncation errors from the expansion interval. The tolerance level selected in the expansion interval has a significant effect on the final error. For instance, a tolerance level of 10^{-6} produces less error compared to a more accurate tolerance level of 10^{-12} .

As previously discussed in Section 4.2, the expansion intervals corresponding to tolerance levels of 10^{-10} and 10^{-12} are too wide when using only 12 training data points in each dimension (i.e. the sampling frequency is too low for these expansion intervals). This leads to a loss of accuracy in the computed Fourier coefficients. To achieve the desired accuracy for those tolerance levels, more training points are required. The computation time increases exponentially in dimension, as depicted in Figure 4.12b.

Next, the influence of an increasing number of training points on the computation time is investigated. To maximize impact of the exponential growth in the number of training points, the six-dimensional case is considered. The number of training points in each dimension will range from $m = 5$ to $m = 12$, which results in a total of $M = m^6$ points. Based on the previous result, the tolerance level for this test in the expansion interval has been set to 10^{-8} . Figure 4.13 shows the exponential convergence of the Fourier coefficients, which has been previously observed, along with an exponential increase in computational time as the number of training points increases. This result illustrates



(a) Error in the first Fourier-coefficient α_0 for different tolerance levels on $[a, b]^N$.

(b) Computation time per dimension.

Figure 4.12: Application of the CPD method on the standard normal density function for dimension 1 up to 6. The error in the first Fourier coefficient for different tolerance levels (a) and the computation time in each dimension (b) are shown. Parameter values used for this test are $K = m = 12$ and $R = 1$.

the impact of the exponential growth of the number of training points, as the addition of only one extra training point per dimension results in a significant increase in the total computational time.

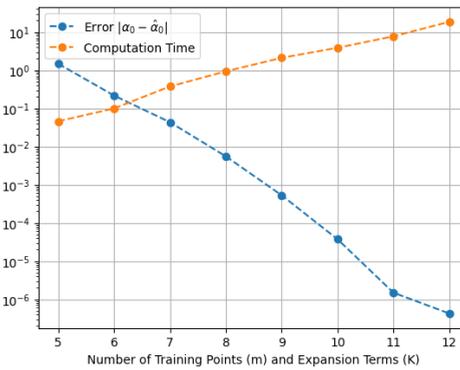


Figure 4.13: Error in the first Fourier coefficient and computation time using CPD on the six-dimensional normal density. For this test we have $K = m$ and $R = 1$.

The impact of different levels of correlation is already exploited in the two-dimensional case. However, in the multi-dimensional case, the number of correlated random variables can also have an impact on the performance of the CPD algorithm. One can expect that the performance will be affected if the number of correlated variables changes. Therefore, a test is conducted using the four-dimensional normal density function with varying numbers of correlated variables. The results shown in Figure 4.14 indicate that the number of correlated variables does indeed have an impact on the final error of the CPD approximation. An increase in the number of correlated variables leads to a decline in performance due to the more complex correlation structure within the density func-

tion. Consequently, the Fourier coefficient tensor becomes more difficult for the CPD algorithm to approximate with a lower-rank tensor.

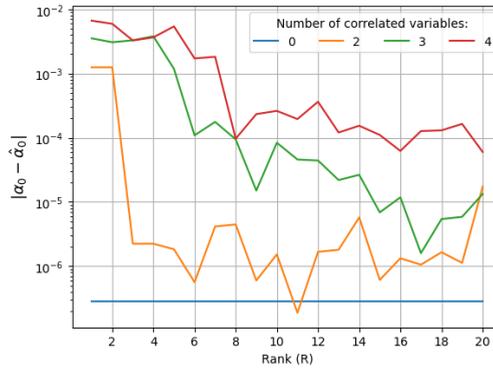


Figure 4.14: Error in the first Fourier coefficient of the four-dimensional normal density function for different number of correlated random variables. Parameters are $K = m = 12$ and all correlations are set to 10%.

5

OUR NOVEL METHOD: THE COS-CPD METHOD

This chapter introduces a new method we developed in this thesis, the COS-CPD method, for evaluating multivariate expectations that is based on the combination of CPD with Fourier-cosine series. The central concepts of the method are the replacement of the joint density function by a dimension-reduced Fourier-cosine series, obtained through the CPD approach, and the use of the COS method in order to derive the distribution of a function of multiple random variables. More precisely, the representation of the Fourier-cosine series obtained via the CPD approach is shown to have a more favorable form when numerical integration is performed than the traditional Fourier-cosine series. The COS method is utilized when the distribution of a function of a random variable has to be recovered from its characteristic function, which is essentially a multivariate expectation. Therefore, the CPD technique can be used to replace the joint density function within the characteristic function, which leads to a reduction in the dimension of the integration. Once the characteristic functions are obtained by solving the dimension-reduced numerical integration, the PDF and CDF of the desired distribution can be derived using the COS method. Hence, we name the method as the COS-CPD.

The first section of this chapter presents an overview of the COS-CPD method in the context of solving multivariate expectations in general. The application of the CPD technique to replace the joint density function with a dimension-reduced Fourier-cosine series is demonstrated.

As this thesis focuses on financial applications, the next sections are devoted to the implementation of the COS-CPD method in the calculation of netting-set level exposure, where the PFE and EE can be computed. It is demonstrated that the COS-CPD method can greatly reduce the dimensionality of the problem by separating the different risk factors involved in the derivative portfolio.

5.1. SOLVING MULTIVARIATE EXPECTATIONS

The focus of this thesis is on applications in the financial sector, specifically exposure computation and option valuation. These problems in essence boil down to the evaluation of multivariate expectations, which is a widely studied topic in various disciplines related to probability theory and statistics. The COS-CPD method, presented in this chapter, provides a solution for efficient computation of multivariate expectations and can therefore be applied to any situation where such evaluations are required. This section briefly shows the basics of the COS-CPD method and its application in a general context for multivariate expectation computation. The remainder of this chapter will focus again on the application of the COS-CPD method in the context of exposure calculation.

To illustrate the general principle of the method, consider a multivariate random variable $\mathbf{X} = (X_1, \dots, X_N)$ with corresponding joint density function $f(\cdot)$. A common problem in probability theory is the computation of the expectation of a target function $g(\cdot)$, defined on this multivariate random variable \mathbf{X} under a certain probability measure, i.e.,

$$\mathbb{E}[g(\mathbf{X})] = \int_{\mathbb{R}^N} g(x_1, \dots, x_N) f(x_1, \dots, x_N) dx_1 \cdots dx_N.$$

The first step in the COS-CPD method is to replace the joint density function with the dimension-reduced Fourier-cosine series obtained via CPD, as given in Equation 3.6 and Equation 3.7, i.e.

$$f(x_1, \dots, x_N) \approx \sum_{r=1}^R \prod_{n=1}^N f_{n,r}(x_n) = \sum_{r=1}^R \prod_{n=1}^N \mathbf{v}_n^T \mathbf{A}_n[:, r], \quad (5.1)$$

with $\{\mathbf{A}_n\}_{n=1}^N$ the factor matrices and $\{\mathbf{v}_n\}_{n=1}^N$ the cosine basis functions. Next, this reduced version of the Fourier-cosine series expansion is inserted into the multivariate expectation to obtain

$$\begin{aligned} \mathbb{E}[g(\mathbf{X})] &\approx \int_{\mathbb{R}^N} g(x_1, \dots, x_N) \sum_{r=1}^R \prod_{n=1}^N \mathbf{v}_n^T \mathbf{A}_n[:, r] dx_1 \cdots dx_N \\ &= \sum_{r=1}^R \int_{\mathbb{R}^N} g(x_1, \dots, x_N) \prod_{n=1}^N \mathbf{v}_n^T \mathbf{A}_n[:, r] dx_1 \cdots dx_N. \end{aligned} \quad (5.2)$$

As the derivation of 5.2 only involves the replacement of the joint density function by its dimension-reduced Fourier-cosine series, we want to emphasize that the COS-CPD method up to this point can be applied to any multivariate expectation problem. Depending on the properties of the target function $g(\cdot)$, the N -dimensional integral in 5.2 can possibly be solved by evaluating only lower dimensional integrals. For example, the characteristic function evaluation in the context of netting-set level exposure can be reduced to only evaluating one- or two-dimensional integrals, which will be shown in the next section.

5.2. COMPUTING POTENTIAL FUTURE EXPOSURE

This section provides the methodology of COS-CPD method in the context of exposure calculation. To obtain the PFE of a derivative portfolio at a future time point, the entire distribution of the exposure has to be computed. Using the COS method, the PDF and CDF profile of the exposure can be approximated by its Fourier-cosine series,

$$f_E(x) \approx \sum_{k=0}^{K-1} F_k \cos\left(k\pi \frac{x-l}{u-l}\right)$$

$$F_E(x) \approx \frac{F_0}{2}(x-l) + \sum_{k=1}^{K-1} F_k \frac{u-l}{k\pi} \sin\left(k\pi \frac{x-l}{u-l}\right),$$

where the coefficients F_k are of the following form,

$$F_k \equiv \frac{2}{u-l} \operatorname{Re} \left\{ \phi_E\left(\frac{k\pi}{u-l}\right) \cdot \exp\left(-i \frac{kl\pi}{u-l}\right) \right\}.$$

Therefore, the problem boils down to the evaluation of the characteristic function of the exposure, ϕ_E , which is defined as

$$\phi_E(\omega) = \int_{\mathbb{R}^N} e^{i\omega E(x_1, \dots, x_N)} f(x_1, \dots, x_N) dx_1 \dots dx_N \quad (5.3)$$

$$= \int_{\mathbb{R}^N} e^{i\omega \max[V(x_1, \dots, x_N), 0]} f(x_1, \dots, x_N) dx_1 \dots dx_N, \quad (5.4)$$

with $f(x_1, \dots, x_N)$ representing the joint probability density function of the driving risk factors of the portfolio. The number of dimensions of this integration is therefore determined by the number of risk factors involved.

The presence of the maximum operator in the expression of the exposure creates a discontinuity in both the PDF and CDF of the exposure. Because of this, the resulting Fourier-cosine series of the exposure PDF and CDF, recovered via the COS method, has to capture this discontinuity behaviour as well. However, the recovered function has to be at least continuous in order for the Fourier-cosine series to converge [50]. Otherwise, the Fourier-cosine series does not converge at the discontinuity points and will have a slower convergence close to these discontinuity points. This discontinuity leads to oscillatory behavior in the Fourier-cosine series around these points, known as the Gibbs phenomenon. The convergence theory on Fourier-cosine series is discussed in Section 6.1, which provides information on the type of convergence for various function classes.

However, in the case of evaluating netting-set level exposure, the maximum operator can be omitted in the calculation of the characteristic function. This results in the recovery of the continuous PDF and CDF functions of the MtM value of the portfolio. The exposure distribution is then obtained by applying the maximum operator to the portfolio distribution afterwards. In this manner, the distribution of the continuous function V is recovered by the Fourier-cosine series in the COS method, leading to a fast convergence of the Fourier-cosine series. For example, the CDF profile of the exposure at the

netting-set level can be found by applying the following transformation,

$$F_E(x) = \begin{cases} F_V(x) & \text{if } x \geq 0 \\ 0 & \text{if } x < 0, \end{cases}$$

where F_V denotes the CDF of the MtM value of the portfolio. Therefore, the remaining analysis focuses on the calculation of the characteristic function of the MtM of the portfolio,

$$\phi_V(\omega) = \int_{\mathbb{R}^N} e^{i\omega V(x_1, \dots, x_N)} f(x_1, \dots, x_N) dx_1 \dots dx_N. \quad (5.5)$$

For a low-dimensional integral ($N \leq 3$), the characteristic function can be approximated directly using a standard numerical integration method such as Trapezoidal rule or Clenshaw-Curtis. This approach has already been studied in a recent thesis done at FF Quant up to 3 dimensions, which shows promising results in computing the PFE and EE with the COS method [36]. However, this direct integration approach is not suitable for high-dimensional integrals ($N > 3$), as these quadrature methods suffer from the so-called "curse of dimension", meaning that the computational complexity grows exponentially with respect to the number of dimensions. Therefore, the COS-CPD method is developed in this thesis to alleviate the curse of dimension.

The COS-CPD approach to compute the exposure distribution takes two fundamental steps. The first step has already been discussed in Section 5.1 in the context of general multivariate expectations. In this step, the joint density function $f(x_1, \dots, x_N)$ of the driving risk factors is decomposed into a dimension-reduced Fourier-cosine series, computed with CPD, i.e.,

$$f(x_1, \dots, x_N) \approx \sum_{r=1}^R \prod_{n=1}^N f_{n,r}(x_n) = \sum_{r=1}^R \prod_{n=1}^N \mathbf{v}_n^T \mathbf{A}_n[:, r],$$

with $\{\mathbf{A}_n\}_{n=1}^N$ the factor matrices and $\{\mathbf{v}_n\}_{n=1}^N$ the cosine basis functions. Inserting this result into 5.5 gives the following expression for ϕ_V ,

$$\begin{aligned} \phi_V(\omega) &\approx \int_{\mathbb{R}^N} e^{i\omega V(x_1, \dots, x_N)} \sum_{r=1}^R \prod_{n=1}^N \mathbf{v}_n^T \mathbf{A}_n[:, r] dx_1 \dots dx_N \\ &= \sum_{r=1}^R \int_{\mathbb{R}^N} e^{i\omega V(x_1, \dots, x_N)} \prod_{n=1}^N \mathbf{v}_n^T \mathbf{A}_n[:, r] dx_1 \dots dx_N. \end{aligned} \quad (5.6)$$

The second step is to observe that the portfolio value is a linear combination of all the paying and receiving legs of the individual contracts. Although the portfolio may be influenced by many risk factors, each contract leg is associated with only one or two risk factors. For instance, a FX forward contract is governed by three risk factors, but its individual legs are influenced by one (for the domestic currency leg) or two (for the foreign currency leg) risk factors. By grouping all the legs that are associated with the same risk factors, the portfolio can be represented as a linear combination of small subportfolios that share one to two common risk factors. If there are M such subportfolios (i.e. M

different combinations of risk factors present in the portfolio legs), the portfolio can be expressed as

$$V(x_1, \dots, x_N) = \sum_{m=1}^M V_m(\mathbf{x}_m),$$

with V_m the total value of all the legs depending on the risk factor combination $\mathbf{x}_m := (x_m^1, \dots, x_m^{\#\mathbf{x}_m})$. The symbol $\#\mathbf{x}_m$ indicates the number of risk factors involved in combination \mathbf{x}_m . To make notation easier in a later stage, we make a distinction between the disjoint and non-disjoint combinations,

$$V(x_1, \dots, x_N) = \sum_{m_1=1}^{M_1} V_{m_1}(\mathbf{x}_{m_1}) + \sum_{m_2=M_1+1}^M V_{m_2}(\mathbf{x}_{m_2}). \quad (5.7)$$

The first M_1 combinations of risk factors are disjoint with all the other combinations, i.e.

$$\mathbf{x}_i \cap \mathbf{x}_j = \emptyset \quad \text{for } i \neq j, \quad \forall i \in \{1, \dots, M_1\}, j \in \{1, \dots, M\}.$$

The other $M - M_1$ combinations contain overlapping risk factors. Next, it is assumed that the first M_1 disjoint combinations contain the risk factors (x_1, \dots, x_{N_1}) , while the remaining combinations consist of the risk factors (x_{N_1+1}, \dots, x_N) . Inserting equation 5.7 for V into 5.6 we obtain

$$\begin{aligned} \phi_V(\omega) &\approx \sum_{r=1}^R \int_{\mathbb{R}^N} e^{i\omega V(x_1, \dots, x_N)} \prod_{n=1}^N \mathbf{v}_n^T \mathbf{A}_n[:, r] dx_1 \dots dx_N \\ &= \sum_{r=1}^R \int_{\mathbb{R}^N} e^{i\omega [\sum_{m_1=1}^{M_1} V_{m_1}(\mathbf{x}_{m_1}) + \sum_{m_2=M_1+1}^M V_{m_2}(\mathbf{x}_{m_2})]} \prod_{n=1}^N \mathbf{v}_n^T \mathbf{A}_n[:, r] dx_1 \dots dx_N \\ &= \sum_{r=1}^R \int_{\mathbb{R}^N} e^{i\omega \sum_{m_1=1}^{M_1} V_{m_1}(\mathbf{x}_{m_1})} e^{i\omega \sum_{m_2=M_1+1}^M V_{m_2}(\mathbf{x}_{m_2})} \prod_{n=1}^N \mathbf{v}_n^T \mathbf{A}_n[:, r] dx_1 \dots dx_N \\ &= \sum_{r=1}^R \int_{\mathbb{R}^{N_1}} e^{i\omega \sum_{m_1=1}^{M_1} V_{m_1}(\mathbf{x}_{m_1})} \prod_{n=1}^{N_1} \mathbf{v}_n^T \mathbf{A}_n[:, r] dx_1 \dots dx_{N_1} \\ &\quad \cdot \int_{\mathbb{R}^{N-N_1}} e^{i\omega \sum_{m_2=M_1+1}^M V_{m_2}(\mathbf{x}_{m_2})} \prod_{n=N_1+1}^N \mathbf{v}_n^T \mathbf{A}_n[:, r] dx_{N_1+1} \dots dx_N \\ &:= \sum_{r=1}^R I(V_1) \cdot I(V_2), \end{aligned}$$

with $I(V_1)$ and $I(V_2)$ given by

$$I(V_1) := \int_{\mathbb{R}^{N_1}} e^{i\omega \sum_{m_1=1}^{M_1} V_{m_1}(\mathbf{x}_{m_1})} \prod_{n=1}^{N_1} \mathbf{v}_n^T \mathbf{A}_n[:, r] dx_1 \dots dx_{N_1}$$

$$I(V_2) := \int_{\mathbb{R}^{N-N_1}} e^{i\omega \sum_{m_2=M_1+1}^M V_{m_2}(\mathbf{x}_{m_2})} \prod_{n=N_1+1}^N \mathbf{v}_n^T \mathbf{A}_n[:, r] dx_{N_1+1} \dots dx_N.$$

The integral $I(V_1)$ regarding the disjoint combinations can be split into a product of integrals of all individual combinations. Since the risk factors (x_1, \dots, x_{N_1}) occur in the disjoint combinations, an ordering of these risk factors can be made over all M_1 risk factor groups, which will be denoted with,

$$\begin{aligned} \mathbf{x}_1 &= \{x_1, \dots, x_{l_1}\} \\ \mathbf{x}_2 &= \{x_{l_1+1}, \dots, x_{l_2}\} \\ &\vdots \\ \mathbf{x}_{m_1} &= \{x_{l_{(m_1-1)}+1}, \dots, x_{l_{m_1}}\} \\ &\vdots \\ \mathbf{x}_{M_1} &= \{x_{l_{(M_1-1)}+1}, \dots, x_{N_1}\}. \end{aligned}$$

Using this notation, the integral $I(V_1)$ can be written as

$$\begin{aligned} I(V_1) &= \int_{\mathbb{R}^{N_1}} e^{i\omega \sum_{m_1=1}^{M_1} V_{m_1}(\mathbf{x}_{m_1})} \prod_{n=1}^{N_1} \mathbf{v}_n^T \mathbf{A}_n[:, r] dx_1 \dots dx_{N_1} \\ &= \int_{\mathbb{R}^{N_1}} \prod_{m_1=1}^{M_1} e^{i\omega V_{m_1}(\mathbf{x}_{m_1})} \prod_{n=1}^{N_1} \mathbf{v}_n^T \mathbf{A}_n[:, r] dx_1 \dots dx_{N_1} \\ &= \prod_{m_1=1}^{M_1} \int_{\mathbb{R}^{\#\mathbf{x}_{m_1}}} e^{i\omega V_{m_1}(\mathbf{x}_{m_1})} \prod_{n=l_{(m_1-1)}+1}^{l_{m_1}} \mathbf{v}_n^T \mathbf{A}_n[:, r] dx_n. \end{aligned} \quad (5.8)$$

Working out the matrix-vector products $\mathbf{v}_n^T \mathbf{A}_n$ results in an element-wise expression for $I(V_1)$,

$$I(V_1) = \prod_{m_1=1}^{M_1} \left\{ \sum_{k_1, \dots, k_{\#\mathbf{x}_{m_1}}=0}^{K-1} \prod_{i=l_{(m_1-1)}+1}^{l_{m_1}} \mathbf{A}_i[k_i, r] \int_{\mathbb{R}^{\#\mathbf{x}_{m_1}}} e^{i\omega V_{m_1}(\mathbf{x}_{m_1})} \prod_{n=l_{(m_1-1)}+1}^{l_{m_1}} \mathbf{v}_n[k_n] dx_n \right\},$$

where the integration variables $\{x_i\}_{n=1}^{N_1}$ are contained in the cosine basis functions $\{\mathbf{v}_n\}_{n=1}^{N_1}$.

The integral $I(V_2)$ containing the overlapping risk factors cannot be split into a product of integrals. However, the dimension of integration can still be reduced due to the characteristics of the portfolio function V . To avoid illegible equations and an excess of notation details, only a simplified example will be provided in which the approach of solving $I(V_2)$ will be clear. Let (x_1, x_2) , (x_1, x_3) be two risk factor combinations occurring in the paying and receiving legs of the portfolio with overlapping risk factor x_1 . Then, the integral $I(V_2)$ that has to be solved is in essence of the following form,

$$\iiint_{\mathbb{R}^3} e^{i\omega V_1(x_1, x_2) + V_2(x_1, x_3)} \cos(x_1) \cos(x_2) \cos(x_3) dx_1 dx_2 dx_3.$$

At first sight, a three-dimensional integral has to be solved. When using quadrature methods to solve this problem numerically, this results in a three-dimensional grid of

all points in which the quadrature rule has to evaluate the integrand. However, this can be avoided by rewriting the integral as follows,

$$\begin{aligned} & \iiint_{\mathbb{R}^3} e^{i\omega V_1(x_1, x_2) + V_2(x_1, x_3)} \cos(x_1) \cos(x_2) \cos(x_3) dx_1 dx_2 dx_3 = \\ & \iiint_{\mathbb{R}^3} e^{i\omega V_1(x_1, x_2)} e^{i\omega V_2(x_1, x_3)} \cos(x_1) \cos(x_2) \cos(x_3) dx_1 dx_2 dx_3 = \\ & \int_{\mathbb{R}} \cos(x_1) \left[\iint_{\mathbb{R}^2} e^{i\omega V_1(x_1, x_2)} e^{i\omega V_2(x_1, x_3)} \cos(x_2) \cos(x_3) dx_2 dx_3 \right] dx_1 = \\ & \int_{\mathbb{R}} \cos(x_1) \left[\int_{\mathbb{R}} e^{i\omega V_1(x_1, x_2)} \cos(x_2) dx_2 \int_{\mathbb{R}} e^{i\omega V_2(x_1, x_3)} \cos(x_3) dx_3 \right] dx_1. \end{aligned}$$

From this, it can be seen that only the two-dimensional grids of (x_1, x_2) and (x_1, x_3) has to be generated, which saves a significant amount of computational costs. For example, if 100 grid points are used for each dimension, a three-dimensional grid consists of $\mathcal{O}(10^6)$ elements, while a two-dimensional grid only has $\mathcal{O}(10^4)$ elements.

The portfolios studied in this thesis will only contain disjoint groups of risk factors,

$$\mathbf{x}_i \cap \mathbf{x}_j = \emptyset \quad \text{for } i \neq j, \quad \forall i, j \in \{1, \dots, M\},$$

and hence we will focus from now on this scenario. When all the risk factor combinations are disjoint, the characteristic function can be simplified using only the expression found in 5.8 for $I(V_1)$ with $M_1 = M$,

$$\begin{aligned} \phi_V(\omega) & \approx \sum_{r=1}^R I(V_1) \\ & = \sum_{r=1}^R \prod_{m_1=1}^M \int_{\mathbb{R}^{\#\mathbf{x}_{m_1}}} e^{i\omega V_{m_1}(\mathbf{x}_{m_1})} \prod_{n=l_{(m_1-1)}+1}^{l_{m_1}} \mathbf{v}_n^T \mathbf{A}_n[:, r] dx_n. \end{aligned} \quad (5.9)$$

Since \mathbf{x}_m only contains one or two risk factors, the N -dimensional integral from 5.6 is simplified to a product of one- and two-dimensional integrals. These integrals can be efficiently computed using fast numerical integration techniques such as Clenshaw-Curtis. Also note from 5.1 that the Fourier-cosine series generated using CPD is expressed as a product of N univariate Fourier-cosine series. This is different from the standard Fourier-series expansion, which involves N nested summations over all the expansion terms. The absence of these nested summations in the CPD expression leads to smaller matrix-vector products as described in 5.9.

The integration range is truncated to the domain on which the Fourier-cosine series for the joint density function is defined,

$$\hat{\phi}_V(\omega) = \sum_{r=1}^R \prod_{m_1=1}^M \int_{\mathbf{a}_{m_1}}^{\mathbf{b}_{m_1}} e^{i\omega V_{m_1}(\mathbf{x}_{m_1})} \prod_{n=l_{(m_1-1)}+1}^{l_{m_1}} \mathbf{v}_n^T \mathbf{A}_n[:, r] dx_n, \quad (5.10)$$

where $[\mathbf{a}_{m_1}, \mathbf{b}_{m_1}] := [a_{l_{(m_1-1)}+1}, b_{l_{(m_1-1)}+1}] \times \dots \times [a_{l_{m_1}}, b_{l_{m_1}}]$. The series coefficients F_k can now be computed via the formula from the COS method,

$$F_k \equiv \frac{2}{u-l} \operatorname{Re} \left\{ \hat{\phi}_V \left(\frac{k\pi}{u-l} \right) \cdot \exp \left(-i \frac{kl\pi}{u-l} \right) \right\}, \quad (5.11)$$

from which the PDF and the CDF of the MtM value can be recovered, i.e.,

$$f_V(x) \approx \sum_{k=0}^{K-1} F_k \cos\left(k\pi \frac{x-l}{u-l}\right), \quad (5.12)$$

$$F_V(x) \approx \frac{F_0}{2}(x-l) + \sum_{k=1}^{K-1} F_k \frac{u-l}{k\pi} \sin\left(k\pi \frac{x-l}{u-l}\right). \quad (5.13)$$

As mentioned before, the exposure distribution can be found by applying the maximum operator afterwards on the recovered PDF and CDF of the MtM value, because the distribution of the total MtM value of a netting set coincides with the distribution of the exposure on $[0, +\infty]$. Once the CDF of the exposure is computed, the PFE can be extracted by taking the 97.5% quantile. The methodology for calculating the PFE using the COS-CPD method is presented in the form of pseudo code in Algorithm 2.

Note that the CPD part can be done "off-line" in advance. As long as the model parameters do not change, which is usually the case in practice for XVA or option valuations, the trained CPD results can be re-used till the next re-calibration of the models.

5

Algorithm 2 COS-CPD for netting-set level PFE

Input: $K, R, M, \mathbf{a}, \mathbf{b}, \max_{CG}, \max_{ALS}, \varepsilon_{CG}, \varepsilon_{ALS}, K_{COS}, N_{quad}, V$
 Compute factor matrices $\{\mathbf{A}_n\}_{n=1}^N$ using Algorithm 1
 Group portfolio V into parts with same risk factors
 Set the COS support $[l, u]$
for $k = 1$ **to** K_{COS} **do**
 Compute $\phi_V(\omega)$ from (5.9) using the Clenshaw–Curtis quadrature method
end for
 Compute coefficients F_k in (5.11)
 Compute CDF of the MtM value with (5.13)
 Derive exposure CDF by flooring the CDF of the MtM value
 Find the 97.5% quantile of the exposure CDF to obtain the PFE
return PFE

5.3. COMPUTING EXPECTED EXPOSURE

This section presents how the COS-CPD method can be used to compute the expected exposure of a derivative portfolio V . The expected exposure is a fundamental quantity when it comes to XVA pricing, including CVA. To begin with, the expression of the expected exposure as in definition 2.2.5 is repeated,

$$EE_t = \mathbb{E}[E(X_t)] = \int_{\mathbb{R}^N} \max[V_t(x_1, \dots, x_N), 0] f(x_1, \dots, x_N) dx_1 \dots dx_N.$$

Instead of considering the risk factors as the integration variables, the integral can also be defined directly in terms of the portfolio V , resulting in the following one-dimensional

integral,

$$EE_t = \int_{\mathbb{R}} \max(v, 0) f_V(v) dv.$$

Computing the expected exposure in this way is most of the time not possible, as it requires the PDF of the whole portfolio, f_V , which has often no closed-form expression. Nevertheless, the Fourier-cosine series of f_V , as described in Equation 5.12, has already been calculated in the preceding section during the calculation of PFE, and therefore, it can be inserted into the formula to attain the expected exposure,

$$\begin{aligned} EE_t &= \int_{\mathbb{R}} \max(v, 0) \sum_{k=0}^{K-1} F_k \cos\left(k\pi \frac{v-l}{u-l}\right) dv \\ &= \sum_{k=0}^{K-1} F_k \int_{\mathbb{R}} \max(v, 0) \cos\left(k\pi \frac{v-l}{u-l}\right) dv. \end{aligned}$$

The integration range is truncated to the interval $[l, u]$ on which the Fourier-cosine series is defined,

$$\begin{aligned} EE_t &= \sum_{k=0}^{K-1} F_k \int_l^u \max(v, 0) \cos\left(k\pi \frac{v-l}{u-l}\right) dv \\ &= \int_{\max(l, 0)}^{\max(u, 0)} v \cos\left(k\pi \frac{v-l}{u-l}\right) dv. \end{aligned} \quad (5.14)$$

Without loss of generality, it is assumed that $u > 0$ (otherwise we directly find $EE_t = 0$). For the forthcoming derivations, we will make the assumption that $l \leq 0$. However, the derivations in case $l > 0$ are identical.

The integral obtained can be solved analytically using integration by parts,

$$\begin{aligned} \int_0^u v \cos\left(k\pi \frac{v-l}{u-l}\right) dv &= \left[v \sin\left(k\pi \frac{v-l}{u-l}\right) \frac{u-l}{k\pi} \right]_{v=0}^u - \int_0^u \sin\left(k\pi \frac{v-l}{u-l}\right) \frac{u-l}{k\pi} dv \\ &= \frac{u-l}{k\pi} [u \sin(k\pi) - 0] + \left(\frac{u-l}{k\pi}\right)^2 \left[\cos\left(k\pi \frac{v-l}{u-l}\right) \right]_{v=0}^u \\ &= \frac{u-l}{k\pi} u \sin(k\pi) + \left(\frac{u-l}{k\pi}\right)^2 \left(\cos(k\pi) - \cos\left(k\pi \frac{-l}{u-l}\right) \right) \\ &= \left(\frac{u-l}{k\pi}\right)^2 \left(\cos(k\pi) - \cos\left(k\pi \frac{-l}{u-l}\right) \right) \\ &= \left(\frac{u-l}{k\pi}\right)^2 \left((-1)^k - \cos\left(k\pi \frac{l}{u-l}\right) \right). \end{aligned}$$

The case $k = 0$ has to be treated separately, as the integrand in this scenario is given by $v \cos\left(0 \cdot \pi \frac{v-l}{u-l}\right) = v$. Hence, the integral for $k = 0$ is given by

$$\int_0^u v dv = \frac{1}{2} u^2.$$

Therefore, the value of the integral can be summarized in the following way,

$$\int_0^u v \cos\left(k\pi \frac{v-l}{u-l}\right) dv = \begin{cases} \frac{1}{2}u^2 & \text{if } k = 0 \\ \left(\frac{u-l}{k\pi}\right)^2 \left((-1)^k - \cos\left(k\pi \frac{l}{u-l}\right)\right) & \text{if } k \neq 0. \end{cases}$$

The results for the case $l > 0$ can be obtained in the same way, and are given by

$$\int_l^u v \cos\left(k\pi \frac{v-l}{u-l}\right) dv = \begin{cases} \frac{1}{2}(u^2 - l^2) & \text{if } k = 0 \\ \left(\frac{u-l}{k\pi}\right)^2 \left((-1)^k - 1\right) & \text{if } k \neq 0. \end{cases}$$

These values can now be inserted into 5.14 to obtain the final expression for the expected exposure,

$$EE_t = \begin{cases} \frac{F_0}{4} u^2 + \sum_{k=1}^{K-1} F_k \left(\frac{u-l}{k\pi}\right)^2 \left((-1)^k - \cos\left(k\pi \frac{l}{u-l}\right)\right) & \text{if } l \leq 0 \\ \frac{F_0}{4} (u^2 - l^2) + \sum_{k=1}^{K-1} F_k \left(\frac{u-l}{k\pi}\right)^2 \left((-1)^k - 1\right) & \text{if } l > 0. \end{cases} \quad (5.15)$$

As a result, the EE can be directly derived from the Fourier-series coefficients F_k that were already computed during the PFE calculation. The pseudo code for computing the EE using the COS-CPD method can be found in Algorithm 3.

