A Fast Quadrature Method for Pricing Basket Default Swaps by Means of Copulae

INAUGURAL – DISSERTATION

zur
Erlangung des Doktorgrades
der Mathematisch-Naturwissenschaftlichen Fakultät
der Universität zu Köln

vorgelegt von
Alexander Schröter
aus Engelskirchen

Köln, 2012

Berichterstatter:	Priv.-Doz. Dr. Pascal Heider Prof. Dr. Rüdiger Seydel
Tag der mündlichen Prüfung:	15.10.2012

Abstract

The computational effort of pricing an m–th to Default Swap highly depends on the size d of the underlying basket. Usually, d different default times are modeled, but in many cases the evaluation only depends on the m–th smallest default time. In this thesis we develop the distribution function $F_{\tau_{\iota_m}}$ of the m–th default time by means of copulae. With the help of this distribution we reduce the dimension of the pricing problem from d to one and break the *curse of dimensionality.* In order to ensure an efficient evaluation of $F_{\tau_{\iota_m}}$ we apply suitable recursion schemes. Independently of the chosen copula, the resulting quadrature offers a very fast convergence and a complexity of at worst $\mathcal{O}\left(N^2 d^2\right)$ by using N nodes. If the underlying m–th to Default Swap does not depend on the m–th smallest default time solely, we will develop new Monte–Carlo methods in this thesis. For this, we extend existing importance sampling methods regarding the Gaussian copula to the usage of any Archimedean copula. Besides the pricing of m–th to Default Swaps, other applications of the presented methods are pricing European max/min options or calculating sensitivities of any kind.

Kurzzusammenfassung

Der Aufwand einer m–th to Default Swap Bewertung hängt stark von der Größe d des zugrunde liegenden Korbs ab. Gewöhnlicherweise werden hierzu d verschiedene Ausfallzeitpunkte modelliert, in vielen Fällen hängt die Bewertung jedoch lediglich vom m–ten Ausfallzeitpunkt ab. In dieser Arbeit leiten wir die Verteilungsfunktion $F_{\tau_{\iota_m}}$ des m–ten Ausfallzeitpunkts mittels Copulae her. Mit Hilfe dieser Verteilung ist es möglich, die Dimension des Bewertungsproblems von d auf eins zu reduzieren und den *Fluch der Dimensionen* zu brechen. Um eine effiziente Auswertung von $F_{\tau_{\iota_m}}$ zu gewährleisten, wenden wir geeignete Rekursionsschemata an. Unabhängig von der gewählten Copula verfügt die resultierende Quadratur über eine sehr schnelle Konvergenz und eine Komplexität von höchstens $\mathcal{O}\left(N^2 d^2\right)$ bei N Stützstellen. Für Fälle, in denen der zu bewertende m–th to Default Swap nicht ausschließlich vom m–ten Ausfallzeitpunkt abhängt, entwickeln wir neue Monte–Carlo Verfahren. Hierzu erweitern wir existierende *importance sampling* Methoden bzgl. der Gauss Copula um die Möglichkeit, jegliche Archimedische Copula anzuwenden. Neben der Bewertung von m–th to Default Swaps können die vorgestellten Methoden zur Bewertung europäischer Max/Min Optionen oder zur Berechnung von Sensitivitäten verwendet werden.

Preface

In recent years, the complexity of financial markets has grown significantly. Markets were flooded by innovative financial products, whose underlying structure was conspicuously more difficult than in earlier years. Unfortunately, many practitioners neglected this fact and traded these products frequently, although they were not able to understand the mathematical background completely or rather they were not able to quote these products properly. In order to overcome such behavior it is very important to develop appropriate pricing methods, which are easily calibrated by market data.

A huge subclass of those financial products are the well–known credit derivatives. Considering the last 10 to 15 years we observed a significantly growing popularity of credit derivatives. Especially during the sub–prime crisis in 2007/2008 the function of contracts like *Mortgage Backed Securities (MBSs)*, *Collateralized Mortgage Obligations (CMOs)* or *Collateralized Debt Obligations (CDOs)* was discussed excessively. However, not only the sub–prime crisis but also the current international debt crisis within the Euro area highly illustrates the relevance of credit derivatives in practice. Nearly every day we find terms like *Credit Default Swaps (CDSs)* or *Basket Default Swaps (BDSs)* (or their corresponding premiums) in the banner headlines. At present, these contracts mostly refer to certain governments, like Portugal, Italy, Greece or Spain (also known as *PIGS*).

In this thesis we mainly focus on *Basket Default Swaps (BDSs)*, which represent a multi–dimensional generalization of the well–known *Credit Default Swaps (CDSs)*. Whereas a *CDS* protects its buyer against a certain default event of the underlying corporation, government or bond, a *BDS* protects its buyer against certain default events within a basket

$$A := \{A_1, \ldots, A_d\}, \ d \in \mathbb{N}_{\geq 2}$$

of underlying assets. The most popular example of a BDS is the widely–used *m–th to Default Swap (mBDS)*, which is considered in detail in this thesis. In brief, an *mBDS* offers protection against the m–th default within A and hence, its payoff only depends on the m–th smallest default time τ_{ι_m}.

In order to ensure a suitable pricing of *mBDSs* it is essential to model the joint distribution of default times. Due to the fact that specific baskets of assets (referring to different obligors, governments or bonds) are less frequently traded, these joint distributions usually cannot be deduced from market data. In contrast, *CDSs* on single reference entities are traded rather liquidly, which means that the marginal distributions of default times can be implied by market data reasonably well (cp. [DS03]). Hence, we follow the approach suggested by Li (cp. [Li00]), who linked the marginal distributions to a joint distribution by means of a copula function. For this, we consider the Gaussian copula and typical Archimedean copulae in this thesis.

To evaluate an *mBDS* properly we also have to establish the underlying pricing model, see [SS01, Sch03]. According to [Li00] we mainly focus on the semi–dynamic pricing model in this thesis. This implicates that the value of an *mBDS* is the expectation of its discounted payoff with respect to the joint distribution of default times. Thus, pricing an *mBDS* equals a d–dimensional quadrature problem. Considering the existing pricing methods we can distinguish between two classes of algorithms. On the one hand it is self–evident to apply a d–dimensional quadrature, i.e. the integral is approximated in each component of the joint distribution. However, these algorithms have to cope with the so–called *curse of dimensionality*, which causes an infeasible effort for an increasing dimension d. On the other hand it is possible to use probabilistic Monte–Carlo algorithms, which draw tuples of default times with respect to the joint distribution and approximate the expectation discretely, see for example [MO88, ELM03, Hof08]. Furthermore, it is possible to apply importance sampling methods in order to accelerate the convergence of these algorithms, see [JK04, CG08, SH12b].

Although the methods mentioned above offer entirely different properties, they all include a huge disadvantage, which is the consideration of d different default times. In truth only the m–th smallest default time τ_{ι_m} is relevant and influences the pricing directly. In this thesis we derive the distribution function $F_{\tau_{\iota_m}}$ of the m–th smallest default time by means of inclusion and exclusion and introduce an innovative quadrature method for pricing an *mBDS* accordingly. Due to the fact that $F_{\tau_{\iota_m}} : \mathbb{R}_{\geq 0} \mapsto [0,1]$ is a one–dimensional mapping, we are able to break the *curse of dimensionality* here. Nevertheless, the plain evaluation of distribution function $F_{\tau_{\iota_m}}$ implicates a complexity of $\mathcal{O}\left(2^d\right)$ and creates a new *curse of dimensionality.* This problem is solved efficiently by applying certain recursion schemes, which reduce the corresponding complexity to $\mathcal{O}\left(d^2\right)$ at worst.

Compared to existing algorithms, this new method offers a significantly better performance, meaning a higher accuracy and a lower CPU time without including any restrictive assumptions. Additionally, the basic principle of our new method can easily be applied to the calculation of sensitivities (also known as *Greeks*) and furthermore, can be transferred to different contexts quickly, like the pricing of high–dimensional European maximum or minimum options.

Alternatively to the quadrature method above, we also introduce two new Monte–Carlo methods within this thesis. For this, we consider the approaches of [JK04, CG08], which are based on the Gaussian copula, and generalize them to the application of any Archimedean copula.

This thesis is structured as follows: In the first part we introduce basic mathematical fundamentals, which are essential for the understanding of the second part. These fundamentals are divided into four chapters (Chapter 1 – 4). In Chapter 1 we present obligatory basics concerning copulae, like their proper definition and the famous Theorem of Sklar. In Chapter 2 the reader gets to know *BDSs* and the corresponding payments are explained in detail. Furthermore, we give information regarding the marginal and the joint distribution of default times. The third chapter of this thesis deals with different possibilities of modeling correlation by means of copulae. On this, we present different correlation measures and structures. In the last chapter of this part we connect the general knowledge of copulae and correlation in order to form special classes of copulae. In particular, we introduce the Gaussian copula and several Archimedean copulae.

In the second part of this thesis we apply the theory from Part I to the context of pricing *mBDSs*. For the purpose of motivation, we firstly present a static pricing model in Chapter 5. Afterwards, we introduce a semi–dynamic pricing model in Chapter 6, which includes the main developments of this thesis. In Section 6.1 we develop a formula for the distribution function $F_{\tau_{\iota_m}}$ of the m–th smallest default time and establish quadrature algorithms for pricing *mBDSs* accordingly. Within this section we distinguish between different copulae and at the end of this section we present the pricing of a *Credit Linked Note*, which is traded at the Stuttgart stock exchange. Having considered these quadrature methods, we present existing Monte–Carlo methods and develop two new methods in Section 6.2. We close this section by giving a detailed comparison of the developed Monte–Carlo methods and the established quadrature methods in Subsection 6.2.3.

In Section 6.3 we transfer the methods from Sections 6.1 and 6.2 to the context of pricing multi–dimensional European maximum or minimum options. We show that this transformation is almost straight forward and does not require any restrictive assumptions. In order to illustrate the relevance of this transformation we price a 13–dimensional minimum put option at the end of this section. The underlying basket of this option consists of 13 different DAX® corporations, whose stock prices satisfy the multi–dimensional Black–Scholes model. In Section 6.4 we again transfer the methods of Sections 6.1 and 6.2. Here, we present the efficient calculation of sensitivities regarding certain parameters which mainly influence the value of any *mBDS*.

In the end of this thesis we conclude the developed achievements in Part III. We give a detailed summary in Chapter 7 and a brief outlook in Chapter 8. Additional information concerning different distributions, algorithms drawing certain random numbers and further tables and figures is to be found in the appendix of this thesis, see Part IV.

Underlying computer hardware and software

Each algorithm within this thesis is implemented in the programming language C++ and is compiled by using the compiler version "gcc (GCC) 4.1.2 20080704 (Red Hat 4.1.2-50)". The particular CPU times for any numerical test were obtained by using a "Dell PowerEdge R710" server, which possesses a random access memory of 96GB and two "Intel Xeon X5570" CPUs, each with 2.93GHz (cp. [Del12]).

Cologne, July 2012 Alexander Schröter

Acknowledgment

My special acknowledgment is owed to my supervisor Priv.-Doz. Dr. Pascal Heider for his excellent assistance and mentoring of any kind. I also express my gratitude to Prof. Dr. Rüdiger Seydel for providing an outstanding research environment and for serving as a referee for this thesis. Additionally, I would like to thank Prof. Dr. Rainer Schrader for acting as a chairman of the disputation examination board and Dr. Markus Schulz for acting as the corresponding secretary. Thanks are also owed to my colleagues Albrecht Budke and Christian Jonen for valuable discussions and advices. Furthermore, I would like to address special thanks to my family and to Julia Jäschke for support of any kind. Additional thanks are given to Jennifer Kühl, Helge Aasen and Michael Tenvenne for auxiliary remarks and suggestions for improvements.

Several parts of this thesis were already presented or discussed at the workshop on computational finance within the “DMV Jahrestagung” in Cologne, Germany (September 2011), and at the “Amsterdam-Cologne Workshop on Computational Finance”, which took place in Amsterdam, The Netherlands (March 2011) and in Cologne, Germany (March 2012). Hence, I would like to thank all participants for helpful advices and comments.

Finally, I would like to express my gratitude to ifb AG, Bayenwerft 14, 50678 Cologne, Germany, especially to Dr. Christoph Schwer, for providing an exemplary credit portfolio, which was used for several tests and trials.

Contents

List of Acronyms

Abbreviations

BDS	Basket Default Swap
CDS	Credit Default Swap
GHQ	Gauss–Hermite quadrature
GKQ	Gauss–Kronrod quadrature
GLQ	Gauss–Legendre quadrature
GLaQ	Gauss–Laguerre quadrature
GKHQ	Gauss–Kronrod quadrature with a nested Gauss–Hermite quadrature
GLHQ	Gauss–Legendre quadrature with a nested Gauss–Hermite quadrature
GKLaQ	Gauss–Kronrod quadrature with a nested Gauss–Laguerre quadrature
(Joint) CDF	Probabilistic (joint) distribution function
(Joint) MDF	Probabilistic (joint) marginal distribution function
(Joint) PDF	Probabilistic (joint) density function
mBDS	m–th to Default Swap
MT	Mersenne Twister
SQ	Sobol's quasi–random number sequence

Mathematical symbols and variables

$\mathbb{R}$	Set of real numbers
$\overline{\mathbb{R}}$	$\mathbb{R} \cup \{-\infty, \infty\}$
$\mathbb{Z}$	Set of integers
$\mathbb{N}$	Set of natural numbers $(= \mathbb{Z}_{\geq 1})$
$\mathcal{D}_{(\cdot)}$	Domain of $(\cdot)$
$\mathcal{I}_{(\cdot)}$	Codomain of $(\cdot)$
$(\cdot)^{\mathrm{tr}}$	Transposed matrix or vector
$\mathrm{diag}_d(\ldots)$	Diagonal matrix with dimension d and entries $(\ldots)$
$\det(B)$	Determinant of any quadratic matrix B
E_d	d–dimensional identity matrix $(= \mathrm{diag}_d(1, \ldots, 1))$
$\boldsymbol{x}$	Vector (bold notation)
$x,\ x_i$	Scalar (non–bold notation)
d	Any natural number (≥ 2) representing the dimension of the underlying context
m	Any natural number $\in \{1, \ldots, d\}$
M	Number of Monte–Carlo runs
C_d	Any d–dimensional copula
c_d	Derivative of C_d $(= \frac{\partial^d C_d}{\partial u_1 \cdots \partial u_d})$
$\check{C}_d$	Sklar's unique d–dimensional copula
P_d	d–dimensional product copula / independence copula
$C^{\mathrm{Gau}}_{d,\Sigma}$	d–dimensional Gaussian copula using correlation matrix Σ
$C^{\mathrm{Arc}}_{d,\phi_\theta}$	d–dimensional Archimedean copula using generator ϕ_θ
$\boldsymbol{\iota}$	$(= \boldsymbol{\iota}\,(\boldsymbol{t}))$ Permutation of the set $\{1, \ldots, d\}$, which induces $t_{\iota_1(\boldsymbol{t})} \leq \ldots \leq t_{\iota_d(\boldsymbol{t})}$
$\mathbb{E}_{\mathbb{P}}$	Expectation with respect to probability measure $\mathbb{P}$

$\mathrm{Var}_{\mathbb{P}}$	Variance with respect to probability measure $\mathbb{P}$
$\mathrm{Cov}_{\mathbb{P}}$	Covariance with respect to probability measure $\mathbb{P}$
ρ	Pearson's correlation coefficient
Σ	Pearson's correlation matrix
Σ_k^{F}	Pearson's correlation matrix showing a k–factor structure
$\mathcal{K}_d$	Set of all possible d–dimensional correlation matrices (Pearson)
$\sim F$	Distributed according to (joint) CDF F
$\mathcal{U}[0,1]^k$	(Joint) CDF of the (indep.) uniform distribution on $[0,1]^k$, $k \in \mathbb{N}$
Φ	Standard Gaussian CDF
$\mathcal{N}(\mu, \sigma^2)$	Gaussian distribution with mean μ and variance σ^2
Φ_{Σ}^{d}	d–dimensional standard Gaussian joint CDF using correlation matrix Σ
τ^{K}	Kendall's Tau, Kendall's τ^{K}
$c_{k,m}$	Coefficient $c_{k,m} = (-1)^{k-m}\binom{k-1}{m-1}$ with $m \in \{1, \ldots, d\}$ and $k \in \{m, \ldots, d\}$

Elements of Basket Default Swaps

A	Basket $\{A_1, \ldots, A_d\}$ of d risky assets
S	Basket $\{S_1, \ldots, S_d\}$ of d obligors referring to A
τ_i	Random variable representing the default time of obligor S_i, $i \in \{1, \ldots, d\}$
$\boldsymbol{\tau}$	Random vector $(\tau_1, \ldots, \tau_d)^{\mathrm{tr}}$ representing the correlated default times of basket S
τ_{ι_m}	Random variable representing the m–th smallest default time within S
F_{τ_i}	CDF of random variable τ_i, $i \in \{1, \ldots, d\}$

$F_{\boldsymbol{\tau}}$	Joint CDF of random vector $\boldsymbol{\tau}$
N_i	Nominal value of asset A_i, $i \in \{1, \dots, d\}$
N	Nominal value of the underlying contract
R_i	Recovery rate of asset A_i, $i \in \{1, \dots, d\}$
h_i	Hazard rate of obligor S_i, $i \in \{1, \dots, d\}$
T	Maturity of the underlying contract
s_m	Premium payment of the underlying mBDS (per nominal value)
$V_m^{(\cdot)}$	Value of the underlying mBDS specified in $(\cdot)$
$\Lambda_m^{(\cdot)}$	Discounted payoff (short form: payoff) of the underlying mBDS specified in $(\cdot)$
$\widetilde{\Lambda}^{(\cdot)}$	Discounted payoff (short form: payoff) of the underlying mBDS, which only depends on the m–th smallest default time and is specified in $(\cdot)$
r_t	Risk free interest rate during the period $[0, t]$, $t \in \mathbb{R}_{>0}$

List of Figures

List of Tables

List of Algorithms

Part I.

Fundamentals

This first part introduces the reader to the mathematical framework of this thesis. Essential terms are presented in detail and are defined properly. For this, we proceed as follows:

In Chapter 1 we derive the definition of a copula and provide some useful copula properties. Furthermore, we present a fundamental theorem concerning copulae, which simultaneously represents their main application. For further information regarding copulae we refer to [Joe97, Nel06].

Next, we introduce a special subclass of credit derivatives named Basket Default Swaps in Chapter 2. On this, we analyze their corresponding payments and their dependence on certain default times. Additionally, we discuss the modeling of marginal and joint distribution of default times. A general introduction to credit derivatives is given in [DS03, Sch03, MRW06].

The correlation describes a crucial part of pricing multi–dimensional products of any kind and is introduced in Chapter 3. In this chapter we get to know different correlation measures and analyze them in terms of special properties. In addition, we introduce factor models, which appear frequently during this thesis. Important discussions regarding the correlation are mentioned in [EMS02, ELM03].

In the last chapter of this part we establish certain copulae. We define the Gaussian copula as well as several Archimedean copulae. Furthermore, these copulae are analyzed regarding special properties, like correlation or symmetry. An extensive survey of existing copulae is given in [Nel06].

1. Copulae

In this chapter we present essential basics dealing with copulae. For this, we develop the definition of a copula at first and afterwards state the main application of copulae by introducing the famous Theorem of Sklar (1973). Finally, we present some useful copula properties and their application to random variables.

1.1. Introduction

Let $d \in \mathbb{N}_{\geq 2}$ denote any natural number and let $\overline{\mathbb{R}}$ denote the set $\overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty, \infty\}$. A d–dimensional copula C_d (abbr.: d–copula) is a real mapping with domain $\mathcal{D}_{C_d} = [0,1]^d$ and codomain $\mathcal{I}_{C_d} = [0,1]$, which meets certain conditions. These conditions are introduced by the following definitions.