Algorithm 3 COS-CPD for netting-set level EE

Input: Coefficients F_k obtained via PFE calculation in Algorithm 2

Compute EE via (5.15)

return EE

6

ERROR ANALYSIS OF THE COS-CPD METHOD

This chapter focuses on a theoretical error analysis of the COS-CPD method to solve multivariate expectations. The convergence of errors from both the CPD part and the COS method will be investigated. Since both the COS method and the CPD part make use of Fourier-cosine series, the first section of this chapter examines the convergence of Fourier series in general. Subsequently, the error resulting from the CPD computation of the dimension-reduced Fourier-cosine series, corresponding to the joint density function, is analyzed. This is done by identifying the various error sources present and, when possible, deriving theoretical bounds for the errors. The analysis of the CPD part is generic for solving any multivariate expectation, as it only concerns the decomposition of the joint density function. The COS method is then employed in the context of exposure quantification to generate the portfolio distribution, which involves evaluating the characteristic function. Similar to the CPD part, the error sources present in the COS method are isolated, after which a theoretical upper bound is derived.

6.1. CONVERGENCE OF FOURIER-COSINE SERIES

This section contains some general theory on the convergence of Fourier-series, mostly taken from [50]. The different types of convergence rates are discussed, together with the corresponding series truncation error. The main result is then found in Proposition 6.1.3, which shows the convergence rate of the Fourier-cosine series for different classes of functions.

Definition 6.1.1 (Algebraic Index of Convergence). *The algebraic index of convergence $n(\geq 0)$ is the largest number for which*

$$\lim_{k \rightarrow \infty} |A_k| k^n < \infty, \quad k \gg 1,$$

where the A_k are the coefficients of the series. An alternative definition is that if the coeffi-

icients of a series, A_k , decay asymptotically as

$$A_k \sim \mathcal{O}\left(\frac{1}{k^n}\right), \quad k \gg 1,$$

then n is the algebraic index of convergence.

Definition 6.1.2 (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*

$$A_k \sim \mathcal{O}\left(e^{-\gamma k^r}\right), \quad k \gg 1,$$

with γ , constant, 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

$$A_k \sim \mathcal{O}\left(k^{-n} e^{-(k/j)\ln(k)}\right),$$

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

$$A_k \sim \mathcal{O}\left(k^{-n} e^{-\gamma k}\right). \quad (6.1)$$

Proposition 6.1.1 (Series truncation error of algebraically converging series). *It can be shown that the series truncation error of an algebraically converging series behaves like*

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

The proof can be found in [63].

Proposition 6.1.2 (Series truncation error of geometrically converging series). *If a series has geometrical convergence, then the error after truncation of the expansion after K terms, $E_T(K-1)$, reads*

$$E_T(K-1) \sim P^* e^{-(K-1)v}.$$

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

$$v = \lim_{n \rightarrow \infty} (-\log|E_T(n)|/n),$$

and $P^* > 0$ denotes a factor which varies less than exponentially with $K-1$.

Proposition 6.1.3 (Convergence of Fourier-cosine series). *If $g(x) \in C^\infty([a, b] \subset \mathbb{R})$, then its Fourier-cosine series expansion on $[a, b]$ has geometric convergence. The constant γ in 6.1 is 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. If a function $g(x)$, or any of its derivatives, is discontinuous, its Fourier-cosine series coefficients show algebraic convergence. Integration-by-parts shows that the algebraic index of convergence, n , is at least as large as n' , with the n' -th derivative of $g(x)$ integrable.*

References to the proof of this proposition are available in [50]. The theory presented here will be used in the upcoming sections to find an upper bound for the truncation error of the Fourier-cosine series in both the CPD part and the COS method.

6.2. ERROR ANALYSIS CPD

The CPD technique is utilized to decompose the joint probability density function $f(x_1, \dots, x_N)$ into a dimensionality reduced version of its Fourier-cosine series expansion. Algorithm 1 is executed to obtain the components of the Fourier coefficient tensor. Along the derivation steps of the algorithm, different errors sources are introduced, which we will track and analyze in the next section.

6.2.1. ERROR SOURCES

The first error arises from the truncation of the infinite Fourier-cosine series expansion, which results in approximation f_1 ,

$$f_1 = \sum_{k_1=0}^{K-1} \cdots \sum_{k_N=0}^{K-1} \alpha_{\mathbf{k}} \prod_{n=1}^N \cos\left(k_n \pi \frac{x_n - a_n}{b_n - a_n}\right),$$

where $\mathbf{k} = \{k_1, \dots, k_N\} \in \mathbb{N}^N$ is a multi-index. Then, the Fourier coefficients $\alpha_{\mathbf{k}}$ are approximated using the CPD algorithm. Assuming the correct rank used in the tensor decomposition (i.e. the rank for which the coefficient tensor is synthesized¹), we obtain approximation f_2 ,

$$f_2 = \sum_{k_1=0}^{K-1} \cdots \sum_{k_N=0}^{K-1} \sum_{r=1}^R \mathbf{a}_r^1[k_1] \cdots \mathbf{a}_r^N[k_N] \prod_{n=1}^N \cos\left(k_n \pi \frac{x_n - a_n}{b_n - a_n}\right).$$

Next, a lower-rank approximation, denoted with \tilde{R} , of the Fourier coefficient tensor is computed. This results in the final approximation of the Fourier-cosine series of the joint density function, given by f_3 ,

$$f(x_1, \dots, x_N) \approx f_3 = \sum_{k_1=0}^{K-1} \cdots \sum_{k_N=0}^{K-1} \sum_{r=1}^{\tilde{R}} \mathbf{a}_r^1[k_1] \cdots \mathbf{a}_r^N[k_N] \prod_{n=1}^N \cos\left(k_n \pi \frac{x_n - a_n}{b_n - a_n}\right).$$

6.2.2. THEORETICAL ERROR BOUNDS

The previous section showed that each of the three steps introduces an error when decomposing the joint density into a dimension-reduced Fourier-cosine series with CPD; the series truncation error, the decomposition error of the Fourier coefficients using CPD and the lower-rank approximation error. That is,

1. The series truncation error,

$$\varepsilon_1 := f - f_1 = \sum_{k_1=K}^{\infty} \cdots \sum_{k_N=K}^{\infty} \alpha_{\mathbf{k}} \prod_{n=1}^N \cos\left(k_n \pi \frac{x_n - a_n}{b_n - a_n}\right).$$

2. The decomposition error using the CPD technique via Algorithm 1 (assuming the correct rank is used),

$$\varepsilon_2 := f_1 - f_2 = \sum_{k_1=0}^{K-1} \cdots \sum_{k_N=0}^{K-1} \left[\alpha_{\mathbf{k}} - \sum_{r=1}^R \mathbf{a}_r^1[k_1] \cdots \mathbf{a}_r^N[k_N] \right] \prod_{n=1}^N \cos\left(k_n \pi \frac{x_n - a_n}{b_n - a_n}\right).$$

¹A tensor \mathcal{X} is synthesized if the CPD decomposition matches the original tensor, i.e. $\mathcal{X} = \sum_{r=1}^R a_r^1 \circ a_r^2 \circ \cdots \circ a_r^N$.

3. The lower-rank approximation error,

$$\varepsilon_3 := f_2 - f_3 = \sum_{k_1=0}^{K-1} \dots \sum_{k_N=0}^{K-1} \left[\sum_{r=1}^R \mathbf{a}_r^1[k_1] \dots \mathbf{a}_r^N[k_N] - \sum_{r=1}^{\tilde{R}} \mathbf{a}_r^1[k_1] \dots \mathbf{a}_r^N[k_N] \right] \prod_{n=1}^N \cos \left(k_n \pi \frac{x_n - a_n}{b_n - a_n} \right).$$

The truncation error of the Fourier-cosine series, denoted as ε_1 , can be approximated by utilizing the theory introduced in Section 6.1. It can be observed that the error converges exponentially in K (assuming that all summations are truncated using the same value K) for density functions belonging to the class $C^\infty([a_1, b_1] \times \dots \times [a_N, b_N])$. This results in an upper bound for ε_1 given by,

$$|\varepsilon_1| < P e^{-(K-1)\nu},$$

with $\nu > 0$ and $P > 0$ being a term that varies less than exponentially with K . A density function with discontinuity in one of its derivatives results in an algebraic convergence of the Fourier-cosine series expansion, for which the error can be bounded as follows,

$$|\varepsilon_1| < \frac{1}{(n-1)(K-1)^{n-1}},$$

with n the algebraic index of convergence of the series, as defined in Definition 6.1.1.

The next error to be examined involves the decomposition of the Fourier coefficient tensor through the CPD algorithm, assuming that the correct rank is utilized for synthesizing the Fourier coefficient tensor. These coefficients are determined by solving the system of equations for the factor matrices $\{\mathbf{A}_n\}_{n=1}^N$ presented in Equation 3.11. As previously noted in the corresponding chapter, the multilinear system of equations is equivalent to a linear least squares problem, which belongs to the class of convex optimization problems. As a result, the system will have a unique solution provided that the number of data points utilized for fitting is greater than or equal to the number of unknown parameters. In this context, a unique solution is then obtained for $K \leq m$. Based on the findings in Section 4.1, it was concluded that the optimal choice for the parameter is to set $K = m$ for each dimension. Therefore, the constraint of $K \leq m$ is satisfied, and we can assume that the system will have a unique solution. As this part of the analysis still assumes the correct rank in the CPD, the error ε_2 depends solely on the error made while solving Equation 3.11 for all matrices $\{\mathbf{A}_n\}_{n=1}^N$. The system is solved by utilizing the ALS algorithm, where each factor matrix is solved iteratively using the CG method. Thus, two parameters control ε_2 : (1) the stopping criterion implemented within the CG solver and (2) the stopping criterion in the outer ALS loop. These stopping criteria are defined by the tolerance levels, ε_{CG} and ε_{ALS} , respectively. The overall error in computing the coefficients is then bounded by

$$\begin{aligned} |\varepsilon_2| &= |\varepsilon_{ALS}| + |\varepsilon_{CG}| \\ &\leq \mathcal{O}(\max(|\varepsilon_{CG}|, |\varepsilon_{ALS}|)). \end{aligned}$$

In practice, the tolerance levels in the CG and ALS procedure are set to a high accuracy level, for example $\mathcal{O}(10^{-10})$. As a result, ε_2 will be dominated by the other error sources.

The final error ε_3 that arises in the CPD procedure is due to the application of a lower-rank approximation for the Fourier coefficient tensor. This error is challenging to quantify as it involves the determination of the correct rank of a tensor, which is an NP-problem [58]. Consequently, no algorithm is available to acquire the correct rank of a tensor. However, the results of the numerical tests presented in Chapter 4, specifically in Figure 4.9, suggest that there is an exponential error convergence with increasing rank for the different correlation levels, up to the point that the series truncation error becomes the dominant error source. Nevertheless, since this is only a single numerical test, further analysis is necessary to confirm the potential exponential convergence of ε_3 .

By combining the results for the three error sources, it is possible to formulate an upper bound for the total error in the CPD computation. For the smooth density functions considered in this thesis, the upper bound is given by

$$|\varepsilon_{CPD}| < P e^{-(K-1)v} + \mathcal{O}(\max(|\varepsilon_{CG}|, |\varepsilon_{ALS}|)) + |\varepsilon_3|. \quad (6.2)$$

6.3. ERROR ANALYSIS COS METHOD

Using the COS method, it is possible to recover the portfolio's density function, f_V , and cumulative distribution function, F_V . The error analysis in this section will be done on the former, and that for the latter can be done in a similar way. We employ the same approach as in [30].

6.3.1. ERROR SOURCES

To obtain the density f_V , multiple steps are involved, each of which introduces an error. The first error arises from the truncation of the infinite Fourier-cosine series expansion, from which we obtain approximation f_V^1 ,

$$\begin{aligned} f_V^1 &= \sum_{k=0}^{K_{\text{COS}}-1} A_k \cos\left(k\pi \frac{v-l}{u-l}\right) \\ &= \sum_{k=0}^{K_{\text{COS}}-1} \frac{2}{u-l} \operatorname{Re} \left\{ \hat{\phi}_V \left(\frac{k\pi}{u-l} \right) \cdot \exp \left(-i \frac{kl\pi}{u-l} \right) \right\} \cos \left(k\pi \frac{v-l}{u-l} \right), \end{aligned}$$

with $\mathbf{k} = \{k_1, \dots, k_N\} \in \mathbb{N}^N$ a multi-index. Then, the Fourier coefficients A_k are replaced by F_k to obtain approximation f_V^2 ,

$$f_V^2 = \sum_{k=0}^{K_{\text{COS}}-1} \frac{2}{u-l} \operatorname{Re} \left\{ \phi_V \left(\frac{k\pi}{u-l} \right) \cdot \exp \left(-i \frac{kl\pi}{u-l} \right) \right\} \cos \left(k\pi \frac{v-l}{u-l} \right).$$

Finally, the characteristic function is numerically integrated with the Clenshaw-Curtis quadrature rule, denoted with ϕ_V^{CC} . Therefore, the final approximation is given by f_V^3 ,

$$f_V(v) \approx f_V^3 = \sum_{k=0}^{K_{\text{COS}}-1} \frac{2}{u-l} \operatorname{Re} \left\{ \phi_V^{CC} \left(\frac{k\pi}{u-l} \right) \cdot \exp \left(-i \frac{kl\pi}{u-l} \right) \right\} \cos \left(k\pi \frac{v-l}{u-l} \right).$$

6.3.2. THEORETICAL ERROR BOUNDS

By examining the error sources present in the COS method, it becomes clear that there are three main factors contributing to the overall error: the series truncation error, the error from the substitution of Fourier coefficients A_k with F_k , and the numerical integration error in the calculation of the characteristic function. Hence, the total error is bounded by the sum of these three individual error components,

1. The series truncation error,

$$\begin{aligned}\varepsilon_1 &:= f_V - f_V^1 = \sum_{k=K_{\text{COS}}}^{\infty} A_k \cos\left(k\pi \frac{v-l}{u-l}\right) \\ &= \sum_{k=K_{\text{COS}}}^{\infty} \frac{2}{u-l} \operatorname{Re} \left\{ \hat{\phi}_V \left(\frac{k\pi}{u-l} \right) \cdot \exp\left(-i \frac{kl\pi}{u-l}\right) \right\} \cos\left(k\pi \frac{v-l}{u-l}\right).\end{aligned}$$

2. Replacement of the coefficients A_k by F_k ,

$$\varepsilon_2 := f_V^1 - f_V^2 = \sum_{k=0}^{K_{\text{COS}}-1} \frac{2}{u-l} \operatorname{Re} \left\{ \int_{\mathbb{R} \setminus [l, u]} e^{ik\pi \frac{v(v-l)}{u-l}} f_V(v) dv \right\} \cos\left(k\pi \frac{v-l}{u-l}\right).$$

3. Numerical integration error of the characteristic function with the Clenshaw-Curtis quadrature rule,

$$\varepsilon_3 := f_V^2 - f_V^3 = \sum_{k=0}^{K_{\text{COS}}-1} \frac{2}{u-l} \operatorname{Re} \left\{ \left[\phi_V \left(\frac{k\pi}{u-l} \right) - \phi_V^{\text{CC}} \left(\frac{k\pi}{u-l} \right) \right] \cdot \exp\left(-i \frac{kl\pi}{u-l}\right) \right\} \cos\left(k\pi \frac{v-l}{u-l}\right).$$

The first two error components are well studied in literature, including the original paper of the COS method by Fang and Oosterlee [30]. Therefore, these error sources will be discussed only briefly. The series truncation error ε_1 is similar to the one described in the CPD error analysis in Section 6.2. Consequently, we can use the theory from Section 6.1 to conclude that the error converges exponentially in K for density functions in the class $C^\infty([a, b])$, resulting in an upper bound for ε_1 given by,

$$|\varepsilon_1| < P e^{-(K-1)v},$$

with $v > 0$ and $P > 0$ a term that varies less than exponentially with K . In case f_V has a discontinuity in one of its derivatives, an algebraic convergence of the Fourier-cosine series expansion will be obtained, for which the error can be bounded as follows,

$$|\varepsilon_1| < \frac{1}{(n-1)(K-1)^{n-1}},$$

with n the algebraic index of convergence of the series, as defined in Definition 6.1.1.

Next, we will analyze the error ε_2 that arises from replacing the coefficients A_k with F_k . This error is affected by the integration truncation errors, as proven in the following lemma.

Lemma 6.3.1. *Error ε_2 consists of integration range truncation errors, and can be bounded by:*

$$|\varepsilon_2| \leq Q \cdot |\varepsilon_4|,$$

where Q is a positive constant dependent on the number of expansion terms K_{COS} and

$$\varepsilon_4 := \int_{\mathbb{R} \setminus [l, u]} f_V(v) dv.$$

Proof. The proof is similar to the one presented in [30]. First, we assume that the function $f_V(v)$ is a real function. Since f_V represent a joint density function, this assumption holds for all our encountered problems. In case f_V is real, we can write ε_2 in the following way,

$$\varepsilon_2 = \sum_{k=0}^{K_{\text{COS}}-1} \frac{2}{u-l} \left\{ \int_{\mathbb{R} \setminus [l, u]} \cos\left(\frac{V(v)-l}{u-l}\right) f_V(v) dv \right\} \cdot \cos\left(k\pi \frac{v-l}{u-l}\right).$$

Next, the triangular inequality is applied to obtain an upper bound for ε_2 . Taking absolute values on both sides we find

$$\begin{aligned} |\varepsilon_2| &\leq \sum_{k=0}^{K_{\text{COS}}-1} \frac{2}{u-l} \left\{ \left| \int_{\mathbb{R} \setminus [l, u]} \cos\left(\frac{V(v)-l}{u-l}\right) f_V(v) dv \right| \right\} \cdot \left| \cos\left(k\pi \frac{v-l}{u-l}\right) \right| \\ &\leq \sum_{k=0}^{K_{\text{COS}}-1} \frac{2}{u-l} \left\{ \left| \int_{\mathbb{R} \setminus [l, u]} \cos\left(\frac{V(v)-l}{u-l}\right) f_V(v) dv \right| \right\}, \end{aligned}$$

where we have used that $|\cos(x)| \leq 1$. Since f_V considers a density function, we have $f_V \geq 0$, so that we find

$$\begin{aligned} |\varepsilon_2| &\leq \sum_{k=0}^{K_{\text{COS}}-1} \frac{2}{u-l} \left\{ \left| \int_{\mathbb{R} \setminus [l, u]} \cos\left(\frac{V(v)-l}{u-l}\right) f_V(v) dv \right| \right\} \\ &\leq \sum_{k=0}^{K_{\text{COS}}-1} \frac{2}{u-l} \left\{ \int_{\mathbb{R} \setminus [l, u]} \left| \cos\left(\frac{V(v)-l}{u-l}\right) \right| f_V(v) dv \right\} \\ &\leq \sum_{k=0}^{K_{\text{COS}}-1} \frac{2}{u-l} \left\{ \int_{\mathbb{R} \setminus [l, u]} f_V(v) dv \right\} \\ &= \frac{2K_{\text{COS}}}{u-l} \cdot \left\{ \int_{\mathbb{R} \setminus [l, u]} f_V(v) dv \right\} \\ &= Q \cdot \varepsilon_4, \end{aligned}$$

where $Q := \frac{2K_{\text{COS}}}{u-l}$ depends on the number cosine expansion terms K_{COS} , and ε_4 represented the integration range truncation error,

$$\varepsilon_4 := \int_{\mathbb{R} \setminus [l, u]} f_V(v) dv.$$

The size of ε_4 error depends on the integration range $[l, u]$. □

Finally, the integration error of the characteristic function via the Clenshaw-Curtis is investigated. Therefore, we present the following lemma, which is adapted from the working paper [64].

Lemma 6.3.2. *The error in the recovered density function f_V due to the numerical integration is bounded by,*

$$|\varepsilon_3| \leq K_{\text{COS}} \cdot |\varepsilon(J, \text{TOL})|,$$

where K_{COS} denotes the number of expansion terms in the Fourier-cosine approximation of f_V , J is the number of points adopted in the Clenshaw-Curtis quadrature rule, TOL is the integration range truncation error, and $\varepsilon(J, \text{TOL})$ is the total error arising from the approximation of the characteristic function, which converges to 0 as $J \rightarrow \infty$ and $\text{TOL} \rightarrow 0$.

Proof. First, the Fourier-cosine series expansion without the numerical integration error is considered, which is given by

$$f_V(v) = \sum_{k=0}^{K_{\text{COS}}-1} F_k \cos\left(k\pi \frac{v-l}{u-l}\right). \quad (6.3)$$

Including the numerical integration error when solving the characteristic function, we have

$$\hat{f}_V(v) = \sum_{k=0}^{K_{\text{COS}}-1} (F_k + \varepsilon(J, \text{TOL}, k)) \cos\left(k\pi \frac{v-l}{u-l}\right), \quad (6.4)$$

where $\varepsilon(J, \text{TOL}, k)$ represents the numerical error due to the Clenshaw-Curtis quadrature rule, utilized with J quadrature points. Note that this error depends on the expansion number k . The total error due to the numerical integration can now be obtained by taking the difference of 6.3 and 6.4,

$$\begin{aligned} |\varepsilon_3| = |\hat{f}_V - f_V| &= \left| \sum_{k=0}^{K_{\text{COS}}-1} \varepsilon(J, \text{TOL}, k) \cos\left(k\pi \frac{v-l}{u-l}\right) \right| \\ &\leq \sum_{k=0}^{K_{\text{COS}}-1} \left| \varepsilon(J, \text{TOL}, k) \cos\left(k\pi \frac{v-l}{u-l}\right) \right| \\ &\leq \sum_{k=0}^{K_{\text{COS}}-1} |\varepsilon(J, \text{TOL}, k)|. \end{aligned}$$

Next, we observe that the set of errors $\{\varepsilon(J, \text{TOL}, k), k \geq 0\}$ has a uniform bound. This can be demonstrated by considering the approximation of the characteristic function $\phi(\omega)$, denoted with $\hat{\phi}(\omega)$. In the context of the COS-CPD method, numerical integration is performed on functions of the form $e^{i\omega V(x)} \cos\left(k\pi \frac{x-a}{b-a}\right)$, where $V(x)$ is the part of the pricing function that corresponds to the risk factor x . For this integrand, we have $|e^{i\omega V(x)}| \leq 1$ for all $\omega = k\pi/(b-a)$, $k \geq 0$ and $|\cos\left(k\pi \frac{x-a}{b-a}\right)| \leq 1$ for all $k \geq 0$. By using these bounds, we can determine that the integration error can be bounded independently of

k . Consequently, there exists a uniform bound, denoted by $\varepsilon(J, TOL)$, that applies to all k . Since $\varepsilon(J, TOL)$ bounds each $\varepsilon(J, TOL, k)$, we can write

$$\begin{aligned} |\varepsilon_3| = |\hat{f}_V - f_V| &\leq \sum_{k=0}^{K_{COS}-1} |\varepsilon(J, TOL, k)| \\ &\leq \sum_{k=0}^{K_{COS}-1} |\varepsilon(J, TOL)| \\ &= K_{COS} \cdot |\varepsilon(J, TOL)|. \end{aligned}$$

□

The integrated functions considered in the COS-CPD method take the form $e^{i\omega V} \cdot \cos(k\pi \frac{v-a}{b-a})$, which are analytic functions. Therefore, we know that the convergence of the Clenshaw-Curtis quadrature rule will be exponential in the number of quadrature points, as can be found in Theorem 19.3 from [65].

Collecting all the results, we can conclude that the total error in the COS method converges exponentially for density functions in $C^\infty([a, b])$, and the upper bound can be formulated as

$$|\varepsilon_{COS}| < P e^{-(K_{COS}-1)v} + Q \cdot |\varepsilon_4| + K_{COS} \cdot |\varepsilon(J, TOL)|. \quad (6.5)$$

With the derivation of the error upper bounds for both the COS and CPD components, we can establish an upper bound for the total error in the COS-CPD method. Since the error in the CPD computation is related with the numerical integration error $\varepsilon(J, TOL)$, we can insert the expression of ε_{CPD} from 6.2 into the upper bound for the COS method in 6.5. This allows us to derive the following upper bound for the total error in the COS-CPD method,

$$|\varepsilon_{COS-CPD}| < P e^{-(K_{COS}-1)v} + Q \cdot |\varepsilon_4| + K_{COS} \cdot |g(\varepsilon(J, TOL), \varepsilon_{CPD})|,$$

where g represents the relation between the errors arising from the numerical integration and the CPD computation.

7

APPLICATION OF COS-CPD IN COUNTERPARTY CREDIT RISK QUANTIFICATIONS

In this chapter, we will examine the application of the COS-CPD method in the context of exposure calculation. The first section covers the modelling and numerical results of a three-dimensional model of risk factors. To accomplish this, the general methodology outlined in Chapter Chapter 5 is adapted to this three-dimensional model. Once the model is established, the numerical results of calculating the PFE and EE profiles using the COS-CPD method will be presented. First, the COS-CPD method will be tested on a single XCS contract to evaluate the validity of the method and conduct a sensitivity analysis on the rank used in the CPD approximation. After determining a suitable rank, the COS-CPD method will be applied to netting-set level exposures by considering portfolios of up to 1000 interest rate and FX derivatives. The accuracy of the COS-CPD method will be analyzed, and the results will be compared with those obtained using the MC approach, for a range of different numbers of simulated paths. Additionally, a comparison will be made between the COS-CPD and MC methods in terms of computation time for generating the exposure profiles.

The second section will present the COS-CPD method for computing the PFE and EE profile in a five-dimensional model of risk factors, in order to demonstrate the application of the method in a high-dimensional setting. The structure of this section will be similar to that of the three-dimensional case. Specifically, the COS-CPD model will be specified for the five-dimensional model, after which numerical results regarding netting-set level exposure will be presented and compared with the MC approach.

7.1. THREE-DIMENSIONAL CASE

7.1.1. MODEL

In the three-dimensional model, the risk factors will include two interest rates from different currencies and the exchange rate that relates these currencies. The G1++ model will be employed to model the short rate processes $r_d(t)$, $r_f(t)$. The subscripts d and f indicate whether the domestic or foreign interest rate process is being considered. Since only the shifted short-rate processes $x_d(t)$, $x_f(t)$ are of significance, only the Ornstein-Uhlenbeck dynamics of these processes will be taken into account. Additionally, a GBM model will be utilized to model the corresponding FX rate. In the context of risk management, the computation of PFE and EE profiles is often performed using the real-world measure \mathbb{P} . Therefore, the dynamics of the three-dimensional system of risk factors are given by

$$\begin{cases} dx_d(t) &= -a_d x_d(t)dt + \sigma_d dW_d^{\mathbb{P}^d}, \\ dx_f(t) &= -a_f x_f(t)dt + \sigma_f dW_f^{\mathbb{P}^f}, \\ dX(t) &= \mu X(t)dt + \sigma X(t)dW_X^{\mathbb{P}^d}, \end{cases} \quad (7.1)$$

with a_d, a_f the mean-reversion coefficients and σ_d, σ_f the volatilities of the domestic and foreign currency. The parameters μ and σ are the drift and volatility in the FX rate process, respectively.

The goal is to compute the PFE and EE of a derivative portfolio V , which contains various interest rate and FX products depending on the risk factors x_d, x_f and X . Using the COS-CPD method, the CDF and PDF of the MtM value of the portfolio will be recovered via the COS method. As presented in Equation 5.5, this requires the evaluation of the following characteristic function,

$$\phi_V(\omega) = \iiint_{\mathbb{R}^3} e^{i\omega V(x_d, x_f, X; t)} f(x_d, x_f, X; t) dx_d dx_f dX,$$

with $f(x_d, x_f, X; t)$ representing the joint density function of the risk factors. The parameter t is included to emphasize that the distribution of the risk factors is time dependent.

It was established in Section 2.4.2 that the FX rate under the GBM model is log-normally distributed. It will be more convenient to work with the normal distribution in the remainder of the computations. Therefore, the logarithm of the FX-rate will be considered, for which the dynamics are given by

$$d \log(X_t) = \left(\mu - \frac{1}{2} \sigma^2 \right) dt + \sigma dW_X^{\mathbb{P}^d}.$$

The characteristic function can be expressed in terms of $\log(X_t)$ as follows,

$$\phi_V(\omega) = \iiint_{\mathbb{R}^3} e^{i\omega V(x_d, x_f, e^{\log(X)}; t)} \tilde{f}(x_d, x_f, \log(X); t) dx_d dx_f d \log(X), \quad (7.2)$$

where the tilde symbol, $\tilde{f}(\cdot)$, indicates a change of the joint density function. Since the short-rate processes x_d and x_f are normally distributed as well, the joint density $\tilde{f}(x_d, x_f, \log(X); t)$ is now transformed into a multivariate normal distribution.

The first step in the COS-CPD method is to decompose the joint density function into its Fourier-cosine series using the dimension-reduced CPD technique. The joint density expressed in 7.2 incorporates all time-dependent parameters of the model. Consequently, it is necessary to execute a new CPD approximation of the density for each future time point at which the PFE and EE have to be computed. This approach can result in a substantial computational cost, particularly when exposure profiles under consideration extend to multiple years in the future. However, it is possible to reduce the impact of the time dependency of the joint density function by re-writing the state variables x_d, x_f and X , using the properties of the normal distribution,

$$\begin{aligned}x_d &\stackrel{d}{\sim} \mu_{x_d} + \sigma_{x_d}(t)Z_d(t), \\x_f &\stackrel{d}{\sim} \mu_{x_f} + \sigma_{x_f}(t)Z_f(t), \\\log(X_t) &\stackrel{d}{\sim} \mu_X(t) + \sigma_X(t)Z_X(t),\end{aligned}$$

where Z_d, Z_f, Z_X are standard normally distributed, i.e. $Z_d, Z_f, Z_X \sim N(0, 1)$. The time parameter t indicates that the correlation is still time-dependent between the new state variables, and are given by

$$\begin{aligned}\text{Cor}(Z_d, Z_f) &= \frac{\frac{\rho_{df}}{a_d + a_f}(1 - e^{-(a_d + a_f)t})}{\sqrt{\frac{1}{2a_d}(1 - e^{-2a_d t})} \sqrt{\frac{1}{2a_f}(1 - e^{-2a_f t})}}, \\\text{Cor}(Z_d, Z_X) &= \frac{\frac{\rho_{dX}}{a_d}(1 - e^{-a_d t})}{\sqrt{\frac{1}{2a_d}(1 - e^{-2a_d t})} \sqrt{t}}, \\\text{Cor}(Z_f, Z_X) &= \frac{\frac{\rho_{fX}}{a_f}(1 - e^{-a_f t})}{\sqrt{\frac{1}{2a_f}(1 - e^{-2a_f t})} \sqrt{t}}.\end{aligned}$$

A proof for these correlation expressions can be found in Appendix B.3. In this way, the joint density function is transformed into the multivariate standard normal density, which only captures the correlation structure of the model. Consequently, a CPD operation needs to be performed only if the correlations structure changes significantly. In order to obtain insight into the factors that contribute to changes in the correlation, the

Taylor series expansion centered around $t = 0$ is examined,

$$\begin{aligned}\text{Cor}(Z_d, Z_f) &\approx \rho_{df} - \frac{\rho_{df}(a_d - a_f)^2}{24} t^2 + \mathcal{O}(t^4), \\ \text{Cor}(Z_d, Z_X) &\approx \rho_{dX} - \frac{\rho_{dX} a_d^2}{24} t^2 + \mathcal{O}(t^4), \\ \text{Cor}(Z_f, Z_X) &\approx \rho_{fX} - \frac{\rho_{fX} a_f^2}{24} t^2 + \mathcal{O}(t^4).\end{aligned}$$

We can see that the correlation change over time is dominated by the magnitude of the mean-reversion coefficients, together with their difference. Hence, these parameter values can provide an indication of the number of CPD operations needed to calculate exposure profiles over a particular time horizon.

Expressing 7.2 in terms of the state variables Z_d, Z_f and Z_X results in

$$\begin{aligned}\phi_V(\omega) &= \iiint_{\mathbb{R}^3} e^{i\omega V(\mu_{x_d} + \sigma_{x_d}(t)Z_d, \mu_{x_f} + \sigma_{x_f}(t)Z_f, \mu_X(t) + \sigma_X(t)Z_X; t)} f(Z_d, Z_f, Z_X; t) dZ_d dZ_f dZ_X \\ &= \iiint_{\mathbb{R}^3} e^{i\omega \tilde{V}(Z_d, Z_f, Z_X; t)} f(Z_d, Z_f, Z_X; t) dZ_d dZ_f dZ_X,\end{aligned}$$

where $\tilde{V}(\cdot)$ indicates that the portfolio function is written in terms of the new state variables Z_d, Z_f, Z_X .

As mentioned before, the first step is to decompose the joint density function into its Fourier-cosine series using CPD. As shown in Equation 3.6, the resulting Fourier-cosine series is of the form

$$f(Z_d, Z_f, Z_X; t) = \sum_{r=1}^R \prod_{n=1}^3 \mathbf{v}_n^T \mathbf{A}_n[:, r],$$

where $\{\mathbf{A}_n\}_{n=1}^3$ the factor matrices and $\{\mathbf{v}_n\}_{n=1}^3$ the cosine basis functions of the state variables Z_d, Z_f and Z_X . The integration range of the characteristic function will be truncated according to the expansion interval of the Fourier-cosine series. This expansion interval is determined by the PPF of the standard normal distribution at a specified level of accuracy. As all state variables have a standard normal distribution, the expansion interval will be the same for each dimension, resulting in the cubic integration domain $[a, b]^3$. After the first COS-CPD step, the characteristic function is given by,

$$\hat{\phi}_V(\omega) = \sum_{r=1}^R \iiint_{[a,b]^3} e^{i\omega \tilde{V}(Z_d, Z_f, Z_X; t)} \prod_{n=1}^3 \mathbf{v}_n^T \mathbf{A}_n[:, r] dZ_d dZ_f dZ_X.$$

The second step in the COS-CPD method is to combine the paying and receiving legs with the same risk factors involved. In this three-dimensional model, two distinct groups of risk factors arise in the portfolio legs; (1) the domestic currency x_d , and (2) the foreign currency x_f together with the exchange rate X . The general formula for the characteristic function in the case of disjoint risk factor groups was derived in Equation 5.8. In this case, it can be expressed as follows,

$$\begin{aligned}
\hat{\phi}_V(\omega) &= \sum_{r=1}^R \iiint_{[a,b]^3} e^{i\omega[\tilde{V}_1(Z_d;t) + \tilde{V}_2(Z_f, Z_X;t)]} \prod_{n=1}^3 \mathbf{v}_n^T \mathbf{A}_n[:, r] dZ_d dZ_f dZ_X \\
&= \sum_{r=1}^R \iiint_{[a,b]^3} e^{i\omega \tilde{V}_1(Z_d;t)} e^{i\omega \tilde{V}_2(Z_f, Z_X;t)} \prod_{n=1}^3 \mathbf{v}_n^T \mathbf{A}_n[:, r] dZ_d dZ_f dZ_X \\
&= \sum_{r=1}^R \int_{[a,b]} e^{i\omega \tilde{V}_1(Z_d;t)} \mathbf{v}_1^T \mathbf{A}_1[:, r] dZ_d \cdot \iint_{[a,b]^2} e^{i\omega \tilde{V}_2(Z_f, Z_X;t)} \mathbf{v}_2^T \mathbf{A}_2[:, r] \mathbf{v}_3^T \mathbf{A}_3[:, r] dZ_f dZ_X.
\end{aligned} \tag{7.3}$$

Equation 7.3 can also be presented element-wise by working out the inner products $\mathbf{v}^T \mathbf{A}[:, r]$,

$$\begin{aligned}
\hat{\phi}_V(\omega) &= \sum_{r=1}^R \left\{ \sum_{k_1=0}^{K-1} \mathbf{A}_1[k_1, r] \int_{[a,b]} e^{i\omega \tilde{V}_1(Z_d;t)} \cos\left(k_1 \pi \frac{Z_d - a}{b - a}\right) dZ_d \right\} \\
&\quad \left\{ \sum_{k_2=0}^{K-1} \sum_{k_3=0}^{K-1} \mathbf{A}_2[k_2, r] \mathbf{A}_3[k_3, r] \iint_{[a,b]^2} e^{i\omega \tilde{V}_2(Z_f, Z_X;t)} \cos\left(k_2 \pi \frac{Z_f - a}{b - a}\right) \cos\left(k_3 \pi \frac{Z_X - a}{b - a}\right) dZ_f dZ_X \right\}.
\end{aligned}$$

This expression can now be inserted in the COS formulas from 2.14 and 2.15 to recover the PDF and CDF of the MtM value of the portfolio, after which the PFE and EE can be derived.

7.1.2. NUMERICAL RESULTS

With the completion of the modeling phase, we can assess the performance of the COS-CPD method through numerical analysis. This analysis starts by calculating the PFE and EE profile for a single XCS. After that, the COS-CPD method is applied at the level of a netting set by computing the exposure profiles of a portfolio comprising up to 1000 derivatives. The results of this analysis are then compared in both accuracy and computation time with those obtained via the standard MC approach.

The selection of model parameters in this study is consistent with the methodology outlined in [66]. Specifically, the parameters for the three-dimensional model described in 7.1 are set as follows:

$$a_d = 1\%, a_f = 5\%, \mu = 0.8\%, \sigma_d = 0.7\%, \sigma_f = 1.2\%, \sigma = 2\%.$$

The domestic currency represents the United States dollar (USD), while the Japanese yen (JPY) is modelled as the foreign currency. The initial exchange rate is set to $X(0) = 105.00$ yen per dollar.

In order to compute the zero-coupon bond (ZCB) price under the G1++ model, the market discount curves for both the domestic and foreign currencies are required. These will be defined as $P_d^M(0, T) = e^{-0.02T}$ and $P_f^M(0, T) = e^{-0.05T}$, respectively. Additionally, the correlation structure between the risk factors is specified as $\rho_{df} = 25\%$, $\rho_{dX} = -15\%$, and $\rho_{fX} = -15\%$.

Next, the parameters used in the CPD model will be outlined. Specifically, the number of training data points per dimension will be set at $m = 12$, resulting in a total of

$M = m^3 = 12^3$ training data points. As determined in the sensitivity analysis of the CPD model in Chapter 4, the number of expansion terms per dimension will be set at $K = m = 12$. Additionally, Figure 4.12a showed that employing a tolerance level of 10^{-8} in the expansion interval of $[a, b]^3$ is a suitable choice in the case $K = m = 12$. This expansion interval is then given by $[-5.612, 5.612]^3$, using the PPF of the standard normal distribution. Furthermore, the tolerance levels in the CG solver and the ALS updating will be set to 10^{-12} , with a maximum of 5 ALS iterations and $K \cdot R$ iterations within the CG solver per factor matrix, where the rank R will be varied in the analysis of the single XCS to determine a suitable value during the netting-set level analysis.

The appropriate number of expansion terms in the COS formula and Clenshaw-Curtis quadrature points is based on the numerical analysis presented in [36]. Therefore, we will use 32 expansion terms in the COS method and 50 quadrature points per dimension in the Clenshaw-Curtis integration.

Before the numerical tests are performed, it is necessary to determine the number of CPD approximations of the joint density function that must be generated due to the time-varying nature of the correlation structure, as was explained in the previous section. From Figure 7.1, it can be observed that for the mean reversion coefficients used in this analysis, the correlation structure remains nearly constant over a 15-year period (the longest duration considered). Therefore, only one CPD execution is performed for the joint density function, with correlation structure $\rho_{df} = 25\%$, $\rho_{dX} = -15\%$ and $\rho_{fX} = -15\%$, which is then used for all time points to compute the PFE and EE profile.

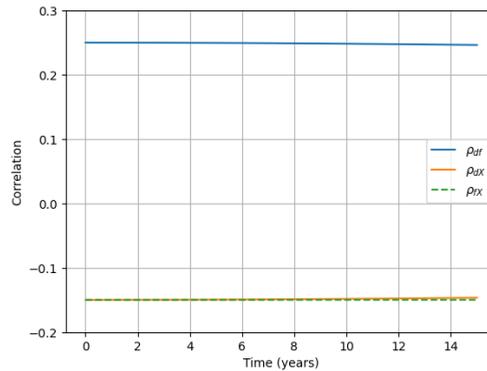


Figure 7.1: Correlation between state variables Z_d , Z_f , and Z_X over a period of 15 years.

CROSS-CURRENCY SWAP

The cross-currency swap (XCS) examined in this analysis has an initiation date of $T_\alpha = 1$ year and concludes at $T_\beta = 10$ years, at which point 10 payments have been made. In order to ensure a fair value of the contract at the initial time $t = 0$, the fixed rate is set to $K = 0.02$. The domestic notional is set to $N_d = 1000$. In determining the COS support $[l, u]$ on which the PDF and CDF will be recovered, the rule-of-thumb provided in [36] will be employed. Specifically, $[l, u] = [\pm N_d/2] = [\pm 500]$ for $t \leq 9$ years and $[\pm 50]$ for $9 < t < 10$.

The PFE and EE of the XCS will be evaluated at 50 distinct time points. At each time point, the PDF and CDF of the MtM value of the XCS will be obtained. Subsequently, the exposure distribution can be obtained by taking the maximum of the MtM value and zero. An illustration of the recovered PDF and CDF profiles for both the MtM value and the exposure at time $t = 3.4$ can be found in Figure 7.2. By applying the maximum operator on the MtM distribution afterwards, there is no Gibbs phenomena occurring at the discontinuity point in the exposure distribution. This outcome is not achievable if the PDF and CDF of the exposure were recovered directly using the maximum operator in the evaluation. Consequently, a higher level of accuracy of the exposure PDF and CDF is obtained by this approach.

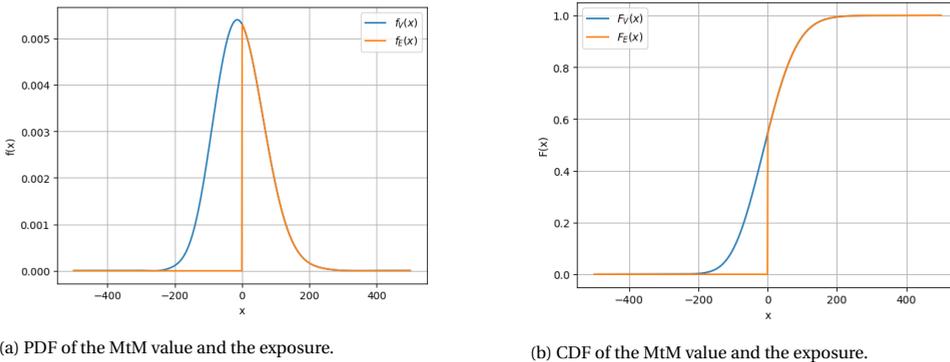


Figure 7.2: Recovered PDF and CDF of the MtM value and the exposure of the XCS at time $t = 3.4$ using the COS-CPD method.

Figure 7.3 illustrates the PFE and EE profile of the XCS using the COS-CPD method, employing a rank 5 approximation in the CPD. The profile was generated using 50 equidistant time points from $t = 0$ up to $t = 11$. The Monte Carlo simulation was used as a benchmark, as it was computed using 5 million paths for each risk factor. The PFE and EE profiles presented in Figure 7.3 are typical for a XCS derivative. The PFE and EE begin with a zero value, as the fixed rate K is chosen such that the initial value of the XCS is zero. Due to the uncertainty in the movement of the risk factors, the PFE and EE increases. The saw-tooth pattern observed is a consequence of intermediate payments over time. Once a payment is settled, the value of the PFE and EE drops as the total potential loss is reduced. Then the exposure rises again due to the uncertainty in the development of the risk factors. At some point, the PFE and EE start to decrease, as most of the payments have been made and hence the remaining value of the XCS decreases as well. After the maturity date $T_\beta = 10$, the PFE and EE profile are zero again. The profile generated using the COS-CPD method is consistent with the MC benchmark, indicating the validity of the mathematical derivation of the COS-CPD method.

In order to determine an appropriate rank within the CPD, it is necessary to quantify the error present in COS-CPD estimations. To this end, the relative error in the COS-CPD for various ranks is presented in Table 7.1. A MC benchmark of five million paths per risk factor serves as the reference value. The relative error then illustrates the difference between the two methods, expressed as a percentage of the MC benchmark. For example,

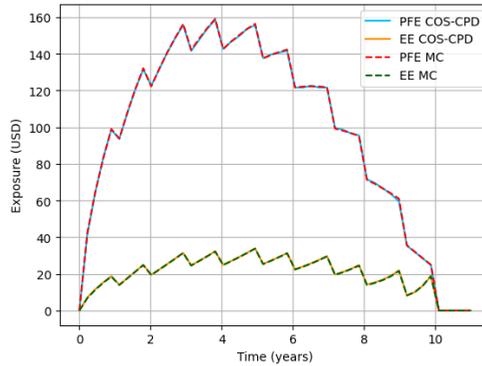


Figure 7.3: The PFE and EE profile of the XCS computed using both the COS-CPD method and a MC benchmark using $5 \cdot 10^6$ simulations. The PFE and EE values are computed for 50 distinct time points.

the relative error in the PFE is computed using the following formula:

$$\text{Relative Error PFE} = \frac{|PFE_{\text{COS-CPD}} - PFE_{\text{MC}}|}{PFE_{\text{MC}}}$$

The maximum error represents the largest relative error at all time points, while the time-averaged error is defined as the average of all relative errors computed for all time points. As seen in Table 7.1, the maximum error in the EE approximation can be decreased with a higher rank approximation. However, the maximum error in the PFE does not improve as the rank increases, suggesting that this error peak may be caused by an improper selection of the COS support for certain time points. Based on the time-averaged errors, a rank 10 approximation in the CPD part is sufficient to achieve acceptable results for both PFE and EE. Therefore, remaining numerical analysis in the three-dimensional case will be carried out using the rank 10 approximation in the CPD.

Table 7.1: Maximum and time-averaged relative error in the PFE and EE computation of the XCS using the COS-CPD method for different rank, using a MC benchmark of $5 \cdot 10^6$ paths. The PFE and EE values are computed for 50 distinct time points.

rank	Max Error (%)		Time-averaged Error (%)	
	PFE	EE	PFE	EE
5	2.38	4.39	0.34	0.65
10	2.38	0.55	0.48	0.24
15	2.38	0.51	0.45	0.24

To highlight the significance of a changing COS support for $t > 9$, a comparison is made assuming the support is fixed at $[-500, 500]$. As the majority of the payments have been made near maturity time $T_\beta = 10$, the MtM value of the XCS will not be affected by most realisations of the risk factors. As a result, the corresponding PDF and CDF will be centered very closely around a single value, resulting in a steep PDF and CDF profile. This steepness of the recovered CDF near maturity is depicted in 7.4a. When using a

COS support of $[-500, 500]$, the number of expansion terms is insufficient to accurately recover the CDF. This leads to oscillatory behavior in the CDF profile and hence an inaccurate approximation of the PFE value, as can be seen from 7.4b. To resolve this issue, either the COS support must be decreased to a smaller interval or the number of expansion terms must be increased. This will reduce the oscillatory behavior and enhance the accuracy of the PFE calculation. As can be observed, a COS support of $[-50, 50]$ is able to capture the steepness of the recovered CDF, resulting in accurate PFE values close to maturity.

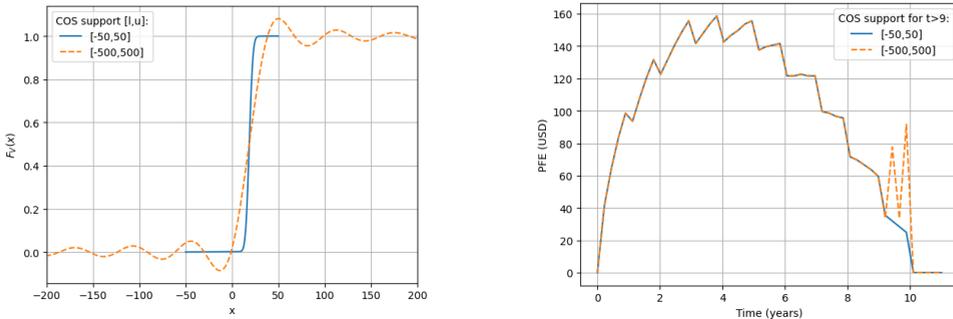
(a) CDF for different COS support at $t = 9.9$.(b) PFE for different COS support at $t > 9.0$.

Figure 7.4: Recovered CDF of the MtM value of the XCS at time $t = 9.9$ for different COS support $[l, u]$, together with the PFE profile when using different COS support for $t > 9.0$.

NETTING-SET LEVEL EXPOSURE

This section focuses on the COS-CPD method in the context of netting-set level exposure. The PFE and EE profile of various portfolio sizes will be examined. The portfolios are generated randomly using a Python script and consist of four distinct types of liquid derivatives, namely FRA, IRS, FX, and XCS products. In all types of products, the payment legs may be in either domestic or foreign currency. The maturity and notional for these products are also generated randomly, as well as the fixed rate and the number of payments (if applicable) until maturity. All the portfolios up to 100 derivatives used in this numerical analysis can be found in Appendix E.

We will analyze the COS-CPD method for portfolio sizes up to 1000 derivatives. Similar to the individual XCS analysis, a MC simulation of 5 million paths will be employed as a benchmark to evaluate the accuracy of the COS-CPD model. Furthermore, all parameter values are identical to the ones used in the XCS analysis. In regards to the rank in the CPD model, a rank 10 approximation will be adopted based on the findings in Table 7.1.

Figure 7.5 illustrates the PFE and EE profile of the netting-set level exposure for a portfolio consisting of 100 derivatives. The PFE and EE values were evaluated at 50 equidistant time points, ranging from $t = 0$ to the longest maturity present, which was $T = 14.7$ for this particular portfolio. In line with the methodology from [36], the COS support is set at $[-3000, 15000]$ at $t = 0$. Additionally, the COS support is altered to $[-1000, 2000]$ when the PFE value drops below 1500. However, an additional change in COS support is implemented here. Starting from $t = 13.7$, which corresponds to the last

year, the COS support is set at $[-125, 125]$, based on the few remaining coupon values that have yet to be exchanged.

The EE profile begins to decline slightly in the initial four years as the derivatives with a short maturity date expire and no longer have an impact on the portfolio exposure. Conversely, the PFE profile increases during these initial years as the uncertainty in the development of the risk factors increases with time. As a result, the recovered exposure PDF will have a wider range of possible outcomes, leading to a higher 97.5% quantile of the distribution and hence an increase in the PFE value. After around five years, both the EE and PFE values decrease rapidly, indicating that a large number of derivatives mature during this period up to year eight. As the final maturity date approaches, the number of active derivatives in the portfolio decreases to zero, resulting in the PFE and EE converging to zero at maturity.

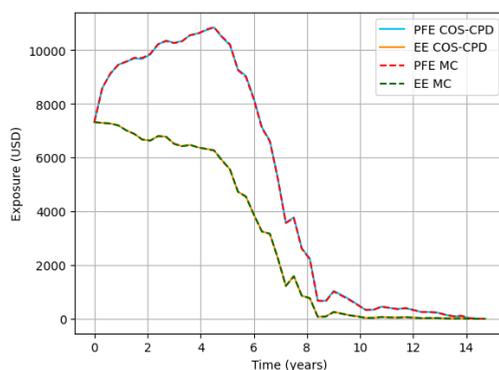
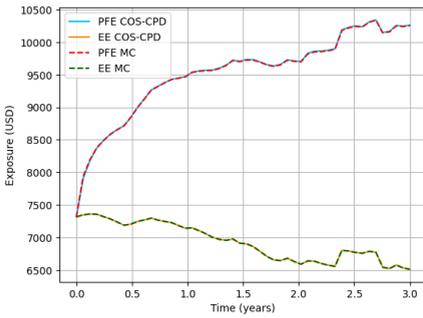
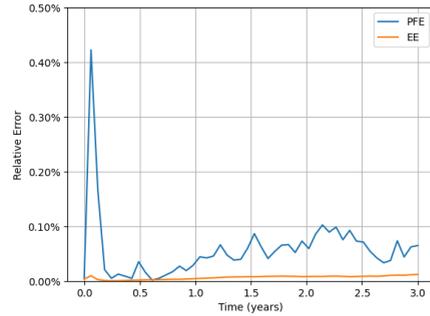


Figure 7.5: PFE and EE profile at a netting-set level for a portfolio consisting of 100 derivatives, computed at 50 equidistant points.

In practice, the computation of PFE and EE profiles is not performed only once for the entire lifetime of the portfolio. Instead, these profiles are computed frequently, with a focus on the exposure values close to the current time point. This is because financial institutions have to allocate a certain amount of capital reserve based on the computed exposure levels. Therefore, the accuracy of the PFE and EE values within the first few years is of more importance, and so the error quantification discussed next will focus on the estimations made in the first three years of the portfolio's lifetime. Again, 50 equidistant time points are considered within this time period, on which the PFE and EE values of the netting-set portfolio are computed. The resulting PFE and EE profiles are shown in Figure 7.6a, from which can be observed that the COS-CPD estimation is again consistent with the MC benchmark of $5 \cdot 10^6$ paths. The relative error is displayed in Figure 7.6b. It shows a peak of 0.40% at the first time point, as the COS support might be too wide for the first time step. Although this level of accuracy is still satisfactory, it could be improved by choosing a more appropriate COS support for the first time steps. After that, the relative error for both the PFE and EE is below 0.10%, which demonstrates the accuracy of the COS-CPD approach compared to the MC benchmark.



(a) PFE and EE profile up to time $t = 3.0$.



(b) Relative error in the COS-CPD method as a percentage of the MC benchmark.

Figure 7.6: The PFE and EE profile of the netting-set level exposure the portfolio consisting of 100 derivatives for the first 3 years, together with the relative error in the resulting PFE and EE values (as a percentage of the MC benchmark).

Table 7.2 illustrates the CPU time¹ and accuracy for both the COS-CPD method and the MC method when utilizing various numbers of paths, ranging from $1 \cdot 10^5$ to $1 \cdot 10^6$. The time-averaged error is calculated using the relative differences over all time points, expressed as a percentage of the MC benchmark of $5 \cdot 10^6$ paths. It is observed that the accuracy of the Monte Carlo estimation improves with an increase in the number of simulations, which is to be expected. Additionally, the computation time of the MC simulation exhibits an approximately linear trend, implying that the computation time doubles when the number of simulations is doubled. The COS-CPD approach is more accurate compared to the MC estimate using $1 \cdot 10^5$ simulations. The other two MC estimates show slightly better accuracy in the PFE compared to the the COS-CPD approach, mainly as a result of the error peak at the first time point previously discussed. However, this difference in accuracy is practically negligible, while the computation time of the COS-CPD method is significantly faster in comparison to all MC estimations. For example, the COS-CPD is a factor of 15 faster when using $5 \cdot 10^5$ MC paths and even 30 times faster than the MC approach utilizing $1 \cdot 10^6$ paths.

Table 7.2: The accuracy and CPU time (in seconds) required to calculate the PFE and EE of netting-set level exposure of the portfolio with 100 derivatives, evaluated at 50 time points up to time $t = 3.0$. The time-averaged error was calculated by averaging the relative errors over the 50 time points.

	CPU Time ¹ (s)	Time-averaged Error (%)	
		PFE	EE
MC ($1 \cdot 10^5$)	53.7	0.116	0.007
MC ($5 \cdot 10^5$)	325.2	0.031	0.001
MC ($1 \cdot 10^6$)	644.5	0.021	0.001
COS-CPD	22.5	0.057	0.006

¹The decomposition of the joint density function into its Fourier-cosine series using CPD is not included here, as this can be done "off-line" and has to be executed only once.

The COS-CPD method offers an additional advantage in that the computation time is less sensitive to the number of derivatives in the portfolio. This in contrast to the MC method, in which the computation time increases significantly as the number of derivatives in the portfolio increases. As shown in Table 7.3, the computation time for the MC method nearly doubles when the portfolio size is doubled, while the COS-CPD method exhibits only a slight increase in computation time. The effect of the portfolio size on the computation time is particularly evident when examining the portfolio consisting of 1000 derivatives. In this scenario, the COS-CPD algorithm still requires less than one minute to calculate the PFE and EE profiles at 50 time points, while the MC method using $5 \cdot 10^5$ simulation paths takes approximately 50 minutes to generate the same profiles.

To understand the different impact on the computation time for both methods, the computational complexity of both methods have to be analysed when an additional derivative is added to the total portfolio. For the MC method, the portfolio has to be evaluated on all simulated paths, which will be denoted by N_{paths} . Adding an extra derivative to the portfolio requires evaluating this derivative on all simulated paths, resulting in an extra N_{paths} evaluations to compute the portfolio value. Therefore, the computational complexity for evaluating an extra derivative in the the MC method is of order $\mathcal{O}(N_{paths})$. Since the number of paths is often chosen in the order of 10^5 or 10^6 , doubling the portfolio size, as in Table 7.3, will have a significant negative impact on the total computation time. On the other hand, The COS-CPD method comes down to the evaluation of multiple characteristic functions, which are approximated by using the Clenshaw-Curtis quadrature method. Therefore, the portfolio is only evaluated on the total number of quadrature points used in the numerical integration, which will be denoted by N_{quad} per dimension. As only one- and two-dimensional integrals have to be computed, adding an extra derivative to the portfolio will require an additional N_{quad} evaluations in the one-dimensional integrals and N_{quad}^2 evaluations for the two-dimensional integrals. Hence, the computational complexity for evaluating an extra derivative in the COS-CPD method is $\mathcal{O}(N_{quad}^2)$. Since the number of quadrature points per dimension is set to 50, we will have $N_{quad}^2 \ll N_{paths}$, and an increase in the portfolio size will have almost no effect on the computation time in the COS-CPD method, while the MC method exhibits a significant increase in computation time.

Table 7.3: Comparison of the CPU time (in seconds) needed to compute the PFE and EE profile for a netting-set portfolio with 10, 20 and 50 time steps, respectively. The MC measurements were performed with $5 \cdot 10^5$ paths.

Number of Derivatives	COS-CPD	MC	COS-CPD	MC	COS-CPD	MC
25	3.7	18.4	7.7	39.3	20.2	103.9
50	3.9	33.7	8.1	71.9	20.9	189.7
100	4.1	57.4	8.7	124.2	22.5	325.2
1000	10.5	546.1	22.1	1163.3	55.9	2995.4

7.2. FIVE-DIMENSIONAL CASE

7.2.1. MODEL

In order to obtain the five-dimensional model of risk factors, the three-dimensional model from 7.1 is extended with an additional short rate process and its exchange rate with the domestic currency. Thus, the five-dimensional model is characterized by

$$\begin{cases} dx_d(t) &= -a_d x_d(t) dt + \sigma_d dW_d^{\mathbb{P}^d}, \\ dx_{f_1}(t) &= -a_{f_1} x_{f_1}(t) dt + \sigma_{f_1} dW_{f_1}^{\mathbb{P}^{f_1}}, \\ dx_{f_2}(t) &= -a_{f_2} x_{f_2}(t) dt + \sigma_{f_2} dW_{f_1}^{\mathbb{P}^{f_2}}, \\ dX_1(t) &= \mu_1 X_1(t) dt + \sigma_1 X_1(t) dW_{X_1}^{\mathbb{P}^d}, \\ dX_2(t) &= \mu_2 X_2(t) dt + \sigma_2 X_2(t) dW_{X_2}^{\mathbb{P}^d}. \end{cases} \quad (7.4)$$

The processes x_d , x_{f_1} and X_1 are consistent with those in the three-dimensional model, where x_d represents the (domestic) USD short rate, x_{f_1} represents the JPY short rate and X_1 models the USD/JPY exchange rate. The second foreign currency x_{f_2} introduced here will be representing the Euro (EUR), and X_2 represents the USD/EUR exchange rate.

In the five-dimensional case, the evaluation of the characteristic function requires to solve a five-dimensional integral,

$$\phi_V(\omega) = \int_{\mathbb{R}^5} e^{i\omega V(x_d, x_{f_1}, x_{f_2}, X_1, X_2; t)} f(x_d, x_{f_1}, x_{f_2}, X_1, X_2; t) dx_d dx_{f_1} dx_{f_2} dX_1 dX_2.$$

In the same way as done for the three-dimensional model, the logarithm of the exchange rates is used to obtain a multivariate normal distribution describing the risk factors in model 7.4. Furthermore, the state variables are again expressed in terms of the standard normal distribution to reduce the time dependency of the joint density function. This leads to the computation of the following characteristic function,

$$\phi_V(\omega) = \int_{\mathbb{R}^5} e^{i\omega \tilde{V}(Z_d, Z_{f_1}, Z_{f_2}, Z_{X_1}, Z_{X_2}; t)} f(Z_d, Z_{f_1}, Z_{f_2}, Z_{X_1}, Z_{X_2}) dZ_d dZ_{f_1} dZ_{f_2} dZ_{X_1} dZ_{X_2}.$$

Consequently, the first step of the COS-CPD method is to decompose the five-dimensional standard normal distribution into its Fourier-cosine series using CPD, which can be expressed as

$$f(Z_d, Z_{f_1}, Z_{f_2}, Z_{X_1}, Z_{X_2}) = \sum_{r=1}^R \prod_{n=1}^5 \mathbf{v}_n^T \mathbf{A}_n[:, r],$$

where $\{\mathbf{A}_n\}_{n=1}^5$ the factor matrices and $\{\mathbf{v}_n\}_{n=1}^5$ the cosine basis functions of the state variables $Z_d, Z_{f_1}, Z_{f_2}, Z_{X_1}$ and Z_{X_2} .

The second step of the COS-CPD method expresses the portfolio as a linear combination of all the individual legs and grouping the legs that share the same risk factors.