Definition 1.1 (H–volume)**.** Let $H : \mathcal{D}_H \mapsto [0,1]$ be a real mapping, in which $\mathcal{D}_H := [\boldsymbol{A}, \boldsymbol{B}] := [A_1, B_1] \times \ldots \times [A_d, B_d] \subseteq \overline{\mathbb{R}}^d$ denotes a d-dimensional interval. Furthermore, let $W_d := [\boldsymbol{a}, \boldsymbol{b}] \subseteq \mathcal{D}_H$ denote another d–dimensional interval and let $\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(2^d)}$ denote the edges of W_d, which are constructed as $\boldsymbol{c}^{(k)} = \left(c_1^{(k)}, \ldots, c_d^{(k)}\right)$ with $c_i^{(k)} \in \{a_i, b_i\}$, $i \in \{1, \ldots, d\}$ and $k \in \{1, \ldots, 2^d\}$. Then, the *H–volume* V_H^{V} with respect to W_d is defined as

$$V_H^{\mathrm{V}}(W_d) := \sum_{k=1}^{2^d} \overline{\operatorname{sgn}}\left(\boldsymbol{c}^{(k)}\right) H\left(\boldsymbol{c}^{(k)}\right),$$

in which $\overline{\operatorname{sgn}}\left(\boldsymbol{c}^{(k)}\right)$ is given by

$$\overline{\operatorname{sgn}}\left(\boldsymbol{c}^{(k)}\right) := \begin{cases} 1, & \text{if } \left|\left\{c_i^{(k)} \in \{c_1^{(k)}, \ldots, c_d^{(k)}\} | c_i^{(k)} = a_i\right\}\right| \text{ is even} \\ -1, & \text{if } \left|\left\{c_i^{(k)} \in \{c_1^{(k)}, \ldots, c_d^{(k)}\} | c_i^{(k)} = a_i\right\}\right| \text{ is odd} \end{cases}.$$

Definition 1.2 (d–increasing)**.** Let $H : \mathcal{D}_H \mapsto [0,1]$ be a real mapping, in which $\mathcal{D}_H := [\boldsymbol{A}, \boldsymbol{B}]^d := [A_1, B_1] \times \ldots \times [A_d, B_d] \subseteq \overline{\mathbb{R}}^d$ denotes a d-dimensional interval. Then, H is called *d–increasing* on $\mathcal{D}_H$, if and only if

$$V_H^{\mathrm{V}}(W_d) \geq 0$$

holds for any d–dimensional interval $W_d \subseteq \mathcal{D}_H$.

Definition 1.3 (Grounded)**.** Let $H : \mathcal{D}_H \mapsto [0,1]$ be a real mapping, in which $\mathcal{D}_H := [\boldsymbol{A}, \boldsymbol{B}]^d := [A_1, B_1] \times \ldots \times [A_d, B_d] \subseteq \overline{\mathbb{R}}^d$ denotes a d-dimensional interval. Furthermore, let the set $\mathcal{G}$ be defined as $\mathcal{G} := \{\boldsymbol{a} \in \mathcal{D}_H | \, \exists \, i \in \{1, \ldots, d\} : a_i = A_i\}$. Then, the mapping H is called *grounded* on $\mathcal{D}_H$, if and only if

$$\forall \boldsymbol{a} \in \mathcal{G} : H(\boldsymbol{a}) = 0$$

holds.

With the help of the definitions above we are able to define a d–dimensional copula properly.

Definition 1.4 (d–copula)**.** The d–dimensional real mapping $C_d : [0,1]^d \mapsto [0,1]$ is called a *d–copula*, if and only if the following conditions are fulfilled:

1. C_d is grounded on $[0,1]^d$.
2. C_d is d–increasing on $[0,1]^d$.
3. C_d has uniformly distributed marginals, i.e. for each $i \in \{1, \ldots, d\}$:

$$\forall u \in [0,1] : \; \forall \boldsymbol{e}_i(u) := (1, \ldots, 1, \underbrace{u}_{i\text{–th pos.}}, 1, \ldots, 1)^{\text{tr}} \in [0,1]^d : C_d\left(\boldsymbol{e}_i(u)\right) = u.$$

If $\frac{\partial^d}{\partial v_1 \cdots \partial v_d} C_d\left(v_1, \ldots, v_d\right)$ exists for any $\boldsymbol{v} \in (0,1)^d$, we will call

$$c_d\left(v_1, \ldots, v_d\right) = \frac{\partial^d}{\partial v_1, \ldots, \partial v_d} C_d\left(v_1, \ldots, v_d\right)$$

the *copula's density.*

Except the continuity from the right, the d–copula C_d meets every condition, which has to be fulfilled in order to form a proper probabilistic joint distribution function on $[0,1]^d$.

Definition 1.5 (Distribution function, marginal distribution)**.** Let $\left(\overline{x}_1, \ldots, \overline{x}_d\right)^{\text{tr}} \in \overline{\mathbb{R}}^d$ be any vector. Then, we can state the following:

1. A d–dimensional *abstract joint distribution function* $F : \overline{\mathbb{R}}^d \mapsto [0,1]$ is a mapping, which is grounded on $\overline{\mathbb{R}}^d$ as well as d–increasing on $\overline{\mathbb{R}}^d$ and additionally meets $F(\infty, \ldots, \infty) = 1$.

2. A d–dimensional *probabilistic joint distribution function* (abbr.: *joint CDF*) $F : \overline{\mathbb{R}}^d \mapsto [0,1]$ is an abstract joint distribution function, which additionally meets
$$\lim_{x_1 \searrow \overline{x}_1, \ldots, x_d \searrow \overline{x}_d} F\left(x_1, \ldots, x_d\right) = F\left(\overline{x}_1, \ldots, \overline{x}_d\right).$$
That is, the limit is interpreted as a component–by–component limit.

3. The i–th *abstract / probabilistic marginal distribution function* $F_i : \overline{\mathbb{R}} \mapsto [0,1]$, $i \in \{1, \ldots, d\}$, (abbr. in the probabilistic case: *MDF*) of a d–dimensional abstract / probabilistic joint distribution function F is defined by
$$F_i(\overline{x}_i) := F(\infty, \ldots, \infty, \underbrace{\overline{x}_i}_{i\text{–th pos.}}, \infty, \ldots, \infty).$$

Consequently, a d–copula is a proper abstract joint distribution function on $[0,1]^d$, but technically spoken it is not a proper probabilistic joint distribution function (joint CDF). In the following section we present an analytical relation between joint distribution functions, copulae and marginal distribution functions with the help of Sklar's famous Theorem.

1.2. Theorem of Sklar

Sklar's Theorem from 1973 illustrates the main application of copulae. It offers its users the possibility of constructing d–dimensional joint CDFs by means of copulae and marginal distribution functions.

Theorem 1.1 (Sklar, abstract)**.** *Let F be a d–dimensional abstract joint distribution function with abstract marginal distribution functions $F_1, \ldots, F_d$. Then, there exists a d–copula $\check{C}_d$, so that*
$$\forall \boldsymbol{x} \in \overline{\mathbb{R}}^d : \; F\left(x_1, \ldots, x_d\right) = \check{C}_d\left(F_1\left(x_1\right), \ldots, F_d\left(x_d\right)\right) \tag{1.1}$$
holds. If $F_1, \ldots, F_d$ are continuous, $\check{C}_d$ from equation (1.1) will be unique.

And vice versa: Let $\check{C}_d$ be any d–copula and let $F_1, \ldots, F_d$ be abstract distribution functions. Then, the d–dimensional mapping F from equation (1.1) defines a proper d–dimensional abstract joint distribution function with abstract marginal distribution functions $F_1, \ldots, F_d$.

Proof. For the proof we refer to [Skl73, Theorem 1]. □

In the following we assume without loss of generality that the abstract / probabilistic marginal distributions $F_1, \ldots, F_d$ are continuous. This is justified by the underlying context in Section 2.2.

The first part of Sklar's Theorem 1.1 will remain unchanged, if we change the abstract (marginal / joint) distribution functions to probabilistic ones, because any probabilistic (marginal / joint) distribution function also represents an abstract one. Nevertheless, in the second part of Theorem 1.1 we have to establish some restrictions to guarantee the applicability of probabilistic (marginal / joint) distribution functions.

Theorem 1.2 (Sklar, probabilistic)**.** *Let F be a d–dimensional joint CDF with MDFs $F_1, \ldots, F_d$. Then, there exists a d–copula $\check{C}_d$, so that*

$$\forall \boldsymbol{x} \in \overline{\mathbb{R}}^d : \; F(x_1, \ldots, x_d) = \check{C}_d \left(F_1(x_1), \ldots, F_d(x_d) \right) \tag{1.2}$$

holds. If $F_1, \ldots, F_d$ are continuous, $\check{C}_d$ from equation (1.2) will be unique.

And vice versa: Let $\check{C}_d$ be any ***continuous*** *d–copula and let $F_1, \ldots, F_d$ be CDFs. Then, the d–dimensional mapping F from equation (1.2) defines a proper d–dimensional joint CDF with MDFs $F_1, \ldots, F_d$.*

Proof. The first part of Theorem 1.2 is implied by Theorem 1.1 directly. Considering the second part we know from Theorem 1.1 that F is "at least" an abstract joint distribution function. Consequently, we just have to show that F is continuous from the right. For this, we use the continuity of $\check{C}_d$ and proof the continuity (from the right) of F by

$$\begin{aligned} \lim_{x_1 \searrow \overline{x}_1, \ldots, x_d \searrow \overline{x}_d} F(x_1, \ldots, x_d) &= \lim_{x_1 \searrow \overline{x}_1, \ldots, x_d \searrow \overline{x}_d} \check{C}_d \left(F_1(x_1), \ldots, F_d(x_d) \right) \\ &= \check{C}_d \left(\lim_{x_1 \searrow \overline{x}_1} F_1(x_1), \ldots, \lim_{x_d \searrow \overline{x}_d} F_d(x_d) \right) \\ &= \check{C}_d \left(F_1(\overline{x}_1), \ldots, F_d(\overline{x}_d) \right) \\ &= F(\overline{x}_1, \ldots, \overline{x}_d), \end{aligned}$$

for any $(\overline{x}_1, \ldots, \overline{x}_d) \in \overline{\mathbb{R}}^d$. □

Especially for the usage of random variables the results of Theorem 1.2 are very important because the distribution function of a random variable is of probabilistic nature. At this, we should remark that the restriction to continuous copulae does not cause any limitation for the further development of this thesis because most copulae used in practice show continuity.

Remark 1.1 (Question of the right copula). Looking at Theorem 1.2 at first sight we have to ask the question: “Which copula is the right one?” i.e. which copula is able to fulfill equation (1.2)? This question is of very high importance for the usage of copulae and consequently it describes a huge area of research, which is not the topic of this thesis. For questions regarding this area we refer to [DNR00, Mel03]. In the following we assume the knowledge of copula $\check{C}_d$ in Theorem 1.2.

Finally, we give some additional definitions, which are useful in the following.

Definition 1.6 (Survival–copula)**.** Let $X_1, \dots, X_d$ be real random variables with corresponding CDFs $F_1(x_1), \dots, F_d(x_d)$, joint CDF

$$F(x_1, \dots, x_d) = \check{C}_d\left(F_1(x_1), \dots, F_d(x_d)\right),\ \boldsymbol{x} \in \overline{\mathbb{R}}^d,$$

and product probability measure $\mathbb{P}$. Then, $\overline{F_1}(x_1) := 1 - F_1(x_1), \dots, \overline{F_d}(x_d) := 1 - F_d(x_d)$ are called the *survival-CDFs* and $\overline{F}(x_1, \dots, x_d) := \mathbb{P}(X_1 > x_1, \dots, X_d > x_d)$ is called the joint *survival–CDF* for any $\boldsymbol{x} \in \overline{\mathbb{R}}^d$. Analogically to Theorem 1.2, we can state the existence of a *survival–copula* $\overline{\check{C}}_d$ via

$$\forall \boldsymbol{x} \in \overline{\mathbb{R}}^d : \overline{F}(x_1, \dots, x_d) = \overline{\check{C}}_d\left(\overline{F}_1(x_1), \dots, \overline{F}_d(x_d)\right)$$

and due to [GLN$^+$01, Theorem 1] mapping $\overline{\check{C}}_d$ defines a proper d–copula. In the special case $d = 2$ we achieve equality

$$\overline{\check{C}}_2(v_1, v_2) = v_1 + v_2 - 1 + \check{C}_2(1 - v_1, 1 - v_2)$$

for any $\boldsymbol{v} \in [0,1]^2$ (cp. [Nel06, Section 2.6]).

By means of survival–copulae we are able to present a special symmetry–property of copulae.

Definition 1.7 (Radial symmetry)**.** Any d–copula C_d with survival–copula $\overline{C}_d$ is called *radial symmetric*, if and only if

$$C_d(\boldsymbol{v}) = \overline{C}_d(\boldsymbol{v})$$

holds for any $\boldsymbol{v} \in [0,1]^d$.

For the special case $d = 2$ we derive equality

$$C_2(v_1, v_2) = v_1 + v_2 - 1 + C_2(1 - v_1, 1 - v_2).$$

This equality illustrates that in case of radial symmetry the C_2–volume (cp. Def. 1.1) of $[0, v_1] \times [0, v_2]$ equals the C_2–volume of $[1 - v_1, 1] \times [1 - v_2, 1]$. This statement is equivalent to

$$\int\limits_0^{v_1} \int\limits_0^{v_2} c_2(x, y)\,\mathrm{d}x\mathrm{d}y = \int\limits_{1-v_1}^{1} \int\limits_{1-v_2}^{1} c_2(x, y)\,\mathrm{d}x\mathrm{d}y.$$

Remark 1.2. Let $\boldsymbol{X} = (X_1, \ldots, X_d)^{\mathrm{tr}}$ denote any real random vector with MDFs $F_1, \ldots, F_d$ and joint CDF F, which are connected via equation (1.2) and d–copula $\check{C}_d$. We have to remark that radial symmetry of $\check{C}_d$ does *not* imply radial symmetry of random vector $\boldsymbol{X}$ (i.e. $\exists \boldsymbol{a} \in \mathbb{R}^d : \forall \boldsymbol{x} \in \overline{\mathbb{R}}^d : F(\boldsymbol{a}+\boldsymbol{x}) = \overline{F}(\boldsymbol{a}-\boldsymbol{x})$ cp. [Nel06, Def. 2.7.1, Theorem 2.7.2]). For this, the MDFs $F_1, \ldots, F_d$ have to be radial symmetric with respect to vector $\boldsymbol{a} \in \mathbb{R}^d$ as well, i.e.

$$\forall t \in \mathbb{R} : F_i\left(a_i + t\right) = \overline{F}_i\left(a_i - t\right)$$

must hold for each $i \in \{1, \ldots, d\}$ (cp. [Nel06, Theorem 2.7.3]).

We can summarize the following: By means of copulae we are able to connect different MDFs to a proper joint CDF. This fact is very useful because joint CDFs are usually unknown at markets.

In practice copulae are often used in risk management and for the evaluation of credit derivatives. Thereby, the joint CDF of default times is modeled with the help of Theorem 1.2. Afterwards, we can evaluate credit portfolios or credit derivatives by means of the resulting joint CDF.

2. Basket Default Swaps

Basket Default Swaps (abbr.: BDSs) build a special subcategory of credit derivatives. The contract, which underlies any BDS, refers to a basket of risky assets. A BDS offers its buyer protection against certain credit events within this basket. Thus, a BDS represents an extension of the one–dimensional Credit Default Swap (abbr.: CDS).

Let $d \in \mathbb{N}_{\geq 2}$ denote the number of different assets A_i, $i \in \{1, \ldots, d\}$, and let

$$A := \{A_1, \ldots, A_d\}$$

denote the basket which consists of these assets. Typically, such a basket contains credits, debt obligations, corporate bonds or government bonds. Each asset A_i belongs to a corresponding obligor S_i, which usually is a borrower, a corporate or a government. The basket consisting of the different obligors S_i, $i \in \{1, \ldots, d^{'}\}$, is denoted by

$$S := \{S_1, \ldots, S_{d'}\}\,,\ d^{'} \leq d.$$

Furthermore, let $\tau_i,\ i \in \{1, \ldots, d\}$, denote a real random variable in the probability space $(\Omega_i, \mathcal{B}_i, \mathbb{P}_i)$, which represents the time of occurrence of a special credit event with respect to asset A_i. If this credit event never arrives, we will define $\tau_i = \infty$. Thus, the random vector $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_d)^{\text{tr}}$ represents the correlated times of occurrences of special credit events and is defined by means of the product probability space $(\Omega, \mathcal{B}, \mathbb{P})$. For a more detailed definition of $\Omega_i, \mathcal{B}_i, \mathbb{P}_i$ and $\Omega, \mathcal{B}, \mathbb{P}$ respectively we refer to [Fel71]. The CDF of τ_i is denoted by F_{τ_i} with its density f_{τ_i}. Additionally, the joint CDF of $\boldsymbol{\tau}$ is denoted by $F_{\boldsymbol{\tau}}$, in which the CDFs $F_{\tau_1}, \ldots, F_{\tau_d}$ act as MDFs.

Remark 2.1 (Special credit events). The special credit events mentioned above are determined before signing the underlying contract. Usually, such credit events are happenings like insolvency of the corresponding obligor S_i, particular payment defaults or payment delays in asset A_i or a restructuring of asset A_i. In the following we assume that the special credit event is determined as the insolvency of the corresponding obligor S_i. Furthermore we assume that each obligor S_i, $i \in \{1, \ldots, d^{'}\}$, only refers to exactly one asset A_i, $i \in \{1, \ldots, d\}$, i.e. $d = d^{'}$. Consequently, random variable τ_i represents the default time of the corresponding obligor S_i.

To offer a more detailed view on the character (payments, kind of protection) of a BDS we have to determine the nature of the underlying contract. A widely used type of contract is the well–known m–th to Default Swap (abbr.: mBDS), $m \in \{1, \ldots, d\}$. During a previously determined time horizon $[0, T]$ an mBDS offers its buyer protection against the m-th default within basket S. A popular example of an mBDS is the First to Default Swap ($m = 1$), which offers its buyer protection against the first default within S.

Important parameters in the context of mBDSs are the following: Each asset A_i has a nominal value N_i, their sum $N = \sum_{i=1}^{d} N_i$ is called the nominal value of the underlying mBDS. Furthermore, each asset A_i possesses a corresponding recovery rate $R_i \in [0, 1]$, which denotes the collateralized fraction of N_i. The risk free interest rate during the period $[0, t]$, $t \in \mathbb{R}_{>0}$, is denoted by r_t.

2.1. Payment flows

Usually, the payments of an mBDS can be divided into two categories, premium payments and default payments. The buyer of an mBDS pays periodical premium payments, whose amounts are determined before signing the contract. If m obligors default before maturity T, the buyer will immediately receive a default payment and will not have to pay any further premium payments. The premiums equal the amount of $s_m \cdot N$, $s_m \in \mathbb{R}$.

Let $\boldsymbol{t} = (t_1, \ldots, t_d)^{\text{tr}} \in \overline{\mathbb{R}}_{\geq 0}^{d}$ be any vector. Then, $\boldsymbol{\iota}(\boldsymbol{t})$ denotes a permutation of the set $\{1, \ldots, d\}$, so that $t_{\iota_1(\boldsymbol{t})} \leq \ldots \leq t_{\iota_d(\boldsymbol{t})}$ holds. In the following we only consider $\boldsymbol{t}$ to be the underlying vector. Therefore, we can neglect the argument $\boldsymbol{t}$ of $\boldsymbol{\iota}(\boldsymbol{t})$ and simply write $\boldsymbol{\iota}$ or ι_i respectively.