In this five-dimensional model, there will be three distinct groups of risk factors present in the portfolio legs: (1) the domestic USD currency x_d , (2) the JPY currency and the USD/JPY exchange rate (x_{f_1}, X_1), and (3) the EUR currency and the USD/EUR exchange rate (x_{f_2}, X_2). Therefore, the characteristic function for this five-dimensional model can be expressed as

$$\begin{aligned}\hat{\phi}_V(\omega) &= \sum_{r=1}^R \int_{[a,b]^5} e^{i\omega [\tilde{v}_1(Z_d;t) + \tilde{v}_2(Z_{f_1}, Z_{X_1};t) + \tilde{v}_3(Z_{f_2}, Z_{X_2};t)]} \prod_{n=1}^5 \mathbf{v}_n^T \mathbf{A}_n[:, r] dZ_d dZ_{f_1} dZ_{f_2} dZ_{X_1} dZ_{X_2} \\ &= \sum_{r=1}^R \int_{[a,b]^5} e^{i\omega \tilde{v}_1(Z_d;t)} e^{i\omega \tilde{v}_2(Z_{f_1}, Z_{X_1};t)} e^{i\omega \tilde{v}_3(Z_{f_2}, Z_{X_2};t)} \prod_{n=1}^5 \mathbf{v}_n^T \mathbf{A}_n[:, r] dZ_d dZ_{f_1} dZ_{f_2} dZ_{X_1} dZ_{X_2} \\ &= \sum_{r=1}^R I(Z_d) \cdot I(Z_{f_1}, X_1) \cdot I(Z_{f_2}, X_2),\end{aligned}\quad (7.5)$$

where $I(Z_d)$, $I(Z_{f_1}, X_1)$ and $I(Z_{f_2}, X_2)$ are defined by

$$\begin{aligned}I(Z_d) &= \int_{[a,b]} e^{i\omega \tilde{v}_1(Z_d;t)} \mathbf{v}_1^T \mathbf{A}_1[:, r] dZ_d, \\ I(Z_{f_1}, X_1) &= \int_{[a,b]^2} e^{i\omega \tilde{v}_2(Z_{f_1}, Z_{X_1};t)} \mathbf{v}_2^T \mathbf{A}_2[:, r] \mathbf{v}_3^T \mathbf{A}_3[:, r] dZ_{f_1} dZ_{X_1}, \\ I(Z_{f_2}, X_2) &= \int_{[a,b]^2} e^{i\omega \tilde{v}_3(Z_{f_2}, Z_{X_2};t)} \mathbf{v}_4^T \mathbf{A}_4[:, r] \mathbf{v}_5^T \mathbf{A}_5[:, r] dZ_{f_2} dZ_{X_2}.\end{aligned}$$

7

Just as in the three-dimensional scenario, the five-dimensional integral for the characteristic function is decomposed into products of only one- and two-dimensional integrals that can be calculated using a direct quadrature rule, such as Clenshaw-Curtis. Once the characteristic functions are computed, the COS method can be applied again to obtain the exposure PDF and CDF, from which the PFE and EE of the portfolio can be obtained.

7.2.2. NUMERICAL RESULTS

The numerical analysis will be equivalent to the one employed in the three-dimensional setting. Therefore, the analysis starts with the calculation of the PFE and EE profile considering only two XCS contracts, in which a sensitivity analysis is preformed on the rank used in the CPD approximation. Once a suitable rank is determined, the COS-CPD method is applied in the context of netting-set level exposure by considering portfolios comprising up to 1000 derivatives. The results of this analysis are again compared in both accuracy and computation time with those obtained via the standard MC approach.

The parameters and correlation values for the processes x_d , x_{f_1} , and X_1 are equivalent to those of the three-dimensional model. For the euro short rate x_{f_2} and USD/EUR exchange rate, the following parameter values are utilized:

$$a_{f_2} = 0.6\%, \mu_2 = 0.6\%, \sigma_{f_2} = 2.0\%, \sigma_2 = 1.5\%.$$

The initial USD/EUR exchange rate is set to $X_2(0) = 1/1.35$ euro per dollar, based on the set-up from [67]. The market discount curve for the euro will be identical to that of the Japanese yen, with $P_{f_2}^M(0, T) = e^{-0.05T}$.

Furthermore, the correlation structures with the new processes x_{f_2} and X_2 must be established. In accordance with the methodology outlined in [66], the correlation level with the new currency and exchange rate is set to:

$$\rho_{d,f_2} = 25\%, \rho_{f_1,f_2} = 25\%, \rho_{d,X_2} = -15\%, \rho_{f_1,X_2} = -15\%$$

Additionally, the correlation between the exchange rates X_1 and X_2 is determined using historical data of the USD/JPY and USD/EUR exchange rates from 01-01-2003 to 01-01-2023, obtained via Yahoo Finance. The correlation based on this data was found to be $\rho_{X_1,X_2} = 60\%$.

In the CPD model, the parameters are equivalent to those of the three-dimensional case, with the exception that a new rank in the CPD approximation will be established in the next section. Again, only one CPD approximating of the joint density function will be used for all time points, as the correlation structure of the state variables is nearly constant over time for the chosen model parameters. The number of expansion terms in the COS method is set to 32, and 50 quadrature points per dimension are used again in the Clenshaw-Curtis integration.

TWO XCS PRODUCTS

The same methodology used in the three-dimensional case is employed here to determine an appropriate rank to use in the CPD approximation. In this case, the PFE and EE of two XCS products combined are computed. One XCS product consists of USD and JPY paying legs, while the other has payments occurring in the USD and EUR currencies, resulting that all five risk factors from the model in 7.4 are being involved in this example. The characteristics of both XCS contracts are presented in Appendix E.4. The analysis of the relative error is again performed by considering only the first three years of the contract's lifetime. Figure 7.7 shows the PFE and EE profile of the two XCS contracts using the COS-CPD method, with a rank of $R = 20$ used in the CPD approximation. The profiles were generated using 20 equidistant time points from $t = 0$ up to $t = 3.0$. The plot demonstrates that the COS-CPD method is also consistent with the MC benchmark in this five-dimensional example, which was again simulated with 5 million paths per risk factors.

The relative error, as a percentage of the MC benchmark, is again utilized to quantify the error made in the COS-CPD method for different values of the rank in the CPD approximation. Table 7.4 displays the maximum and time-averaged error for four different number of rank, ranging from $R = 10$ up to $R = 25$. The maximum error in both the PFE and EE reduces for every increase in the number of rank. The time averaged-error shows similar results for the rank 10 and 15 approximation, but decreases when a rank 20 or 25 is used. Since an increase of rank does not have a significant impact on the computation time when evaluating the characteristic function in Equation 7.5, the rank-25 approximation in the CPD part will be used in the next section when the netting-set level exposure is discussed.

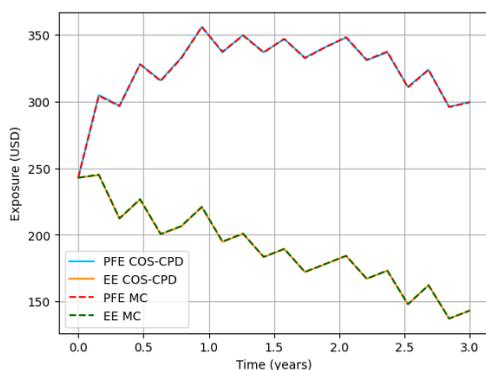


Figure 7.7: The PFE and EE profile of two XCS derivatives computed using both the COS-CPD method and a MC benchmark using $5 \cdot 10^6$ simulations. The profile is generated using 20 equidistant time points up to $t = 3.0$

Table 7.4: Relative error in the PFE and EE computation of two XCS products up to time $t = 3.0$ with 20 time steps using the COS-CPD method for different rank, using a MC benchmark of $5 \cdot 10^6$ paths.

rank	Max Error (%)		Time-averaged Error (%)	
	PFE	EE	PFE	EE
10	0.67	0.68	0.22	0.37
15	0.53	0.51	0.26	0.42
20	0.33	0.22	0.12	0.08
25	0.20	0.12	0.09	0.08

NETTING-SET LEVEL EXPOSURE

The netting-set level exposure for portfolio sizes up to 1000 derivatives will now be considered for the five-dimensional model. The portfolios are constructed in a similar way to the ones used in the three-dimensional section, but now also consisting of derivatives with paying legs in the euro currency. The portfolios used during this analysis can be found in Appendix E. Based on the findings from Table 7.4, the rank-25 approximation will be adopted in the CPD model.

First, the PFE and EE profiles were constructed for a portfolio consisting of 100 derivatives. For the entire lifespan of the portfolio, 50 equidistant time points were used, ranging from $t = 0$ to the maturity date of the final derivative in the portfolio, which was $T = 14.7$ for this specific portfolio. The results for both the COS-CPD and MC benchmark are illustrated in Figure 7.8, showing again a complete overlap of the COS-CPD estimates with the MC benchmark values. Regarding the COS support, a rule-of-thumb based on the one utilized in the three-dimensional case is used. Therefore, the COS support is set at $[-3000, 10000]$ at starting date $t = 0$. Additionally, the COS support is altered to $[-1000, 2000]$ when the PFE value drops below 1500. For the last year of the portfolio's lifetime, starting from $t = 13.7$, the COS support is set at $[-125, 125]$ to cope with the remaining coupon values that have yet to be exchanged.

The error quantification will be equivalent to the three-dimensional analysis. Therefore, only the first 3 years of the portfolio's lifetime are considered in the error analysis.

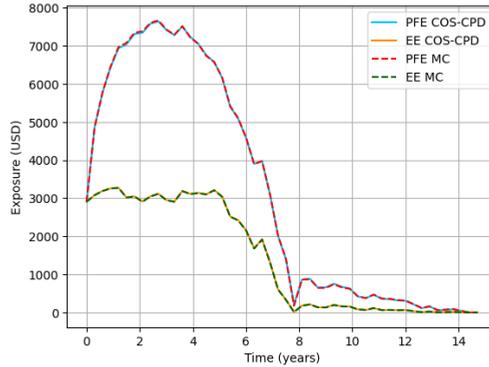
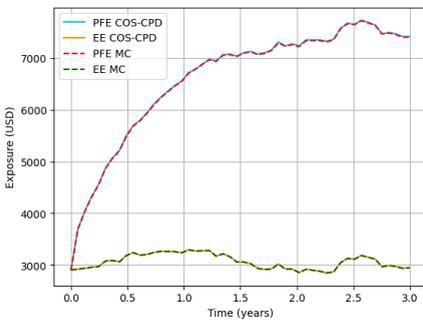
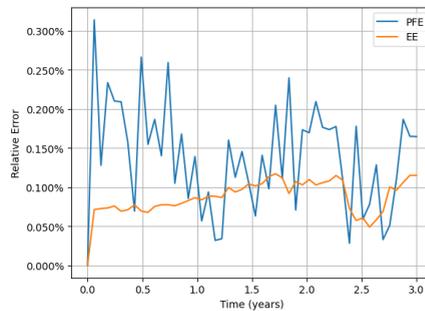


Figure 7.8: PFE and EE profile at a netting-set level for a portfolio consisting of 100 derivatives, computed at 50 equidistant time points. The MC benchmark was generated using $5 \cdot 10^6$ simulated paths.

The PFE and EE values within the first 3 years of the netting-set portfolio are computed for 50 equidistant time points. The resulting PFE and EE profiles are shown in Figure 7.9a, while the relative error is displayed in Figure 7.9b. The error in the PFE values fluctuates between 0.05% and 0.25% for the majority of the time points, while the EE error is bounded by 0.15% for all time points. The relative error in the EE shows less volatile behavior, which is to be expected, as the EE is computed by taking the average over the exposure distribution. Consequently, the under- and over estimations for this distribution can be cancelled out. As the maximum error over all time points is around 0.30%, these results show that the COS-CPD approximation in this five-dimensional model, even with high correlation structures up to 60%, still has a sufficient accuracy to match the PFE and EE profiles of the MC benchmark within acceptable range.



(a) PFE and EE profile up to time $t = 3.0$.



(b) Relative error in the COS-CPD method as a percentage of the MC benchmark.

Figure 7.9: The PFE and EE profile of the netting-set level exposure the portfolio consisting of 100 derivatives for the first 3 years, together with the relative error in the resulting PFE and EE values (as a percentage of the MC benchmark).

Table 7.5 displays the CPU time¹ and relative error for both the COS-CPD method and the MC method, utilizing different number of simulated paths. As expected, the ac-

curacy of the Monte Carlo estimation improves as the number of simulations increases. Additionally, the linear trend in the computation time can be observed again. The COS-CPD approach exhibits a comparable accuracy to the MC estimation with 10^5 paths. Similar to the three-dimensional case, the other two MC estimates show slightly better accuracy in the PFE and EE compared to the the COS-CPD approach. The error in the COS-CPD method is increased a minimal amount compared to the three-dimensional results, which might be caused by an increase in error of the CPD approximation. As already mentioned, the increase in dimension and a high correlation structure of 60% make it more difficult to approximate the Fourier-cosine series of the density function using CPD, as was observed during the numerical analysis done in Chapter 4. Despite these slightly less accurate estimates, which are still satisfactory, the computation time of the COS-CPD method is again significantly faster in comparison to all MC estimations. For this five-dimensional case, the COS-CPD method is still a factor of 10 faster when using $5 \cdot 10^5$ MC paths and almost 20 times faster than the MC approach utilizing $1 \cdot 10^6$ paths.

Table 7.5: The accuracy and CPU time (in seconds) required to calculate the PFE and EE of netting-set-level exposure of the portfolio with 100 derivatives evaluated at 50 time points up to time $t = 3.0$. The time-averaged error was calculated by averaging the relative differences over the 50 time points.

	CPU Time ¹ (s)	Time-averaged Error (%)	
		PFE	EE
MC ($1 \cdot 10^5$)	51.4	0.117	0.024
MC ($5 \cdot 10^5$)	312.5	0.054	0.006
MC ($1 \cdot 10^6$)	623.7	0.039	0.003
COS-CPD	36.9	0.137	0.087

As previously demonstrated through the three-dimensional analysis, the performance of the COS-CPD and MC methods are again compared for various portfolio sizes utilizing 10, 20, or 50 time steps. The results of this comparison are presented in Table 7.6. It can be observed that the computation time grows linearly in the number of time steps for both methods. This outcome is to be expected, as the computation time for the PFE and EE value per time step should be equal for both the COS-CPD and MC method. Similarly to the three-dimensional case, the increase in computation time for an increasing number of derivatives is significantly smaller for the COS-CPD method. This is due to the computational complexity of evaluating the portfolio, as explained in the three-dimensional case. Adding an extra derivative requires to evaluate this derivative on all simulated paths in case of the MC method, which is $5 \cdot 10^5$ in this example. Consequently, the number of evaluations increases significantly in the MC method for an increase in portfolio size, resulting in a substantial impact on the computation time. The computation of the COS method primarily consists of evaluating of the characteristic functions on the quadrature points. The total number of quadrature points for the two-dimensional integrals is $50^2 = 2500$, which means that evaluating extra derivatives on these quadrature points does not have a significant impact on the computation time. Therefore, The COS-CPD method is more efficient in handling portfolios that contain a large number of derivatives.

Table 7.6: Comparison of the CPU time (in seconds) needed to compute the PFE and EE profile for a netting-set portfolio with 10, 20 and 50 time steps, respectively. For the MC method, $5 \cdot 10^5$ paths are considered

Number of Derivatives	COS-CPD	MC	COS-CPD	MC	COS-CPD	MC
25	7.5	17.2	16.0	38.1	41.2	101.8
50	7.7	32.9	16.5	71.2	42.6	187.6
100	8.2	54.8	17.4	118.4	44.4	312.5
1000	16.0	504.5	33.9	1110.3	87.4	2945.0

8

APPLICATION OF COS-CPD IN MULTI-DIMENSIONAL OPTION PRICING

The COS-CPD method is a technique for evaluating multivariate expectations and can be applied to any field that deals with such expectations. To illustrate the generality of the COS-CPD method, it will also be applied in the context of option pricing, which is another relevant topic studied in the field of finance. Similar to the exposure calculations in the previous chapter, evaluating multivariate expectations plays a central role when valuating options.

This chapter focuses on the application of the COS-CPD method to a six-dimensional basket option, where the option's payoff is dependent on six different stock prices. The COS-CPD method can effectively reduce the dimensionality of the problem, as demonstrated in the context of exposure calculation in the previous chapter. Consequently, the COS-CPD method can be an efficient method for evaluating high-dimensional options.

First, the methodology for applying the COS-CPD method to the six-dimensional basket option will be discussed, including the modeling process. After that, some numerical results are presented, comparing the accuracy and computation time of the COS-CPD method with a recently introduced sparse grid method in [35] for evaluating multi-asset options.

8.1. SIX-DIMENSIONAL BASKET OPTION

8.1.1. MODEL

In this section, we will examine basket options that are equivalent to those discussed in [35]. This will allow us to make a comparison with the recently proposed method that employs sparse grid techniques. The basket options under consideration are six-

dimensional and have a payoff given by:

$$H_{basket}^p(\mathbf{S}(T); K) = \max\left(K - \frac{1}{6} \sum_{i=1}^6 S_i(T), 0\right),$$

where $\mathbf{S} = (S_1, \dots, S_6)$ represents the prices of the six stocks, K is the strike price, T is the maturity time of the option, and the superscript p denotes that we are dealing with a put option. It is important to note that as we are only considering European options, meaning that they can only be exercised at maturity. Similar to the exchange rate processes, the stock prices will be modeled using a GBM,

$$dS_i = \mu_i S_i dt + \sigma_i S_i dW_i^{\mathbb{P}},$$

where μ_i and σ_i denote the drift factor and the volatility of stock S_i , respectively. In the context of option pricing, the dynamics of the underlying stochastic processes have to be in accordance with the no-arbitrage principle. Therefore, a change of measure is required to the risk-neutral measure, often denoted with \mathbb{Q} . Under the risk-neutral measure, the dynamics of the stock prices are given by

$$dS_i = r S_i dt + \sigma_i S_i dW_i^{\mathbb{Q}},$$

where r represents the (constant) interest rate. The stock prices are modelled using GBM, which results in lognormal distributions. To make the calculations more convenient, we will consider the dynamics of the logarithm of the stock prices, represented by $\mathbf{X} = (X_1, \dots, X_6)$. As a result, \mathbf{X} follows a multivariate normal distribution, and the payoff can be expressed in terms of the logarithmic stock prices as

$$H_{basket}^p(\mathbf{X}(T); K) = \max\left(K - \frac{1}{6} \sum_{i=1}^6 e^{X_i(T)}, 0\right).$$

8

The value of this six-dimensional basket option can now be obtained by evaluating the discounted expected payoff. Assuming a constant interest rate r , the option value at time t is given by

$$V_t = e^{-r(T-t)} \mathbb{E}^{\mathbb{Q}} [H_{basket}^p(\mathbf{X}(T); K)] = e^{-r(T-t)} \mathbb{E}^{\mathbb{Q}} \left[\max\left(K - \frac{1}{6} \sum_{i=1}^6 e^{X_i(T)}, 0\right) \right],$$

where the superscript \mathbb{Q} indicates that the expectation is taken under the risk-neutral measure. Computing the multivariate expectation directly requires to solve a six-dimensional integral. As noted in the calculation of exposures in the previous chapter, direct integration methods as Clenshaw-Curtis cannot be utilized in these high-dimensional scenarios, due to the presence of the "curse of dimensionality" in these quadrature techniques. As a result, alternative methods such as Monte Carlo and sparse grids techniques have been developed to mitigate this exponential growth in high-dimensional situations.

The COS-CPD method provides an alternative way to circumvent the high-dimensionality of the problem. In fact, it can be simplified to the evaluation of only one-dimensional integrals, as will be shown in the upcoming derivations. Instead of directly computing

the expectation, the density function of the payoff, denoted with f_H , is approximated with its Fourier-cosine series using the COS formula,

$$f_H(x) \approx \sum_{n=0}^{N-1} F_n \cos\left(n\pi \frac{x-l}{u-l}\right),$$

where the coefficients F_k are of the following form,

$$F_n \equiv \frac{2}{u-l} \operatorname{Re} \left\{ \phi_H\left(\frac{n\pi}{u-l}\right) \cdot \exp\left(-i \frac{n\pi}{u-l}\right) \right\}. \quad (8.1)$$

Therefore, the characteristic function ϕ_H has to be evaluated, which is of the form

$$\begin{aligned} \phi_H(\omega) &= \int_{\mathbb{R}^6} e^{i\omega H_{basket}^P(\mathbf{X}(T);K)} f(\mathbf{X}(T)) d\mathbf{X}(T) \\ &= \int_{\mathbb{R}^6} e^{i\omega \cdot \max(K - \frac{1}{6} \sum_{i=1}^6 e^{X_i(T)}, 0)} f(\mathbf{X}(T)) d\mathbf{X}(T). \end{aligned} \quad (8.2)$$

Expression 8.2 is equivalent with the characteristic function of the portfolio exposure in Equation 5.3. Therefore, the derivations from now on are analogous to the methodology described in Chapter 5 on exposure computation. The same technique can be used by omitting the maximum operator during the computation of the characteristic function. In this way, the contract is considered first as it was a forward contract, with payoff

$$\tilde{H}_{forward} = K - \frac{1}{6} \sum_{i=1}^6 e^{X_i(T)}.$$

Once the PDF of the payoff regarding the forward contract is obtained, the maximum operator can be applied afterwards to find the PDF of the option payoff. The characteristic function in case of the forward payoff is given by

$$\phi_{\tilde{H}}(\omega) = \int_{\mathbb{R}^6} e^{i\omega \cdot (K - \frac{1}{6} \sum_{i=1}^6 e^{X_i(T)})} f(\mathbf{X}(T)) d\mathbf{X}(T). \quad (8.3)$$

The first step in the COS-CPD is to decompose the six-dimensional joint density function into its Fourier-cosine series using CPD. Before that, the density function is made independent of the maturity time T , so that a single CPD approximation can be utilized for all possible maturity times. This is achieved by expressing the multivariate random variable \mathbf{X} in terms of the standard normal distribution as follows:

$$X_i(T) \sim \mu_i^X(T) + \sigma_i^X(T) Z_i, \quad i = 1, \dots, 6,$$

with $Z_i = \frac{1}{\sqrt{T}} W_i(T) \sim N(0, 1)$. The mean μ_i^X and standard deviation σ_i^X are given by

$$\begin{aligned} \mu_i^X &= \log(X_i(0)) + \left(r - \frac{\sigma_i^2}{2}\right) T, \\ \sigma_i^X &= \sqrt{\sigma_i^2 T}. \end{aligned}$$

The time-independence of the correlations structure of the new state variables $\mathbf{Z} := (Z_1, \dots, Z_6)$ can be proven as follows,

$$\begin{aligned} \text{Cor}(Z_i, Z_j) &= \text{Cov}(Z_i, Z_j) \\ &= \text{Cov}\left(\frac{1}{\sqrt{T}}W_i(T), \frac{1}{\sqrt{T}}W_j(T)\right) \\ &= \frac{\text{Cov}(W_i(T), W_j(T))}{T} \\ &= \frac{\rho_{ij}T}{T} \\ &= \rho_{ij}, \end{aligned}$$

where ρ_{ij} denotes the correlation between stock S_i and S_j . Therefore, only one CPD approximation of the joint density can be used for all maturity times T .

With this transformation, the characteristic function in 8.3 can be written as

$$\phi_{\tilde{H}}(\omega) = \int_{\mathbb{R}^6} e^{i\omega \cdot \left(K - \frac{1}{6} \sum_{i=1}^6 e^{\mu_i^X + \sigma_i^X Z_i}\right)} f(\mathbf{Z}) d\mathbf{Z},$$

with $\mathbf{Z} := (Z_1, \dots, Z_6)$ the standard normal state variables. Once the density function $f(\mathbf{Z})$ is decomposed into its Fourier-cosine series using CPD, it can be expressed as

$$f(\mathbf{Z}) = \sum_{r=1}^R \prod_{n=1}^6 \mathbf{v}_n^T \mathbf{A}_n[:, r],$$

where $\{\mathbf{A}_n\}_{n=1}^6$ the factor matrices and $\{\mathbf{v}_n\}_{n=1}^6$ the cosine basis functions of the state variables \mathbf{Z} . Inserting this back into the characteristic function we obtain

8

$$\begin{aligned} \phi_{\tilde{H}}(\omega) &= \int_{\mathbb{R}^6} e^{i\omega \cdot \left(K - \frac{1}{6} \sum_{i=1}^6 e^{\mu_i^X + \sigma_i^X Z_i}\right)} \sum_{r=1}^R \prod_{n=1}^6 \mathbf{v}_n^T \mathbf{A}_n[:, r] d\mathbf{Z} \\ &= \sum_{r=1}^R \int_{\mathbb{R}^6} e^{i\omega \cdot \left(K - \frac{1}{6} \sum_{i=1}^6 e^{\mu_i^X + \sigma_i^X Z_i}\right)} \prod_{n=1}^6 \mathbf{v}_n^T \mathbf{A}_n[:, r] d\mathbf{Z} \\ &= \sum_{r=1}^R e^{i\omega \cdot K} \int_{\mathbb{R}^6} \prod_{i=1}^6 e^{i\omega \cdot \left(-\frac{1}{6} e^{\mu_i^X + \sigma_i^X Z_i}\right)} \prod_{n=1}^6 \mathbf{v}_n^T \mathbf{A}_n[:, r] d\mathbf{Z} \\ &= \sum_{r=1}^R e^{i\omega \cdot K} \prod_{n=1}^6 \int_{\mathbb{R}} e^{i\omega \cdot \left(-\frac{1}{6} e^{\mu_n^X + \sigma_n^X Z_n}\right)} \mathbf{v}_n^T \mathbf{A}_n[:, r] dZ_n. \end{aligned} \quad (8.4)$$

Therefore, the six-dimensional problem is simplified to evaluating only one-dimensional integrals, which is a significant dimension reduction. The Clenshaw-Curtis quadrature method can once again be applied to the one-dimensional integrals to obtain an approximation for the characteristic function. Expression 8.4 can then be inserted in the Fourier coefficients F_k in 8.1 to compute the PDF of the forward payoff. To obtain the expected payoff of the basket option, the exact same methodology as described Section 5.3 for the EE computation can be applied. Therefore, the expected payoff of the basket

option, and thus its value, can be found by applying formula 5.15 with the obtained coefficients F_k . For example, in case $l \leq 0$, the option value at time t can be computed with the following formula (note the discount factor),

$$V_t = e^{-r(T-t)} \left\{ \frac{F_0}{4} u^2 + \sum_{n=1}^{N-1} F_n \left(\frac{u-l}{n\pi} \right)^2 \left((-1)^n - \cos \left(n\pi \frac{l}{u-l} \right) \right) \right\}. \quad (8.5)$$

An additional advantage when using the COS-CPD method, is that the strike price K can be extracted from the integral computations, as can be seen from Equation 8.4. Consequently, the option value can be computed for multiple strike prices simultaneously with no significant additional computational cost. This is in contrast to methods that directly evaluate the multivariate expectations, such as the sparse grid techniques, which can only evaluate for one strike price at a time. Furthermore, the joint density function $f(\mathbf{Z})$ only captures the correlation structures between the different stocks, so that the CPD approximation can be used as long as the same correlation structure is present. As previously stated, the CPD part can be carried out "off-line", and the results can be re-used until the next re-calibration of the model parameters.

8.1.2. NUMERICAL RESULTS

This section presents the numerical outcomes for the accuracy and computation of the COS-CPD method in calculating a six-dimensional basket put option. As the results will be compared with the sparse grid method proposed in [35], we will employ the same setup here to ensure a fair comparison of the accuracy results. Accordingly, we will set the constant interest rate to $r = 0$, the option's maturity time to $T = 1$, the strike price to $K = 60$, and the initial stock price for all stocks to $S_i(0) = 100$, $i = 1, \dots, 6$. The numerical results will examine two sets of volatility parameters, which are given by $\boldsymbol{\sigma}_1 = (0.4, 0.4, 0.4, 0.4, 0.4, 0.4)$ and $\boldsymbol{\sigma}_2 = (0.2, 0.3, 0.4, 0.5, 0.6, 0.7)$. Furthermore, we assume that the stocks are independent.

Regarding the CPD component, the parameters used are similar to those utilized in the exposure distribution calculations in Chapter 7. Hence, the number of expansion terms and training points per dimension is set to $K_{CPD} = m = 12$, with a tolerance level of 10^{-8} for the expansion interval $[a, b]^6$. However, since the stocks are modelled independently, a rank $R = 1$ approximation is already sufficient to synthesize the Fourier coefficient tensor.

The Clenshaw-Curtis algorithm with 50 quadrature points is used to assess the one-dimensional integrals. Additionally, the number of cosine expansion terms employed in the COS approximation of the payoff density is varied from $N = 32$ up to $N = 128$, so that the impact for different expansion sizes on the accuracy of the resulting option value can be assessed. For all results generated, the COS support is set to the interval $[l, u] = [-400, 200]$.

With the COS-CPD method, the complete payoff distribution is generated first, from which the final option value can be derived. For the parameter values described above, the payoff density is illustrated in Figure 8.1. Only the strike price is changed to $K = 100$ to make the truncation of the forward payoff density more visible.

The COS-CPD method generates the complete payoff distribution of the option, from which the final option value can be obtained. Figure 8.1 illustrates the recovered payoff

density of the basket forward contract, which is truncated to zero for negative outcomes to find the payoff density of the basket put option. The results are generated with the parameter values described above. Only the strike price is changed to $K = 100$ for this specific numerical test to enhance the visibility of the truncation of the forward payoff density.

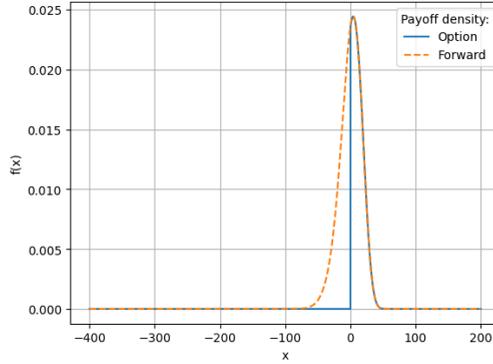


Figure 8.1: Recovered payoff density of the six-dimensional basket put option with the COS-CPD method, utilizing $N = 64$ expansion terms in the COS approximation. Regarding the option, the parameters used were $r = 0$, $T = 1$, $K = 100$, $\sigma_1 = (0.4, 0.4, 0.4, 0.4, 0.4, 0.4)$ and $S_i(0) = 100$, $i = 1, \dots, 6$.

After computing the payoff density, Equation 8.5 is used to determine the option value. The option value will be computed using 32, 64 and 128 expansion terms in the COS approximation. The results are then compared for accuracy and computation time with those obtained using the sparse grid method described in [35]. The results for two sets of volatility parameters, σ_1 and σ_2 , are presented in Table 8.1 and Table 8.2, respectively. The relative error for both methods is determined by taking a Monte Carlo estimation of 10^9 paths as the reference value. As expected, the accuracy of the COS-CPD method improves with an increasing number of expansion terms. Clearly, 32 expansion terms are insufficient to provide an accurate approximation of the option value. However, using 64 expansion terms already leads to a factor 10 more accurate result for the σ_1 parameter set compared to the sparse grid method. This accuracy is slightly improved again when 128 expansion terms are used. It is important to note that the comparison of computation times between the two methods must be done with caution, as they were generated using different computers¹. Nonetheless, these findings suggest that the COS-CPD method provides a significantly faster computation time compared to the sparse grid evaluation, even when using a less powerful computer.