Definition 2.1 (Premium payment)**.** Let us consider any m–th to Default Swap with maturity T and underlying basket of obligors $S = \{S_1, \ldots, S_d\}$. Furthermore, the periodic premium $s_m \cdot N$ is paid at dates $0 < t_1^{\text{PP}} < \ldots < t_\kappa^{\text{PP}} \leq T$, $\kappa \in \mathbb{N}$. Let $\boldsymbol{t} = (t_1, \ldots, t_d)^{\text{tr}} \in \overline{\mathbb{R}}_{\geq 0}^{d}$ denote the default times of obligors $S_1, \ldots, S_d$. Then, today's ($t = 0$) value of the *premium payment* of an m-th to Default Swaps is given by

$$\Lambda_m^{\mathrm{PP}}(\boldsymbol{t}) :=$$
$$\begin{cases} \sum\limits_{i=1}^{j} s_m N \exp\left(-r_{t_i^{\mathrm{PP}}} t_i^{\mathrm{PP}}\right) + \left(t_{\iota_m} - t_j^{\mathrm{PP}}\right) s_m N \exp\left(-r_{t_{\iota_m}} t_{\iota_m}\right), & t_j^{\mathrm{PP}} \leq t_{\iota_m} \leq T^* \\ \sum\limits_{i=1}^{\kappa} s_m N \exp\left(-r_{t_i^{\mathrm{PP}}} t_i^{\mathrm{PP}}\right), & \text{else} \end{cases},$$

in which $j \in \{0, \ldots, \kappa\}$, $T^* = \min\{T, t_{j+1}^{\mathrm{PP}}\}$, $t_0^{\mathrm{PP}} := 0$ and $t_{\kappa+1}^{\mathrm{PP}} := \infty$ hold.

Definition 2.2 (Default payment)**.** Let us consider any m–th to Default Swap with maturity T and underlying basket of obligors $S = \{S_1, \ldots, S_d\}$. Furthermore, let $\boldsymbol{t} = (t_1, \ldots, t_d)^{\mathrm{tr}} \in \overline{\mathbb{R}}_{\geq 0}^{d}$ denote the default times of obligors $S_1, \ldots, S_d$. Then, today's $(t = 0)$ value of the *default payment* of an m-th to Default Swap is given by

$$\Lambda_m^{\mathrm{DP}}(\boldsymbol{t}) := \begin{cases} (1 - R_{\iota_m}) N_{\iota_m} \exp\left(-r_{t_{\iota_m}} t_{\iota_m}\right), & t_{\iota_m} \leq T \\ 0, & \text{else} \end{cases}. \tag{2.1}$$

At this, R_{ι_m} denotes the recovery rate and N_{ι_m} denotes the nominal value of asset A_{ι_m}, which causes the m-th default event.

Of course, today $(t = 0)$ the exact default times $t_1, \ldots, t_d$ are unknown. Thus, we have to calculate the expectations $\mathbb{E}_{\mathbb{P}}\left[\Lambda_m^{\mathrm{DP}}(\boldsymbol{\tau})\right]$ and $\mathbb{E}_{\mathbb{P}}\left[\Lambda_m^{\mathrm{PP}}(\boldsymbol{\tau})\right]$ of the corresponding payment flows. Using these expectations we are able to calculate a *fair premium.*

Definition 2.3 (mBDS value, fair premium)**.** Let us consider any m–th to Default Swap with periodic premium $s_m \cdot N$ and maturity T. Then, today's $(t = 0)$ *value* of this mBDS equals the expectation

$$V_m := \mathbb{E}_{\mathbb{P}}\left[\Lambda_m^{\mathrm{PP}}(\tau_1, \ldots, \tau_d) - \Lambda_m^{\mathrm{DP}}(\tau_1, \ldots, \tau_d)\right].$$

Furthermore, today's $(t = 0)$ *fair premium* $s_m \neq 0$ equals the expectation

$$s_m := \frac{\mathbb{E}_{\mathbb{P}}\left[\Lambda_m^{\mathrm{DP}}(\tau_1, \ldots, \tau_d)\right]}{\mathbb{E}_{\mathbb{P}}\left[\widehat{\Lambda}_m^{\mathrm{PP}}(\tau_1, \ldots, \tau_d)\right]},$$

in which $\widehat{\Lambda}_m^{\mathrm{PP}}(\boldsymbol{t}) := \Lambda_m^{\mathrm{PP}}(\boldsymbol{t}) / s_m$ holds for any $\boldsymbol{t} \in \overline{\mathbb{R}}_{\geq 0}^{d}$.

2.2. Marginal distributions of default times

The CDFs of the random variables $\tau_1, \ldots, \tau_d$ can efficiently be calibrated by market data. Especially CDS and bond markets are very helpful in this context. By this means we can model a hazard rate $h_i(t_i)$ for each obligor S_i (cp. e.g. [MRW06]).

Using hazard rate h_i we are finally able to formulate the CDF of the random variable τ_i via an intensity model, cp. [DS03]. Then, we can state the CDF of τ_i as

$$F_{\tau_i}\left(t_i\right) := \mathbb{P}_i\left(\tau_i \leq t_i\right) = 1 - \exp\left(-\int\limits_0^{t_i} h_i(u)\mathrm{d}u\right) \tag{2.2}$$

and its corresponding probabilistic density function (abbr.: PDF) as

$$f_{\tau_i}\left(t_i\right) = h_i\left(t_i\right)\exp\left(-\int\limits_0^{t_i} h_i(u)\mathrm{d}u\right)$$

for any $t_i \in \overline{\mathbb{R}}_{\geq 0}$ and for each $i \in \{1, \ldots, d\}$.

2.3. Joint distribution function of default times

In contrast to the MDFs of $\boldsymbol{\tau}$, it is almost impossible to calibrate the joint CDF $F_{\boldsymbol{\tau}}$ by market data because the required products on particular baskets are illiquid. Nevertheless, due to Theorem 1.2 we know that a copula $\check{C}_d$ exists, which is able to state this joint CDF $F_{\boldsymbol{\tau}}$ by means of its MDFs $F_{\tau_1}, \ldots, F_{\tau_d}$. For this, we define $F_{\boldsymbol{\tau}}$ as

$$\forall \boldsymbol{t} \in \overline{\mathbb{R}}^d_{\geq 0}: \; F_{\boldsymbol{\tau}}\left(\boldsymbol{t}\right) := \check{C}_d\left(F_{\tau_1}\left(t_1\right), \ldots, F_{\tau_d}\left(t_d\right)\right) \tag{2.3}$$

with known copula $\check{C}_d$ and known MDFs $F_{\tau_1}, \ldots, F_{\tau_d}$. The choice of $\check{C}_d$ is a widely discussed problem, but is not considered in this thesis (cp. Remark 1.1). In general, copula $\check{C}_d$ should be chosen with respect to the context and to the user. In Chapter 4 we present a short survey of different copulae.

Considering equation (2.3) we can easily recognize that the correlation occurring within $F_{\boldsymbol{\tau}}$ is completely induced by copula $\check{C}_d$. Due to this fact it is essential to analyze any copula regarding its implied correlation structure. We have to introduce robust parameters for quantifying the prevailing correlation, which we get to know in the next chapter. We close this section with a useful remark.

Remark 2.2 (Multi–dimensional marginal distributions). For any $S_I \subset S$, with index set $I := \{I_1, \ldots, I_k\} \subset \{1, \ldots, d\}$, $I_1 < \ldots < I_k$, copula $\check{C}_d$ induces a joint CDF $F_{\boldsymbol{\tau}_I}$ with respect to the random vector $\boldsymbol{\tau}_I := (\tau_j)_{j\in I}$ (cp. [Nel06, Section 2.10]). Let $\boldsymbol{t} := (t_1, \ldots, t_d)^{\mathrm{tr}} \in \overline{\mathbb{R}}^d_{\geq 0}$ and $\boldsymbol{t}_I := (t_{I_1}, \ldots, t_{I_k})^{\mathrm{tr}} \in \overline{\mathbb{R}}^k_{\geq 0}$ be vectors. Then, the joint CDF $F_{\boldsymbol{\tau}_I}$ is given by

$$F_{\boldsymbol{\tau}_I}\left(\boldsymbol{t}_I\right) := F_{\boldsymbol{\tau}}\left(\boldsymbol{t}^{I\to d}\right) = \check{C}_d\left(\boldsymbol{F}^{I\to d}\right) = \check{C}^I_k\left(F_{\tau_{I_1}}(t_{I_1}), \ldots, F_{\tau_{I_k}}(t_{I_k})\right)$$

in which

$$\boldsymbol{t}^{I\to d} = \left(t_j^{I\to d}\right)_{j\in\{1,\ldots,d\}} := \begin{cases} t_j, & j \in I \\ \infty, & \text{else} \end{cases}$$

and

$$\boldsymbol{F}^{I\to d} = \left(F_j^{I\to d}\right)_{j\in\{1,\ldots,d\}} := \left(F_{\tau_j}\left(t_j^{I\to d}\right)\right)_{j\in\{1,\ldots,d\}} = \begin{cases} F_{\tau_j}(t_j), & j \in I \\ 1, & \text{else} \end{cases}$$

hold. If $|I| = d$ holds, the only possible index set will equal $I = \{1, \ldots, d\}$, which justifies the notation $\check{C}_d^I =: \check{C}_d$ and $C_d^I =: C_d$ respectively.

3. Correlation

In addition to the MDFs, the correlation represents another crucial component, which is essential for modeling the joint CDF of default times by means of copulae. In order to quantify the prevailing correlation as well as possible we have to choose a suitable correlation measure first. For this, we can find five desirable properties in [EMS02], which a suitable correlation measure should possess. Unfortunately, [EMS02, Proposition 3] shows that no correlation measure is able to fulfill these five properties simultaneously.

Let Ω_{RV} be the set of all real random variables and let $X \in \Omega_{\text{RV}}$ and $Y \in \Omega_{\text{RV}}$ be elements out of this set. Furthermore, F_X and F_Y denote their CDFs and $F_{X,Y}$ denotes their joint CDF. Then, the correlation measure $\mathbb{K} : \Omega_{\text{RV}} \times \Omega_{\text{RV}} \mapsto [-1, 1]$ should meet as many of the following five intuitive conditions as possible.

(K1) $\mathbb{K}(X, Y) = \mathbb{K}(Y, X)$ *(symmetry)*

(K2) $\mathbb{K}(X, Y) \in [-1, 1]$ *(scaling)*

(K3) $\mathbb{K}(X, Y) = 1 \Leftrightarrow F_{X,Y}(x, y) = \min\{F_X(x), F_Y(y)\}$ *(comonotony)*
$\mathbb{K}(X, Y) = -1 \Leftrightarrow F_{X,Y}(x, y) = \max\{F_X(x) + F_Y(y) - 1, 0\}$ *(countermonotony)*

(K4) For a strictly monotonic mapping $\mathcal{T} : \Omega_{\text{RV}} \mapsto \Omega_{\text{RV}}$[1]:

$$\mathbb{K}\left(\mathcal{T}(X), Y\right) = \begin{cases} \mathbb{K}(X, Y), & \text{if } \mathcal{T} \text{ is increasing} \\ -\mathbb{K}(X, Y), & \text{if } \mathcal{T} \text{ is decreasing} \end{cases}$$

(invariance under strict transformations)

(K5) $\mathbb{K}(X, Y) = 0 \Leftrightarrow X$ and Y are independent *(equivalence of independence)*

For a better illustration of property (K3) we add: Assuming $\mathbb{K}(X, Y) = 1$ and $X < x$ hold, then, a unique $\widehat{y} \in \mathbb{R}$ should exist, so that $X < x \Leftrightarrow Y < \widehat{y}$ and $F_X(x) = F_Y(\widehat{y}) = F_{X,Y}(x, \widehat{y})$ hold. Analogically, we can state: Let $Y < y$ hold.

[1] Use $\mathcal{T}$ as a real–valued mapping to apply the concept of monotonicity, cp. Example 3.1(b) for a better understanding

Then, a unique $\widehat{x} \in \mathbb{R}$ should exist, so that $Y < y \Leftrightarrow X < \widehat{x}$ and $F_Y(y) = F_X(\widehat{x}) = F_{X,Y}(\widehat{x}, y)$ hold. Because of equality $\mathbb{K}(X,Y) = 1$ either $x \leq \widehat{x} \wedge \widehat{y} \leq y$ and $F_{X,Y}(x,y) = F_{X,Y}(x,\widehat{y})$ or $x > \widehat{x} \wedge \widehat{y} > y$ and $F_{X,Y}(x,y) = F_{X,Y}(\widehat{x},y)$ hold. Consequently, $F_{X,Y}(x,y) = \min\{F_X(x), F_Y(y)\}$ follows. The joint CDF for the special case $\mathbb{K}(X,Y) = -1$ is developed analogically.

In this thesis we discuss two different measures for modeling correlation. On the one hand we introduce Pearson's correlation coefficient ρ, which meets the conditions (K1) and (K2) and on the other hand we introduce the rank correlation coefficient Kendall's Tau τ^{K}, which meets the conditions (K1) to (K4).

3.1. Pearson's correlation coefficient ρ

In the following let $(X_1,\ldots,X_d)^{\mathrm{tr}}$ be a real random vector with MDFs $F_{X_1},\ldots,F_{X_d}$.

Definition 3.1 (Pearson's correlation coefficient, correlation matrix). *Pearson's correlation coefficient* ρ_{X_i,X_j} (also: correlation coefficient) of two random variables X_i and X_j with joint CDF F_{X_i,X_j}, $i,j \in \{1,\ldots,d\}$ with $i \neq j$, is given by

$$\begin{aligned}\rho_{X_i,X_j} &:= \frac{\mathrm{Cov}\left(X_i,X_j\right)}{\sqrt{\mathrm{Var}\left(X_i\right)}\sqrt{\mathrm{Var}\left(X_j\right)}} = \frac{\mathbb{E}\left(X_i \cdot X_j\right) - \mathbb{E}\left(X_i\right)\cdot\mathbb{E}\left(X_j\right)}{\sqrt{\mathrm{Var}\left(X_i\right)}\sqrt{\mathrm{Var}\left(X_j\right)}} \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{F_{X_i,X_j}\left(t_i,t_j\right) - F_{X_i}\left(t_i\right)\cdot F_{X_j}\left(t_j\right)}{\sqrt{\mathrm{Var}\left(X_i\right)}\sqrt{\mathrm{Var}\left(X_j\right)}}\mathrm{d}t_i\mathrm{d}t_j, \end{aligned} \tag{3.1}$$

in which the functions $\mathbb{E}(\cdot)$, $\mathrm{Cov}(\cdot)$ and $\mathrm{Var}(\cdot)$ are calculated with respect to the particular (product) probability measure of X_i and X_j. Equality (3.1) was presented in [Hoe40] first and was proven in [DG96, Lemma 1] for example. The positive semi–definite and symmetric matrix

$$\Sigma = \left(\Sigma_{i,j}\right)_{i,j\in\{1,\ldots,d\}} := \begin{pmatrix} 1 & \rho_{X_1,X_2} & \rho_{X_1,X_3} & \cdots & \rho_{X_1,X_d} \\ \rho_{X_2,X_1} & 1 & \rho_{X_2,X_3} & \cdots & \rho_{X_2,X_d} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \rho_{X_{d-1},X_1} & \cdots & \rho_{X_{d-1},X_{d-2}} & 1 & \rho_{X_{d-1},X_d} \\ \rho_{X_d,X_1} & \cdots & \rho_{X_d,X_{d-2}} & \rho_{X_d,X_{d-1}} & 1 \end{pmatrix} \tag{3.2}$$

is called *Pearson's correlation matrix* (also: correlation matrix) of random vector $(X_1,\ldots,X_d)^{\mathrm{tr}}$.

Remark 3.1 (Correlation coefficient ρ of a copula). Let $\check{C}_2$ be the 2–copula, so that

$$F_{X_i,X_j}\left(x_i,x_j\right) = \check{C}_2\left(F_{X_i}\left(x_i\right), F_{X_j}\left(x_j\right)\right)$$

holds for any $(x_i, x_j)^{\mathrm{tr}} \in \overline{\mathbb{R}}^2$. Then, correlation coefficient ρ_{X_i,X_j} is calculated by means of 2–copula $\check{C}_2$ as

$$\rho_{X_i,X_j} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{\check{C}_2\left(F_{X_i}(x_i), F_{X_j}(x_j)\right) - \check{C}_2\left(F_{X_i}(x_i), 1\right)\cdot \check{C}_2\left(1, F_{X_j}(x_j)\right)}{\sqrt{\operatorname{Var}(X_i)}\sqrt{\operatorname{Var}(X_j)}} \mathrm{d}x_i \mathrm{d}x_j.$$

The correlation coefficient ρ_{X_i,X_j} (and correlation matrix Σ respectively) is widely used in practice because of its occurrence in the (multivariate) Gaussian distribution. Nevertheless, we have to mind some disadvantages, which come along with the usage of correlation coefficient ρ_{X_i,X_j}.

By definition we know that any correlation matrix has to be positive semi–definite. However, we can easily show that not any symmetric dependence matrix Σ' with $\Sigma'_{i,i} = 1$ and $\Sigma'_{i,j} \in [-1, 1]$, $i \neq j$, is positive semi–definite and is consequently not a proper correlation matrix. For each entry $\Sigma_{i,j} = \rho_{X_i,X_j}$ of a proper correlation matrix Σ there exist bounds $-1 \leq \rho^{\min}_{X_i,X_j} < 0$ and $0 < \rho^{\max}_{X_i,X_j} \leq 1$, so that

$$\rho_{X_i,X_j} \in \left[\rho^{\min}_{X_i,X_j}, \rho^{\max}_{X_i,X_j}\right]$$

holds (cp. [EMS02, Theorem 4]). Thus, if we estimate a dependence matrix Σ' by market data, we will not be able to ensure that the estimated dependence matrix is a proper correlation matrix, i.e. the estimated dependence matrix could be useless. In [Hig02, QS07, BH10] the authors discuss this problem and present efficient approaches to solve this problem.

A further disadvantage of Pearson's correlation coefficient ρ_{X_i,X_j} is its ignorance of non–linear relations between X_i and X_j. It is only able to quantify linear dependences. For a better illustration we introduce the following example.

Example 3.1 (Failures of Pearson's correlation coefficient ρ)**.** Let $X \sim \Phi$ be a normally distributed random variable (mean = 0, variance = 1, cp. Definition 4.1) and let Y be any other real random variable.

(a) If $Y = a \cdot X + b$ holds, with $a > 0$ and $b \in \mathbb{R}$, equality $\rho_{X,Y} = 1$ will result. This result is independent of coefficients a and b and is founded by the perfect positive dependence of X and Y.

(b) If $Y = a \cdot X^3 + b$ holds, with $a > 0$ and $b \in \mathbb{R}$, equality $\rho_{X,Y} = 3/\sqrt{15}$ will result, although the dependence of X and Y is still perfectly positive. However, the dependence additionally involves non–linear relations.

Furthermore, we can observe that property (K3) is not fulfilled, because we can easily show the implication $F_{X,Y}(x,y) = \min\{F_X(x), F_Y(y)\} \not\Rightarrow \rho_{X,Y} = 1$.

A common consideration of the cases (a) and (b) reveals the fact that Pearson's correlation coefficient does not meet the condition (K4) neither (choose for example $a := 1,\ b := 0 \Rightarrow \mathcal{T}(x) = x^3$).

(c) If $Y = X^2$ holds, a further disadvantage of Pearson's correlation coefficient will become obvious. In this case we can state equality $\rho_{X,Y} = 0$, although random variables X and Y are in no way independent. Thus, in general the following implication does not hold: $\rho_{X,Y} = 0 \Rightarrow X$ and Y are independent, which causes a violation of the condition (K5).