As previously mentioned, the COS-CPD method has the additional advantage that the option value can be generated for multiple strike prices K simultaneously, without having significant extra computational costs. To demonstrate this advantage, a numerical test is performed in which the COS-CPD method is employed to value the six-

¹The COS-CPD results are generated with a Intel(R) Core(TM) i7-10700F CPU @ 2.90GHz (8 CPU cores), with the code written in Python 3.8.10. For the sparse grid method in [35], a cluster machine is used with the following characteristics: clock speed 2.1 GHz, #CPU cores: 72, and memory per node 256 GB. Furthermore, the computer code is written in MATLAB (version R2021b).

Table 8.1: Comparison in computation time (in seconds) and accuracy of the COS-CPD method with the sparse grid method from [35] in pricing a six-dimensional basket put option with volatility parameters $\sigma_1 = (0.4, 0.4, 0.4, 0.4, 0.4, 0.4)$. The relative error for both methods is derived by using a Monte Carlo benchmark utilizing 10^9 paths.

Method	Option Value	CPU Time ¹	Relative Error
COS-CPD ($N = 32$)	0.013777	0.008	2.36
COS-CPD ($N = 64$)	0.004120	0.017	0.0054
COS-CPD ($N = 128$)	0.004116	0.033	0.0043
Sparse Grid	-	2.0	0.029

Table 8.2: Comparison in computation time (in seconds) and accuracy of the COS-CPD method with the sparse grid method from [35] in pricing a six-dimensional basket put option with volatility parameters $\sigma_2 = (0.2, 0.3, 0.4, 0.5, 0.6, 0.7)$. The relative error for both methods is derived by using a Monte Carlo benchmark utilizing 10^9 paths.

Method	Option Value	CPU Time	Relative Error
COS-CPD ($N = 32$)	0.007646	0.047	0.3981
COS-CPD ($N = 64$)	0.012729	0.091	0.0021
COS-CPD ($N = 128$)	0.012721	0.181	0.0015
Sparse Grid	-	2.1	0.0033

dimensional basket option for the strike price vector $\mathbf{K} = (75, 76, \dots, 100)$. The resulting option value curve is presented in Figure 8.2, together with a Monte Carlo estimation using 10^8 simulated paths. Table 8.3 shows the computation time required to generate this option value curve, as well as the maximum relative error made in the computed option prices. Similar to the previous findings, utilizing 64 and 128 expansion terms provide a significantly more accurate result compared to the estimations with 32 expansion terms. However, the most crucial point to note is that the computation time required to generate the option values for the strike price vector \mathbf{K} is almost identical to the computation times presented in Table 8.2. These findings show that the increase in computational costs is indeed almost negligible when evaluating the option value for a range of strike prices simultaneously. This advantage is significant compared to the sparse grid-related method, which can only compute the option value for one strike price at a time.

Table 8.3: Computation time (in seconds) and accuracy of the COS-CPD method to price the six-dimensional basket put option for the strike price vector $\mathbf{K} = (75, 76, \dots, 100)$. The volatility parameters are $\sigma_2 = (0.2, 0.3, 0.4, 0.5, 0.6, 0.7)$ and the relative error is derived by using a Monte Carlo benchmark utilizing 10^8 paths.

	CPU Time	Max Relative Error
COS-CPD ($N = 32$)	0.046	0.01100
COS-CPD ($N = 64$)	0.091	0.00059
COS-CPD ($N = 128$)	0.194	0.00057

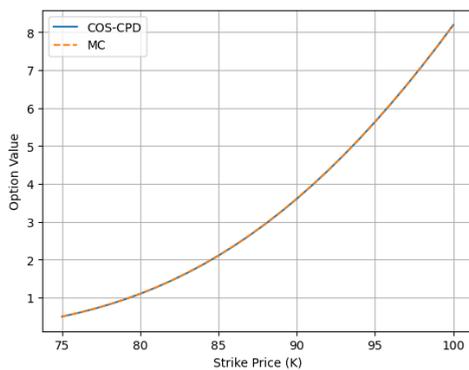


Figure 8.2: Value of the six-dimensional basket put option for the strike vector $\mathbf{K} = (75, 76, \dots, 100)$. The volatility parameters are $\sigma_2 = (0.2, 0.3, 0.4, 0.5, 0.6, 0.7)$ and $N = 64$ COS expansion terms are utilized.

9

CONCLUSIONS AND FUTURE RESEARCH

In this thesis, we developed a generic and efficient solver for multivariate expectation problems, with two detailed examples of its application in the field of quantitative finance. The solver, which is referred to as the COS-CPD method, comprises two main components: the COS method and a tensor decomposition technique called CPD.

The primary focus of this thesis was the application of the COS-CPD method in quantifying Counterparty Credit Risk, more precisely, the netting-set level PFE and EE profiles of liquid derivative portfolios. Although a previous Master's thesis at FF Quant [36] had replaced the Monte Carlo method with the COS method to calculate exposure profiles, this method had limitations in the number of dimensions, i.e. it has difficulties to handle portfolios with more than three risk factors.

We showed that the COS-CPD method can tackle this limitation, as it leads to a reduction of the dimension of integration for the characteristic function. This reduction is achieved through two steps. First, the joint density function of the risk factors is replaced by a dimension-reduced Fourier-cosine series expansion. The CPD technique is utilized to decompose the Fourier coefficient tensor, which lowers the computational complexity from exponential to linear with respect to the number of dimensions. Note that this step is generic and is applicable to any multivariate expectation problem. Secondly, the portfolio structure was exploited such that the portfolio can be divided into segments that share the same risk factors. Together with the first step, this reduces the evaluation of the characteristic function, as defined by a multidimensional integral, to the calculation of a few one- and two-dimensional integrals. These integrals can then be solved by the Clenshaw-Curtis quadrature rule. As a result, the COS-CPD method is applicable to portfolios with more than three risk factors.

The CPD approximation of the joint density function is derived using the Hidden Tensor Factorization approach, which circumvents the need to compute the entire Fourier coefficient tensor. With the application of CPD, the Fourier-cosine series expansion of the joint density could be represented as a product of univariate Fourier-cosine series,

which gives great computational benefit, as effectively the dimensionality of the Fourier-cosine series expansion is greatly reduced. Furthermore, the CPD approximation of the joint density function of the risk factors can be done "off-line" in advance. As long as the model parameters of the risk factors do not change, the trained CPD coefficients can be re-used. In other words, the CPD only needs to be updated in the next re-calibration of the risk factor models. Extensive numerical tests and analyses were performed to gain insights into the impact of various parameters on and their sensitivity to the accuracy of the COS-CPD method. Based on these results, an optimal parameter setting could be determined for the COS-CPD method either in the context of CCR quantification or option pricing.

A careful theoretical error analysis was also conducted for the COS-CPD method, wherein the COS and CPD components were individually examined. The analysis showed that the COS part exhibits exponential convergence in case of smooth density functions. For the CPD part, the error resulting from the low-rank approximation of the tensor was not theoretically quantified, as determining the correct rank of a tensor is an NP-hard problem. However, numerical experiments showed that the CPD converges for increasing rank to the optimal accuracy level for different correlation levels, up to the point that the series truncation error becomes the dominant error source.

The accuracy and computational time of the COS-CPD method in calculating the PFE and EE profiles were compared to those of the Monte Carlo method. Portfolios with three and five risk factors were considered, each consisting up to 1000 derivatives. The results showed that for the same level of accuracy, the COS-CPD method greatly outperforms the MC method in computation time. More importantly, the time difference increases with larger portfolio sizes due to the fact that the computational complexity of the COS-CPD method is not driven by the number of trades, while MC is. Therefore, when considering portfolios up to five risk factors, the COS-CPD method can be a much more efficient alternative for the MC method, particularly for large portfolios.

To demonstrate the generic nature of the COS-CPD method, another example was selected from the financial industry: multi-dimensional option pricing. More precisely, the COS-CPD method was used to price a six-dimensional basket option. The numerical results were compared with those of a recently developed sparse grid method to price multi-asset options. This comparison showed that the COS-CPD method outperformed the sparse grid method both in accuracy and computational time. Furthermore, the COS-CPD method allows the computation of the option value for multiple strike prices simultaneously, with no significant additional computational cost.

In the example of CCR quantification, the COS-CPD method was only applied on netting-set level. However, counterparty-level CCR quantification is also common in practice. The computation of counterparty-level exposure brings an extra challenge, as the maximum operator in the evaluation of the characteristic functions cannot be eliminated. Consequently, it is not possible to directly divide the portfolios into groups with common risk factors. A future research focus could be on how to utilize the COS-CPD method in this case.

The portfolios considered in this thesis consist of at most five risk factors. In practice, the number of risk factors can easily be higher. Therefore, future research also includes the extension of the application of the COS-CPD method for portfolios with more than

five risk factors, by first testing the limitation of the CPD algorithm in terms of the maximum number of dimensions it can handle. One approach to further reduce the computational cost in the CPD calibration step, is to identify and exclude cosine basis functions with negligible coefficients. The idea then is to filter in only the principal basis functions for each risk factor by, for example, examining the univariate Fourier-cosine series expansions of the marginal distributions of the risk factors. This could greatly reduce the computational complexity of the CPD computation, while having a minimal impact on the accuracy.

Finally, the COS support $[l, u]$ used in the numerical tests was mainly based on rule-of-thumbs. As a too wide or too small COS support can lead to significant errors in the recovered distribution, it is important to find out how the COS support should be defined. This is particularly relevant in the context of exposure distributions, as the distribution changes over time, and therefore, the COS support has to be adapted accordingly. Since the portfolio contains a lot of information already on the potential exposure, one possible solution could be to use this portfolio information to find a suitable COS support.

BIBLIOGRAPHY

- [1] Basel Committee on Banking Supervision, “Counterparty credit risk definitions and terminology,” *Bank For International Settlements*, 2019. [Online]. Available: https://www.bis.org/basel_framework/chapter/CRE/50.htm.
- [2] S. H. Zhu and M. Pykhtin, “A Guide to Modelling Counterparty Credit Risk,” *Global Association of Risk Professionals*, 2007. [Online]. Available: <https://ssrn.com/abstract=1032522>.
- [3] C. W. Clenshaw and A. R. Curtis, “A method for numerical integration on an automatic computer,” *Numerische Mathematik*, vol. 2, no. 1, pp. 197–205, Dec. 1960, ISSN: 0029-599X. DOI: [10.1007/BF01386223](https://doi.org/10.1007/BF01386223). [Online]. Available: <http://link.springer.com/10.1007/BF01386223>.
- [4] S. Smolyak, “Quadrature and Interpolation Formulas for Tensor Products of Certain Classes of Functions,” *Soviet Mathematics Doklady*, vol. 4, pp. 240–243, 1963.
- [5] F. A. Longstaff and E. S. Schwartz, “Valuing American Options by Simulation: A Simple Least-Squares Approach,” *Review of Financial Studies*, vol. 14, no. 1, pp. 113–147, Jan. 2001, ISSN: 0893-9454. DOI: [10.1093/rfs/14.1.113](https://doi.org/10.1093/rfs/14.1.113). [Online]. Available: <https://academic.oup.com/rfs/article-lookup/doi/10.1093/rfs/14.1.113>.
- [6] Y. Krepkiy, A. Lakhany, and A. Zhang, “Efficient Least Squares Monte-Carlo Technique for PFE/EE Calculations,” 2021. [Online]. Available: <http://arxiv.org/abs/2105.07061>.
- [7] S. Ghamami and B. Zhang, “Efficient Monte Carlo Counterparty Credit Risk Pricing and Measurement,” *SSRN Electronic Journal*, 2014, ISSN: 1556-5068. DOI: [10.2139/ssrn.2544692](https://doi.org/10.2139/ssrn.2544692). [Online]. Available: <http://www.ssrn.com/abstract=2544692>.
- [8] Y. T. Trinh and B. Hanzon, “Option Pricing and CVA Calculations using the Monte Carlo-Tree (MC-Tree) Method,” Feb. 2022. [Online]. Available: <http://arxiv.org/abs/2202.00785>.
- [9] S. Jain and C. W. Oosterlee, “The Stochastic Grid Bundling Method: Efficient pricing of Bermudan options and their Greeks,” *Applied Mathematics and Computation*, vol. 269, pp. 412–431, Oct. 2015, ISSN: 00963003. DOI: [10.1016/j.amc.2015.07.085](https://doi.org/10.1016/j.amc.2015.07.085). [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0096300315010073>.
- [10] C. S. de Graaf, Q. Feng, D. Kandhai, and C. W. Oosterlee, “Efficient Computation of Exposure Profiles for Counterparty Credit Risk,” *SSRN Electronic Journal*, 2014, ISSN: 1556-5068. DOI: [10.2139/ssrn.2395183](https://doi.org/10.2139/ssrn.2395183). [Online]. Available: <http://www.ssrn.com/abstract=2395183>.

- [11] P. Karlsson, S. Jain, and C. W. Oosterlee, “Counterparty Credit Exposures for Interest Rate Derivatives using the Stochastic Grid Bundling Method,” *Applied Mathematical Finance*, vol. 23, no. 3, pp. 175–196, May 2016, ISSN: 1350-486X. DOI: [10.1080/1350486X.2016.1226144](https://doi.org/10.1080/1350486X.2016.1226144). [Online]. Available: <https://www.tandfonline.com/doi/full/10.1080/1350486X.2016.1226144>.
- [12] Q. Feng, S. Jain, P. Karlsson, D. Kandhai, and C. Oosterlee, “Efficient computation of exposure profiles on real-world and risk-neutral scenarios for Bermudan swaptions,” *The Journal of Computational Finance*, vol. 20, no. 1, pp. 139–172, 2016, ISSN: 14601559. DOI: [10.21314/JCF.2017.337](https://doi.org/10.21314/JCF.2017.337). [Online]. Available: <http://www.risk.net/journal-of-computational-finance/technical-paper/2464610/efficient-computation-of-exposure-profiles-on-real-world-and-risk-neutral-scenarios-for-bermudan-swaptions>.
- [13] M. Zeron Medina Laris and I. Ruiz, “Chebyshev Methods for Ultra-Efficient Risk Calculations,” *SSRN Electronic Journal*, 2018, ISSN: 1556-5068. DOI: [10.2139/ssrn.3165563](https://doi.org/10.2139/ssrn.3165563). [Online]. Available: <https://www.ssrn.com/abstract=3165563>.
- [14] K. Glau, R. Pachon, and C. Pötz, “Fast Calculation of Credit Exposures for Barrier and Bermudan options using Chebyshev interpolation,” May 2019. [Online]. Available: <http://arxiv.org/abs/1905.00238>.
- [15] K. Glau, M. Mahlstedt, and C. Pötz, “A new approach for American option pricing: The Dynamic Chebyshev method,” Jun. 2018. [Online]. Available: <http://arxiv.org/abs/1806.05579>.
- [16] C. S. de Graaf, D. Kandhai, and P. M. Sloot, “Efficient Estimation of Sensitivities for Counterparty Credit Risk with the Finite Difference Monte-Carlo Method,” *SSRN Electronic Journal*, 2014, ISSN: 1556-5068. DOI: [10.2139/ssrn.2521431](https://doi.org/10.2139/ssrn.2521431). [Online]. Available: <http://www.ssrn.com/abstract=2521431>.
- [17] M. Broadie and P. Glasserman, “Estimating Security Price Derivatives Using Simulation,” *Management Science*, vol. 42, no. 2, pp. 269–285, Feb. 1996, ISSN: 0025-1909. DOI: [10.1287/mnsc.42.2.269](https://doi.org/10.1287/mnsc.42.2.269). [Online]. Available: <http://pubsonline.informs.org/doi/10.1287/mnsc.42.2.269>.
- [18] T. Gerstner, B. Harrach, and D. Roth, “Monte Carlo pathwise sensitivities for barrier options,” Apr. 2018. DOI: [10.21314/JCF.2020.385](https://doi.org/10.21314/JCF.2020.385). [Online]. Available: <http://arxiv.org/abs/1804.03975> <http://dx.doi.org/10.21314/JCF.2020.385>.
- [19] J. Prenio and J. Yong, “Humans keeping AI in check – emerging regulatory expectations in the financial sector,” *Bank For International Settlements*, 2021. [Online]. Available: <https://www.bis.org/fsi/publ/insights35.pdf?fbclid=IwAR2wd2L4FUaQDbxURePvoahJua8rSXBSHJFDC9l19ptZKsKSCtL20506sZw>.
- [20] S. Crépey and M. Dixon, “Gaussian Process Regression for Derivative Portfolio Modeling and Application to CVA Computations,” Jan. 2019. [Online]. Available: <http://arxiv.org/abs/1901.11081>.
- [21] L. Goudenege, A. Molent, and A. Zanette, “Computing XVA for American basket derivatives by Machine Learning techniques,” Sep. 2022. [Online]. Available: <http://arxiv.org/abs/2209.06485>.

- [22] M. Ludkovski, “Kriging Metamodels and Experimental Design for Bermudan Option Pricing,” Sep. 2015. [Online]. Available: <http://arxiv.org/abs/1509.02179>.
- [23] B. Lapeyre and J. Lelong, “Neural network regression for Bermudan option pricing,” Jul. 2019. [Online]. Available: <http://arxiv.org/abs/1907.06474>.
- [24] S. Zhu, J. Chan, and D. Bright, “Applying Machine Learning for Troubleshooting Credit Exposure and xVA Profiles,” *SSRN Electronic Journal*, 2019, ISSN: 1556-5068. DOI: [10.2139/ssrn.3404863](https://doi.org/10.2139/ssrn.3404863). [Online]. Available: <https://www.ssrn.com/abstract=3404863>.
- [25] A. Gnoatto, C. Reisinger, and A. Picarelli, “Deep xVA Solver – A Neural Network Based Counterparty Credit Risk Management Framework,” *SSRN Electronic Journal*, 2020, ISSN: 1556-5068. DOI: [10.2139/ssrn.3594076](https://doi.org/10.2139/ssrn.3594076). [Online]. Available: <https://www.ssrn.com/abstract=3594076>.
- [26] J.-H. She and D. Greco, “Neural Network for CVA: Learning Future Values,” Nov. 2018. [Online]. Available: <http://arxiv.org/abs/1811.08726>.
- [27] S. Welack, “Artificial Neural Network Approach to Counterparty Credit Risk and XVA,” *SSRN Electronic Journal*, 2019, ISSN: 1556-5068. DOI: [10.2139/ssrn.3312944](https://doi.org/10.2139/ssrn.3312944). [Online]. Available: <https://www.ssrn.com/abstract=3312944>.
- [28] I. Frod e, V. Sambergs, and S. Zhu, “Neural Networks for Credit Risk and xVA in a Front Office Pricing Environment,” *SSRN Electronic Journal*, 2022, ISSN: 1556-5068. DOI: [10.2139/ssrn.4136123](https://doi.org/10.2139/ssrn.4136123). [Online]. Available: <https://www.ssrn.com/abstract=4136123>.
- [29] K. Andersson and C. W. Oosterlee, “Deep learning for CVA computations of large portfolios of financial derivatives,” *Applied Mathematics and Computation*, vol. 409, p. 126399, Nov. 2021, ISSN: 00963003. DOI: [10.1016/j.amc.2021.126399](https://doi.org/10.1016/j.amc.2021.126399). [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0096300321004884>.
- [30] F. Fang and C. W. Oosterlee, “A Novel Pricing Method for European Options Based on Fourier-Cosine Series Expansions,” *SIAM Journal on Scientific Computing*, vol. 31, no. 2, pp. 826–848, Jan. 2009, ISSN: 1064-8275. DOI: [10.1137/080718061](https://doi.org/10.1137/080718061). [Online]. Available: <http://epubs.siam.org/doi/10.1137/080718061>.
- [31] D. Duffie, D. Filipovi c, and W. Schachermayer, “Affine processes and applications in finance,” *The Annals of Applied Probability*, vol. 13, no. 3, Aug. 2003, ISSN: 1050-5164. DOI: [10.1214/aoap/1060202833](https://doi.org/10.1214/aoap/1060202833). [Online]. Available: <https://projecteuclid.org/journals/annals-of-applied-probability/volume-13/issue-3/Affine-processes-and-applications-in-finance/10.1214/aoap/1060202833.full>.
- [32] C. Bayer, C. B. Hammouda, and R. Tempone, “Hierarchical adaptive sparse grids and quasi Monte Carlo for option pricing under the rough Bergomi model,” Dec. 2018. DOI: [10.1080/14697688.2020.1744700](https://doi.org/10.1080/14697688.2020.1744700). [Online]. Available: <http://arxiv.org/abs/1812.08533> <http://dx.doi.org/10.1080/14697688.2020.1744700>.

- [33] C. Bayer, C. B. Hammouda, and R. Tempone, “Multilevel Monte Carlo combined with numerical smoothing for robust and efficient option pricing and density estimation,” Mar. 2020. [Online]. Available: <http://arxiv.org/abs/2003.05708>.
- [34] C. Bayer, C. B. Hammouda, and R. Tempone, “Numerical Smoothing with Hierarchical Adaptive Sparse Grids and Quasi-Monte Carlo Methods for Efficient Option Pricing,” Nov. 2021. DOI: [10.1080/14697688.2022.2135455](https://doi.org/10.1080/14697688.2022.2135455). [Online]. Available: <http://arxiv.org/abs/2111.01874><http://dx.doi.org/10.1080/14697688.2022.2135455>.
- [35] C. Bayer, C. B. Hammouda, A. Papapantoleon, M. Samet, and R. Tempone, “Optimal Damping with Hierarchical Adaptive Quadrature for Efficient Fourier Pricing of Multi-Asset Options in Lévy Models,” Mar. 2022. [Online]. Available: <http://arxiv.org/abs/2203.08196>.
- [36] G. Mast, *Replacing the Monte Carlo Simulation with the COS Method for PFE (Potential Future Exposure) Calculations (MSc Thesis)*, 2022. [Online]. Available: <https://repository.tudelft.nl/islandora/object/uuid%3A90df250b-189f-4d0f-8b1f-eb14cadd2009>.
- [37] C. W. Oosterlee and L. A. Grzelak, *Mathematical Modeling and Computation in Finance*. WORLD SCIENTIFIC (EUROPE), Nov. 2019, ISBN: 978-1-78634-794-7. DOI: [10.1142/q0236](https://doi.org/10.1142/q0236). [Online]. Available: <https://www.worldscientific.com/worldscibooks/10.1142/q0236>.
- [38] R. L. Karandikar and B. V. Rao, *Introduction to Stochastic Calculus* (Indian Statistical Institute Series). Singapore: Springer Singapore, 2018, ISBN: 978-981-10-8317-4. DOI: [10.1007/978-981-10-8318-1](https://doi.org/10.1007/978-981-10-8318-1). [Online]. Available: <http://link.springer.com/10.1007/978-981-10-8318-1>.
- [39] J. Gregory, *Counterparty Credit Risk - The New Challenge for Global Financial Markets*, 1st ed. Wiley, 2010, ISBN: 978-0-470-68576-1.
- [40] A. Green, *XVA: Credit, Funding and Capital Valuation Adjustments*. Wiley, Oct. 2015, ISBN: 9781118556788. DOI: [10.1002/9781119161233](https://doi.org/10.1002/9781119161233). [Online]. Available: <https://onlinelibrary.wiley.com/doi/book/10.1002/9781119161233>.
- [41] Basel Committee on Banking Supervision, “Basel III: A global regulatory framework for more resilient banks and banking systems,” *Bank For International Settlements*, 2011. [Online]. Available: <https://www.bis.org/publ/bcbs189.pdf>.
- [42] D. Brigo and F. Mercurio, *Interest rate models-theory and practice: with smile, inflation and credit*. 2006, vol. 82, p. 541, ISBN: 9783540221494. [Online]. Available: http://books.google.com/books?hl=en&lr=&id=2sGwSAfA8eAC&oi=fnd&pg=PR7&dq=mathematical+models+of+financial+derivatives&ots=Ir2b9BWzw_&sig=F9X5JzDPF5fP99faBnhX9tVJpgQ%5Cnhttp://scholar.google.com/scholar?hl=en&btnG=Search&q=intit.
- [43] O. Vasicek, “An equilibrium characterization of the term structure,” *Journal of Financial Economics*, vol. 5, no. 2, pp. 177–188, Nov. 1977, ISSN: 0304405X. DOI: [10.1016/0304-405X\(77\)90016-2](https://doi.org/10.1016/0304-405X(77)90016-2). [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/0304405X77900162>.

- [44] J. C. Cox, J. E. Ingersoll, and S. A. Ross, "A Theory of the Term Structure of Interest Rates," *Econometrica*, vol. 53, no. 2, p. 385, Mar. 1985, ISSN: 00129682. DOI: [10.2307/1911242](https://doi.org/10.2307/1911242). [Online]. Available: <https://www.jstor.org/stable/1911242?origin=crossref>.
- [45] J. Hull and A. White, "Pricing Interest-Rate-Derivative Securities," *Review of Financial Studies*, vol. 3, no. 4, pp. 573–592, Oct. 1990, ISSN: 0893-9454. DOI: [10.1093/rfs/3.4.573](https://doi.org/10.1093/rfs/3.4.573). [Online]. Available: <https://academic.oup.com/rfs/article-lookup/doi/10.1093/rfs/3.4.573>.
- [46] M. Di Francesco, "A General Gaussian Interest Rate Model Consistent with the Current Term Structure," *ISRN Probability and Statistics*, vol. 2012, pp. 1–16, Sep. 2012, ISSN: 2090-472X. DOI: [10.5402/2012/673607](https://doi.org/10.5402/2012/673607). [Online]. Available: <https://www.hindawi.com/journals/isrn/2012/673607/>.
- [47] "Statistical release: OTC derivatives statistics at end- June 2021," *Bank For International Settlements*, 2021. [Online]. Available: <https://www.bis.org/publ/otchy2111.pdf>.
- [48] D. Higham, *An Introduction to Financial Option Valuation*. Cambridge University Press, Apr. 2004, ISBN: 9780521547574. DOI: [10.1017/CB09780521547574](https://doi.org/10.1017/CB09780521547574). [Online]. Available: <https://www.cambridge.org/core/product/identifier/9780521547574/type/book>.
- [49] F. Black and M. Scholes, "The Pricing of Options and Corporate Liabilities," *Journal of Political Economy*, vol. 81, no. 3, pp. 637–654, 1973. [Online]. Available: <http://www.jstor.org/stable/1831029>.
- [50] J. P. Boyd, *Chebyshev and Fourier Spectral Methods* (Lecture Notes in Engineering). Berlin, Heidelberg: Springer Berlin Heidelberg, 1989, vol. 49, ISBN: 978-3-540-51487-9. DOI: [10.1007/978-3-642-83876-7](https://doi.org/10.1007/978-3-642-83876-7). [Online]. Available: <http://link.springer.com/10.1007/978-3-642-83876-7>.
- [51] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor Decomposition for Signal Processing and Machine Learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, Jul. 2017, ISSN: 1053-587X. DOI: [10.1109/TSP.2017.2690524](https://doi.org/10.1109/TSP.2017.2690524). [Online]. Available: <http://ieeexplore.ieee.org/document/7891546/>.
- [52] E. E. Papalexakis, C. Faloutsos, and N. D. Sidiropoulos, "Tensors for Data Mining and Data Fusion," *ACM Transactions on Intelligent Systems and Technology*, vol. 8, no. 2, pp. 1–44, Mar. 2017, ISSN: 2157-6904. DOI: [10.1145/2915921](https://doi.org/10.1145/2915921). [Online]. Available: <https://dl.acm.org/doi/10.1145/2915921>.
- [53] N. Kargas, N. D. Sidiropoulos, and X. Fu, "Tensors, Learning, and "Kolmogorov Extension" for Finite-Alphabet Random Vectors," *IEEE Transactions on Signal Processing*, vol. 66, no. 18, pp. 4854–4868, Sep. 2018, ISSN: 1053-587X. DOI: [10.1109/TSP.2018.2862383](https://doi.org/10.1109/TSP.2018.2862383). [Online]. Available: <https://ieeexplore.ieee.org/document/8424892/>.

- [54] T. G. Kolda and B. W. Bader, "Tensor Decompositions and Applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, Aug. 2009, ISSN: 0036-1445. DOI: [10.1137/07070111X](https://doi.org/10.1137/07070111X). [Online]. Available: <http://epubs.siam.org/doi/10.1137/07070111X>.
- [55] H. Lu, K. N. Plataniotis, and A. Venetsanopoulos, *Multilinear Subspace Learning*. Chapman and Hall/CRC, Dec. 2013, ISBN: 9780429108099. DOI: [10.1201/b16252](https://doi.org/10.1201/b16252). [Online]. Available: <https://www.taylorfrancis.com/books/9781439857298>.
- [56] R. You, Y. Yao, and J. Shi, "Tensor-based ultrasonic data analysis for defect detection in fiber reinforced polymer (FRP) composites," *Chemometrics and Intelligent Laboratory Systems*, vol. 163, pp. 24–30, Apr. 2017, ISSN: 01697439. DOI: [10.1016/j.chemolab.2017.02.007](https://doi.org/10.1016/j.chemolab.2017.02.007). [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0169743917300473>.
- [57] T. Kolda, "Multilinear operators for higher-order decompositions.," Sandia National Laboratories (SNL), Albuquerque, NM, and Livermore, CA (United States), Tech. Rep., Apr. 2006. DOI: [10.2172/923081](https://doi.org/10.2172/923081). [Online]. Available: <https://www.osti.gov/servlets/purl/923081/>.
- [58] J. Håstad, "Tensor rank is NP-complete," *Journal of Algorithms*, vol. 11, no. 4, pp. 644–654, Dec. 1990, ISSN: 01966774. DOI: [10.1016/0196-6774\(90\)90014-6](https://doi.org/10.1016/0196-6774(90)90014-6). [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/0196677490900146>.
- [59] Z. Cheng and M. Brands, "Finding Dimension-reduced Fourier-cosine Series Expansion via Supervised Machine-Learning and its Application In Multi-dimensional Integration," Breukelen, 2023.
- [60] Z. Cheng, *Dimension Reduction Techniques for Multi-dimensional Numerical Integrations Based on Fourier-cosine Series Expansion (MSc thesis)*, 2022. [Online]. Available: <https://repository.tudelft.nl/islandora/object/uuid:f6ffbcd6-df14-425b-84bb-63e4e105205c>.
- [61] N. Kargas and N. D. Sidiropoulos, "Supervised Learning and Canonical Decomposition of Multivariate Functions," *IEEE Transactions on Signal Processing*, vol. 69, pp. 1097–1107, 2021, ISSN: 1053-587X. DOI: [10.1109/TSP.2021.3055000](https://doi.org/10.1109/TSP.2021.3055000). [Online]. Available: <https://ieeexplore.ieee.org/document/9340610/>.
- [62] M. J. Ruijter and C. W. Oosterlee, "Two-Dimensional Fourier Cosine Series Expansion Method for Pricing Financial Options," *SIAM Journal on Scientific Computing*, vol. 34, no. 5, B642–B671, Jan. 2012, ISSN: 1064-8275. DOI: [10.1137/120862053](https://doi.org/10.1137/120862053). [Online]. Available: <http://epubs.siam.org/doi/10.1137/120862053>.
- [63] C. M. Bender and S. A. Orszag, *Advanced Mathematical Methods for Scientists and Engineers I*. New York, NY: Springer New York, 1999, ISBN: 978-1-4419-3187-0. DOI: [10.1007/978-1-4757-3069-2](https://doi.org/10.1007/978-1-4757-3069-2). [Online]. Available: <http://link.springer.com/10.1007/978-1-4757-3069-2>.
- [64] F. Fang, X. Shen, and C. Qiu, "A New and Efficient Fourier Method for Risk Quantification and Allocation of Credit Portfolios," *SSRN Electronic Journal*, 2022, ISSN: 1556-5068. DOI: [10.2139/ssrn.4153482](https://doi.org/10.2139/ssrn.4153482). [Online]. Available: <https://www.ssrn.com/abstract=4153482>.