If we neglect the violation of the conditions (K3) – (K5) and choose Pearson's correlation coefficient for quantifying the underlying correlation, it will make sense to use (linear) factor models. Hereby, we can significantly reduce the estimation effort.

3.1.1. (Linear) factor models

Let $(X_1, \ldots, X_d)^{\mathrm{tr}}$ be a real random vector. Normally, the estimation of the corresponding correlation matrix $\Sigma \in [-1,1]^{d\times d}$ causes an effort of estimating $d(d-1)/2$ covariances (assuming known variances $\mathrm{Var}\,(X_i)$). However, if the random variables $X_1, \ldots, X_d$ only depend on $k < d$ common factors $F_1, \ldots, F_k$, it will make sense to define these correlation coefficients with a factor model. The usage of factor models reduces the estimation effort and makes the calculation of correlation easier. Popular factor models are *Capital Asset Pricing Model* (cp. [Sha64]), *Arbitrage Pricing Theory* (cp. [Ros76]) and its special case *Fama–French three–factor model* (cp. [FF92, FF93]).

In a linear k–factor model (according to [GS07, Section 3]) we can state

$$X_i := \sum_{j=1}^{k} \beta_{i,j} \cdot F_j + \vartheta_i \epsilon_i \tag{3.3}$$

for each $i \in \{1, \ldots, d\}$, in which notations $\boldsymbol{\beta} := (\beta_{i,j}) \in \mathbb{R}^{d\times k}$, $\vartheta_i \geq 0$ and $\mathrm{Cov}\,(F_j, F_m) = \mathrm{Cov}\,(\epsilon_i, F_j) = \mathrm{Cov}\,(\epsilon_i, \epsilon_l) = 0$ for $j, m \in \{1, \ldots, k\},\ j \neq m$ and $l \in \{1, \ldots, d\},\ l \neq i$, hold. Of course, $F_1, \ldots, F_k$ as well as $\epsilon_1, \ldots, \epsilon_d$ represent real random variables.

Thus, we only have to estimate $k \cdot d + k$ different parameters (matrix $\boldsymbol{\beta}$ and factor variances $\mathrm{Var}\,(F_l)$, assuming known variances $\mathrm{Var}\,(X_i)$) and can calculate the correlation coefficient ρ_{X_i,X_j} as

$$\rho_{X_i,X_j} = \frac{\operatorname{Cov}(X_i, X_j)}{\sqrt{\operatorname{Var}(X_i)}\sqrt{\operatorname{Var}(X_j)}} = \frac{\sum_{l=1}^{k} \beta_{i,l} \cdot \beta_{j,l} \cdot \operatorname{Var}(F_l)}{\sqrt{\operatorname{Var}(X_i)}\sqrt{\operatorname{Var}(X_j)}} = \sum_{l=1}^{k} \rho_{X_i,F_l}\rho_{X_j,F_l}. \tag{3.4}$$

Using the factor model above, correlation matrix Σ has the structure (also known as *k–factor–structure*)

$$\Sigma = \Sigma_k^{\mathrm{F}} := \left(\rho_{X_i,X_j}\right)_{i,j\in\{1,\ldots,d\}} = \operatorname{diag}_d\left(1 - \sum_{l=1}^{k}\psi_{1,l}^2, \ldots, 1 - \sum_{l=1}^{k}\psi_{d,l}^2\right) + \boldsymbol{\psi}\cdot\boldsymbol{\psi}^{\mathrm{tr}}, \tag{3.5}$$

in which notations $\boldsymbol{\psi} \in \mathbb{R}^{d\times k}$ and $\boldsymbol{\psi} = (\psi_{i,l}) := \beta_{i,l} \cdot \frac{\sqrt{\operatorname{Var}(F_l)}}{\sqrt{\operatorname{Var}(X_i)}}$, $i \in \{1,\ldots,d\}$, $l \in \{1,\ldots,k\}$ hold. Parameters $\beta_{i,j}$ are known as the popular *β–factors* or *β–factor loadings.*

The problem of estimating a non–proper correlation matrix is simplified significantly by using a factor model. For example we can ensure that Σ_k^{F} is a proper correlation matrix as long as

$$\forall i \in \{1,\ldots,d\} : \sum_{j=1}^{k}\psi_{i,j}^2 \leq 1$$

holds (cp. [BHR10, Section 1]). In this case correlation matrix Σ_k^{F} is the sum of two positive semi–definite matrices and therefore a positive semi–definite matrix itself.

Example 3.2 (Factor model based on the German stock index DAX®)**.** For a better illustration we introduce an example for estimating stock returns by means of a factor model. For this let $r_{\mathrm{DAX}}, r_{\mathrm{A}}, r_{\mathrm{B}}, r_{\mathrm{E}}$ and r_{S} be the stock returns of the stock index DAXK[(2)] and of the corporations ALV[(3)], BAS[(4)], EOAN[(5)] and SIE[(6)]. In this context the returns refer to price stock quotations, which have to be distinguished from the performance stock quotations. If we calculate the correlation matrix of the random vector $(r_{\mathrm{A}}, r_{\mathrm{B}}, r_{\mathrm{E}}, r_{\mathrm{S}})^{\mathrm{tr}}$ with respect to equation (3.2), we will get

$$\Sigma := \begin{pmatrix} 1.0 & 0.71 & 0.79 & 0.72 \\ 0.71 & 1.0 & 0.60 & 0.81 \\ 0.79 & 0.60 & 1.0 & 0.61 \\ 0.72 & 0.81 & 0.61 & 1.0 \end{pmatrix}.$$

[(2)] DAX® – German stock index (price index)

[(3)] Allianz SE, Königinstraße 28, 80802 München, Germany

[(4)] BASF SE, Carl-Bosch-Straße 38, 67056 Ludwigshafen, Germany

[(5)] E.ON, E.ON Platz 1, 40479 Düsseldorf, Germany

[(6)] Siemens AG, Wittelsbacherplatz 2, 80333 München, Germany

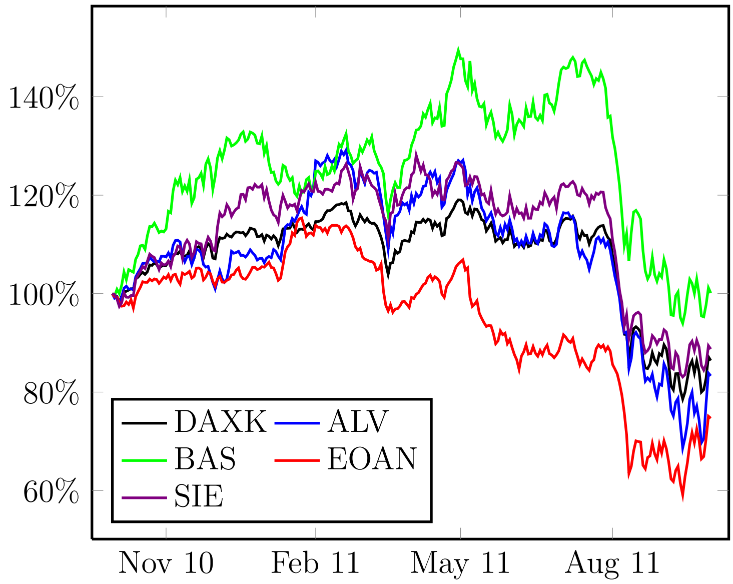


Figure 3.1.: Daily closing prices from 29/09/2010 to 28/09/2011 (cp. Example 3.2), 100% correspond to the particular daily closing price on 29/09/2010.

For this, we used 257 day closing prices from 29/09/2010 to 28/09/2011 (cp. [com11]). Besides this approach it is also possible to model the correlation matrix by means of a factor model. Here we use an one–factor model, in which index return r_{DAX} acts as the only factor. Using this approach in connection with equations (3.3) – (3.5) we obtain the following β–factors and correlation matrix

$$\boldsymbol{\beta} = \begin{pmatrix} 1.23 \\ 1.14 \\ 1.07 \\ 0.98 \end{pmatrix} \qquad \Sigma_1^{\mathrm{F}} := \begin{pmatrix} 1.0 & 0.76 & 0.69 & 0.76 \\ 0.76 & 1.0 & 0.69 & 0.77 \\ 0.69 & 0.69 & 1.0 & 0.69 \\ 0.76 & 0.77 & 0.69 & 1.0 \end{pmatrix} \approx \Sigma.$$

The correlation matrix Σ_1^{F} above is a good approximation of the *real* correlation matrix Σ. We have to keep in mind that we use only one factor and that the time frame equals just one year. Furthermore, the stock returns make a very volatile progress (cp. Figure 3.1) within this time frame, which is due to the economic situation. Finally, we have to add that both correlation matrices are based on historical data. Thus, the question “Which correlation matrix is more suitable for a future pricing of m–th to Default Swaps?” has to be answered carefully.

Besides the effort reduction regarding the estimation, a correlation matrix in factor structure (cp. equation (3.5)) offers several other advantages, for example a simplified evaluation of the multivariate Gaussian distribution (cp. Definition 4.2). In [BHR10] the authors present different approaches for transforming a given correlation matrix to a new approximate correlation matrix showing a factor structure.

3.2. Kendall's Tau τ^{K}

A robust parameter for quantifying correlation, which is able to handle linear relations as well as non–linear relations, is the rank correlation coefficient Kendall's Tau. Let $\left(X_1, \dots, X_d\right)^{\mathrm{tr}}$ and $\left(X_1', \dots, X_d'\right)^{\mathrm{tr}}$ denote two independently and identically distributed random vectors.

Definition 3.2 (Kendall's Tau)**.** The rank correlation coefficient *Kendall's Tau* (abbr.: *Kendall's* $\tau^{\mathrm{K}}_{X_i,X_j}$) with respect to random variables X_i and X_j is given by the probability

$$\tau^{\mathrm{K}}_{X_i,X_j} := \mathbb{P}\left(\left(X_i - X_i'\right) \cdot \left(X_j - X_j'\right) > 0\right) - \mathbb{P}\left(\left(X_i - X_i'\right) \cdot \left(X_j - X_j'\right) < 0\right),$$

in which $\mathbb{P}$ denotes the product probability measure of X_i and X_j, $i, j \in \{1, \dots, d\}$ with $i \neq j$.

Let $\mathcal{S} := \left\{(x_{i,1}, x_{j,1})^{\mathrm{tr}}, \dots, (x_{i,n}, x_{j,n})^{\mathrm{tr}}\right\}$ be a given sample of random vector $(X_i, X_j)^{\mathrm{tr}}$. Then, we can calculate Kendall's $\tau^{\mathrm{K}}_{X_i,X_j}$ as

$$\tau^{\mathrm{K}}_{X_i,X_j} := \frac{|\mathcal{K}| - \left|\overline{\mathcal{K}}\right|}{\binom{n}{2}},$$

in which

$$\mathcal{K} := \left\{\left\{(x_{i,l}, x_{j,l})^{\mathrm{tr}}, (x_{i,m}, x_{j,m})^{\mathrm{tr}}\right\} \subset \mathcal{S} \;\middle|\; (x_{i,l} - x_{i,m}) \cdot (x_{j,l} - x_{j,m}) > 0,\ l < m\right\}$$

denotes the set of all concordant pairs and

$$\overline{\mathcal{K}} := \left\{\left\{(x_{i,l}, x_{j,l})^{\mathrm{tr}}, (x_{i,m}, x_{j,m})^{\mathrm{tr}}\right\} \subset \mathcal{S} \;\middle|\; (x_{i,l} - x_{i,m}) \cdot (x_{j,l} - x_{j,m}) < 0,\ l < m\right\}$$

denotes the set of all discordant pairs.

If we use Kendall's τ^{K} in connection with copulae, we will be able to prove the following theorem easily.

Theorem 3.1 (Kendall's τ^{K} with respect to copulae)**.** *Assuming the joint CDF of random vector $(X_i, X_j)^{tr}$ is given by*

$$\forall\, (x_i, x_j)^{tr} \in \overline{\mathbb{R}}^2 :\ F_{X_i,X_j}(x_i, x_j) = \check{C}_2\left(F_{X_i}(x_i), F_{X_j}(x_j)\right). \tag{3.6}$$

Then, Kendall's $\tau^{K}_{X_i,X_j}$ is calculated via

$$\tau^{K}_{X_i,X_j} := 4 \int_0^1 \int_0^1 \check{C}_2(u, v)\, \mathrm{d}\check{C}_2(u, v) = 4 \int_0^1 \int_0^1 \check{C}_2(u, v) \cdot \check{c}_2(u, v)\, \mathrm{d}u\mathrm{d}v.$$

Proof. See [Nel06, Theorem 5.1.3]. □

Remark 3.2 (Marginal copulae). The constriction to a 2–copula in Theorem 3.1 is not at all a restriction of generality. If we consider d different random variables $X_1, \dots, X_d$, whose joint CDF is given by d–copula $\check{C}_d$, we will be able to construct the corresponding 2–copula from equation (3.6) as

$$\forall v_i, v_j \in [0,1]: \; \check{C}_2(v_i, v_j) = \check{C}_d(1, \dots, 1, \underbrace{v_i}_{i\text{–th pos.}}, 1, \dots, 1, \underbrace{v_j}_{j\text{–th pos.}}, 1, \dots, 1).$$

Reviewing conditions (K1) – (K5) from page 17, Kendall's τ^{K} is obviously able to meet the conditions (K1) and (K2). Furthermore, we can show the validity of the conditions (K3) and (K4) by using [EMS02, Theorem 3]. Nevertheless and analogically to Pearson's correlation coefficient ρ, Kendall's τ^{K} is not able to meet equivalence (K5). Only the direction "The independence of the real random variables X_i and X_j implies $\tau^{\mathrm{K}}_{X_i,X_j} = 0$" holds. For a violation with respect to the other direction see Example 3.1(c).

4. Different types of copulae

In this chapter we analyze different types of copulae and present their properties in detail. Additionally, we discuss the evaluation of different copulae and present algorithms, which sample copula distributed random numbers. First, we introduce two copula bounds.

Lemma 4.1 (Copula bounds)**.** *Let $d \in \mathbb{N}_{\geq 2}$ be any number and let C_d be any d–copula (cp. Definition 1.4). Then, for any vector $\boldsymbol{v} \in [0,1]^d$ inequalities*

$$\max\left(1 - d + \sum_{i=1}^{d} v_i, 0\right) =: L_d\left(\boldsymbol{v}\right) \leq C_d(\boldsymbol{v}) \leq U_d(\boldsymbol{v}) := \min\left(v_1, \ldots, v_d\right)$$

hold. At this, $L_d\left(\boldsymbol{v}\right)$ denotes the "Fréchet-Hoeffding lower bound" and $U_d(\boldsymbol{v})$ denotes the "Fréchet-Hoeffding upper bound". Furthermore, mapping U_d represents a proper d–copula for any $d \in \mathbb{N}_{\geq 2}$, whereas mapping L_d only describes a proper d–copula for the choice $d = 2$.

Proof. See [Nel06, Theorems 2.10.11 and 2.10.12]. □

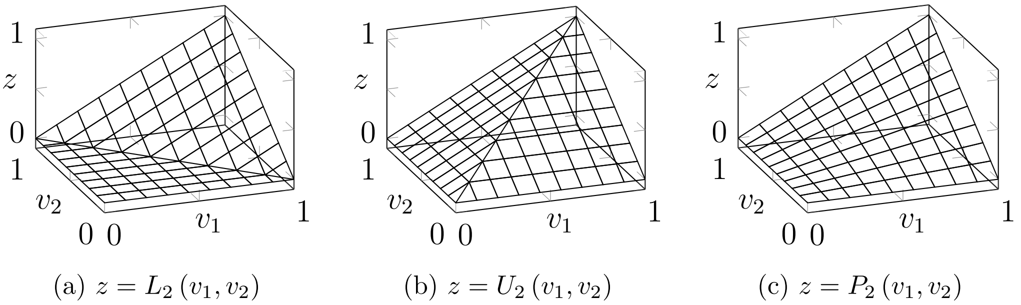
Another copula, which is of high importance in theory as well as in practice, is the well–known *product copula P_d*. The product copula describes a proper copula for any $d \in \mathbb{N}_{\geq 2}$ and maps any vector $\boldsymbol{v} \in [0,1]^d$ to the product of its entries, i.e.

$$P_d\left(\boldsymbol{v}\right) := \prod_{i=1}^{d} v_i.$$

The product copula is often called "independence copula", which is justified by the following lemma.

Lemma 4.2 (Independence copula)**.** *Let $X_1, \ldots, X_d$ be real random variables with CDFs $F_{X_1}, \ldots, F_{X_n}$ and joint CDF $F_{\boldsymbol{X}}$. Then, the following equivalence holds:*

$$\forall \boldsymbol{x} \in \overline{\mathbb{R}}^d : F_{\boldsymbol{X}}\left(x_1, \ldots, x_d\right) = P_d\left(F_{X_1}\left(x_1\right), \ldots, F_{X_d}\left(x_d\right)\right)$$
$$\Leftrightarrow X_1, \ldots, X_d \textit{ are independent.}$$



(a) $z = L_2(v_1, v_2)$ (b) $z = U_2(v_1, v_2)$ (c) $z = P_2(v_1, v_2)$

Figure 4.1.: Fréchet-Hoeffding lower bound $L_2(\boldsymbol{v})$, Fréchet-Hoeffding upper bound $U_2(\boldsymbol{v})$ and product copula $P_2(\boldsymbol{v})$ for any $\boldsymbol{v} \in [0,1]^2$.

Proof. See [Nel06, Theorem 2.10.13]. □

In Figure 4.1 a graphical illustration of the copulae above is shown for the special case $d = 2$.

The most frequently used copula in practice is the well–known Gaussian copula, which is based on the Gaussian (normal) distribution. Alternatively to the Gaussian copula, practitioners often use Archimedean copulae as well. Hence, we briefly introduce both types of copulae in the following sections.

4.1. Gaussian copula

Before we are able to define the Gaussian copula we have to introduce two distributions.

Definition 4.1 (Standard Gaussian distribution)**.** Let $x \in \overline{\mathbb{R}}$ be any number. Then, the mapping

$$\Phi(x) := \int\limits_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^2}{2}\right) \mathrm{d}u$$

is called the CDF of the *standard Gaussian distribution* (also: *standard normal distribution*). Its PDF $\phi(x) := \mathrm{d}\Phi/\mathrm{d}x$ is given by

$$\phi(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right).$$

Remark 4.1 (Evaluation of CDF Φ and its inverse Φ^{-1}). In each of the following algorithms we use the approaches of Hastings (cp. [HWW55, AS70]) and Marsaglia et al. (cp. [MZM94]) for evaluating the CDF Φ of the standard Gaussian distribution. The corresponding inverse CDF Φ^{-1} is evaluated with the help of Moro's algorithm (cp. [Mor95]), which can easily be improved by the application of a few Newton steps (cp. [Gla04, Section 2.3.2]).