- [65] L. N. Trefethen, *Approximation Theory and Approximation Practice, Extended Edition*. Philadelphia, PA: Society for Industrial and Applied Mathematics, Jan. 2019, ISBN: 978-1-61197-593-2. DOI: [10.1137/1.9781611975949](https://doi.org/10.1137/1.9781611975949). [Online]. Available: <https://epubs.siam.org/doi/book/10.1137/1.9781611975949>.
- [66] V. Piterbarg, "A Multi-currency Model with FX Volatility Skew," *SSRN Electronic Journal*, 2005, ISSN: 1556-5068. DOI: [10.2139/ssrn.685084](https://doi.org/10.2139/ssrn.685084). [Online]. Available: <http://www.ssrn.com/abstract=685084>.
- [67] L. A. Grzelak and C. W. Oosterlee, "On Cross-Currency Models with Stochastic Volatility and Correlated Interest Rates," *SSRN Electronic Journal*, 2010, ISSN: 1556-5068. DOI: [10.2139/ssrn.1618684](https://doi.org/10.2139/ssrn.1618684). [Online]. Available: <http://www.ssrn.com/abstract=1618684>.
- [68] D. Rosen, "CVA, Basel III and Wrong-Way Risk," *R2 Financial Technologies*, 2011.
- [69] D. Rosen and D. Saunders, "CVA the Wrong Way," *Journal of Risk Management in Financial Institutions*, vol. 5, no. 3, pp. 252–272, 2012.
- [70] K. Pan, "Credit valuation adjustment wrong-way risk in a Gaussian copula model," *Journal of Credit Risk*, Dec. 2019, ISSN: 17446619. DOI: [10.21314/JCR.2019.256](https://doi.org/10.21314/JCR.2019.256). [Online]. Available: <https://www.risk.net/journal-of-credit-risk/7241601/credit-valuation-adjustment-wrong-way-risk-in-a-gaussian-copula-model>.
- [71] J. Hull and A. White, "CVA and Wrong Way Risk," *Financial Analysts Journal*, vol. 68, no. 5, pp. 58–69, 2012.
- [72] A. Kumar, L. Markus, and N. Hari, "Arbitrage-free pricing of CVA for cross-currency swap with wrong-way risk under stochastic correlation modeling framework," Jul. 2021. [Online]. Available: <http://arxiv.org/abs/2107.06349>.
- [73] K. Pan and C. Khandrika, "An Analytical Expression for Credit Valuation Adjustment Pricing with Wrong-Way Risk," *Economics*, vol. 10, no. 3, p. 94, 2021, ISSN: 2376-659X. DOI: [10.11648/j.eco.20211003.14](https://doi.org/10.11648/j.eco.20211003.14). [Online]. Available: <http://www.sciencepublishinggroup.com/journal/paperinfo?journalid=177&doi=10.11648/j.eco.20211003.14>.
- [74] D. Filipovic, *Term-Structure Models*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, ISBN: 978-3-540-09726-6. DOI: [10.1007/978-3-540-68015-4](https://doi.org/10.1007/978-3-540-68015-4). [Online]. Available: <https://link.springer.com/10.1007/978-3-540-68015-4>.

A

ADDITIONAL LITERATURE REVIEW

A.1. WRONG-WAY RISK

Often, the assumption is made that the exposure and probability of default are independent when computing CVA. This results in convenient models to work with, but the so-called wrong-way risk cannot be captured in this way. Wrong-way risk is an important aspect to consider when it comes to pricing CVA. In the context of CVA, Wrong-way risk occurs whenever there is an adverse co-dependence between the exposure to a counterparty and the risk that the counterparty will default, while right-way risk occurs whenever there is a favourable co-dependence [40]. Two main approaches to include wrong-way risk is either using copula techniques or stochastic credit models to relate the probability of default and exposure.

Copula techniques incorporating wrong-way risk are described [68]–[70]. Often the Gaussian Copula is used for convenience, but the theory is general and any copula can be used. The application of copulas requires the marginal distribution of both the exposure and the probability of default. For the probability of default, an analytic form on the distribution is often imposed and hence the marginal distribution is known. However, the marginal distribution of the exposure is only known in analytical form for a few simple products. Generally, the marginal distribution has to be derived empirically from the Monte Carlo simulation.

Another approach similar to the copulas is to model the hazard rate as a function of a risk factor, for example a short rate, associated with the expected exposure. This idea described in [71] is very straightforward and introduces a dependence between the exposure and the hazard rate in an intuitive way. The calibration of the model, however, is difficult.

Stochastic credit models assume some dynamics for the hazard rate, for example an Ornstein-Uhlenbeck process with jumps [40] or an Cox-Ingersoll-Ross (CIR) model [72]. The advantages of using a stochastic process for credit in right-way/wrong-way risk models are considerable. Firstly it is a dynamical model for a process that is clearly dynamic as credit spreads are very volatile. The doubly stochastic nature of the credit

A

process is therefore fully captured. Secondly it is more intuitive as the credit process is directly linked to other assets through diffusive correlations and jump relationships. Calibration is still problematic as the volatility of the credit process must be chosen.

An analytical expression for CVA with wrong-way risk for basic futures and forward contracts is derived in [73]. Instead of using the default dynamics, the credit deterioration dynamics are used.

B

MATHEMATICAL PROOFS

B.1. SOLUTION ORNSTEIN-UHLENBECK PROCESS

Proposition B.1.1 (Dynamics Ornstein-Uhlenbeck Process). *Let the Ornstein-Uhlenbeck process $x(t)$ be defined by the following stochastic differential equation,*

$$dx(t) = -ax(t)dt + \sigma dW(t), \quad x(t_0) = x_0,$$

where a is a positive constant, σ the volatility and $W(t)$ is a standard Brownian motion. An analytical solution to this stochastic differential equation can be derived, which is given by

$$x(t) = x_0 e^{-a(t-t_0)} + \sigma \int_{t_0}^t e^{-a(t-s)} dW(s).$$

Proof. The solution to the Ornstein-Uhlenbeck process can be found by applying Itô's formula to the function $f(t, x) = e^{at}x$, which gives the following result,

$$\begin{aligned} de^{at}x(t) &= \frac{\partial f}{\partial t} dt + \frac{\partial f}{\partial x} dx + \frac{1}{2} \frac{\partial^2 f}{\partial x^2} d[x] \\ &= ae^{at}x dt + e^{at} dx + \frac{1}{2} \cdot 0 \cdot d[x] \\ &= ae^{at}x dt + e^{at} [-ax dt + \sigma dW] \\ &= e^{at} \sigma dW. \end{aligned}$$

The solution can now be obtained by integrating both sides from t_0 to t ,

$$\begin{aligned}\int_{t_0}^t d e^{as} x(s) &= \sigma \int_{t_0}^t e^{as} dW(s) \\ e^{at} x(t) - e^{at_0} x(t_0) &= \sigma \int_{t_0}^t e^{as} dW(s) \\ e^{at} x(t) &= e^{at_0} x(t_0) + \sigma \int_{t_0}^t e^{as} dW(s) \\ x(t) &= x_0 e^{-a(t-t_0)} + \sigma \int_{t_0}^t e^{-a(t-s)} dW(s).\end{aligned}$$

□

B.2. ZERO-COUPON BOND FORMULA UNDER THE G1++ MODEL

Proposition B.2.1 (Zero-Coupon Bond Formula G1++ Model). *Let the short rate process be modelled according to the G1++ dynamics,*

$$r(t) = x(t) + \beta(t),$$

where the shifted short rate process $x(t)$ follows an Ornstein-Uhlenbeck process,

$$dx(t) = -ax(t)dt + \sigma dW(t), \quad x(0) = 0,$$

Then, the ZCB formula can be expressed as

$$P(t, T) = A(t, T) e^{-B(t, T)x(t)},$$

where

$$A(t, T) = \frac{P^M(0, T)}{P^M(0, t)} e^{\frac{1}{2}[V(t, T) - V(0, T) + V(0, t)]}$$

$$B(t, T) = \frac{1}{a} [1 - e^{-a(T-t)}].$$

$V(t, T)$ is the variance of $\int_t^T x(s)ds$ conditional on the natural filtration \mathcal{F}_t of the Brownian motion, and is given by

$$V(t, T) = \frac{\sigma^2}{a^2} \left(T - t - 2 \frac{1 - e^{-a(T-t)}}{a} + \frac{1 - e^{-2a(T-t)}}{2a} \right).$$

Proof. The general formula for the ZCB price is presented in Definition 2.3.3, and is repeated here,

$$P(t, T) = \mathbb{E}^Q \left[e^{-\int_t^T r(s)ds} \middle| \mathcal{F}_t \right].$$

Inserting the G1++ model for the short rate process yields

$$\begin{aligned} P(t, T) &= \mathbb{E}^{\mathbb{Q}} \left[e^{-\int_t^T x(s) + \beta(s) ds} \middle| \mathcal{F}_t \right] \\ &= e^{-\int_t^T \beta(s) ds} \mathbb{E}^{\mathbb{Q}} \left[e^{-\int_t^T x(s)} \middle| \mathcal{F}_t \right]. \end{aligned} \quad (\text{B.1})$$

To evaluate the expectation, the integration of the short rate process is considered first. The solution of the short rate process $x(t)$ was derived in Proposition B.1.1. Applying Fubini's Theorem for stochastic integral (see [74]), the integration of the short rate process gives the following result,

$$\begin{aligned} \int_t^T x(s) ds &= \int_t^T \left[x(t) e^{-a(s-t)} + \sigma \int_t^s e^{-a(s-u)} dW(u) \right] ds \\ &= -\frac{x(t)}{a} [e^{-a(T-t)} - 1] + \sigma \int_t^T \int_t^s e^{-a(s-u)} dW(u) ds \\ (\text{Fubini}) &= -\frac{x(t)}{a} [e^{-a(T-t)} - 1] + \sigma \int_t^T \int_u^T e^{-a(s-u)} ds dW(u) \\ &= -\frac{x(t)}{a} [e^{-a(T-t)} - 1] + \sigma \int_t^T -\frac{1}{a} [e^{-a(T-u)} - 1] dW(u) \\ &= \frac{x(t)}{a} [1 - e^{-a(T-t)}] + \frac{\sigma}{a} \int_t^T [1 - e^{-a(T-u)}] dW(u). \end{aligned}$$

By the properties of the Itô integral, we find that $\int_t^T x(s) ds$ conditional on \mathcal{F}_t is again normally distributed with mean and variance

$$\begin{aligned} \mathbb{E}^{\mathbb{Q}} \left[\int_t^T x(s) ds \middle| \mathcal{F}_t \right] &= \frac{x(t)}{a} [1 - e^{-a(T-t)}], \\ \text{Var} \left[\int_t^T x(s) ds \middle| \mathcal{F}_t \right] &= \text{Var} \left[\frac{\sigma}{a} \int_t^T [1 - e^{-a(T-u)}] dW(u) \right] \\ &= \frac{\sigma^2}{a^2} \mathbb{E}^{\mathbb{Q}} \left[\left(\int_t^T [1 - e^{-a(T-u)}] dW(u) \right)^2 \middle| \mathcal{F}_t \right] \\ (\text{Itô's Isometry}) &= \frac{\sigma^2}{a^2} \int_t^T (1 - e^{-a(T-u)})^2 du \\ &= \frac{\sigma^2}{a^2} \left(T - t - 2 \frac{1 - e^{-a(T-t)}}{a} + \frac{1 - e^{-2a(T-t)}}{2a} \right) \\ &:= V(t, T) \end{aligned} \quad (\text{B.2})$$

Since $\int_t^T x(s) ds$ is normally distributed, the moment generating function can be utilized to find the expectation in B.1. If $Z \sim N(\mu, \sigma^2)$, the moment generating function gives $\mathbb{E}[e^{tZ}] = e^{t\mu + \frac{1}{2}\sigma^2 t^2}$. Using this property we obtain

$$\mathbb{E}^{\mathbb{Q}} \left[e^{-\int_t^T x(s)} \middle| \mathcal{F}_t \right] = e^{-\frac{x(t)}{a} [1 - e^{-a(T-t)}] + \frac{1}{2} V(t, T)}. \quad (\text{B.3})$$

The ZCB price should fit the market discount curve for all maturities T . Consequently, the ZCB formula has to satisfy $P(0, T) = P^M(0, T)$, where $P^M(0, T)$ denotes the discount factor observed in the market. Evaluating the ZCB formula at $t = 0$ yields

$$\begin{aligned} P(0, T) &= e^{-\int_0^T \beta(s) ds} \mathbb{E}^{\mathbb{Q}} \left[e^{-\int_0^T x(s)} \middle| \mathcal{F}_t \right] \\ &= e^{-\int_0^T \beta(s) ds} e^{-\frac{x(0)}{a} [1 - e^{-aT}] + \frac{1}{2} V(0, T)} \\ (x(0) = 0) &= e^{-\int_0^T \beta(s) ds} e^{\frac{1}{2} V(0, T)}. \end{aligned}$$

Therefore, the following relation with the market discount curve is obtained,

$$P^M(0, T) = e^{-\int_0^T \beta(s) ds} e^{\frac{1}{2} V(0, T)} \implies e^{-\int_0^T \beta(s) ds} = P^M(0, T) e^{-\frac{1}{2} V(0, T)}. \quad (\text{B.4})$$

Finally, the expressions from B.4 and B.3 are used to obtain the ZCB formula under the G1++ short rate model,

$$\begin{aligned} P(t, T) &= e^{-\int_t^T \beta(s) ds} \mathbb{E}^{\mathbb{Q}} \left[e^{-\int_t^T x(s)} \middle| \mathcal{F}_t \right] \\ &= e^{-\int_0^T \beta(s) ds + \int_0^t \beta(s) ds} \cdot e^{-\frac{x(t)}{a} [1 - e^{-a(T-t)}] + \frac{1}{2} V(t, T)} \\ &= \frac{P^M(0, T)}{P^M(0, t)} e^{-\frac{1}{2} V(0, T) + \frac{1}{2} V(0, t)} \cdot e^{-\frac{x(t)}{a} [1 - e^{-a(T-t)}] + \frac{1}{2} V(t, T)} \\ &= A(t, T) e^{-B(t, T)x(t)}, \end{aligned}$$

with

$$\begin{aligned} A(t, T) &= \frac{P^M(0, T)}{P^M(0, t)} e^{\frac{1}{2} [V(t, T) - V(0, T) + V(0, t)]} \\ B(t, T) &= \frac{1}{a} [1 - e^{-a(T-t)}]. \end{aligned}$$

$V(t, T)$ was derived in B.2, and is the variance of $\int_t^T x(s) ds$ conditional on the natural filtration \mathcal{F}_t of the Brownian motion, and is given by

$$V(t, T) = \frac{\sigma^2}{a^2} \left(T - t - 2 \frac{1 - e^{-a(T-t)}}{a} + \frac{1 - e^{-2a(T-t)}}{2a} \right).$$

□

B.3. CORRELATION STRUCTURE OF STATE VARIABLES

Proposition B.3.1. *The correlation structure of the standard normally distributed state variables Z_d , Z_f , and Z_X , derived from the model in 7.1, is given by*

$$\begin{aligned} \text{Cor}(Z_d, Z_f) &= \frac{\frac{\rho_{df}}{a_d + a_f}(1 - e^{-(a_d + a_f)t})}{\sqrt{\frac{1}{2a_d}(1 - e^{-2a_d t})} \sqrt{\frac{1}{2a_f}(1 - e^{-2a_f t})}}, \\ \text{Cor}(Z_d, Z_X) &= \frac{\frac{\rho_{dX}}{a_d}(1 - e^{-a_d t})}{\sqrt{\frac{1}{2a_d}(1 - e^{-2a_d t})} \sqrt{t}}, \\ \text{Cor}(Z_f, Z_X) &= \frac{\frac{\rho_{fX}}{a_f}(1 - e^{-a_f t})}{\sqrt{\frac{1}{2a_f}(1 - e^{-2a_f t})} \sqrt{t}}. \end{aligned}$$

Proof. We start with the solution of the processes x_d , x_f and X ,

$$\begin{aligned} x_d(t) &= x_d(0)e^{-a_d t} + \sigma_d \int_0^t e^{-a_d(t-s)} dW_d(s), \\ x_f(t) &= x_f(0)e^{-a_f t} + \sigma_f \int_0^t e^{-a_f(t-s)} dW_f(s), \\ \log(X(t)) &= \log(X(0)) + \left(\mu - \frac{1}{2}\sigma^2 \right) t + \sigma W_X(t). \end{aligned}$$

These processes consist of a deterministic and stochastic part. The stochastic part due to the Brownian motion will be denoted with the state variables $\hat{Z}_d(t)$, $\hat{Z}_f(t)$ and $\hat{Z}_X(t)$, so that we find

$$\begin{aligned} x_d(t) &= x_d(0)e^{-a_d t} + \sigma_d \hat{Z}_d(t), \\ x_f(t) &= x_f(0)e^{-a_f t} + \sigma_f \hat{Z}_f(t), \\ \log(X(t)) &= \log(X(0)) + \left(\mu - \frac{1}{2}\sigma^2 \right) t + \sigma \hat{Z}_X(t). \end{aligned}$$

From the properties of the Brownian motion and the Itô integral, we know that $\hat{Z}_d(t)$, $\hat{Z}_f(t)$ and $\hat{Z}_X(t)$ are normally distributed with mean zero. Therefore, normalising these state variables yields

$$\begin{aligned} \hat{Z}_d(t) &= \int_0^t e^{-a_d(t-s)} dW_d(s) = \sigma_{\hat{Z}_d} Z_d(t), \\ \hat{Z}_f(t) &= \int_0^t e^{-a_f(t-s)} dW_f(s) = \sigma_{\hat{Z}_f} Z_f(t), \\ \hat{Z}_X(t) &= W_X(t) = \sigma_{\hat{Z}_X} Z_X(t), \end{aligned}$$

where $Z_d(t)$, $Z_f(t)$ and $Z_X(t)$ have a standard normal distribution, i.e. $Z_d, Z_f, Z_X \sim N(0,1)$. The time parameter t indicates the time-dependence of the correlation structure.

To obtain the the correlation between Z_d and Z_f , the following expression has to be computed,

$$\begin{aligned}
 \text{Cor}(Z_d, Z_f) &= \frac{\text{Cov}(Z_d, Z_f)}{\sigma_{Z_d} \sigma_{Z_f}} \\
 &= \text{Cov}(Z_d, Z_f) \\
 &= \text{Cov}\left(\frac{1}{\sigma_{\hat{Z}_d}} \hat{Z}_d, \frac{1}{\sigma_{\hat{Z}_f}} \hat{Z}_f\right) \\
 &= \frac{\text{Cov}(\hat{Z}_d, \hat{Z}_f)}{\sigma_{\hat{Z}_d} \sigma_{\hat{Z}_f}}.
 \end{aligned} \tag{B.5}$$

The covariance between \hat{Z}_d and \hat{Z}_f can be obtained using Itô's Isometry property. If ρ_{df} represents the correlation between the Brownian motions of the processes, then the covariance is given by

$$\begin{aligned}
 \text{Cov}(\hat{Z}_d, \hat{Z}_f) &= \text{Cov}\left(\int_0^t e^{-a_d(t-s)} dW_d(s), \int_0^t e^{-a_f(t-s)} dW_f(s)\right) \\
 \text{(Itô's Isometry)} &= \int_0^t e^{-(a_d+a_f)(t-s)} d[W_d, W_f]_s \\
 &= \rho_{df} \int_0^t e^{-(a_d+a_f)(t-s)} ds \\
 &= \rho_{df} e^{-(a_d+a_f)t} \left[\frac{1}{a_d+a_f} e^{(a_d+a_f)s} \right]_{s=0}^t \\
 &= \frac{\rho_{df}}{a_d+a_f} (1 - e^{-(a_d+a_f)t}).
 \end{aligned}$$

The standard deviation of the processes \hat{Z}_d and \hat{Z}_f is obtained by computing the corresponding variance. Making use of the Itô Isometry property again we find

$$\begin{aligned}
 \sigma_{\hat{Z}_d}^2 &= \text{Var}\left(\int_0^t e^{-a_d(t-s)} dW_d(s)\right) \\
 &= \int_0^t e^{-2a_d(t-s)} ds \\
 &= \frac{1}{2a_d} e^{-2a_d t} [e^{2a_d s}]_{s=0}^t \\
 &= \frac{1}{2a_d} (1 - e^{-2a_d t}).
 \end{aligned} \tag{B.6}$$

The variance for \hat{Z}_f can be found in a similar way. Therefore, the correlation between Z_d

and Z_f is given by

$$\text{Cor}(Z_d, Z_f) = \frac{\text{Cov}(\hat{Z}_d, \hat{Z}_f)}{\sigma_{\hat{Z}_d} \sigma_{\hat{Z}_f}} = \frac{\frac{\rho_{df}}{a_d + a_f} (1 - e^{-(a_d + a_f)t})}{\sqrt{\frac{1}{2a_d} (1 - e^{-2a_d t})} \sqrt{\frac{1}{2a_f} (1 - e^{-2a_f t})}}.$$

The correlation between Z_d and Z_X has a similar expression as found in B.5,

$$\text{Cor}(Z_d, Z_X) = \frac{\text{Cov}(\hat{Z}_d, \hat{Z}_X)}{\sigma_{\hat{Z}_d} \sigma_{\hat{Z}_X}}.$$

If ρ_{dX} denotes the correlation between the Brownian motions W_d and W_X , the covariance has the following form,

$$\begin{aligned} \text{Cov}(\hat{Z}_d, \hat{Z}_X) &= \text{Cov}\left(\int_0^t e^{-a_d(t-s)} dW_d(s), W_X(t)\right) \\ &= \text{Cov}\left(\int_0^t e^{-a_d(t-s)} dW_d(s), \int_0^t dW_X(s)\right) \\ &= \rho_{dX} \int_0^t e^{-(a_d(t-s))} ds \\ &= \frac{\rho_{dX}}{a_d} (1 - e^{-a_d t}). \end{aligned}$$

The variance of the Brownian motion $W_X(t)$ is t , so that $\sigma_{\hat{Z}_X} = \sqrt{t}$. As $\sigma_{\hat{Z}_d}^2$ was already computed in B.6, the correlation can now be computed,

$$\text{Cor}(Z_d, Z_X) = \frac{\text{Cov}(\hat{Z}_d, \hat{Z}_X)}{\sigma_{\hat{Z}_d} \sigma_{\hat{Z}_X}} = \frac{\frac{\rho_{dX}}{a_d} (1 - e^{-a_d t})}{\sqrt{\frac{1}{2a_d} (1 - e^{-2a_d t})} \sqrt{t}}.$$

The correlation between Z_f and Z_X can be found in the exact same way, so that we have

$$\text{Cor}(Z_f, Z_X) = \frac{\text{Cov}(\hat{Z}_f, \hat{Z}_X)}{\sigma_{\hat{Z}_f} \sigma_{\hat{Z}_X}} = \frac{\frac{\rho_{fX}}{a_f} (1 - e^{-a_f t})}{\sqrt{\frac{1}{2a_f} (1 - e^{-2a_f t})} \sqrt{t}}.$$

□

C

CPD AS DIRECT INTEGRATION METHOD

This section shows how the technique of CPD in the context of Fourier-cosine series can be used to compute a multivariate expectation in a direct way. Therefore, consider a multivariate random variable $\mathbf{X} = (X_1, \dots, X_N)$ with corresponding joint density function $f(\cdot)$. The multivariate expectation of $g(\cdot)$ is defined as

$$\mathbb{E}[g(\mathbf{X})] = \int_{\mathbb{R}^N} g(x_1, \dots, x_N) f(x_1, \dots, x_N) dx_1 \cdots dx_N. \quad (\text{C.1})$$

Instead of decomposing only the joint density function with CPD into its Fourier-cosine series, as is done within the COS-CPD method, this technique is now applied on the full integrand. Consequently, the dimension-reduced Fourier-cosine series of the function $g(\mathbf{x})f(\mathbf{x})$ is generated with CPD. Equation 3.5 shows that this approximation takes the following form,

$$g(\mathbf{x})f(\mathbf{x}) \approx \sum_{k_1=0}^{K-1} \dots \sum_{k_N=0}^{K-1} \sum_{r=1}^R \prod_{n=1}^N \mathbf{a}_r^n[k_n] \mathbf{v}_n[k_n],$$

with $\{\mathbf{a}_r^n\}_{n=1}^N$ the columns of the factorized matrices $\{\mathbf{A}_n\}_{n=1}^N$ and $\{\mathbf{v}_n\}_{n=1}^N$ the cosine basis functions (i.e. $\mathbf{v}_n[k_n] = \cos(k_n \pi \frac{x_n - a}{b - a})$). Assuming that the Fourier-cosine series expan-

sion is constructed on $[a, b]^N$, the expectation in (C.1) can be computed as follows,

$$\begin{aligned}
 \mathbb{E}[g(\mathbf{X})] &= \int_{\mathbb{R}^N} g(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} \\
 &\approx \int_{[a, b]^N} \sum_{k_1=0}^{K-1} \dots \sum_{k_N=0}^{K-1} \sum_{r=1}^R \prod_{n=1}^N \mathbf{a}_r^n[k_n] \mathbf{v}_n[k_n] d\mathbf{x} \\
 &= \sum_{k_1=0}^{K-1} \dots \sum_{k_N=0}^{K-1} \sum_{r=1}^R \int_{[a, b]^N} \prod_{n=1}^N \mathbf{a}_r^n[k_n] \mathbf{v}_n[k_n] d\mathbf{x} \\
 &= \sum_{k_1=0}^{K-1} \dots \sum_{k_N=0}^{K-1} \sum_{r=1}^R \mathbf{a}_r^1[k_1] \dots \mathbf{a}_r^N[k_N] \int_{[a, b]^N} \prod_{n=1}^N \mathbf{v}_n[k_n] d\mathbf{x}. \tag{C.2}
 \end{aligned}$$

The integral that has to be evaluated, consists of the product of the cosine basis functions. Therefore, the integrand is given by

$$\prod_{n=1}^N \mathbf{v}_n[k_n] = \cos\left(k_1 \pi \frac{x_1 - a}{b - a}\right) \dots \cos\left(k_N \pi \frac{x_N - a}{b - a}\right).$$

Using the fact that this product of cosine basis functions is a separable function (i.e. it can be written in the form $f(x_1)g(x_2)\dots h(x_N)$), the N -dimensional integral can be written as product of N one-dimensional integrals,

$$\begin{aligned}
 \int_{[a, b]^N} \prod_{n=1}^N \mathbf{v}_n[k_n] dx_1 dx_2 \dots dx_N &= \int_{[a, b]^N} \cos\left(k_1 \pi \frac{x_1 - a}{b - a}\right) \dots \cos\left(k_N \pi \frac{x_N - a}{b - a}\right) dx_1 \dots dx_N \\
 &= \int_a^b \cos\left(k_1 \pi \frac{x_1 - a}{b - a}\right) dx_1 \dots \int_a^b \cos\left(k_N \pi \frac{x_N - a}{b - a}\right) dx_N \\
 &= \prod_{n=1}^N \int_a^b \cos\left(k_n \pi \frac{x_n - a}{b - a}\right) dx_n. \tag{C.3}
 \end{aligned}$$

When integrating the cosine basis functions, a distinction has to be made between $k_n = 0$ and $k_n \neq 0$. For $k_n = 0$, the cosine basis functions are

$$v_n[0] = \cos\left(0 \cdot \pi \frac{x_n - a}{b - a}\right) = 1,$$

while for $k_n \neq 0$ we have

$$v_n[k_n] = \cos\left(k_n \cdot \pi \frac{x_n - a}{b - a}\right).$$

Consequently, integrating the cosine basis functions over the domain $[a, b]$ we obtain

$$\int_a^b \cos\left(k_n \pi \frac{x_n - a}{b - a}\right) dx_n = \begin{cases} (b - a), & \text{if } k_n = 0 \\ \frac{b-a}{k_n \pi} \sin(k_n \pi), & \text{if } k_n = 1, 2, \dots, K-1. \end{cases}$$

Since $\sin(k_n \pi) = 0$ for $k_n \in \mathbb{N}$, all integral values are zero for $k_n = 1, 2, \dots, K-1$,

$$\int_a^b \cos\left(k_n \pi \frac{x_n - a}{b - a}\right) dx_n = \begin{cases} (b - a), & \text{if } k_n = 0 \\ 0, & \text{if } k_n = 1, 2, \dots, K-1 \end{cases}$$

Therefore, the product of the integrals in (C.3) is non-zero only for $k_1 = k_2 = \dots = k_N = 0$. This results in a great simplification of the formula for the expectation in (C.2):

$$\begin{aligned}\mathbb{E}[g(\mathbf{X})] &= \sum_{k_1=0}^{K-1} \dots \sum_{k_N=0}^{K-1} \sum_{r=1}^R \mathbf{a}_r^1[k_1] \dots \mathbf{a}_r^N[k_N] \int_a^b \prod_{n=1}^N \mathbf{v}_n[k_n] d\mathbf{x} \\ &= \sum_{r=1}^R \mathbf{a}_r^1[0] \dots \mathbf{a}_r^N[0] (b-a)^N \\ &= (b-a)^N \sum_{r=1}^R \mathbf{a}_r^1[0] \dots \mathbf{a}_r^N[0].\end{aligned}$$

So the final expression to approximate the expectation of $g(\cdot)$ on the domain $[a, b]^N$ is given by,

$$\mathbb{E}[g(\mathbf{X})] \approx (b-a)^N \sum_{r=1}^R \mathbf{a}_r^1[0] \dots \mathbf{a}_r^N[0].$$

This formula can be easily generalized to different grid sizes $[a_n, b_n]$ for $n = 1, \dots, N$. Then, the approximation of the expectation on the domain $[a_1, b_1] \times \dots \times [a_N, b_N]$ is as follows,

$$\mathbb{E}[g(\mathbf{X})] \approx (b_1 - a_1) \dots (b_N - a_N) \sum_{r=1}^R \mathbf{a}_r^1[0] \dots \mathbf{a}_r^N[0].$$

Note that for this direct integration approach with CPD, only the first rows the factor matrices $\{\mathbf{A}_n\}_{n=1}^N$ are required to compute the expectation value.

D

CLENSHAW-CURTIS QUADRATURE

Throughout this thesis, the Clenshaw-Curtis quadrature is used as numerical integration method. Typically, the algorithm is described for integration of a function $f(x)$ over the interval $[-1, 1]$. The first step is to employ the variable substitution $x = \cos(\theta)$, which will transform the integration problem into the following form,

$$\int_{-1}^1 f(x) dx = \int_0^\pi f(\cos(\theta)) \sin(\theta) d\theta.$$

Then, this integral can be solved by considering the Fourier-cosine series of $f(\cos(\theta))$,

$$f(\cos(\theta)) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos(k\theta).$$

Inserting the Fourier-cosine series into the integral we find,

$$\begin{aligned} \int_0^\pi f(\cos(\theta)) \sin(\theta) d\theta &= \int_0^\pi \left[\frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos(k\theta) \right] \sin(\theta) d\theta \\ &= a_0 + \sum_{k=1}^{\infty} \frac{2a_{2k}}{1 - (2k)^2}. \end{aligned}$$

Consequently, the integration problem is translated to the computation of the Fourier coefficients, which are defined by

$$a_k = \frac{2}{\pi} \int_0^\pi f(\cos(\theta)) \cos(k\theta) d\theta, \quad k \in \mathbb{N}.$$

The coefficients can be approximated using the type-I discrete cosine transform (DCT-I). If $N + 1$ quadrature points are employed, the coefficients can be expressed as

$$a_k \approx \frac{2}{N} \left[\frac{f(1)}{2} + \frac{f(-1)}{2} (-1)^k + \sum_{n=1}^{N-1} f(\cos(n\pi/N)) \cos(nk\pi/N) \right],$$

for $k = 0, \dots, N$. The advantage of using this approach is that the coefficients can be computed efficiently with FFT algorithms, which reduces the computational complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log N)$. Furthermore, the function $f(\cos(\theta))$ is even and periodic by construction, such that the Fourier-cosine series converges exponentially if the original function $f(x)$ is sufficiently smooth.