Definition 4.2 (Multivariate standard Gaussian distribution)**.** Let the set $\mathcal{K}_d$ be defined as

$$\mathcal{K}_d := \left\{ M \in [-1,1]^{d\times d} \middle| M \text{ pos. semi–def.}, \ M = M^{\text{tr}}, \ \forall i \in \{1,\dots,d\} : M_{ii} = 1 \right\},$$

let $\boldsymbol{x} \in \overline{\mathbb{R}}^d$ be any vector and let $\Sigma \in \mathcal{K}_d$ be any correlation matrix. Then, the mapping

$$\Phi_\Sigma^d(\boldsymbol{x}) := \int_{-\infty}^{x_1} \cdots \int_{-\infty}^{x_d} \frac{1}{(2\pi)^{\frac{d}{2}} \sqrt{\det \Sigma}} \exp\left(-\frac{1}{2}\boldsymbol{v}^{\text{tr}}\Sigma^{-1}\boldsymbol{v}\right) \mathrm{d}v_1 \cdots \mathrm{d}v_d$$

is called the joint CDF of the *d–dimensional standard Gaussian distribution* with Pearson's correlation matrix Σ. The corresponding joint PDF is given by

$$\phi_\Sigma^d(\boldsymbol{x}) := \frac{\partial^d}{\partial x_1 \cdots \partial x_d} \Phi_\Sigma^d(\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{d}{2}} \sqrt{\det \Sigma}} \exp\left(-\frac{1}{2}\boldsymbol{x}^{\text{tr}}\Sigma^{-1}\boldsymbol{x}\right). \tag{4.1}$$

By using the definitions above we are now able to define the Gaussian copula properly:

Definition 4.3 (Gaussian copula)**.** Let $\Sigma \in \mathcal{K}_d$ be any correlation matrix and let $\boldsymbol{u} \in [0,1]^d$ be any vector. Then, the mapping

$$C_{d,\Sigma}^{\text{Gau}}(u_1,\dots,u_d) := \Phi_\Sigma^d\left(\Phi^{-1}(u_1), \dots, \Phi^{-1}(u_d)\right)$$

is a proper d–copula and is called the d–dimensional *Gaussian copula.* Its density function is denoted by $c_{d,\Sigma}^{\text{Gau}}$.

Consequently, the Gaussian copula is continuous as well as radial symmetric (cp. Definition 1.7 and e.g. [ELM03, Section 5.2]). Furthermore, a Gaussian copula creates a linear correlation, which is quantified by Pearson's correlation coefficient and which is based on the underlying correlation matrix Σ. For an illustration of density function $c_{d,\Sigma}^{\text{Gau}}$ we refer to Figure 4.2.

Let X_i und X_j be two real random variables, which are distributed according to the Gaussian copula, i.e. equality $F_{X_i,X_j} = C_{2,\Sigma}^{\text{Gau}}\left(F_{X_i}, F_{X_j}\right)$ holds. Then, we can calculate the corresponding rank correlation coefficient Kendall's $\tau_{X_i,X_j}^{\text{K}}$ by means of

$$\tau_{X_i,X_j}^{\text{K}} := \frac{2}{\pi} \arcsin\left(\rho_{X_i,X_j}\right), \tag{4.2}$$

in which ρ_{X_i,X_j} represents the correlation matrix entry $\rho_{X_i,X_j} = \Sigma_{1,2} = \Sigma_{2,1}$ (cp. [LMS03, Theorem 2]). We have to note that equation (4.2) simply rescales the linear correlation to the parameter Kendall's τ^{K}. The Gaussian copula is still not able to create a non–linear correlation.

Nevertheless, we can easily observe the validity of the following lemma, which exploits the application of the Gaussian copula as a joint CDF

Lemma 4.3 (Special correlation properties of the Gaussian copula)**.** *Let X_i and X_j be real random variables with CDFs F_{X_i} and F_{X_j} and with joint CDF $F_{X_i,X_j} = C^{Gau}_{2,\Sigma}\left(F_{X_i}, F_{X_j}\right)$. Then, Pearson's correlation coefficient $\rho_{X_i,X_j} = \Sigma_{1,2} = \Sigma_{2,1}$ fulfills the conditions (K3) and (K5) from page 17.*

Proof. To prove the validity of the condition (K3), we introduce an alternative definition of the 2–dimensional Gaussian copula as in [JRR04, Section 2]. For this, let $u, v \in [0,1]$ denote numbers and let $\Sigma \in \mathcal{K}_2$ denote any correlation matrix. Then, equality

$$\begin{aligned} C^{\mathrm{Gau}}_{2,\Sigma}(u,v) &= \int\limits_0^u \Phi\left(\frac{\Phi^{-1}(v) - \Sigma_{12}\Phi^{-1}(w)}{\sqrt{1-\Sigma_{12}^2}}\right) \mathrm{d}w \\ &= \int\limits_0^v \Phi\left(\frac{\Phi^{-1}(u) - \Sigma_{12}\Phi^{-1}(w)}{\sqrt{1-\Sigma_{12}^2}}\right) \mathrm{d}w \end{aligned}$$

holds. Now, let $\rho_{X_i,X_j} = 1$ hold. Because the term $C^{\mathrm{Gau}}_{2,\Sigma}$ is not well–defined in this case we have to consider the limit value $\rho_{X_i,X_j} \nearrow 1$. For this, let $u, v \in [0,1]$ be numbers with $u \leq v$. Then, we can easily prove equality

$$\lim_{\rho_{X_i,X_j}\nearrow 1} C^{\mathrm{Gau}}_{2,\Sigma}(u,v) = \lim_{\rho_{X_i,X_j}\nearrow 1} \int\limits_0^u \Phi\left(\frac{\Phi^{-1}(v) - \rho_{X_i,X_j}\Phi^{-1}(w)}{\sqrt{1-\rho^2_{X_i,X_j}}}\right) \mathrm{d}w = u.$$

Analogically, we can prove

$$\lim_{\rho_{X_i,X_j}\nearrow 1} C^{\mathrm{Gau}}_{2,\Sigma}(u,v) = \lim_{\rho_{X_i,X_j}\nearrow 1} \int\limits_0^v \Phi\left(\frac{\Phi^{-1}(u) - \rho_{X_i,X_j}\Phi^{-1}(w)}{\sqrt{1-\rho^2_{X_i,X_j}}}\right) \mathrm{d}w = v,$$

for $u > v$. Thus, we obtain $F_{X_i,X_j}(x_i,x_j) = \min\left\{F_{X_i}(x_i), F_{X_j}(x_j)\right\}$ for any $x_i, x_j \in \overline{\mathbb{R}}$, i.e. property (K3) (comonotony) holds. The derivation of the countermonotony follows analogically (consider the limit value $\rho_{X_i,X_j} \searrow -1$).

Observing Definition 4.2 in detail we can easily see the validity of the condition (K5). Let $\rho_{X_i,X_j} = 0$ hold. Then, we can directly show equivalences

$$\begin{aligned}\rho_{X_i,X_j} = 0 &\Leftrightarrow \Sigma = E_2\\ &\Leftrightarrow \forall x_i, x_j \in \overline{\mathbb{R}} : \Phi_\Sigma^2\left(x_i, x_j\right) = \Phi(x_i) \cdot \Phi(x_j)\\ &\Leftrightarrow \forall u_i, u_j \in [0,1] : C_{2,\Sigma}^{\text{Gau}}\left(u_i, u_j\right) = P_2\left(u_i, u_j\right)\\ &\Leftrightarrow X_i \text{ and } X_j \text{ are independent,}\end{aligned} \tag{4.3}$$

in which E_2 denotes the 2–dimensional identity matrix. For the last equivalence (4.3) we refer to Lemma 4.2. □

Finally, the usage of Pearson's correlation coefficient ρ in connection with the Gaussian copula ensures the validity of the greatest possible number of condition (K1) – (K5) from page 17.

4.1.1. Evaluating the Gaussian copula

Let $X_1, \ldots, X_d$ be real random variables with CDFs $F_{X_1}, \ldots, F_{X_d}$ and joint CDF

$$\begin{aligned}F_{\boldsymbol{X}}\left(x_1, \ldots, x_d\right) &= C_{d,\Sigma}^{\text{Gau}}\left(F_{X_1}\left(x_1\right), \ldots, F_{X_d}\left(x_d\right)\right)\\ &= \Phi_\Sigma^d\left(\Phi^{-1}\left(F_{X_1}\left(x_1\right)\right), \ldots, \Phi^{-1}\left(F_{X_d}\left(x_d\right)\right)\right)\\ = \int_{-\infty}^{\Phi^{-1}\left(F_{X_1}(x_1)\right)} \cdots \int_{-\infty}^{\Phi^{-1}\left(F_{X_d}(x_d)\right)} &\frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{\det \Sigma}} \exp\left(-\frac{1}{2}\boldsymbol{v}^{\text{tr}}\Sigma^{-1}\boldsymbol{v}\right) \mathrm{d}v_1 \cdots \mathrm{d}v_d,\end{aligned} \tag{4.4}$$

in which $\Sigma \in \mathcal{K}_d$ denotes any correlation matrix and $\boldsymbol{x} \in \overline{\mathbb{R}}^d$ denotes any vector. Thus, to evaluate the joint CDF $F_{\boldsymbol{X}}$ we have to evaluate the d–dimensional Gaussian copula. For this, we have to calculate the d–dimensional integral (cp. equation (4.4)), which is known as the multivariate standard Gaussian CDF. Unfortunately, its evaluation is in no case trivial. Only for special cases (cp. [GB09, Section 2]) there exist efficient algorithms for solving this integral. One of them is introduced in the following theorem.

Theorem 4.1 (Evaluating a Gaussian copula with factor structure)**.** *Let $X_1, \ldots, X_d$ be real random variables with joint CDF $F_{\boldsymbol{X}}(\boldsymbol{x}) = C_{d,\Sigma}^{Gau}(F_{X_1}\left(x_1\right), \ldots, F_{X_d}\left(x_d\right))$ and CFDs $F_{X_1}(x_1), \ldots, F_{X_d}(x_d)$ for any vector $\boldsymbol{x} \in \overline{\mathbb{R}}^d$ and any correlation matrix $\Sigma \in \mathcal{K}_d$. Additionally, we assume Σ to show a factor structure (cp. Subsection 3.1.1, especially equation (3.5)), i.e. equation*

$$\Sigma = \Sigma_k^F = diag_d\left(1 - \sum_{l=1}^{k} \psi_{1,l}^2, \ldots, 1 - \sum_{l=1}^{k} \psi_{d,l}^2\right) + \boldsymbol{\psi} \cdot \boldsymbol{\psi}^{tr} =: D + \boldsymbol{\psi} \cdot \boldsymbol{\psi}^{tr}$$

holds for any $\boldsymbol{\psi} \in \mathbb{R}^{d \times k}$ and $k < d$. Then, the joint CDF $F_{\boldsymbol{X}}$ is calculated as

$$
\begin{aligned}
&F_{\boldsymbol{X}}\left(x_1,\ldots,x_d\right) = C^{Gau}_{d,\Sigma_k^F}\left(F_{X_1}\left(x_1\right),\ldots,F_{X_d}\left(x_d\right)\right)\\
&= \int\limits_{-\infty}^{\Phi^{-1}\left(F_{X_1}(x_1)\right)} \cdots \int\limits_{-\infty}^{\Phi^{-1}\left(F_{X_d}(x_d)\right)} \frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{\det\left(\Sigma_k^F\right)}} \exp\left(-\frac{1}{2}\boldsymbol{v}^{tr}\left(\Sigma_k^F\right)^{-1}\boldsymbol{v}\right) \mathrm{d}v_1\cdots\mathrm{d}v_d\\
&= \int\limits_{\mathbb{R}} \cdots \int\limits_{\mathbb{R}} \phi^k_{E_k}(\boldsymbol{y}) \prod_{i=1}^{d} \Phi\left(\frac{\Phi^{-1}\left(F_{X_i}\left(x_i\right)\right) - \sum\limits_{l=1}^{k}\psi_{i,l}\cdot y_l}{\sqrt{D_{i,i}}}\right) \mathrm{d}y_1\cdots\mathrm{d}y_k.
\end{aligned}
$$

Here, mapping $\phi^k_{E_k}$ denotes the joint PDF of the k–dimensional multivariate standard Gaussian distribution (cp. equation (4.1)) and E_k denotes the k–dimensional identity matrix.

Proof. See [CD62, Mar63] or [GB09, Section 2.3.1]. □

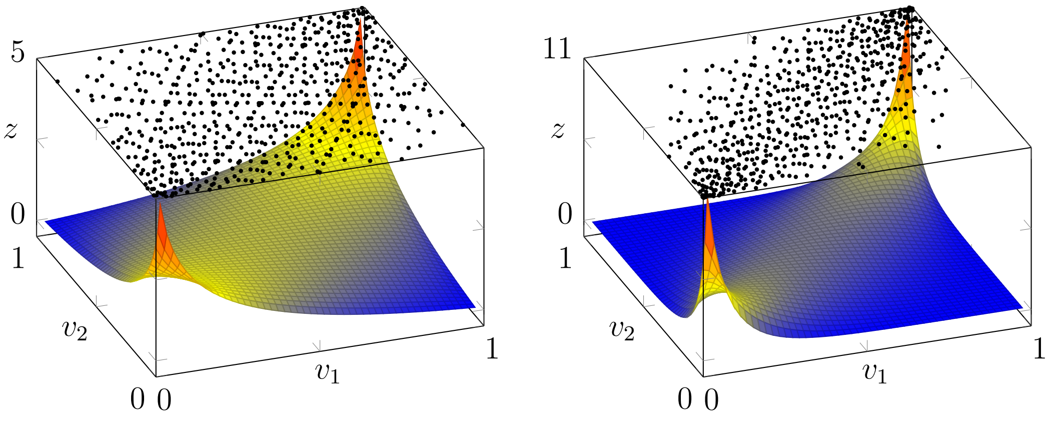
Thus, via Theorem 4.1 it is possible to evaluate the Gaussian copula in higher dimensions efficiently. The only assumption we have to make is the application of a k–factor structure with $k \ll d$ different factors. However, in contexts, in which this assumption is not justified, it is very helpful to draw high–dimensional random tuples, which are distributed according to the Gaussian copula (e.g. for establishing a Monte–Carlo integration).

4.1.2. Drawing random tuples according to the Gaussian copula

Let $X_1,\ldots,X_d$ be real random variables with CDFs $F_{X_1},\ldots,F_{X_d}$ and joint CDF $F_{\boldsymbol{X}}\left(\boldsymbol{x}\right) = C^{\mathrm{Gau}}_{d,\Sigma}\left(F_{X_1}\left(x_1\right),\ldots,F_{X_d}\left(x_d\right)\right)$. Furthermore, let $\boldsymbol{x} \in \overline{\mathbb{R}}^d$ be any vector and let $\Sigma \in \mathcal{K}_d$ be any correlation matrix. The aim of this subsection is drawing a random tuple $\boldsymbol{v} = \left(v_1,\ldots,v_d\right)^{\mathrm{tr}}$, which is distributed according to the Gaussian copula $C^{\mathrm{Gau}}_{d,\Sigma}$. Using this tuple and the transformation $\hat{\boldsymbol{x}} = \left(F^{-1}_{X_1}\left(v_1\right),\ldots,F^{-1}_{X_d}\left(v_d\right)\right)^{\mathrm{tr}}$ we are subsequently able to create a tuple $\hat{\boldsymbol{x}} = \left(\hat{x}_1,\ldots,\hat{x}_d\right)^{\mathrm{tr}}$, which is distributed according to the joint CDF $F_{\boldsymbol{X}}$.

A sequence, which creates tuples $\boldsymbol{v}$, is shown in Algorithm 4.1 (cp. e.g. [ELM03, MV04, Sey12]). For this, we just have to draw an independently and uniformly distributed tuple $\boldsymbol{u} \sim \mathcal{U}[0,1]^d$ and have to calculate the Cholesky decomposition of the correlation matrix Σ. Then, tuple $\boldsymbol{v}$ is calculated by a few matrix–vector multiplications and distribution evaluations. In Figure 4.2 we show 600 different tuples $\boldsymbol{v} = \left(v_1,v_2\right)^{\mathrm{tr}} \sim C^{\mathrm{Gau}}_{2,\Sigma}$.

Remark 4.2 (Uniformly distributed random numbers $u_i \sim \mathcal{U}\left[0,1\right]$). For drawing independently and uniformly distributed random numbers u_i on $[0,1]$ we use two



(a) $z = c^{\text{Gau}}_{2,\Sigma_1}(v_1, v_2)$

(b) $z = c^{\text{Gau}}_{2,\Sigma_2}(v_1, v_2)$

Figure 4.2.: Density function $c^{\text{Gau}}_{2,\Sigma}$ using correlation matrices $\Sigma_1 = \begin{pmatrix} 1.0 & 0.5 \\ 0.5 & 1.0 \end{pmatrix}$ and $\Sigma_2 = \begin{pmatrix} 1.0 & 0.8 \\ 0.8 & 1.0 \end{pmatrix}$. Additionally, 600 tuples $(v_1, v_2)^{\text{tr}} \sim C^{\text{Gau}}_{2,\Sigma_1}$ and $(v_1, v_2)^{\text{tr}} \sim C^{\text{Gau}}_{2,\Sigma_2}$ are shown on top of the particular figure, which were drawn by means of Sobol's sequence.

different methods in this thesis. On the one hand we use the *Mersenne Twister* (cp. [MN98]) for creating "pure" random numbers and on the other hand we use Sobol's sequences (cp. [Sob76]) for creating quasi–random numbers.

Algorithm 4.1: Drawing a tuple $\boldsymbol{v} = (v_1, \ldots, v_d)^{\text{tr}} \sim C^{\text{Gau}}_{d,\Sigma}$

```
Input  : $\boldsymbol{u} = (u_1, \ldots, u_d)^{\text{tr}}$ with $u_i \sim \mathcal{U}[0,1]$
Output: $\boldsymbol{v} = (v_1, \ldots, v_d)^{\text{tr}} \sim C^{\text{Gau}}_{d,\Sigma}$
Global : $d$, $\Sigma$

for $i \leftarrow 1$ to $d$ do
    $z_i \leftarrow \Phi^{-1}(u_i)$ // $z_i \sim \Phi$
$A \leftarrow$Cholesky$(d, \Sigma)$ // $AA^{\text{tr}} = \Sigma$
$\boldsymbol{w} = (w_1, \ldots, w_d)^{\text{tr}} \leftarrow A\boldsymbol{z}$
for $i \leftarrow 1$ to $d$ do
    $v_i \leftarrow \Phi(w_i)$
```

If the chosen correlation matrix Σ additionally shows a factor structure (cp. Subsection 3.1.1, especially equation (3.5)), we will be able to simplify Algorithm 4.1, because we do not have to calculate a Cholesky decomposition anymore. Let us assume Σ shows a k–factor structure like

$$\Sigma = \Sigma^{\text{F}}_k = \text{diag}_d\left(1 - \sum_{l=1}^{k} \psi^2_{1,l}, \ldots, 1 - \sum_{l=1}^{k} \psi^2_{d,l}\right) + \boldsymbol{\psi} \cdot \boldsymbol{\psi}^{\text{tr}} =: D + \boldsymbol{\psi} \cdot \boldsymbol{\psi}^{\text{tr}}.$$

Then, we just have to draw $d+k$ independently and uniformly distributed random numbers u_i and can calculate the desired tuple $\boldsymbol{v} \sim C^{\mathrm{Gau}}_{d,\Sigma^{\mathrm{F}}_k}$ via a few matrix–vector multiplications and distribution evaluations. We have to add that the random variables $F_1, \dots, F_k$ and $\epsilon_1, \dots, \epsilon_d$ from equation (3.3) will be independently and standard normally distributed, if we apply the Gaussian copula with underlying factor structure. The resulting program acts analogically to Algorithm 4.1 and is shown in Algorithm 4.2 (cp. [CG08, Section 5]).

Algorithm 4.2: Drawing a tuple $\boldsymbol{v} = (v_1, \dots, v_d)^{\mathrm{tr}} \sim C^{\mathrm{Gau}}_{d,\Sigma^{\mathrm{F}}_k}$

```
Input  : $\boldsymbol{u}' = (u_1, \dots, u_d, u_{d+1}, \dots, u_{d+k})^{\mathrm{tr}}$ with $u_i \sim \mathcal{U}[0,1]$
Output: $\boldsymbol{v} = (v_1, \dots, v_d)^{\mathrm{tr}} \sim C^{\mathrm{Gau}}_{d,\Sigma^{\mathrm{F}}_k}$
Global : $d$, $\boldsymbol{\psi}$

for $i \leftarrow 1$ to $d+k$ do
    $z_i \leftarrow \Phi^{-1}(u_1)$ // $z_i \sim \Phi$
$\boldsymbol{w} = (w_1, \dots, w_d)^{\mathrm{tr}} \leftarrow \boldsymbol{\psi} \cdot (z_{d+1}, \dots, z_{d+k})^{\mathrm{tr}} + D \cdot (z_1, \dots, z_d)^{\mathrm{tr}}$
for $i \leftarrow 1$ to $d$ do
    $v_i \leftarrow \Phi(w_i)$
```

Because of its intuitive and simple structure, which is based on the (multivariate) standard Gaussian distribution, the Gaussian copula is widely used in practice. Additionally, the Gaussian copula in connection with Pearson's correlation coefficient ρ meets the greatest possible number of conditions (K1) – (K5) (cp. page 17). Nevertheless, the Gaussian copula is not able to map non–linear relations and its efficient evaluation requires restrictions like factor models. Due to this disadvantages we introduce Archimedean copulae in the next section.

4.2. Archimedean copulae

Archimedean copulae represent an own type of copulae. In contrast to the Gaussian copula Archimedean copulae do not consist of nested CDF evaluations, but they are based on a fundamentally different approach. They consist of a chaining sum of completely monotone functions, which can be evaluated easily. Furthermore, they are able to create a non–linear correlation, which is why it is inevitable to use Kendall's τ^{K} for quantifying the prevailing correlation structure.

Definition 4.4 (Completely monotone)**.** A mapping $g : [0,\infty) \mapsto \mathbb{R}$ will be called *completely monotone* on $[0,\infty)$, if it is continuous on $[0,\infty)$ and if additionally

$$\forall k \in \mathbb{N} \cup \{0\} : \forall t \in (0,\infty) : (-1)^k \left. \frac{\mathrm{d}^k}{\mathrm{d}s^k} g(s) \right|_{s=t} \geq 0$$

holds.

This property illustrates enough information for a proper definition of an Archimedean copula.

Definition 4.5 (Archimedean copula)**.** Let $\phi_\theta : [0,1] \mapsto [0,\infty)$, $\theta \in \mathcal{D}_\theta \subseteq \overline{\mathbb{R}}$, be a continuous and strictly decreasing mapping with inverse mapping $\phi_\theta^{-1} : [0,\infty) \mapsto [0,1]$ and the properties $\phi_\theta(0) = \infty$ and $\phi_\theta(1) = 0$. Then, mapping

$$C^{\mathrm{Arc}}_{d,\phi_\theta}(u_1,\ldots,u_d) := \phi_\theta^{-1}\left(\phi_\theta(u_1) + \ldots + \phi_\theta(u_d)\right) \tag{4.5}$$

is a proper d–copula for any $\boldsymbol{u} \in [0,1]^d$, if and only if ϕ_θ^{-1} is completely monotone on $[0,\infty)$ (cp. [Kim74, Nel06]). In this case the d–dimensional mapping $C^{\mathrm{Arc}}_{d,\phi_\theta} : [0,1]^d \mapsto [0,1]$ is called *Archimedean copula* with *generator* ϕ_θ. The corresponding copula density function is denoted by $c^{\mathrm{Arc}}_{d,\phi_\theta}$.

Consequently, Archimedean copulae are continuous, but in general not radial symmetric (cp. Definition 1.7). In order to ensure a huge variety of modelings the generators ϕ_θ are usually defined as families of functions with particular domains $\theta \in \mathcal{D}_\theta$ for parameter θ. We have to note that only within these domains the complete monotony is guaranteed. Widely used examples of Archimedean copulae and their generators are shown in Table 4.1.

Copula	$\mathcal{D}_\theta$	$\phi_\theta(t)$	$\phi_\theta^{-1}(t)$
Clayton $C^{\mathrm{Cla}}_{d,\phi_\theta}$	$(0,\infty)$	$t^{-\theta} - 1$	$(1+t)^{-1/\theta}$
Frank $C^{\mathrm{Fra}}_{d,\phi_\theta}$	$(0,\infty)$	$-\ln\left(\frac{\exp(-\theta t)-1}{\exp(-\theta)-1}\right)$	$-\frac{1}{\theta}\ln\left(1-\left(1-\exp\left(-\theta\right)\right)\exp\left(-t\right)\right)$
Gumbel $C^{\mathrm{Gum}}_{d,\phi_\theta}$	$[1,\infty)$	$\left(-\ln\left(t\right)\right)^\theta$	$\exp\left(-t^{1/\theta}\right)$

Table 4.1.: Examples of Archimedean copulae.

Because generators usually show a closed form, the evaluation of an Archimedean copula is trivial in the most cases and is not discussed here. In Figure 4.3 the density function of the well–known Clayton copula is shown.

For quantifying the correlation structure, which is created by any Archimedean copula, we use the rank correlation coefficient Kendall's τ^{K} (cp. Section 3.2). In connection with Archimedean copulae Kendall's τ^{K} meets the conditions (K1) – (K4) from page 17. In contrast to the usage of Pearson's correlation coefficient ρ in connection with the Gaussian copula, we cannot ensure equivalence (K5) for any Archimedean copula. For calculating Kendall's τ^{K} regarding Archimedean copulae the following lemma is very important.

Lemma 4.4 (Kendall's τ^{K} of any Archimedean copula)**.** *Let X_1 and X_2 be real random variables, whose joint CDF is given by any Archimedean copula C^{Arc}_{2,ϕ_θ} according to Theorem 1.2. Then, the rank correlation coefficient Kendall's $\tau^K_{X_1,X_2}$ is given by*

$$\tau^K_{X_1,X_2} = 1 + 4 \int\limits_0^1 \frac{\phi_\theta(t)}{\phi'_\theta(t)} \mathrm{d}t. \tag{4.6}$$

Proof. Equation (4.6) is implicated by Theorem 3.1 and equation (4.5), see [Nel06, Corollary 5.1.4]. □

With the help of the Lemma 4.4 above it is possible to calculate Kendall's τ^{K} analytically for certain Archimedean copulae (depending on ϕ_θ). Thus, it is also possible to calculate a codomain $\mathcal{I}_{\tau^{\mathrm{K}}}$ for any Archimedean copula representing all possible values of τ^{K}, cp. Table 4.2.

Copula	$\mathcal{D}_\theta$	τ^{K}	$\mathcal{I}_{\tau^{\mathrm{K}}}$
Clayton $C^{\mathrm{Cla}}_{d,\phi_\theta}$	$(0,\infty)$	$\frac{\theta}{\theta+2}$	$(0,1)$
Frank $C^{\mathrm{Fra}}_{d,\phi_\theta}$	$(0,\infty)$	$1-\frac{4}{\theta}\left[1-D_1(\theta)\right]$ (7)	$(0,1)$
Gumbel $C^{\mathrm{Gum}}_{d,\phi_\theta}$	$[1,\infty)$	$1-\frac{1}{\theta}$	$[0,1)$

Table 4.2.: Calculation and codomain of Kendall's τ^{K} with respect to selected Archimedean copulae.

4.2.1. Drawing random tuples according to Archimedean copulae

In this subsection we present efficient algorithms for drawing random tuples $\boldsymbol{v} \sim C^{\mathrm{Arc}}_{d,\phi_\theta}$, which are distributed according to any Archimedean copula (cp. also Subsection 4.1.2 for applying the Gaussian copula). At this, we especially consider the Archimedean copulae from Table 4.1.

(7) cp. [BK06, Table 2], $D_k(\theta)$ denotes the Debye–function (cp. [AS70, Section 27.1])

According to Bernstein's famous Theorem from 1928 (cp. [Fel71, Section XIII.4]) a function $g : [0, \infty) \mapsto \mathbb{R}$ with $g(0) = 1$ is completely monotone (cp. Definition 4.4), if and only if it equals the Laplace–Stieltjes transformation of a CDF G, i.e.

$$\forall y \in \mathbb{R}_{\geq 0} : \ g(y) = \int_0^\infty \exp\left(-yx\right) \mathrm{d}G(x).$$

In this manner we can create an Archimedean copula out of any CDF G and vice versa a CDF G out of any Archimedean copula (cp. [Joe97, Section 4.2]). Furthermore, we are able to establish a very simple algorithm for sampling tuples according to Archimedean copulae (cp. [MO88, Section 5]).

Let $X_1, \ldots, X_d$ be real random variables, whose joint CDF is given by any Archimedean copula $C^{\mathrm{Arc}}_{d,\phi_\theta}$. Then, we can easily see that random variable X_i follows a one–factor model, i.e. $X_1, \ldots, X_d$ are conditionally independent under a known factor V^{Arc}. In addition, we can easily show that factor V^{Arc} is distributed according to CDF F^{Arc}, which is calculated via the inverse Laplace–Stieltjes transformation of the inverse generator ϕ_θ^{-1} of copula $C^{\mathrm{Arc}}_{d,\phi_\theta}$ (cp. [MFE05, McN08]).

For creating the desired tuple $\boldsymbol{v} \sim C^{\mathrm{Arc}}_{d,\phi_\theta}$ we consequently have to draw d independently and uniformly distributed random numbers $u_1, \ldots, u_d \sim \mathcal{U}[0,1]$ and a random factor $\hat{v}^{\mathrm{Arc}} \sim F^{\mathrm{Arc}}$. The corresponding program is shown in Algorithm 4.3 and the required CDF F^{Arc} of the factor V^{Arc} is given in Table 4.3. For sampling a random number $\hat{v}^{\mathrm{Arc}}$ according to CDF F^{Arc} we refer to [Knu00, PFTV07, Nol12]. The corresponding programs are shown in Appendix B as Algorithms B.2 – B.4.

Copula	Factor	CDF F^{Arc} of factor V^{Arc}
$C^{\mathrm{Cla}}_{d,\theta}$	V^{Cla}	Gamma distribution $\mathcal{G}_{1,1/\theta}$ (cp. Def. A.1)
$C^{\mathrm{Fra}}_{d,\theta}$	V^{Fra}	Logarithmic distribution $\mathcal{L}_{1-\exp(-\theta)}$ (cp. Def. A.2)
$C^{\mathrm{Gum}}_{d,\theta}$	V^{Gum}	Stable distribution $\mathcal{S}\left(\frac{1}{\theta}, 1, \left(\cos\left(\frac{\pi}{2\theta}\right)\right)^\theta, 0; 1\right)$ (cp. Def. A.3)

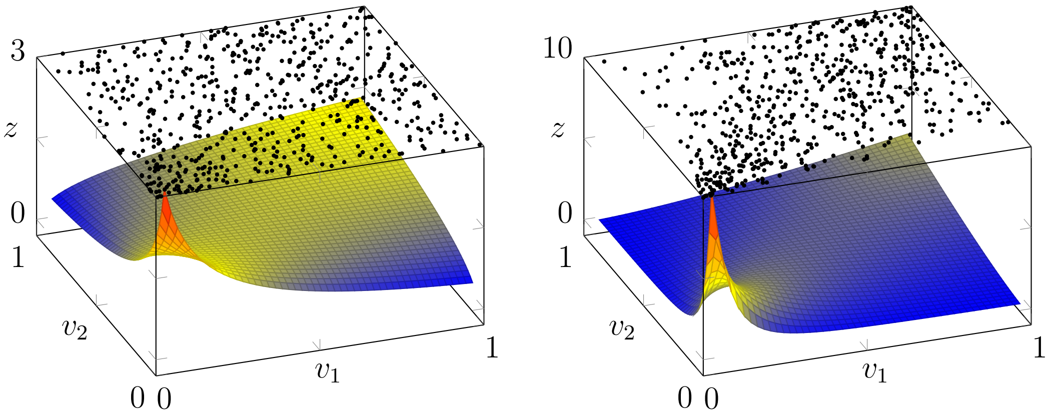
Table 4.3.: CDF of factor V^{Arc} for different Archimedean copulae.

With the help of Algorithm 4.3 we can easily and efficiently draw random tuples distributed to any Archimedean copula. We only have to assume the knowledge of the inverse Laplace-Stieltjes transformation of its inverse generator ϕ_θ^{-1}. A huge list of (inverse) Laplace–Stieltjes transformations is found in [OB73]. In Figure 4.3 600 different random tuples $\boldsymbol{v} = (v_1, v_2)^{\mathrm{tr}} \sim C^{\mathrm{Cla}}_{2,\theta}$ are shown.

Algorithm 4.3: Drawing a tuple $\boldsymbol{v} = (v_1, \dots, v_d)^{\mathrm{tr}} \sim C^{\mathrm{Arc}}_{d,\phi_\theta}$

```
Input  : $\boldsymbol{u} = (u_1, \dots, u_d)^{\mathrm{tr}}$ with $u_i \sim \mathcal{U}[0,1]$
Output: $\boldsymbol{v} = (v_1, \dots, v_d)^{\mathrm{tr}} \sim C^{\mathrm{Arc}}_{d,\phi_\theta}$
Global : $d,\ \theta$

Sample $\hat{v}^{\mathrm{Arc}} \sim F^{\mathrm{Arc}}$ // Assumption:  $F^{\mathrm{Arc}}$ is known
for $i \leftarrow 1$ to $d$ do
    $v_i \leftarrow \phi_\theta^{-1}\left(\frac{-\ln(u_i)}{\hat{v}^{\mathrm{Arc}}}\right)$
```



(a) $z = c^{\mathrm{Cla}}_{2,\theta_1}(v_1, v_2)$

(b) $z = c^{\mathrm{Cla}}_{2,\theta_2}(v_1, v_2)$

Figure 4.3.: Density function $c^{\mathrm{Cla}}_{2,\theta}$ using parameters $\theta_1 = 0.4$ and $\theta_2 = 1.5$. Additionally, 600 tuples $(v_1, v_2)^{\mathrm{tr}} \sim C^{\mathrm{Cla}}_{2,\theta}$ are shown on top of the particular figure, which were drawn by means of Sobol's sequence.

Part II.

Evaluation models based on copulae

In this part we apply the fundamental methods of Part I to different evaluation models based on copulae. Thereby, we distinguish between different types of evaluation models as well as between different types of copulae.

Evaluation models based on copulae are frequently applied in banks, insurances or business consultancies. Especially in risk management and in investment banking they are widely used. Here, their area of application ranges from the modeling of currency–, credit– or liquidity–risks to the evaluation of derivatives of any kind (e.g. credit derivatives, stock or real options).

In any evaluation model a copula is used for modeling the joint CDF of certain random variables (cp. Theorem 1.2). The difference between various evaluation models takes place in the modeling and interacting of the particular MDFs (cp. Section 2.2). The underlying random variables usually represent default times (for modeling credit events). However, it is easily possible to expand a copula based evaluation model to another context like option pricing (cp. Section 6.3).

The first evaluation model based on copulae was presented in [Li00], but it was only applied to the Gaussian copula. Later on, in [SS01], this approach was generalized to the usage of any copula. A detailed survey on different evaluation models in connection with different copulae is found in [Sch03, MFE05].

In this thesis we firstly introduce a static evaluation model acting as a motivation for a more complex model. After this, we present the semi–dynamic evaluation model in detail and give numerous suggestions for improvements. Furthermore, we present applications and some numerical tests according to this model. Finally, we give a brief outlook to the dynamic evaluation model.

5. Motivation: The static evaluation model

Regarding the static evaluation model we model the MDFs as discrete distribution functions with codomains consisting of only three different values. Then, the underlying random variables only possess two different values. If these random variables represent default times of obligors, they will only indicate: “default” or “no default”. We do not consider the exact times of defaults here.

This concept is often used for evaluating homogeneous credit portfolios. In these portfolios each asset is scaled to a fixed maturity T (typically $T = 1$ year) and only the probabilities that defaults occur in this time interval $[0, T]$ are of interest. Based on this information, we can easily calculate statistics like “expected loss” or “Value at Risk”[(8)]. This concept is not applicable for evaluating credit derivatives or similar contracts, in which accurate default times are essential.

Let $[0, T]$ be any fixed time horizon and let $S := \{S_1, \ldots, S_d\}$ be a basket of obligors. We assume that for each obligor S_i a probability of default p_i is known (default within $[0, T]$). Either this probability is calculated via (cp. Section 2.2 and in detail equation (2.2))

$$p_i := 1 - \exp\left(- \int\limits_0^T h_i(u) \mathrm{d}u \right) \tag{5.1}$$

or it is estimated by ratings or other market data (for each $i \in \{1, \ldots, d\}$). Furthermore, we define a random variable X_i, $i \in \{1, \ldots, d\}$, according to equation (5.1) as

$$X_i := \begin{cases} 0, & \text{with probability } 1 - p_i \\ 1, & \text{with probability } p_i \end{cases} . \tag{5.2}$$

This random variable acts as an indicator whether obligor S_i defaults within $[0, T]$ or not. Here, 0 represents “no default” and 1 represents “default”. Consequently,

[(8)] For a definition of the coefficient “Value at Risk” (abbr.: VaR) cp. [Hul09] the CDF of X_i is given by

$$F_{X_i}(x) := \begin{cases} 0, & x < 0 \\ 1 - p_i, & 0 \leq x < 1 \\ 1, & x \geq 1 \end{cases} \tag{5.3}$$

for any $x \in \overline{\mathbb{R}}$. Let $\check{C}_d$ represent the d–copula stating the joint CDF $F_{\boldsymbol{X}}$ by means of Sklar's Theorem 1.2. Then, $F_{\boldsymbol{X}}$ is given by

$$\begin{aligned} F_{\boldsymbol{X}}\left(x_1, \ldots, x_d\right) &= \check{C}_d\left(F_{X_1}\left(x_1\right), \ldots, F_{X_d}\left(x_d\right)\right) \qquad (5.4) \\ &= \check{C}_d\left(\begin{cases} 0, & x_1 < 0 \\ 1 - p_1, & 0 \leq x_1 < 1 \\ 1, & x_1 \geq 1 \end{cases}, \ldots, \begin{cases} 0, & x_d < 0 \\ 1 - p_d, & 0 \leq x_d < 1 \\ 1, & x_d \geq 1 \end{cases}\right) \end{aligned}$$

for any $\boldsymbol{x} \in \overline{\mathbb{R}}^d$. Consequently, the joint CDF is a so called "step function", whose step heights depend on the underlying copula $\check{C}_d$. However, the step coordinates are independent of the chosen copula. Additionally, we have to remark that the copula $\check{C}_d$ needs not to be unique, because its arguments F_{X_i} are not continuous within this evaluation model (cp. Theorem 1.2).

In Figure 5.1 two different joint CDFs are shown. These functions were calculated with equation (5.4) by using two different copulae and $d = 2$. At this, we can easily see that four of five shown surfaces are equivalent by definition. Only the surface $(x_1, x_2)^{\text{tr}} \in [0,1]^2$ depends on the chosen copula.