Because only the even coefficients contribute to the integral value, an alternative approach of computing the coefficients is to use a DCT-I of order $N/2$ (assuming that N is an even number),

$$a_{2k} \approx \frac{2}{N} \left[\frac{f(1) + f(-1)}{2} + f(0)(-1)^k + \sum_{n=1}^{N/2-1} \{f(\cos(n\pi/N)) + f(-\cos(n\pi/N))\} \cos(nk\pi/(N/2)) \right].$$

Once the Fourier coefficients are computed, the Clenshaw-Curtis quadrature approximation of the integral is given by

$$\int_{-1}^1 f(x) dx \approx a_0 + \sum_{\substack{k=1 \\ k \text{ even}}}^N \frac{2a_{2k}}{1 - (2k)^2}.$$

The derivations presented above are based on the integration interval $[-1, 1]$. However, the same technique can be employed for any domain $[a, b]$ by utilizing the following change of variables,

$$x = \frac{b-a}{2} \tilde{x} + \frac{b+a}{2}.$$

In that case, the integration can be transformed again to the interval $[-1, 1]$ on which the methodology described above can be applied again,

$$\begin{aligned} \int_a^b f(x) dx &= \int_{-1}^1 f\left(\frac{b-a}{2} \tilde{x} + \frac{b+a}{2}\right) \frac{b-a}{2} d\tilde{x} \\ &= \frac{b-a}{2} \int_{-1}^1 f\left(\frac{b-a}{2} \tilde{x} + \frac{b+a}{2}\right) d\tilde{x}. \end{aligned}$$

E

PORTFOLIOS

This appendix contains the portfolios up to 100 derivatives used in the numerical experiments of Chapter 7 to compute netting-set level PFE and EE profiles.

E.1. PORTFOLIO 100 DERIVATIVES 3D

Tradefld	ProductType	PayOrReceive	Ccy	Notional	IsFixed	StartDate	Coupon	CouponFrequency	NumberOfCoupons	Maturity
0	FRA	-1	USD	968	ONWAAR	0,255556	IBOR			1,777778
0	FRA	1	USD	968	WAAR	0,255556	0,011			1,777778
1	FRA	-1	USD	2160	ONWAAR	1,438889	IBOR			7,019444
1	FRA	1	USD	2160	WAAR	1,438889	0,033			7,019444
2	FRA	-1	JPY	68563	ONWAAR	0,511111	IBOR			5,072222
2	FRA	1	JPY	68563	WAAR	0,511111	0,023			5,072222
3	FRA	-1	JPY	106856	ONWAAR	0,086111	0,012			8,713889
3	FRA	1	JPY	106856	ONWAAR	0,086111	IBOR			8,713889
4	FRA	-1	USD	1910	WAAR	1,525	0,054			6,6
4	FRA	1	USD	1910	ONWAAR	1,525	IBOR			6,6
5	FRA	1	JPY	244011	ONWAAR	0,341667	IBOR			2,372222
5	FRA	-1	JPY	244011	WAAR	0,341667	0,043			2,372222
6	FRA	-1	JPY	110295	WAAR	0,761111	0,021			9,894444
6	FRA	1	JPY	110295	ONWAAR	0,761111	IBOR			9,894444
7	FRA	-1	JPY	154244	WAAR	0,172222	0,005			7,275
7	FRA	1	JPY	154244	ONWAAR	0,172222	IBOR			7,275
8	FRA	-1	JPY	198682	ONWAAR	0,341667	IBOR			5,919444
8	FRA	1	JPY	198682	WAAR	0,341667	0,012			5,919444
9	FRA	-1	USD	1591	WAAR	0,425	0,033			8,033333
9	FRA	1	USD	1591	ONWAAR	0,425	IBOR			8,033333
10	FRA	1	JPY	148287	WAAR	0,425	0,016			6,005556
10	FRA	-1	JPY	148287	ONWAAR	0,425	IBOR			6,005556
11	FRA	-1	JPY	245248	WAAR	0,761111	0,018			1,777778
11	FRA	1	JPY	245248	ONWAAR	0,761111	IBOR			1,777778
12	FRA	1	JPY	101599	WAAR	0,425	0,023			4,483333
12	FRA	-1	JPY	101599	ONWAAR	0,425	IBOR			4,483333
13	FRA	-1	USD	811	ONWAAR	1,186111	IBOR			6,763889
13	FRA	1	USD	811	WAAR	1,186111	0,048			6,763889
14	FRA	1	JPY	94341	ONWAAR	0,341667	IBOR			0,844444
14	FRA	-1	JPY	94341	WAAR	0,341667	0,028			0,844444
15	FRA	1	JPY	75371	WAAR	1,438889	0,026			4,988889
15	FRA	-1	JPY	75371	ONWAAR	1,438889	IBOR			4,988889
16	FRA	1	USD	1791	WAAR	1,013889	0,043			1,525
16	FRA	-1	USD	1791	ONWAAR	1,013889	IBOR			1,525
17	FRA	1	USD	2459	WAAR	0,675	0,048			5,244444
17	FRA	-1	USD	2459	ONWAAR	0,675	IBOR			5,244444
18	FRA	1	JPY	105362	ONWAAR	1,438889	IBOR			4,483333
18	FRA	-1	JPY	105362	WAAR	1,438889	0,031			4,483333
19	FRA	-1	USD	658	WAAR	0,341667	0,022			7,444444
19	FRA	1	USD	658	ONWAAR	0,341667	IBOR			7,444444
20	FRA	-1	USD	1936	ONWAAR	1,1	IBOR			8,713889
20	FRA	1	USD	1936	WAAR	1,1	0,008			8,713889
21	FRA	-1	JPY	229981	ONWAAR	1,186111	IBOR			2,705556
21	FRA	1	JPY	229981	WAAR	1,186111	0,054			2,705556
22	FRA	1	USD	1479	WAAR	1,013889	0,05			8,116667
22	FRA	-1	USD	1479	ONWAAR	1,013889	IBOR			8,116667
23	FRA	1	USD	2455	WAAR	1,691667	0,007			4,733333
23	FRA	-1	USD	2455	ONWAAR	1,691667	IBOR			4,733333
24	FRA	1	JPY	196459	ONWAAR	0,675	IBOR			7,275
24	FRA	-1	JPY	196459	WAAR	0,675	0,043			7,275
25	FRA	1	USD	689	ONWAAR	0,675	IBOR			7,275
25	FRA	-1	USD	689	WAAR	0,675	0,036			7,275
26	FRA	-1	JPY	169941	WAAR	1,611111	0,034			2,627778
26	FRA	1	JPY	169941	ONWAAR	1,611111	IBOR			2,627778
27	FRA	1	USD	1948	ONWAAR	0,172222	IBOR			8,791667
27	FRA	-1	USD	1948	WAAR	0,172222	0,034			8,791667
28	FRA	1	JPY	211128	ONWAAR	1,438889	IBOR			8,541667
28	FRA	-1	JPY	211128	WAAR	1,438889	0,022			8,541667
29	FRA	1	JPY	124135	WAAR	1,525	0,053			7,613889
29	FRA	-1	JPY	124135	ONWAAR	1,525	IBOR			7,613889
30	IRS	1	JPY	109990	ONWAAR	0,341667	JPY_3M	3	7	1,861111
30	IRS	-1	JPY	109990	WAAR	0,341667	0,02	6	4	1,861111
31	IRS	-1	USD	1214	ONWAAR	0,930556	USD_3M	3	7	2,455556
31	IRS	1	USD	1214	WAAR	0,930556	0,048	3	7	2,455556
32	IRS	-1	JPY	86571	WAAR	1,691667	0,05	1	11	2,541667
32	IRS	1	JPY	86571	ONWAAR	1,691667	JPY_1M	1	11	2,541667
33	IRS	-1	USD	1634	WAAR	0,930556	0,014	1	5	1,269444
33	IRS	1	USD	1634	ONWAAR	0,930556	USD_1M	1	5	1,269444
34	IRS	-1	USD	2149	ONWAAR	0,425	USD_6M	6	22	11,07778
34	IRS	1	USD	2149	WAAR	0,425	0,016	6	22	11,07778
35	IRS	-1	USD	2489	ONWAAR	0,597222	USD_3M	3	5	1,611111
35	IRS	1	USD	2489	WAAR	0,597222	0,046	3	5	1,611111
36	IRS	-1	JPY	210504	ONWAAR	1,611111	JPY_1M	1	20	3,216667
36	IRS	1	JPY	210504	WAAR	1,611111	0,014	1	20	3,216667
37	IRS	1	JPY	192237	ONWAAR	1,691667	JPY_3M	3	20	6,513889
37	IRS	-1	JPY	192237	WAAR	1,691667	0,028	3	20	6,513889

Tradeld	ProductType	PayOrReceive	Ccy	Notional	IsFixed	StartDate	Coupon	CouponFrequency	NumberOfCoupons	Maturity
38	IRS	-1	JPY	185564	ONWAAR	1,438889	JPY_6M		6	9 5,497222
38	IRS	1	JPY	185564	WAAR	1,438889	0,017		6	9 5,497222
39	IRS	-1	USD	1045	WAAR	0,511111	0,029		3	24 6,344444
39	IRS	1	USD	1045	ONWAAR	0,511111	USD_3M		3	24 6,344444
40	IRS	-1	USD	1611	WAAR	1,013889	0,04		6	12 6,6
40	IRS	1	USD	1611	ONWAAR	1,013889	USD_6M		6	12 6,6
41	IRS	-1	USD	1163	WAAR	1,438889	0,04		12	14 14,63056
41	IRS	1	USD	1163	ONWAAR	1,438889	USD_12M		12	14 14,63056
42	IRS	-1	USD	1567	WAAR	0,425	0,036		6	21 10,57222
42	IRS	1	USD	1567	ONWAAR	0,425	USD_3M		3	41 10,57222
43	IRS	1	USD	2272	ONWAAR	1,525	USD_6M		6	27 14,71667
43	IRS	-1	USD	2272	WAAR	1,525	0,025		12	14 14,71667
44	IRS	1	USD	1217	WAAR	1,611111	0,032		3	21 6,686111
44	IRS	-1	USD	1217	ONWAAR	1,611111	USD_3M		3	21 6,686111
45	IRS	1	JPY	60121	ONWAAR	1,355556	JPY_3M		3	11 3,888889
45	IRS	-1	JPY	60121	WAAR	1,355556	0,055		6	6 3,888889
46	IRS	-1	USD	1982	WAAR	1,691667	0,007		6	8 5,244444
46	IRS	1	USD	1982	ONWAAR	1,691667	USD_3M		3	15 5,244444
47	IRS	-1	USD	744	ONWAAR	0,172222	USD_6M		6	9 4,230556
47	IRS	1	USD	744	WAAR	0,172222	0,045		6	9 4,230556
48	IRS	1	USD	2002	ONWAAR	1,611111	USD_3M		3	20 6,430556
48	IRS	-1	USD	2002	WAAR	1,611111	0,041		3	20 6,430556
49	IRS	1	USD	1687	WAAR	0,172222	0,027		6	14 6,763889
49	IRS	-1	USD	1687	ONWAAR	0,172222	USD_6M		6	14 6,763889
50	FX	-1	USD	1832	WAAR					2,541667
50	FX	1	JPY	192360	WAAR					2,541667
51	FX	-1	JPY	169451	WAAR					5,583333
51	FX	1	USD	1613,819	WAAR					5,583333
52	FX	-1	USD	870	WAAR					9,641667
52	FX	1	JPY	91350	WAAR					9,641667
53	FX	1	JPY	201390	WAAR					2,541667
53	FX	-1	USD	1918	WAAR					2,541667
54	FX	-1	JPY	85705	WAAR					4,569444
54	FX	1	USD	816,2381	WAAR					4,569444
55	FX	-1	USD	1158	WAAR					1,525
55	FX	1	JPY	121590	WAAR					1,525
56	FX	-1	USD	1895	WAAR					4,569444
56	FX	1	JPY	198975	WAAR					4,569444
57	FX	1	JPY	155295	WAAR					7,102778
57	FX	-1	USD	1479	WAAR					7,102778
58	FX	-1	USD	2439	WAAR					7,613889
58	FX	1	JPY	256095	WAAR					7,613889
59	FX	-1	USD	1457	WAAR					5,072222
59	FX	1	JPY	152985	WAAR					5,072222
60	FX	-1	JPY	192469	WAAR					3,555556
60	FX	1	USD	1833,038	WAAR					3,555556
61	FX	-1	USD	2231	WAAR					4,569444
61	FX	1	JPY	234255	WAAR					4,569444
62	FX	-1	JPY	233123	WAAR					2,030556
62	FX	1	USD	2220,219	WAAR					2,030556
63	FX	1	JPY	229005	WAAR					10,14722
63	FX	-1	USD	2181	WAAR					10,14722
64	FX	1	USD	819,6857	WAAR					7,613889
64	FX	-1	JPY	86067	WAAR					7,613889
65	FX	-1	JPY	218258	WAAR					1,525
65	FX	1	USD	2078,648	WAAR					1,525
66	FX	1	USD	1955,229	WAAR					7,613889
66	FX	-1	JPY	205299	WAAR					7,613889
67	FX	1	USD	625,3714	WAAR					4,058333
67	FX	-1	JPY	65664	WAAR					4,058333
68	FX	1	USD	1690,495	WAAR					2,030556
68	FX	-1	JPY	177502	WAAR					2,030556
69	FX	1	JPY	194460	WAAR					4,569444
69	FX	-1	USD	1852	WAAR					4,569444
70	FX	-1	USD	1091	WAAR					9,641667
70	FX	1	JPY	114555	WAAR					9,641667
71	FX	-1	USD	2315	WAAR					8,627778
71	FX	1	JPY	243075	WAAR					8,627778
72	FX	-1	JPY	159930	WAAR					1,013889
72	FX	1	USD	1523,143	WAAR					1,013889
73	FX	-1	JPY	121809	WAAR					1,013889
73	FX	1	USD	1160,086	WAAR					1,013889
74	FX	1	JPY	88935	WAAR					7,613889
74	FX	-1	USD	847	WAAR					7,613889
75	FX	-1	JPY	139698	WAAR					7,102778
75	FX	1	USD	1330,457	WAAR					7,102778
76	FX	1	USD	2094,029	WAAR					0,511111
76	FX	-1	JPY	219873	WAAR					0,511111
77	FX	1	USD	2309,581	WAAR					8,627778
77	FX	-1	JPY	242506	WAAR					8,627778

Tradeld	ProductType	PayOrReceive	Ccy	Notional	IsFixed	StartDate	Coupon	CouponFrequency	NumberOfCoupons	Maturity
78	FX	1	USD	1728,952	WAAR					8,627778
78	FX	-1	JPY	181540	WAAR					8,627778
79	FX	1	JPY	192570	WAAR					8,116667
79	FX	-1	USD	1834	WAAR					8,116667
80	XCS	1	USD	1742	WAAR	1,269444	0,05	3	16	5,072222
80	XCS	-1	JPY	182910	ONWAAR	1,269444	JPY_3M	3	16	5,072222
81	XCS	-1	JPY	79631	WAAR	0,172222	0,033	12	15	14,37778
81	XCS	1	USD	758,3905	ONWAAR	0,172222	USD_12M	12	15	14,37778
82	XCS	-1	USD	2283	ONWAAR	1,1	USD_6M	6	20	10,74444
82	XCS	1	JPY	239715	WAAR	1,1	0,006	6	20	10,74444
83	XCS	-1	USD	1296	ONWAAR	0,341667	USD_12M	12	6	5,413889
83	XCS	1	JPY	136080	WAAR	0,341667	0,043	6	11	5,413889
84	XCS	-1	JPY	126486	ONWAAR	1,525	JPY_6M	6	7	4,569444
84	XCS	1	USD	1204,629	WAAR	1,525	0,044	3	13	4,569444
85	XCS	-1	JPY	255527	WAAR	1,525	0,053	3	15	5,072222
85	XCS	1	USD	2433,59	ONWAAR	1,525	USD_3M	3	15	5,072222
86	XCS	-1	USD	635	WAAR	0,172222	0,052	3	25	6,261111
86	XCS	1	JPY	66675	ONWAAR	0,172222	JPY_3M	3	25	6,261111
87	XCS	1	USD	2294,257	ONWAAR	1,438889	USD_3M	3	13	4,483333
87	XCS	-1	JPY	240897	WAAR	1,438889	0,011	3	13	4,483333
88	XCS	-1	USD	738	WAAR	1,355556	0,041	3	4	2,116667
88	XCS	1	JPY	77490	ONWAAR	1,355556	JPY_3M	3	4	2,116667
89	XCS	-1	JPY	93759	ONWAAR	0,425	JPY_12M	12	10	9,555556
89	XCS	1	USD	892,9429	WAAR	0,425	0,054	12	10	9,555556
90	XCS	-1	USD	528	ONWAAR	0,425	USD_3M	3	9	2,455556
90	XCS	1	JPY	55440	WAAR	0,425	0,008	3	9	2,455556
91	XCS	1	JPY	120540	ONWAAR	0,597222	JPY_12M	12	8	7,7
91	XCS	-1	USD	1148	WAAR	0,597222	0,007	12	8	7,7
92	XCS	1	JPY	168210	ONWAAR	0,172222	JPY_6M	6	5	2,202778
92	XCS	-1	USD	1602	WAAR	0,172222	0,008	6	5	2,202778
93	XCS	1	USD	1162,971	ONWAAR	1,269444	USD_1M	1	8	1,861111
93	XCS	-1	JPY	122112	WAAR	1,269444	0,014	1	8	1,861111
94	XCS	-1	USD	1273	WAAR	1,186111	0,042	3	15	4,733333
94	XCS	1	JPY	133665	ONWAAR	1,186111	JPY_3M	3	15	4,733333
95	XCS	1	JPY	110040	WAAR	1,013889	0,034	6	17	9,130556
95	XCS	-1	USD	1048	ONWAAR	1,013889	USD_6M	6	17	9,130556
96	XCS	1	USD	531,0857	ONWAAR	0,597222	USD_3M	3	29	7,7
96	XCS	-1	JPY	55764	WAAR	0,597222	0,05	6	15	7,7
97	XCS	1	JPY	240975	WAAR	1,611111	0,014	12	8	8,713889
97	XCS	-1	USD	2295	ONWAAR	1,611111	USD_12M	12	8	8,713889
98	XCS	1	JPY	252735	WAAR	1,186111	0,051	3	25	7,275
98	XCS	-1	USD	2407	ONWAAR	1,186111	USD_3M	3	25	7,275
99	XCS	-1	JPY	252329	WAAR	0,844444	0,049	6	10	5,413889
99	XCS	1	USD	2403,133	ONWAAR	0,844444	USD_3M	3	19	5,413889

E.2. PORTFOLIO 50 DERIVATIVES 3D

TradeId	ProductType	PayOrReceive	Ccy	Notional	IsFixed	StartDate	Coupon	CouponFrequency	NumberOfCoupons	Maturity
0	FRA	-1	USD	658	WAAR	0,341667	0,022			7,444444
0	FRA	1	USD	658	ONWAAR	0,341667	IBOR			7,444444
1	FRA	-1	USD	1936	ONWAAR	1,1	IBOR			8,713889
1	FRA	1	USD	1936	WAAR	1,1	0,008			8,713889
2	FRA	-1	JPY	229981	ONWAAR	1,186111	IBOR			2,705556
2	FRA	1	JPY	229981	WAAR	1,186111	0,054			2,705556
3	FRA	1	USD	1479	WAAR	1,013889	0,05			8,116667
3	FRA	-1	USD	1479	ONWAAR	1,013889	IBOR			8,116667
4	FRA	1	USD	2455	WAAR	1,691667	0,007			4,733333
4	FRA	-1	USD	2455	ONWAAR	1,691667	IBOR			4,733333
5	FRA	1	JPY	196459	ONWAAR	0,675	IBOR			7,275
5	FRA	-1	JPY	196459	WAAR	0,675	0,043			7,275
6	FRA	1	USD	689	ONWAAR	0,675	IBOR			7,275
6	FRA	-1	USD	689	WAAR	0,675	0,036			7,275
7	FRA	-1	JPY	169941	WAAR	1,611111	0,034			2,627778
7	FRA	1	JPY	169941	ONWAAR	1,611111	IBOR			2,627778
8	FRA	1	USD	1948	ONWAAR	0,172222	IBOR			8,791667
8	FRA	-1	USD	1948	WAAR	0,172222	0,034			8,791667
9	FRA	1	JPY	211128	ONWAAR	1,438889	IBOR			8,541667
9	FRA	-1	JPY	211128	WAAR	1,438889	0,022			8,541667
10	FRA	1	JPY	124135	WAAR	1,525	0,053			7,613889
10	FRA	-1	JPY	124135	ONWAAR	1,525	IBOR			7,613889
11	FRA	1	JPY	244011	ONWAAR	0,341667	IBOR			2,372222
11	FRA	-1	JPY	244011	WAAR	0,341667	0,043			2,372222
12	FRA	-1	JPY	110295	WAAR	0,761111	0,021			9,894444
12	FRA	1	JPY	110295	ONWAAR	0,761111	IBOR			9,894444
13	IRS	-1	USD	1567	WAAR	0,425	0,036	6	21	10,57222
13	IRS	1	USD	1567	ONWAAR	0,425	USD_3M	3	3	10,57222
14	IRS	1	USD	2272	ONWAAR	1,525	USD_6M	6	27	14,71667
14	IRS	-1	USD	2272	WAAR	1,525	0,025	12	14	14,71667
15	IRS	1	USD	1217	WAAR	1,611111	0,032	3	21	6,686111
15	IRS	-1	USD	1217	ONWAAR	1,611111	USD_3M	3	21	6,686111
16	IRS	1	JPY	60121	ONWAAR	1,355556	JPY_3M	3	11	3,888889
16	IRS	-1	JPY	60121	WAAR	1,355556	0,055	6	6	3,888889
17	IRS	-1	USD	1982	WAAR	1,691667	0,007	6	8	5,244444
17	IRS	1	USD	1982	ONWAAR	1,691667	USD_3M	3	15	5,244444
18	IRS	-1	USD	744	ONWAAR	0,172222	USD_6M	6	9	4,230556
18	IRS	1	USD	744	WAAR	0,172222	0,045	6	9	4,230556
19	IRS	1	USD	2002	ONWAAR	1,611111	USD_3M	3	20	6,430556
19	IRS	-1	USD	2002	WAAR	1,611111	0,041	3	20	6,430556
20	IRS	1	USD	1687	WAAR	0,172222	0,027	6	14	6,763889
20	IRS	-1	USD	1687	ONWAAR	0,172222	USD_6M	6	14	6,763889
21	IRS	-1	USD	1045	WAAR	0,511111	0,029	3	24	6,344444
21	IRS	1	USD	1045	ONWAAR	0,511111	USD_3M	3	24	6,344444
22	IRS	-1	USD	1611	WAAR	1,013889	0,04	6	12	6,6
22	IRS	1	USD	1611	ONWAAR	1,013889	USD_6M	6	12	6,6
23	IRS	1	JPY	192237	ONWAAR	1,691667	JPY_3M	3	20	6,513889
23	IRS	-1	JPY	192237	WAAR	1,691667	0,028	3	20	6,513889
24	IRS	-1	JPY	185564	ONWAAR	1,438889	JPY_6M	6	9	5,497222
24	IRS	1	JPY	185564	WAAR	1,438889	0,017	6	9	5,497222
25	FX	1	USD	1955,229	WAAR					7,613889
25	FX	-1	JPY	205299	WAAR					7,613889
26	FX	1	USD	625,3714	WAAR					4,058333
26	FX	-1	JPY	65664	WAAR					4,058333
27	FX	1	USD	1690,495	WAAR					2,030556
27	FX	-1	JPY	177502	WAAR					2,030556
28	FX	1	JPY	194460	WAAR					4,569444
28	FX	-1	USD	1852	WAAR					4,569444
29	FX	-1	USD	1091	WAAR					9,641667
29	FX	1	JPY	114555	WAAR					9,641667
30	FX	-1	USD	2315	WAAR					8,627778
30	FX	1	JPY	243075	WAAR					8,627778
31	FX	-1	JPY	159930	WAAR					1,013889
31	FX	1	USD	1523,143	WAAR					1,013889
32	FX	-1	JPY	121809	WAAR					1,013889
32	FX	1	USD	1160,086	WAAR					1,013889
33	FX	1	JPY	88935	WAAR					7,613889
33	FX	-1	USD	847	WAAR					7,613889
34	FX	-1	JPY	139698	WAAR					7,102778
34	FX	1	USD	1330,457	WAAR					7,102778
35	FX	1	USD	2094,029	WAAR					0,511111
35	FX	-1	JPY	219873	WAAR					0,511111
36	FX	1	USD	2309,581	WAAR					8,627778
36	FX	-1	JPY	242506	WAAR					8,627778
37	FX	1	USD	1728,952	WAAR					8,627778
37	FX	-1	JPY	181540	WAAR					8,627778

Tradeld	ProductType	PayOrReceive	Ccy	Notional	IsFixed	StartDate	Coupon	CouponFrequency	NumberOfCoupons	Maturity
38	XCS	-1	USD	2283	ONWAAR	1,1	USD_6M	6	20	10,74444
38	XCS	1	JPY	239715	WAAR	1,1	0,006	6	20	10,74444
39	XCS	-1	USD	1296	ONWAAR	0,341667	USD_12M	12	6	5,413889
39	XCS	1	JPY	136080	WAAR	0,341667	0,043	6	11	5,413889
40	XCS	-1	JPY	126486	ONWAAR	1,525	JPY_6M	6	7	4,569444
40	XCS	1	USD	1204,629	WAAR	1,525	0,044	3	13	4,569444
41	XCS	-1	JPY	255527	WAAR	1,525	0,053	3	15	5,072222
41	XCS	1	USD	2433,59	ONWAAR	1,525	USD_3M	3	15	5,072222
42	XCS	-1	USD	635	WAAR	0,172222	0,052	3	25	6,261111
42	XCS	1	JPY	66675	ONWAAR	0,172222	JPY_3M	3	25	6,261111
43	XCS	1	USD	2294,257	ONWAAR	1,438889	USD_3M	3	13	4,483333
43	XCS	-1	JPY	240897	WAAR	1,438889	0,011	3	13	4,483333
44	XCS	-1	USD	738	WAAR	1,355556	0,041	3	4	2,116667
44	XCS	1	JPY	77490	ONWAAR	1,355556	JPY_3M	3	4	2,116667
45	XCS	-1	JPY	93759	ONWAAR	0,425	JPY_12M	12	10	9,555556
45	XCS	1	USD	892,9429	WAAR	0,425	0,054	12	10	9,555556
46	XCS	-1	USD	528	ONWAAR	0,425	USD_3M	3	9	2,455556
46	XCS	1	JPY	55440	WAAR	0,425	0,008	3	9	2,455556
47	XCS	1	JPY	120540	ONWAAR	0,597222	JPY_12M	12	8	7,7
47	XCS	-1	USD	1148	WAAR	0,597222	0,007	12	8	7,7
48	XCS	1	JPY	168210	ONWAAR	0,172222	JPY_6M	6	5	2,202778
48	XCS	-1	USD	1602	WAAR	0,172222	0,008	6	5	2,202778
49	XCS	1	USD	1162,971	ONWAAR	1,269444	USD_1M	1	8	1,861111
49	XCS	-1	JPY	122112	WAAR	1,269444	0,014	1	8	1,861111

E.3. PORTFOLIO 25 DERIVATIVES 3D

TradeId	ProductType	PayOrReceive	Ccy	Notional	IsFixed	StartDate	Coupon	CouponFrequency	NumberOfCoupons	Maturity	
0	FRA	1	USD	1479	WAAR	1,013889	0,05			8,116667	
0	FRA	-1	USD	1479	ONWAAR	1,013889	IBOR			8,116667	
1	FRA	1	USD	2455	WAAR	1,691667	0,007			4,733333	
1	FRA	-1	USD	2455	ONWAAR	1,691667	IBOR			4,733333	
2	FRA	1	JPY	196459	ONWAAR	0,675	IBOR			7,275	
2	FRA	-1	JPY	196459	WAAR	0,675	0,043			7,275	
3	FRA	-1	JPY	110295	WAAR	0,761111	0,021			9,894444	
3	FRA	1	JPY	110295	ONWAAR	0,761111	IBOR			9,894444	
4	FRA	-1	JPY	169941	WAAR	1,611111	0,034			2,627778	
4	FRA	1	JPY	169941	ONWAAR	1,611111	IBOR			2,627778	
5	FRA	1	USD	1948	ONWAAR	0,172222	IBOR			8,791667	
5	FRA	-1	USD	1948	WAAR	0,172222	0,034			8,791667	
6	FRA	-1	JPY	245248	WAAR	0,761111	0,018			1,777778	
6	FRA	1	JPY	245248	ONWAAR	0,761111	IBOR			1,777778	
7	IRS	-1	USD	1634	WAAR	0,930556	0,014		1	5	1,269444
7	IRS	1	USD	1634	ONWAAR	0,930556	USD_1M		1	5	1,269444
8	IRS	-1	USD	2149	ONWAAR	0,425	USD_6M		6	22	11,07778
8	IRS	1	USD	2149	WAAR	0,425	0,016		6	22	11,07778
9	IRS	-1	USD	2489	ONWAAR	0,597222	USD_3M		3	5	1,611111
9	IRS	1	USD	2489	WAAR	0,597222	0,046		3	5	1,611111
10	IRS	-1	JPY	210504	ONWAAR	1,611111	JPY_1M		1	20	3,216667
10	IRS	1	JPY	210504	WAAR	1,611111	0,014		1	20	3,216667
11	IRS	1	JPY	192237	ONWAAR	1,691667	JPY_3M		3	20	6,513889
11	IRS	-1	JPY	192237	WAAR	1,691667	0,028		3	20	6,513889
12	IRS	-1	USD	1567	WAAR	0,425	0,036		6	21	10,57222
12	IRS	1	USD	1567	ONWAAR	0,425	USD_3M		3	41	10,57222
13	IRS	-1	USD	1045	WAAR	0,511111	0,029		3	24	6,344444
13	IRS	1	USD	1045	ONWAAR	0,511111	USD_3M		3	24	6,344444
14	FX	-1	USD	870	WAAR						9,641667
14	FX	1	JPY	91350	WAAR						9,641667
15	FX	1	JPY	201390	WAAR						2,541667
15	FX	-1	USD	1918	WAAR						2,541667
16	FX	-1	JPY	85705	WAAR						4,569444
16	FX	1	USD	816,2381	WAAR						4,569444
17	FX	1	JPY	229005	WAAR						10,14722
17	FX	-1	USD	2181	WAAR						10,14722
18	FX	-1	USD	1895	WAAR						4,569444
18	FX	1	JPY	198975	WAAR						4,569444
19	FX	1	JPY	155295	WAAR						7,102778
19	FX	-1	USD	1479	WAAR						7,102778
20	FX	-1	USD	2439	WAAR						7,613889
20	FX	1	JPY	256095	WAAR						7,613889
21	XCS	-1	USD	1296	ONWAAR	0,341667	USD_12M		12	6	5,413889
21	XCS	1	JPY	136080	WAAR	0,341667	0,043		6	11	5,413889
22	XCS	-1	JPY	126486	ONWAAR	1,525	JPY_6M		6	7	4,569444
22	XCS	1	USD	1204,629	WAAR	1,525	0,044		3	13	4,569444
23	XCS	-1	JPY	255527	WAAR	1,525	0,053		3	15	5,072222
23	XCS	1	USD	2433,59	ONWAAR	1,525	USD_3M		3	15	5,072222
24	XCS	-1	USD	635	WAAR	0,172222	0,052		3	25	6,261111
24	XCS	1	JPY	66675	ONWAAR	0,172222	JPY_3M		3	25	6,261111