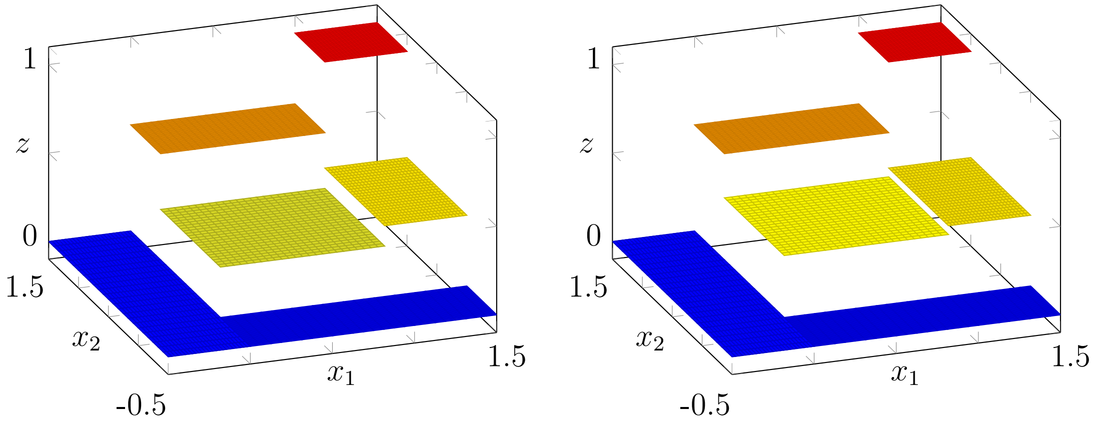
For sampling a tuple $\widehat{\boldsymbol{x}} \sim F_{\boldsymbol{X}}$, which is distributed according to the joint CDF $F_{\boldsymbol{X}}$, we first have to draw a sample $\boldsymbol{v} \sim \check{C}_d$, which is distributed according to the chosen copula $\check{C}_d$. For this, we use Algorithms 4.1 – 4.3 and afterwards we calculate $\widehat{\boldsymbol{x}}$ as

$$\widehat{x}_i := \begin{cases} 0, & v_i < 1 - p_i \\ 1, & v_i \geq 1 - p_i \end{cases}$$

for each $i \in \{1, \ldots, d\}$.

Example 5.1 (Evaluating a credit portfolio). In order to apply the static evaluation model to practice we evaluate a credit portfolio in the following. For this, we use an exemplary credit portfolio given by ifb group AG[(9)]. This portfolio is shown in Table C.5 and consists of 75 obligors from the health care business. In this table each obligor S_i possesses parameters $p_i \in (0,1)$ (probability of default), N_i (nominal value) and R_i (recovery rate).

[(9)] ifb AG, Bayenwerft 14, 50678 Cologne, Germany



(a) $z = C_{2,\Sigma}^{\text{Gau}}\left(F_{X_1}\left(x_1\right), F_{X_2}\left(x_2\right)\right)$

(b) $z = C_{2,\phi_\theta}^{\text{Gum}}\left(F_{X_1}\left(x_1\right), F_{X_2}\left(x_2\right)\right)$

Figure 5.1.: Joint CDF $z := F_{X_1,X_2}\left(x_1, x_2\right)$ according to eq. (5.4) with $d = 2$, $p_1 = 0.4$, $p_2 = 0.6$, $\Sigma_{1,2} = \Sigma_{2,1} = 0.3$ (see (a)), $\theta = 2.0$ (see (b)) and $\left(x_1, x_2\right)^{\text{tr}} \in [-0.5, 1.5]^2$.

Now, let $X_1, \ldots, X_{75}$ denote those random variables, which indicate the defaults of the corresponding obligors $S_1, \ldots, S_{75}$ (cp. equation (5.2)). Their CDFs $F_{X_1}, \ldots, F_{X_{75}}$ are given by equation (5.3) and their joint CDF $F_{\boldsymbol{X}}$ is given by equation (5.4). If X_i indicates the default of obligor S_i, the corresponding loss L_i will be calculated by $L_i\left(X_i\right) = N_i \cdot \left(1 - R_i\right) \cdot X_i$. Consequently, we can calculate the total loss L of S as the sum $L\left(\boldsymbol{X}\right) = \sum_{i=1}^{75} L_i\left(X_i\right)$.

Combining these information we can calculate the expected loss of S as the expectation

$$\mathbb{E}_{\mathbb{P}}\left[L\left(\boldsymbol{X}\right)\right] := \int\limits_{\mathbb{R}^{75}} L\left(\boldsymbol{x}\right) \mathrm{d}F_{\boldsymbol{X}}\left(\boldsymbol{x}\right),$$

in which $\mathbb{P}$ represents the proper product probability measure of random vector $\boldsymbol{X}$.

Copula ↓	Expected loss			VaR		
M →	1000	10000	100000	1000	10000	100000
Gauss	0.2849	0.3028	0.3020	5.0300	5.0576	5.0240
Gumbel	0.2861	0.3032	0.3024	5.7952	7.2282	7.1908

Table 5.1.: Expected loss and Value at Risk regarding the credit portfolio from Table C.5.

In Table 5.1 we calculate the expected loss by means of two different copulae. Furthermore, we calculate the Value at Risk of L with a confidence level of $\alpha = 0.99$. For this, we use a Monte–Carlo simulation, which samples tuples $\widehat{\boldsymbol{x}}$ as described above. We apply M simulations and a homogeneous correlation of $\tau^{\text{K}} = 0.4$. Although the expected losses are similar to each other, it is obvious that the Value at Risks differ significantly. This is caused by the different correlation structures of the chosen copulae.

Due to the fact that credit portfolios normally consist of obligors with a small probability of default ($0 < p_i \ll 1$), more than half of the executed simulations return the value $L = 0$. This causes a high variance of the underlying Monte–Carlo simulation. To reduce the variance, we can easily apply variance reduction techniques, which are introduced in Section 6.2.

6. The semi–dynamic evaluation model

The semi–dynamic evaluation model is rather similar to the static evaluation model discussed in Chapter 5. The crucial difference is the fact that the semi–dynamic evaluation model is based on proper default times and not only on default indicators as the static evaluation model.

The semi–dynamic evaluation model was introduced in [Li00] for the first time and is widely used ever since. Analogically to the static evaluation model, the semi–dynamic evaluation model is used for pricing credit portfolios. Additionally, we can evaluate derivatives with payments depending on individual default times of a given portfolio (e.g. Basket Default Swaps or m-th to Default Swaps).

Let $d \in \mathbb{N}_{\geq 2}$ be any number and let $S = \{S_1, \ldots, S_d\}$ be a basket of obligors. Furthermore, let $\tau_1, \ldots, \tau_d$ denote real random variables, which represent the default times of obligors $S_1, \ldots, S_d$. The CDF of τ_i is modeled easily with an intensity model (cp. [DS03] and equation (2.2)) as

$$\forall t_i \in \overline{\mathbb{R}}_{\geq 0} : F_{\tau_i}(t_i) := 1 - \exp\left(-\int\limits_0^{t_i} h_i(u)\mathrm{d}u\right),\ i \in \{1, \ldots, d\}. \tag{6.1}$$

At this, mapping h_i denotes the hazard rate of S_i and is assumed to be known and deterministic.

Now, let $F_{\boldsymbol{\tau}}$ denote the joint CDF of random vector $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_d)^{\mathrm{tr}}$ and let $\check{C}_d$ be the d–copula that meets

$$\forall \boldsymbol{t} = (t_1, \ldots, t_d)^{\mathrm{tr}} \in \overline{\mathbb{R}}^d_{\geq 0} : F_{\boldsymbol{\tau}}(\boldsymbol{t}) = \check{C}_d\left(F_{\tau_1}(t_1), \ldots, F_{\tau_d}(t_d)\right). \tag{6.2}$$

Then, the value V_m of any m-th to Default Swap with *discounted payoff* $\Lambda_m(t_1, \ldots, t_d)$ equals the expectation

$$V_m := \mathbb{E}_{\mathbb{P}}\left[\Lambda_m(\boldsymbol{\tau})\right] = \int\limits_0^{\infty} \cdots \int\limits_0^{\infty} \Lambda_m(t_1, \ldots, t_d)\,\mathrm{d}F_{\boldsymbol{\tau}}(\boldsymbol{t}), \tag{6.3}$$

in which $\mathbb{P}$ denotes the product probability measure of $\boldsymbol{\tau}$.

Remark 6.1 (Discounted payoff). In this thesis we only consider discounted payoffs. Therefore, we neglect the add–on "discounted" and just write "payoff" in the following.

At first sight we can identify two possibilities for solving the integral above. On the one hand we can use a d–dimensional quadrature, which is very inefficient for huge dimensions d. On the other hand we can use a Monte–Carlo integration creating tuples $\hat{\boldsymbol{t}} \sim F_{\boldsymbol{\tau}}$ as follows:

1. Draw a tuple $\boldsymbol{v} = (v_1, \ldots, v_d)^{\mathrm{tr}} \sim \check{C}_d$, which is distributed according to $\check{C}_d$ (cp. Algorithms 4.1 – 4.3).
2. Calculate $\hat{\boldsymbol{t}} = \left(\hat{t}_1, \ldots, \hat{t}_d\right)^{\mathrm{tr}}$ as $\hat{t}_i = F_{\tau_i}^{-1}(v_i)$ for each $i \in \{1, \ldots, d\}$.

Unfortunately, a Monte–Carlo integration shows a significantly lower order of convergence than a quadrature method. In the following Section 6.1 we present a method, which offers a convergence like a quadrature method and a complexity (for huge dimensions d) like a Monte–Carlo method.

6.1. The distribution of the m–th smallest default time

As already mentioned before, with the semi–dynamic evaluation model we are able to price products, whose value depends on the individual default times or rather on the sequence of defaults. The most popular example for such products is the well–known m–th to Default Swap (abbr.: mBDS, cp. Chapter 2).

Let V_m denote the value of any mBDS with payoff function Λ_m and let $\boldsymbol{\tau}$, F_{τ_i} and $F_{\boldsymbol{\tau}}$ be defined analogically to the introduction above. Then, V_m is given by

$$\begin{aligned} V_m := & \int_0^\infty \cdots \int_0^\infty \Lambda_m(t_1, \ldots, t_d)\, \mathrm{d}F_{\boldsymbol{\tau}}(\boldsymbol{t}) \\ = & \int_0^\infty \cdots \int_0^\infty \Lambda_m(t_1, \ldots, t_d) \cdot f_{\boldsymbol{\tau}}(\boldsymbol{t})\, \mathrm{d}\boldsymbol{t}. \end{aligned} \tag{6.4}$$

Here, mapping $f_{\boldsymbol{\tau}} = \frac{\partial^d}{\partial t_1 \cdots \partial t_d} F_{\boldsymbol{\tau}}$ denotes the joint PDF of $F_{\boldsymbol{\tau}}$.

In order to calculate V_m from equation (6.4), the application of a d–dimensional quadrature seems to be obvious. Unfortunately, this approach causes infeasible effort for huge dimensions d and is therefore neglected in the following. Alternatively, we develop a quadrature approach, which uses the distribution of the m–th smallest default time and is consequently able to reduce the dimension of equation (6.4) significantly.

Let $\boldsymbol{t} = (t_1, \ldots, t_d)^{\mathrm{tr}}$ be a realization of random vector $\boldsymbol{\tau}$ and let $\boldsymbol{\iota}(\boldsymbol{t})$ (abbr.: $\boldsymbol{\iota}$) be a permutation of index set $\{1, \ldots, d\}$, which induces $t_{\iota_1(\boldsymbol{t})} \leq \ldots \leq t_{\iota_d(\boldsymbol{t})}$.

We can observe that payments Λ_m^{DP} and Λ_m^{PP} (cp. Chapter 2) are only depending on the m–th smallest default time t_{ι_m} (we neglect the dependence on R_{ι_m} and N_{ι_m} here). Let $\Lambda_m(t_1, \ldots, t_d)$ be any mBDS payoff function. Then, we can simplify Λ_m to

$$\widetilde{\Lambda}_m(t_{\iota_m}) := \Lambda_m(t_1, \ldots, t_d)$$

for any $\boldsymbol{t} \in \overline{\mathbb{R}}^d_{\geq 0}$.

Remark 6.2 (Dependence on m). In fact, mapping $\widetilde{\Lambda}_m(t_{\iota_m})$ does not depend on m itself, but only on its argument t_{ι_m}. Consequently we define $\widetilde{\Lambda}(t_{\iota_m}) := \widetilde{\Lambda}_m(t_{\iota_m})$.

Definition 6.1 (Perpetual payoff)**.** Obviously, payoff $\Lambda_m(\boldsymbol{t})$ and also payoff $\widetilde{\Lambda}(t_{\iota_m})$ remain constant for any tuple $\boldsymbol{t} = (t_1, \ldots, t_d)^{\mathrm{tr}} \in \overline{\mathbb{R}}^d_{\geq 0}$ meeting $t_{\iota_m} > T$. Thus, we define

$$\Lambda_{m,\infty} := \Lambda_m(\boldsymbol{t})$$

for any $\boldsymbol{t} \in (T, \infty]^d$ and

$$\widetilde{\Lambda}_\infty := \widetilde{\Lambda}(t)$$

for any $t \in (T, \infty]$. Especially the limit values $\lim_{t \nearrow \infty} \widetilde{\Lambda}(t)$ and $\lim_{t \searrow T} \widetilde{\Lambda}(t)$ are included. Despite the different notation, $\Lambda_{m,\infty} = \widetilde{\Lambda}_\infty$ holds, of course.

Let τ_{ι_m} denote the real random variable, which represents the m–th smallest default time within S. In addition, let $F_{\tau_{\iota_m}}$ denote its CDF and $f_{\tau_{\iota_m}}$ its PDF. The aim of this section is the development of CDF $F_{\tau_{\iota_m}}$ and the simplification of any mBDS evaluation by means of

$$V_m = \int_0^\infty \cdots \int_0^\infty \Lambda_m(t_1, \ldots, t_d)\, \mathrm{d}F_{\boldsymbol{\tau}}(\boldsymbol{t}) \overset{?}{=} \int_0^\infty \widetilde{\Lambda}(t) \mathrm{d}F_{\tau_{\iota_m}}(t). \tag{6.5}$$

In order to prove equation (6.5) we have to derive the CDF $F_{\tau_{\iota_m}}$ first, which is done by means of the famous principle of inclusion and exclusion, see the following lemma.

Lemma 6.1 (Principle of inclusion and exclusion)**.** *Let M_i denote the set $\{\tau_i \leq t\}$ (short form of $\{\omega_i \in \Omega_i | \tau_i(\omega_i) \leq t\}$) for any fixed $t \in [0, \infty]$ and $i \in \{1, \ldots, d\}$. Then, the CDF $F_{\tau_{\iota_m}}$ of random variable τ_{ι_m} is given by*

$$\begin{aligned} F_{\tau_{\iota_m}}(t) &= \mathbb{P}\left(\bigcup_{l=1}^{\binom{d}{m}} \bigcap_{\substack{i \in I_l, |I_l| = m \\ I_l \subseteq \{1,\ldots,d\}}} M_i \right) = \sum_{k=m}^{d} (-1)^{k-m} \binom{k-1}{m-1} \sum_{\substack{I \subseteq \{1,\ldots,d\} \\ |I| = k}} \mathbb{P}\left(\bigcap_{i \in I} M_i \right) \\ &= \sum_{k=m}^{d} (-1)^{k-m} \binom{k-1}{m-1} \sum_{\substack{I \subseteq \{1,\ldots,d\} \\ |I| = k}} F_{\boldsymbol{\tau}_I}(\boldsymbol{t}_k) \\ &= \sum_{k=m}^{d} (-1)^{k-m} \binom{k-1}{m-1} \sum_{\substack{I \subseteq \{1,\ldots,d\} \\ |I| = k}} \check{C}_k^I \left(F_{\tau_{I_1}}(t), \ldots, F_{\tau_{I_k}}(t) \right), \end{aligned}$$

in which notations $\boldsymbol{\tau}_I := (\tau_j)_{j \in I}$ and $\boldsymbol{t}_k = (t, \ldots, t)^{tr} \in \overline{\mathbb{R}}_{\geq 0}^k$ hold.

Proof. See pages 50ff.. □

Now, we are able to prove equality (6.5) with the help of Lemma 6.1. In order to avoid the cumbersome calculation of $f_{\tau_{\iota_m}} = \frac{\mathrm{d}}{\mathrm{d}t} F_{\tau_{\iota_m}}$ we introduce the following proposition.

Proposition 6.1 (Integration by parts)**.** Let $\widetilde{\Lambda}(t)$ be continuously differentiable on $(0, \infty)$. Then, we can simplify

$$\int_0^\infty \widetilde{\Lambda}(t) \mathrm{d}F_{\tau_{\iota_m}}(t) = \int_0^\infty \widetilde{\Lambda}(t) \cdot f_{\tau_{\iota_m}}(t) dt = \widetilde{\Lambda}(t) \cdot F_{\tau_{\iota_m}}(t) \Big|_0^\infty - \int_0^\infty \widetilde{\Lambda}'(t) \cdot F_{\tau_{\iota_m}}(t) \mathrm{d}t \quad (6.6)$$

by means of integration by parts. Additionally, we know that payoff $\widetilde{\Lambda}(t)$ is constant for any $t > T$ (cp. Definition 6.1). Thus, we can further simplify equation (6.6) to

$$\begin{aligned} \int_0^\infty \widetilde{\Lambda}(t) \mathrm{d}F_{\tau_{\iota_m}}(t) &= \widetilde{\Lambda}(t) \cdot F_{\tau_{\iota_m}}(t) \Big|_0^\infty - \int_0^\infty \widetilde{\Lambda}'(t) \cdot F_{\tau_{\iota_m}}(t) \mathrm{d}t \\ &= \lim_{t \nearrow \infty} \widetilde{\Lambda}(t) \cdot \lim_{t \nearrow \infty} F_{\tau_{\iota_m}}(t) - \widetilde{\Lambda}(0) \cdot F_{\tau_{\iota_m}}(0) - \int_0^T \widetilde{\Lambda}'(t) \cdot F_{\tau_{\iota_m}}(t) \mathrm{d}t \\ &= \widetilde{\Lambda}_\infty - \int_0^T \widetilde{\Lambda}'(t) \cdot F_{\tau_{\iota_m}}(t) \mathrm{d}t. \end{aligned} \quad (6.7)$$

Remark 6.3 (Continuous differentiability)*.* To assume continuous differentiability of $\widetilde{\Lambda}$ on $(0, \infty)$ does not imply any restrictions. Usually, the payoff of an mBDS is at least piecewise continuously differentiable. Consequently, there exists any $\gamma \in \mathbb{N}$ and a partition $\{t_0 = 0, t_1, \ldots, t_\gamma \leq T, t_{\gamma+1} = \infty\}$, so that $\widetilde{\Lambda}$ is continuously differentiable on (t_i, t_{i+1}) for each $i \in \{0, \dots, \gamma\}$. Then, we can define the mapping $\widetilde{\Lambda}_i' : [t_i, t_i + 1] \mapsto \mathbb{R},\ i \in \{0, \dots, \gamma\}$ as

$$\widetilde{\Lambda}_i'(t) := \begin{cases} \lim\limits_{h \searrow 0} \frac{\widetilde{\Lambda}(t_i+h) - \lim\limits_{t \searrow t_i} \widetilde{\Lambda}(t)}{h}, & t = t_i \\ \frac{\mathrm{d}}{\mathrm{d}t}\widetilde{\Lambda}(t), & t \in (t_i, t_{i+1}) \\ \lim\limits_{h \searrow 0} \frac{\lim\limits_{t \nearrow t_{i+1}} \widetilde{\Lambda}(t) - \widetilde{\Lambda}(t_{i+1}-h)}{h}, & t = t_{i+1} \end{cases}$$

and can consequently write

$$\begin{aligned} V_m &= \int\limits_0^\infty \widetilde{\Lambda}(t) \cdot f_{\tau_{\iota_m}}(t)\mathrm{d}t \\ &= \sum_{i=0}^{\gamma} \lim_{t_l \searrow t_i} \lim_{t_u \nearrow t_{i+1}} \int\limits_{t_l}^{t_u} \widetilde{\Lambda}(t) \cdot f_{\tau_{\iota_m}}(t)\mathrm{d}t \\ &= \sum_{i=0}^{\gamma} \lim_{t_l \searrow t_i} \lim_{t_u \nearrow t_{i+1}} \widetilde{\Lambda}\left(t\right) F_{\tau_{\iota_m}}\left(t\right)\Big|_{t_l}^{t_u} - \int\limits_{t_i}^{t_{i+1}} \widetilde{\Lambda}_i'(t) \cdot F_{\tau_{\iota_m}}(t)\mathrm{d}t \\ &= \sum_{i=1}^{\gamma} \left(\lim_{t_u \nearrow t_i} \widetilde{\Lambda}\left(t_u\right) - \lim_{t_l \searrow t_i} \widetilde{\Lambda}\left(t_l\right) \right) F_{\tau_{\iota_m}}\left(t_i\right) - \int\limits_{t_{i-1}}^{t_i} \widetilde{\Lambda}_i'(t) \cdot F_{\tau_{\iota_m}}(t)\mathrm{d}t + \widetilde{\Lambda}_\infty. \end{aligned}$$

The last equality holds as long as the MDFs F_{τ_i} and d–copula $\check{C}_d$ are continuous.