E.4. Two XCS 5D

TradeId	ProductType	PayOrReceive	Ccy	Notional	IsFixed	StartDate	Coupon	CouponFrequency	NumberOfCoupons	Maturity
0	XCS	1	USD	1742	WAAR	0,23424	0,05	3	16	5,754
0	XCS	-1	JPY	182910	ONWAAR	0,23424	JPY_3M	3	16	5,754
1	XCS	-1	USD	1000	WAAR	0,031	0,02	3	20	8,1
1	XCS	1	EUR	740,74	ONWAAR	0,031	EUR_3M	3	20	8,1

E.5. PORTFOLIO 100 DERIVATIVES 5D

TradedId	ProductType	PayOrReceive	Ccy	Notional	IsFixed	StartDate	Coupon	CouponFrequency	NumberOfCoupons	Maturity
0	FRA	-1	JPY	198682	ONWAAR	0,341667	IBOR			5,919444
0	FRA	1	JPY	198682	WAAR	0,341667		0,012		5,919444
1	FRA	-1	USD	1591	WAAR	0,425		0,033		8,033333
1	FRA	1	USD	1591	ONWAAR	0,425	IBOR			8,033333
2	FRA	1	JPY	148287	WAAR	0,425		0,016		6,005556
2	FRA	-1	JPY	148287	ONWAAR	0,425	IBOR			6,005556
3	FRA	-1	JPY	245248	WAAR	0,761111		0,018		1,777778
3	FRA	1	JPY	245248	ONWAAR	0,761111	IBOR			1,777778
4	FRA	1	JPY	101599	WAAR	0,425		0,023		4,483333
4	FRA	-1	JPY	101599	ONWAAR	0,425	IBOR			4,483333
5	FRA	-1	USD	811	ONWAAR	1,186111		IBOR		6,763889
5	FRA	1	USD	811	WAAR	1,186111		0,048		6,763889
6	FRA	1	JPY	94341	ONWAAR	0,341667	IBOR			0,844444
6	FRA	-1	JPY	94341	WAAR	0,341667		0,028		0,844444
7	FRA	1	JPY	75371	WAAR	1,438889		0,026		4,988889
7	FRA	-1	JPY	75371	ONWAAR	1,438889	IBOR			4,988889
8	FRA	1	USD	1791	WAAR	1,013889		0,043		1,525
8	FRA	-1	USD	1791	ONWAAR	1,013889	IBOR			1,525
9	FRA	1	USD	2459	WAAR	0,675		0,048		5,244444
9	FRA	-1	USD	2459	ONWAAR	0,675	IBOR			5,244444
10	FRA	1	JPY	105362	ONWAAR	1,438889	IBOR			4,483333
10	FRA	-1	JPY	105362	WAAR	1,438889		0,031		4,483333
11	FRA	-1	USD	658	WAAR	0,341667		0,022		7,444444
11	FRA	1	USD	658	ONWAAR	0,341667	IBOR			7,444444
12	FRA	-1	USD	1936	ONWAAR	1,1	IBOR			8,713889
12	FRA	1	USD	1936	WAAR	1,1		0,008		8,713889
13	FRA	-1	JPY	229981	ONWAAR	1,186111	IBOR			2,705556
13	FRA	1	JPY	229981	WAAR	1,186111		0,054		2,705556
14	FRA	1	USD	1479	WAAR	1,013889		0,05		8,116667
14	FRA	-1	USD	1479	ONWAAR	1,013889	IBOR			8,116667
15	FRA	1	USD	2455	WAAR	1,691667		0,007		4,733333
15	FRA	-1	USD	2455	ONWAAR	1,691667	IBOR			4,733333
16	FRA	-1	USD	2160	ONWAAR	1,438889	IBOR			7,019444
16	FRA	1	USD	2160	WAAR	1,438889		0,033		7,019444
17	FRA	-1	JPY	68563	ONWAAR	0,511111	IBOR			5,072222
17	FRA	1	JPY	68563	WAAR	0,511111		0,023		5,072222
18	FRA	1	JPY	124135	WAAR	1,525		0,053		7,613889
18	FRA	-1	JPY	124135	ONWAAR	1,525	IBOR			7,613889
19	FRA	-1	USD	1910	WAAR	1,525		0,054		6,6
19	FRA	1	USD	1910	ONWAAR	1,525	IBOR			6,6
20	FRA	1	JPY	244011	ONWAAR	0,341667	IBOR			2,372222
20	FRA	-1	JPY	244011	WAAR	0,341667		0,043		2,372222
21	FRA	1	EUR	1570	ONWAAR	1,238	IBOR			8,3456
21	FRA	-1	EUR	1570	WAAR	1,238		0,026		8,3456
22	FRA	1	EUR	754	ONWAAR	0,678	IBOR			0,844444
22	FRA	-1	EUR	754	WAAR	0,678		0,031		0,844444
23	FRA	1	EUR	987	ONWAAR	0,551	IBOR			8,3321
23	FRA	-1	EUR	987	WAAR	0,551		0,044		8,3321
24	FRA	1	EUR	832	ONWAAR	1,134	IBOR			4,6645
24	FRA	-1	EUR	832	WAAR	1,134		0,036		4,6645
25	FRA	1	EUR	1234	ONWAAR	1,005	IBOR			7,8843
25	FRA	-1	EUR	1234	WAAR	1,005		0,05		7,8843
26	FRA	1	EUR	728	ONWAAR	0,355	IBOR			3,441
26	FRA	-1	EUR	728	WAAR	0,355		0,048		3,441
27	FRA	1	EUR	505	ONWAAR	1,112	IBOR			4,496
27	FRA	-1	EUR	505	WAAR	1,112		0,026		4,496
28	IRS	-1	JPY	86571	WAAR	1,691667		0,05	1	11 2,541667
28	IRS	1	JPY	86571	ONWAAR	1,691667	JPY_1M		1	11 2,541667
29	IRS	-1	USD	1634	WAAR	0,930556		0,014	1	5 1,269444
29	IRS	1	USD	1634	ONWAAR	0,930556	USD_1M		1	5 1,269444
30	IRS	-1	USD	2149	ONWAAR	0,425	USD_6M		6	22 11,07778
30	IRS	1	USD	2149	WAAR	0,425		0,016	6	22 11,07778
31	IRS	-1	USD	2489	ONWAAR	0,597222	USD_3M		3	5 1,611111
31	IRS	1	USD	2489	WAAR	0,597222		0,046	3	5 1,611111
32	IRS	-1	JPY	210504	ONWAAR	1,611111	JPY_1M		1	20 3,216667
32	IRS	1	JPY	210504	WAAR	1,611111		0,014	1	20 3,216667
33	IRS	1	JPY	192237	ONWAAR	1,691667	JPY_3M		3	20 6,513889
33	IRS	-1	JPY	192237	WAAR	1,691667		0,028	3	20 6,513889
34	IRS	-1	JPY	185564	ONWAAR	1,438889	JPY_6M		6	9 0,56768
34	IRS	1	JPY	185564	WAAR	1,438889		0,017	6	9 0,56768
35	IRS	-1	USD	1045	WAAR	0,511111		0,029	3	24 6,344444
35	IRS	1	USD	1045	ONWAAR	0,511111	USD_3M		3	24 6,344444
36	IRS	-1	USD	1611	WAAR	1,013889		0,04	6	12 6,6
36	IRS	1	USD	1611	ONWAAR	1,013889	USD_6M		6	12 6,6
37	IRS	-1	USD	1214	ONWAAR	0,930556	USD_3M		3	7 2,455556
37	IRS	1	USD	1214	WAAR	0,930556		0,048	3	7 2,455556

Tradeld	ProductType	PayOrReceive	Ccy	Notional	IsFixed	StartDate	Coupon	CouponFrequency	NumberOfCoupons	Maturity
38	IRS	-1	USD	1567	WAAR	0,425	0,036		6	21 10,57222
38	IRS	1	USD	1567	ONWAAR	0,425	USD_3M		3	41 10,57222
39	IRS	1	USD	2272	ONWAAR	1,525	USD_6M		6	27 14,71667
39	IRS	-1	USD	2272	WAAR	1,525	0,025		12	14 14,71667
40	IRS	1	USD	1217	WAAR	1,611111	0,032		3	21 6,686111
40	IRS	-1	USD	1217	ONWAAR	1,611111	USD_3M		3	21 6,686111
41	IRS	1	JPY	60121	ONWAAR	1,355556	JPY_3M		3	11 3,888889
41	IRS	-1	JPY	60121	WAAR	1,355556	0,055		6	6 3,888889
42	IRS	1	EUR	843	ONWAAR	1,7743	EUR_3M		3	15 5,23453
42	IRS	-1	EUR	843	WAAR	1,7743	0,045		6	8 5,23453
43	IRS	1	EUR	612	ONWAAR	0,67775	EUR_3M		3	17 4,8932
43	IRS	-1	EUR	612	WAAR	0,67775	0,045		6	9 4,8932
44	IRS	-1	EUR	1004	ONWAAR	0,96543	EUR_6M		6	23 6,5432
44	IRS	1	EUR	1004	WAAR	0,96543	0,045		12	12 6,5432
45	IRS	1	EUR	964	ONWAAR	1,5543	EUR_12M		12	7 3,9555
45	IRS	-1	EUR	964	WAAR	1,5543	0,045		6	13 3,9555
46	IRS	-1	EUR	2134	ONWAAR	1,2355	EUR_1M		1	13 8,5434
46	IRS	1	EUR	2134	WAAR	1,2355	0,045		1	13 8,5434
47	IRS	-1	EUR	786	ONWAAR	0,75443	EUR_3M		3	8 0,9865
47	IRS	1	EUR	786	WAAR	0,75443	0,045		3	8 0,9865
48	IRS	1	EUR	659	ONWAAR	1,6566	EUR_3M		3	15 7,5456
48	IRS	-1	EUR	659	WAAR	1,6566	0,045		6	8 7,5456
49	IRS	1	EUR	1890	ONWAAR	0,4323	EUR_1M		1	20 13,3454
49	IRS	-1	EUR	1890	WAAR	0,4323	0,045		1	20 13,3454
50	IRS	-1	EUR	755	ONWAAR	1,1453	EUR_6M		6	17 9,3432
50	IRS	1	EUR	755	WAAR	1,1453	0,045		6	17 9,3432
51	FX	-1	USD	1158	WAAR					1,525
51	FX	1	JPY	121590	WAAR					1,525
52	FX	-1	USD	1895	WAAR					4,569444
52	FX	1	JPY	198975	WAAR					4,569444
53	FX	1	JPY	155295	WAAR					7,102778
53	FX	-1	USD	1479	WAAR					7,102778
54	FX	-1	USD	2439	WAAR					7,613889
54	FX	1	JPY	256095	WAAR					7,613889
55	FX	-1	USD	1457	WAAR					5,072222
55	FX	1	JPY	152985	WAAR					5,072222
56	FX	-1	JPY	192469	WAAR					3,555556
56	FX	1	USD	1833,038	WAAR					3,555556
57	FX	-1	USD	2231	WAAR					0,564354
57	FX	1	JPY	234255	WAAR					0,564354
58	FX	-1	JPY	233123	WAAR					2,030556
58	FX	1	USD	2220,219	WAAR					2,030556
59	FX	1	JPY	229005	WAAR					10,14722
59	FX	-1	USD	2181	WAAR					10,14722
60	FX	1	USD	819,6857	WAAR					7,613889
60	FX	-1	JPY	86067	WAAR					7,613889
61	FX	-1	JPY	218258	WAAR					1,525
61	FX	1	USD	2078,648	WAAR					1,525
62	FX	1	USD	1955,229	WAAR					7,613889
62	FX	-1	JPY	205299	WAAR					7,613889
63	FX	1	USD	625,3714	WAAR					4,058333
63	FX	-1	JPY	65664	WAAR					4,058333
64	FX	1	USD	1690,495	WAAR					2,030556
64	FX	-1	JPY	177502	WAAR					2,030556
65	FX	1	JPY	194460	WAAR					4,569444
65	FX	-1	USD	1852	WAAR					4,569444
66	FX	-1	USD	1091	WAAR					9,641667
66	FX	1	JPY	114555	WAAR					9,641667
67	FX	-1	USD	2315	WAAR					8,627778
67	FX	1	JPY	243075	WAAR					8,627778
68	FX	-1	JPY	159930	WAAR					1,013889
68	FX	1	USD	1523,143	WAAR					1,013889
69	FX	-1	JPY	121809	WAAR					1,013889
69	FX	1	USD	1160,086	WAAR					1,013889
70	FX	1	JPY	88935	WAAR					7,613889
70	FX	-1	USD	847	WAAR					7,613889
71	FX	-1	JPY	139698	WAAR					7,102778
71	FX	1	USD	1330,457	WAAR					7,102778
72	FX	-1	EUR	1130	WAAR					8,095432
72	FX	1	USD	1525,5	WAAR					8,095432
73	FX	-1	USD	1230	WAAR					6,668454
73	FX	1	EUR	911,111	WAAR					6,668454
74	FX	-1	JPY	180545	WAAR					9,754562
74	FX	1	EUR	1273,686	WAAR					9,754562
75	FX	-1	EUR	790	WAAR					6,356744
75	FX	1	USD	1066,5	WAAR					6,356744
76	FX	-1	EUR	1354	WAAR					4,597553
76	FX	1	JPY	191929,5	WAAR					4,597553
77	FX	-1	USD	689	WAAR					10,87653
77	FX	1	EUR	510,37	WAAR					10,87653

Tradeld	ProductType	PayOrReceive	Ccy	Notional	IsFixed	StartDate	Coupon	CouponFrequency	NumberOfCoupons	Maturity
78	FX	-1	JPY	115765	WAAR					5,785532
78	FX	1	EUR	816,684	WAAR					5,785532
79	FX	-1	EUR	935	WAAR					8,854324
79	FX	1	USD	1262,25	WAAR					8,854324
80	FX	-1	USD	1774	WAAR					3,754346
80	FX	1	EUR	1314,074	WAAR					3,754346
81	XCS	-1	JPY	126486	ONWAAR	1,525	JPY_6M	6	7	4,569444
81	XCS	1	USD	1204,629	WAAR	1,525	0,044	3	13	4,569444
82	XCS	-1	JPY	255527	WAAR	1,525	0,053	3	15	5,072222
82	XCS	1	USD	2433,59	ONWAAR	1,525	USD_3M	3	15	5,072222
83	XCS	-1	USD	635	WAAR	0,172222	0,052	3	25	1,06535
83	XCS	1	JPY	66675	ONWAAR	0,172222	JPY_3M	3	25	1,06535
84	XCS	1	USD	2294,257	ONWAAR	1,438889	USD_3M	3	13	4,483333
84	XCS	-1	JPY	240897	WAAR	1,438889	0,011	3	13	4,483333
85	XCS	-1	USD	738	WAAR	1,355556	0,041	3	4	2,116667
85	XCS	1	JPY	77490	ONWAAR	1,355556	JPY_3M	3	4	2,116667
86	XCS	-1	JPY	93759	ONWAAR	0,425	JPY_12M	12	10	9,555556
86	XCS	1	USD	892,9429	WAAR	0,425	0,054	12	10	9,555556
87	XCS	-1	USD	528	ONWAAR	0,425	USD_3M	3	9	2,455556
87	XCS	1	JPY	55440	WAAR	0,425	0,008	3	9	2,455556
88	XCS	1	JPY	120540	ONWAAR	0,597222	JPY_12M	12	8	7,7
88	XCS	-1	USD	1148	WAAR	0,597222	0,007	12	8	7,7
89	XCS	1	JPY	168210	ONWAAR	0,172222	JPY_6M	6	5	2,202778
89	XCS	-1	USD	1602	WAAR	0,172222	0,008	6	5	2,202778
90	XCS	1	USD	1162,971	ONWAAR	1,269444	USD_1M	1	8	1,861111
90	XCS	-1	JPY	122112	WAAR	1,269444	0,014	1	8	1,861111
91	XCS	-1	USD	1273	WAAR	1,186111	0,042	3	15	4,733333
91	XCS	1	JPY	133665	ONWAAR	1,186111	JPY_3M	3	15	4,733333
92	XCS	1	JPY	110040	WAAR	1,013889	0,034	6	17	9,130556
92	XCS	-1	USD	1048	ONWAAR	1,013889	USD_6M	6	17	9,130556
93	XCS	1	USD	531,0857	ONWAAR	0,597222	USD_3M	3	29	7,7
93	XCS	-1	JPY	55764	WAAR	0,597222	0,05	6	15	7,7
94	XCS	1	JPY	240975	WAAR	1,611111	0,014	12	8	8,713889
94	XCS	-1	USD	2295	ONWAAR	1,611111	USD_12M	12	8	8,713889
95	XCS	1	EUR	1245	WAAR	0,86543	0,034	6	15	4,765469
95	XCS	-1	USD	1680,75	ONWAAR	0,86543	USD_6M	6	15	4,765469
96	XCS	-1	EUR	978	WAAR	0,99437	0,025	12	11	5,867678
96	XCS	1	JPY	138631,5	ONWAAR	0,99437	JPY_6M	6	21	5,867678
97	XCS	1	USD	1432	WAAR	1,43532	0,053	6	13	0,786887
97	XCS	-1	EUR	1060,741	ONWAAR	1,43532	EUR_3M	3	25	0,786887
98	XCS	-1	JPY	224565	WAAR	0,24743	0,044	1	9	8,965343
98	XCS	1	EUR	1584,233	ONWAAR	0,24743	EUR_1M	1	9	8,965343
99	XCS	1	EUR	774	WAAR	1,04342	0,021	3	17	10,54321
99	XCS	-1	JPY	109714,5	ONWAAR	1,04342	JPY_3M	3	17	10,54321

E.6. PORTFOLIO 50 DERIVATIVES 5D

Tradefld	ProductType	PayOrReceive	Ccy	Notional	IsFixed	StartDate	Coupon	CouponFrequency	NumberOfCoupons	Maturity	
0	FRA	-1	JPY	245248	WAAR	0,761111	0,018			1,777778	
0	FRA	1	JPY	245248	ONWAAR	0,761111	IBOR			1,777778	
1	FRA	-1	JPY	101599	WAAR	0,425	0,023			4,483333	
1	FRA	-1	JPY	101599	ONWAAR	0,425	IBOR			4,483333	
2	FRA	-1	USD	811	ONWAAR	1,186111	IBOR			6,763889	
2	FRA	1	USD	811	WAAR	1,186111	0,048			6,763889	
3	FRA	1	JPY	94341	ONWAAR	0,341667	IBOR			0,844444	
3	FRA	-1	JPY	94341	WAAR	0,341667	0,028			0,844444	
4	FRA	1	JPY	75371	WAAR	1,438889	0,026			4,988889	
4	FRA	-1	JPY	75371	ONWAAR	1,438889	IBOR			4,988889	
5	FRA	-1	USD	1936	ONWAAR	1,1	IBOR			8,713889	
5	FRA	1	USD	1936	WAAR	1,1	0,008			8,713889	
6	FRA	-1	JPY	229981	ONWAAR	1,186111	IBOR			2,705556	
6	FRA	1	JPY	229981	WAAR	1,186111	0,054			2,705556	
7	FRA	1	USD	1479	WAAR	1,013889	0,05			8,116667	
7	FRA	-1	USD	1479	ONWAAR	1,013889	IBOR			8,116667	
8	FRA	1	EUR	754	ONWAAR	0,678	IBOR			6,05323	
8	FRA	-1	EUR	754	WAAR	0,678	0,031			6,05323	
9	FRA	1	EUR	550	ONWAAR	0,551	IBOR			8,3321	
9	FRA	-1	EUR	550	WAAR	0,551	0,044			8,3321	
10	FRA	1	EUR	832	ONWAAR	1,134	IBOR			4,6645	
10	FRA	-1	EUR	832	WAAR	1,134	0,036			4,6645	
11	FRA	1	EUR	435	ONWAAR	1,005	IBOR			7,8843	
11	FRA	-1	EUR	435	WAAR	1,005	0,05			7,8843	
12	FRA	1	EUR	728	ONWAAR	0,355	IBOR			3,441	
12	FRA	-1	EUR	728	WAAR	0,355	0,048			3,441	
13	FRA	1	EUR	505	ONWAAR	1,112	IBOR			4,496	
13	FRA	-1	EUR	505	WAAR	1,112	0,026			4,496	
14	IRS	-1	USD	2489	ONWAAR	0,597222	USD_3M	3	5	1,611111	
14	IRS	1	USD	2489	WAAR	0,597222	0,046		3	5	1,611111
15	IRS	-1	JPY	210504	ONWAAR	1,611111	JPY_1M	1	20	3,216667	
15	IRS	1	JPY	210504	WAAR	1,611111	0,014		1	20	3,216667
16	IRS	1	JPY	192237	ONWAAR	1,691667	JPY_3M	3	20	6,513889	
16	IRS	-1	JPY	192237	WAAR	1,691667	0,028		3	20	6,513889
17	IRS	-1	JPY	185564	ONWAAR	1,438889	JPY_6M	6	9	5,497222	
17	IRS	1	JPY	185564	WAAR	1,438889	0,017		6	9	5,497222
18	IRS	-1	USD	1045	WAAR	0,511111	0,029		3	24	6,344444
18	IRS	1	USD	1045	ONWAAR	0,511111	USD_3M	3	24	6,344444	
19	IRS	-1	USD	1611	WAAR	1,013889	0,04		6	12	6,6
19	IRS	1	USD	1611	ONWAAR	1,013889	USD_6M	6	12	6,6	
20	IRS	-1	USD	1214	ONWAAR	0,930556	USD_3M	3	7	2,455556	
20	IRS	1	USD	1214	WAAR	0,930556	0,048		3	7	2,455556
21	IRS	-1	USD	1567	WAAR	0,425	0,036		6	21	10,57222
21	IRS	1	USD	1567	ONWAAR	0,425	USD_3M	3	41	10,57222	
22	IRS	1	USD	2272	ONWAAR	1,525	USD_6M	6	27	14,71667	
22	IRS	-1	USD	2272	WAAR	1,525	0,025		12	14	14,71667
23	IRS	-1	EUR	432	ONWAAR	1,2355	EUR_1M	1	13	8,5434	
23	IRS	1	EUR	432	WAAR	1,2355	0,045		1	13	8,5434
24	IRS	-1	EUR	786	ONWAAR	0,75443	EUR_3M	3	8	5,73453	
24	IRS	1	EUR	786	WAAR	0,75443	0,045		3	8	5,73453
25	IRS	1	EUR	659	ONWAAR	1,6566	EUR_3M	3	15	7,5456	
25	IRS	-1	EUR	659	WAAR	1,6566	0,045		6	8	7,5456
26	IRS	1	EUR	845	ONWAAR	0,4323	EUR_1M	1	20	13,3454	
26	IRS	-1	EUR	845	WAAR	0,4323	0,045		1	20	13,3454
27	FX	-1	USD	2439	WAAR					7,613889	
27	FX	1	JPY	256095	WAAR					7,613889	
28	FX	-1	USD	1457	WAAR					5,072222	
28	FX	1	JPY	152985	WAAR					5,072222	
29	FX	-1	JPY	192469	WAAR					3,555556	
29	FX	1	USD	1833,038	WAAR					3,555556	
30	FX	-1	USD	2231	WAAR					4,569444	
30	FX	1	JPY	234255	WAAR					4,569444	
31	FX	-1	JPY	233123	WAAR					2,030556	
31	FX	1	USD	2220,219	WAAR					2,030556	
32	FX	1	JPY	229005	WAAR					10,14722	
32	FX	-1	USD	2181	WAAR					10,14722	
33	FX	1	USD	819,6857	WAAR					7,613889	
33	FX	-1	JPY	86067	WAAR					7,613889	
34	FX	-1	JPY	218258	WAAR					1,525	
34	FX	1	USD	2078,648	WAAR					1,525	
35	FX	-1	JPY	180545	WAAR					9,754562	
35	FX	1	EUR	1273,686	WAAR					9,754562	
36	FX	-1	EUR	790	WAAR					6,356744	
36	FX	1	USD	1066,5	WAAR					6,356744	
37	FX	-1	EUR	1354	WAAR					4,597553	
37	FX	1	JPY	191929,5	WAAR					4,597553	

Tradeld	ProductType	PayOrReceive	Ccy	Notional	IsFixed	StartDate	Coupon	CouponFrequency	NumberOfCoupons	Maturity
38	FX	-1	USD	689	WAAR					10,87653
38	FX	1	EUR	510,37	WAAR					10,87653
39	XCS	-1	USD	635	WAAR	0,172222	0,052	3	25	6,261111
39	XCS	1	JPY	66675	ONWAAR	0,172222	JPY_3M	3	25	6,261111
40	XCS	1	USD	2294,257	ONWAAR	1,438889	USD_3M	3	13	4,483333
40	XCS	-1	JPY	240897	WAAR	1,438889	0,011	3	13	4,483333
41	XCS	-1	USD	738	WAAR		0,041	3	4	2,116667
41	XCS	1	JPY	77490	ONWAAR	1,355556	JPY_3M	3	4	2,116667
42	XCS	-1	JPY	93759	ONWAAR	0,425	JPY_12M	12	10	9,555556
42	XCS	1	USD	892,9429	WAAR	0,425	0,054	12	10	9,555556
43	XCS	-1	USD	528	ONWAAR	0,425	USD_3M	3	9	2,455556
43	XCS	1	JPY	55440	WAAR	0,425	0,008	3	9	2,455556
44	XCS	1	JPY	120540	ONWAAR	0,597222	JPY_12M	12	8	7,7
44	XCS	-1	USD	1148	WAAR	0,597222	0,007	12	8	7,7
45	XCS	1	JPY	168210	ONWAAR	0,172222	JPY_6M	6	5	2,202778
45	XCS	-1	USD	1602	WAAR	0,172222	0,008	6	5	2,202778
46	XCS	1	JPY	110040	WAAR	1,013889	0,034	6	17	9,130556
46	XCS	-1	USD	1048	ONWAAR	1,013889	USD_6M	6	17	9,130556
47	XCS	1	USD	531,0857	ONWAAR	0,597222	USD_3M	3	29	7,7
47	XCS	-1	JPY	55764	WAAR	0,597222	0,05	6	15	7,7
48	XCS	1	JPY	240975	WAAR	1,611111	0,014	12	8	8,713889
48	XCS	-1	USD	2295	ONWAAR	1,611111	USD_12M	12	8	8,713889
49	XCS	1	EUR	1245	WAAR	0,86543	0,034	6	15	4,765469
49	XCS	-1	USD	1680,75	ONWAAR	0,86543	USD_6M	6	15	4,765469

E.7. PORTFOLIO 25 DERIVATIVES 5D

TradeId	ProductType	PayOrReceive	Ccy	Notional	IsFixed	StartDate	Coupon	CouponFrequency	NumberOfCoupons	Maturity	
0	FRA	-1	USD	2160	ONWAAR	1,438889	IBOR			7,019444	
0	FRA	1	USD	2160	WAAR	1,438889	0,033			7,019444	
1	FRA	-1	JPY	68563	ONWAAR	0,511111	IBOR			5,072222	
1	FRA	1	JPY	68563	WAAR	0,511111	0,023			5,072222	
2	FRA	1	JPY	124135	WAAR	1,525	0,053			7,613889	
2	FRA	-1	JPY	124135	ONWAAR	1,525	IBOR			7,613889	
3	FRA	-1	USD	1910	WAAR	1,525	0,054			6,6	
3	FRA	1	USD	1910	ONWAAR	1,525	IBOR			6,6	
4	FRA	1	EUR	435	ONWAAR	1,005	IBOR			7,8843	
4	FRA	-1	EUR	435	WAAR	1,005	0,05			7,8843	
5	FRA	1	EUR	728	ONWAAR	0,355	IBOR			3,441	
5	FRA	-1	EUR	728	WAAR	0,355	0,048			3,441	
6	FRA	1	EUR	505	ONWAAR	1,112	IBOR			4,496	
6	FRA	-1	EUR	505	WAAR	1,112	0,026			4,496	
7	IRS	-1	USD	2149	ONWAAR	0,425	USD_6M	6	22	11,07778	
7	IRS	1	USD	2149	WAAR	0,425	0,016		6	22	11,07778
8	IRS	-1	USD	2489	ONWAAR	0,597222	USD_3M	3	5	1,611111	
8	IRS	1	USD	2489	WAAR	0,597222	0,046		3	5	1,611111
9	IRS	-1	JPY	210504	ONWAAR	1,611111	JPY_1M	1	20	3,216667	
9	IRS	1	JPY	210504	WAAR	1,611111	0,014		1	20	3,216667
10	IRS	1	JPY	192237	ONWAAR	1,691667	JPY_3M	3	20	6,513889	
10	IRS	-1	JPY	192237	WAAR	1,691667	0,028		3	20	6,513889
11	IRS	1	EUR	964	ONWAAR	1,5543	EUR_12M	12	7	3,9555	
11	IRS	-1	EUR	964	WAAR	1,5543	0,045		6	13	3,9555
12	IRS	-1	EUR	432	ONWAAR	1,2355	EUR_1M	1	13	8,5434	
12	IRS	1	EUR	432	WAAR	1,2355	0,045		1	13	8,5434
13	FX	-1	USD	1457	WAAR					5,072222	
13	FX	1	JPY	152985	WAAR					5,072222	
14	FX	-1	JPY	192469	WAAR					3,555556	
14	FX	1	USD	1833,038	WAAR					3,555556	
15	FX	-1	USD	2231	WAAR					4,569444	
15	FX	1	JPY	234255	WAAR					4,569444	
16	FX	-1	EUR	790	WAAR					6,356744	
16	FX	1	USD	1066,5	WAAR					6,356744	
17	FX	-1	EUR	1354	WAAR					4,597553	
17	FX	1	JPY	191929,5	WAAR					4,597553	
18	FX	-1	USD	1774	WAAR					3,754346	
18	FX	1	EUR	1314,074	WAAR					3,754346	
19	XCS	-1	USD	738	WAAR	1,355556	0,041	3	4	2,116667	
19	XCS	1	JPY	77490	ONWAAR	1,355556	JPY_3M	3	4	2,116667	
20	XCS	-1	JPY	93759	ONWAAR	0,425	JPY_12M	12	10	9,555556	
20	XCS	1	USD	892,9429	WAAR	0,425	0,054	12	10	9,555556	
21	XCS	-1	USD	528	ONWAAR	0,425	USD_3M	3	9	2,455556	
21	XCS	1	JPY	55440	WAAR	0,425	0,008	3	9	2,455556	
22	XCS	1	JPY	120540	ONWAAR	0,597222	JPY_12M	12	8	7,7	
22	XCS	-1	USD	1148	WAAR	0,597222	0,007	12	8	7,7	
23	XCS	-1	EUR	978	WAAR	0,99437	0,025	12	11	5,867678	
23	XCS	1	JPY	138631,5	ONWAAR	0,99437	JPY_6M	6	21	5,867678	
24	XCS	1	USD	1432	WAAR	1,43532	0,053	6	13	12,65334	
24	XCS	-1	EUR	1060,741	ONWAAR	1,43532	EUR_3M	3	25	12,65334	