Example 6.1 (Piecewise continuously differentiable payoff)**.** Let V_m be the value of an mBDS with payoff $\widetilde{\Lambda}$ and maturity T. Furthermore, we assume $\widetilde{\Lambda}$ to be not continuously differentiable at $0 < t_1 < t_2 = T$. Then, we can calculate V_m as

$$\begin{aligned} V_m &= \int\limits_0^\infty \widetilde{\Lambda}(t) \cdot f_{\tau_{\iota_m}}(t)dt \\ &= \widetilde{\Lambda}_\infty + \left(\lim_{t_u \nearrow t_1} \widetilde{\Lambda}\left(t_u\right) - \lim_{t_l \searrow t_1} \widetilde{\Lambda}\left(t_l\right) \right) F_{\tau_{\iota_m}}\left(t_1\right) - \int\limits_0^{t_1} \widetilde{\Lambda}_i'(t) \cdot F_{\tau_{\iota_m}}(t)\mathrm{d}t \\ &\quad + \left(\lim_{t_u \nearrow T} \widetilde{\Lambda}\left(t_u\right) - \lim_{t_l \searrow T} \widetilde{\Lambda}\left(t_l\right) \right) F_{\tau_{\iota_m}}\left(T\right) - \int\limits_{t_1}^{T} \widetilde{\Lambda}_i'(t) \cdot F_{\tau_{\iota_m}}(t)\mathrm{d}t, \end{aligned}$$

in which the notation from Remark 6.3 holds.

Using Lemma 6.1 in connection with Proposition 6.1, we are finally able to prove equation (6.5) (cp. the following Theorem 6.1). Due to the unspecified copula $\check{C}_d$, this result is free of any model risk.

Theorem 6.1 (Evaluation of m-th to Default Swaps)**.** *The value V_m of any m-th to Default Swap with continuously differentiable payoff $\widetilde{\Lambda}$ and maturity T is given by*

$$V_m = \widetilde{\Lambda}_\infty - \int\limits_0^T \widetilde{\Lambda}'(t) \cdot F_{\tau_{\iota_m}}(t) \mathrm{d}t, \tag{6.8}$$

in which $F_{\tau_{\iota_m}}(t)$ is given explicitly by Lemma 6.1.

Proof. See below. □

In the following we prove Lemma 6.1 as well as Theorem 6.1 starting with the special case $m = 1$.

Proofs of Lemma 6.1 and Theorem 6.1

Proof (Lemma 6.1, $m = 1$). We have to show that

$$F_{\tau_{\iota_1}}(t) = \mathbb{P}\left(\bigcup_{l=1}^{d} M_i\right) = \sum_{k=1}^{d} (-1)^{k-1} \sum_{\substack{I \subseteq \{1,\dots,d\} \\ |I|=k}} \mathbb{P}\left(\bigcap_{i\in I} M_i\right)$$

holds. The first equality is trivially true because we can write

$$\begin{aligned} F_{\tau_{\iota_1}}(t) &= \mathbb{P}\left(\tau_{\iota_1} \le t\right) = \mathbb{P}\left((\tau_1 \vee \ldots \vee \tau_d) \le t\right) \\ &= \mathbb{P}\left(\{\tau_1 \le t\} \cup \ldots \cup \{\tau_d \le t\}\right) = \mathbb{P}\left(\bigcup_{i=1}^{d} M_i\right). \end{aligned}$$

The second equality is known as the famous sieve formula of Poincaré and Sylvester, cp. [Tak67, Theorem 2] or [Gal11, Chapter 4]. □

Proof (Theorem 6.1, $m = 1$). According to equations (6.7) and (6.8) we have to show

$$\int\limits_0^\infty \cdots \int\limits_0^\infty \Lambda_m(t_1,\dots,t_d) \cdot f_{\tau_1,\dots,\tau_d}(t_1,\dots,t_d) \mathrm{d}t_1 \cdots \mathrm{d}t_d = \int\limits_0^\infty \widetilde{\Lambda}(t) \cdot f_{\tau_{\iota_m}}(t) \mathrm{d}t.$$

For a better illustration we only consider the special case $d = 2$ in the following. The general case $d > 2$ is proven analogically.

$$\begin{aligned} \int\limits_0^\infty \int\limits_0^\infty \Lambda_1(t_1,t_2) \cdot f_{\tau_1,\tau_2}(t_1,t_2) \mathrm{d}t_1 \mathrm{d}t_2 &= \int\limits_0^\infty \int\limits_{t_1}^\infty \Lambda_1(t_1,t_2) \cdot f_{\tau_1,\tau_2}(t_1,t_2) \mathrm{d}t_2 \mathrm{d}t_1 \\ &\quad + \int\limits_0^\infty \int\limits_{t_2}^\infty \Lambda_1(t_1,t_2) \cdot f_{\tau_1,\tau_2}(t_1,t_2) \mathrm{d}t_1 \mathrm{d}t_2 \end{aligned}$$

$$
\begin{aligned}
&= \int_0^\infty \widetilde{\Lambda}(t_1) \int_{t_1}^\infty f_{\tau_1,\tau_2}(t_1,t_2)\mathrm{d}t_2\mathrm{d}t_1 + \int_0^\infty \widetilde{\Lambda}(t_2) \int_{t_2}^\infty f_{\tau_1,\tau_2}(t_1,t_2)\mathrm{d}t_1\mathrm{d}t_2 \\
&= \int_0^\infty \widetilde{\Lambda}(t_1) \left[\int_0^\infty f_{\tau_1,\tau_2}(t_1,t_2)\mathrm{d}t_2 - \int_0^{t_1} f_{\tau_1,\tau_2}(t_1,t_2)\mathrm{d}t_2 \right] \mathrm{d}t_1 \\
&\qquad + \int_0^\infty \widetilde{\Lambda}(t_2) \left[\int_0^\infty f_{\tau_1,\tau_2}(t_1,t_2)\mathrm{d}t_1 - \int_0^{t_2} f_{\tau_1,\tau_2}(t_1,t_2)\mathrm{d}t_1 \right] \mathrm{d}t_2 \\
&= \int_0^\infty \widetilde{\Lambda}(t_1) f_{\tau_1}(t_1)\mathrm{d}t_1 - \int_0^\infty \widetilde{\Lambda}(t_1) \int_0^{t_1} f_{\tau_1,\tau_2}(t_1,t_2)\mathrm{d}t_2\mathrm{d}t_1 \\
&\qquad + \int_0^\infty \widetilde{\Lambda}(t_2) f_{\tau_2}(t_2)\mathrm{d}t_2 - \int_0^\infty \widetilde{\Lambda}(t_2) \int_0^{t_2} f_{\tau_1,\tau_2}(t_1,t_2)\mathrm{d}t_1\mathrm{d}t_2 \\
&= \int_0^\infty \widetilde{\Lambda}(t)\mathrm{d}F_{\tau_1}(t) - \int_0^\infty \widetilde{\Lambda}(t_1) \left. \frac{\partial F_{\tau_1,\tau_2}(t_1,t_2)}{\partial t_1} \right|_{t_2=t_1} \mathrm{d}t_1 \\
&\qquad + \int_0^\infty \widetilde{\Lambda}(t)\mathrm{d}F_{\tau_2}(t) - \int_0^\infty \widetilde{\Lambda}(t_2) \left. \frac{\partial F_{\tau_1,\tau_2}(t_1,t_2)}{\partial t_2} \right|_{t_1=t_2} \mathrm{d}t_2 \\
&= \int_0^\infty \widetilde{\Lambda}(t)\mathrm{d}F_{\tau_1}(t) + \int_0^\infty \widetilde{\Lambda}(t)\mathrm{d}F_{\tau_2}(t) \\
&\qquad - \int_0^\infty \widetilde{\Lambda}(t) \left[\left. \frac{\partial F_{\tau_1,\tau_2}(t,t_2)}{\partial t} \right|_{t_2=t} + \left. \frac{\partial F_{\tau_1,\tau_2}(t_1,t)}{\partial t} \right|_{t_1=t} \right] \mathrm{d}t \\
&= \int_0^\infty \widetilde{\Lambda}(t)\mathrm{d}F_{\tau_1}(t) + \int_0^\infty \widetilde{\Lambda}(t)\mathrm{d}F_{\tau_2}(t) - \int_0^\infty \widetilde{\Lambda}(t)\mathrm{d}F_{\tau_1,\tau_2}(t,t) \\
&= \int_0^\infty \widetilde{\Lambda}(t) \cdot d\left[F_{\tau_1}(t) + F_{\tau_2}(t) - F_{\tau_1,\tau_2}(t,t)\right] \\
&\overset{\text{Lemma 6.1}}{=} \int_0^\infty \widetilde{\Lambda}(t) \cdot \mathrm{d}F_{\tau_{\iota_1}}(t) = \int_0^\infty \widetilde{\Lambda}(t) \cdot f_{\tau_{\iota_1}}(t)\mathrm{d}t
\end{aligned}
$$

□

Example 6.2 (First to Default Swap from [Li00]). Let V_1^{Li} be the value of the First to Default Swap given in [Li00, Exhibit 6]. For this, let $\tau_1, \ldots, \tau_d$ be independent random variables and let each hazard rate $h_i(t)$ be constant and equal to each other, $h :\equiv h_i(t)$. The underlying payoff is denoted by $\widetilde{\Lambda}^{\mathrm{Li}}$ and is defined as $\widetilde{\Lambda}^{\mathrm{Li}}(t) = 1 \cdot \exp\left(-rt\right)$. Li stated the value V_1^{Li} as

$$
V_1^{\mathrm{Li}} = \frac{dh}{r+dh}\left(1 - \exp(-T(r+dh))\right), \tag{6.9}
$$

in which r denotes a constant risk free interest rate and T denotes any maturity. In the following this analytical formula (6.9) is proven by using Lemma 6.1 and Theorem 6.1.

First, the joint CDF of random vector $\boldsymbol{\tau}$ is given by means of the product copula (cp. Lemma 4.2) as

$$\begin{aligned} F_{\boldsymbol{\tau}}(\boldsymbol{t}) &= \check{C}_d\left(F_{\tau_1}(t_1), \ldots, F_{\tau_d}(t_d)\right) \\ &= P_d\left(F_{\tau_1}(t_1), \ldots, F_{\tau_d}(t_d)\right) \\ &= \prod_{i=1}^{d} F_{\tau_i}(t_i) \end{aligned}$$

for any $\boldsymbol{t} \in \overline{\mathbb{R}}^d_{\geq 0}$. Due to the usage of constant hazard rates $h_i(t) \equiv h$ we obtain the MDFs

$$F(t) := F_{\tau_1}(t) = \ldots = F_{\tau_d}(t) = 1 - \exp(-ht)$$

for any $t \in \overline{\mathbb{R}}_{\geq 0}$. Thus, Lemma 6.1 results in

$$\begin{aligned} F_{\tau_{\iota_1}}(t) &= \sum_{k=1}^{d} (-1)^{k-1} \sum_{\substack{I \subseteq \{1,\ldots,d\} \\ |I|=k}} \mathbb{P}\left(\bigcap_{i\in I} \{\tau_i \leq t\}\right) \\ &= \sum_{k=1}^{d} (-1)^{k-1} \sum_{\substack{I \subseteq \{1,\ldots,d\} \\ |I|=k}} P_k^I\left(F(t), \ldots, F(t)\right) \\ &= \sum_{k=1}^{d} (-1)^{k-1} \sum_{\substack{I \subseteq \{1,\ldots,d\} \\ |I|=k}} F(t)^k = \sum_{k=1}^{d} (-1)^{k-1} \binom{d}{k} F(t)^k \\ &= -\sum_{k=0}^{d} \binom{d}{k} (1)^{d-k} (-F(t))^k + 1 = 1 - (1 - F(t))^d \end{aligned}$$

and

$$\begin{aligned} f_{\tau_{\iota_1}}(t) &= d(1 - F(t))^{d-1} \frac{\mathrm{d}F(t)}{\mathrm{d}t} = d(\exp(-ht))^{d-1} h \exp(-ht) \\ &= hd \exp(-dht) \end{aligned}$$

holds. Finally, the usage of Theorem 6.1 leads to

$$\begin{aligned} V_1^{\mathrm{Li}} &= \int_0^T \exp(-rt) \cdot f_{\tau_{\iota_1}}(t)\mathrm{d}t = dh \int_0^T \exp(-t(r + dh))\mathrm{d}t \\ &= \frac{dh}{r + dh}\left(1 - \exp(-T(r + dh))\right), \end{aligned}$$

which is equivalent to equation (6.9). Alternatively to the approach above, we can avoid the calculation of $f_{\tau_{\iota_1}}$ by means of Proposition 6.1.

After considering this short example we continue with the proofs of Lemma 6.1 and Theorem 6.1 by considering the general case of $m \in \{1, \ldots, d\}$.

Proof (Lemma 6.1, $m \in \{1,\dots,d\}$). In order to prove Lemma 6.1 for the general case of $m \in \{1,\dots,d\}$ we have to show

$$F_{\tau_{\iota_m}}(t) = \mathbb{P}\left(\bigcup_{l=1}^{\binom{d}{m}} \bigcap_{\substack{i\in I_l, |I_l|=m \\ I_l\subseteq\{1,\dots,d\}}} M_i\right) = \sum_{k=m}^{d} (-1)^{k-m}\binom{k-1}{m-1} \sum_{\substack{I\subseteq\{1,\dots,d\} \\ |I|=k}} \mathbb{P}\left(\bigcap_{i\in I} M_i\right),$$

in which M_i denotes the set $\{\tau_i \le t\}$ again. The first equality is proven in the same way as special case $m = 1$ and is also trivially true. The second equality is proven by means of the *general probability theorem* of [Tak67, Theorem 2] or [Gal11, Chapter 4], which states the probability of observing **exactly** m defaults within d obligors by

$$\begin{aligned}\mathbb{P}\left(\#\text{ of defaults} = m\right) &= \mathbb{P}\left(\dot{\bigcup_{l\in\left\{1,\dots,\binom{d}{m}\right\}}} \bigcap_{\substack{i\in I_l, |I_l|=m \\ I_l\subseteq\{1,\dots,d\}}} M_i\right) \\ &= \sum_{k=m}^{d} (-1)^{k-m}\binom{k}{m} \sum_{\substack{I\subseteq\{1,\dots,d\} \\ |I|=k}} \mathbb{P}\left(\bigcap_{i\in I} M_i\right), \qquad (6.10)\end{aligned}$$

in which $\dot{\bigcup}$ denotes a disjoint union. Consequently, for calculating the probability of observing **at least** m defaults within d obligors we only have to sum up equation (6.10). This leads to

$$\begin{aligned}&\mathbb{P}\left(\tau_{\iota_m} \le t\right) = \mathbb{P}\left(\bigcup_{l=1}^{\binom{d}{m}} \bigcap_{\substack{i\in I_l, |I_l|=m \\ I_l\subseteq\{1,\dots,d\}}} M_i\right) = \sum_{j=m}^{d} \mathbb{P}(\#\text{ of defaults} = j) \\ &= \sum_{j=m}^{d}\sum_{k=j}^{d}(-1)^{k-j}\binom{k}{j} \sum_{\substack{I\subseteq\{1,\dots,d\} \\ |I|=k}} \mathbb{P}\left(\bigcap_{i\in I} M_i\right) \\ &= \sum_{k=m}^{d}\sum_{j=m}^{k}(-1)^{k-j}\binom{k}{j} \sum_{\substack{I\subseteq\{1,\dots,d\} \\ |I|=k}} \mathbb{P}\left(\bigcap_{i\in I} M_i\right) \\ &= \sum_{k=m}^{d} c_{k,m} \sum_{\substack{I\subseteq\{1,\dots,d\} \\ |I|=k}} \mathbb{P}\left(\bigcap_{i\in I} M_i\right),\end{aligned}$$

with

$$\begin{aligned}c_{k,m} := \sum_{j=m}^{k}(-1)^{k-j}\binom{k}{j} &= (-1)^{k-m}\binom{k}{m} + 1 + \sum_{j=m+1}^{k-1}(-1)^{k-j}\binom{k}{j} \\ &= (-1)^{k-m}\binom{k}{m} + 1 + \sum_{j=m+1}^{k-1}(-1)^{k-j}\left(\binom{k-1}{j-1} + \binom{k-1}{j}\right) \\ &= (-1)^{k-m}\binom{k}{m} + 1 + (-1)^{k-(m+1)}\binom{k-1}{m} - 1\end{aligned}$$

$$= (-1)^{k-m}\left(\binom{k}{m} - \binom{k-1}{m}\right) = (-1)^{k-m}\binom{k-1}{m-1}.$$

□

Proof (Theorem 6.1, $m \in \{1, \ldots, d\}$). In order to complete the proof of Theorem 6.1 we have to show

$$\int\limits_0^\infty \cdots \int\limits_0^\infty \Lambda_m(t_1, \ldots, t_d) \cdot f_{\boldsymbol{\tau}}(t_1, \ldots, t_d)\mathrm{d}t_1 \cdots \mathrm{d}t_d = \int\limits_0^\infty \widetilde{\Lambda}(t) \cdot f_{\tau_{\iota_m}}(t)\mathrm{d}t.$$

This can be shown by applying Lemma 6.1 with $m \in \{1, \ldots, d\}$ and proceeding analogically to the proof of Theorem 6.1 with $m = 1$. □

The proofs above show that we are able to simplify the pricing integral from equation (6.4) to an one–dimensional problem for any $m \in \{1, \ldots, d\}$. Nevertheless, we have to pay attention to the complexity of $F_{\tau_{\iota_m}}$, which we do in the following.

6.1.1. Computational simplifications

The calculation of $F_{\tau_{\iota_m}}$ consists of $\mathcal{O}\left(2^d\right)$ function calls in the worst case. Hence, we will only be able to calculate $F_{\tau_{\iota_m}}$ straight forward, if dimension d is moderate. A significant reduction of complexity is achieved by using radial symmetric copulae, though.

Remark 6.4 (Reflecting the distributions). By reflecting PDFs $f_{\tau_1}, \ldots, f_{\tau_d}$ and payoff $\widetilde{\Lambda}$ with respect to the y–axis we can create new PDFs $\widehat{f}_{\tau_1}(t), \ldots, \widehat{f}_{\tau_d}(t)$ with corresponding CDFs $\widehat{F}_{\tau_1}(t) = 1 - F_{\tau_1}(-t), \ldots, \widehat{F}_{\tau_d}(t) = 1 - F_{\tau_d}(-t)$ and a new payoff $\widehat{\Lambda}(t)$, in which $t \in (-\infty, 0]$ holds.

Thus, we can state the CDF $F_{\tau_{\iota_m}}$ of the m–th smallest default time by means of the transformed CDF $\widehat{F}_{\tau_{\iota_{d-m+1}}}$ of the $(d - m + 1)$–smallest default time. For a better illustration we show the special case $d = 2$ and $m = 1$ in the following (at $(*)$ we use the radial symmetry property).

$$\begin{aligned} V_1 &= \int\limits_0^\infty \widetilde{\Lambda}(t) f_{\tau_{\iota_1}}(t)\mathrm{d}t = \widetilde{\Lambda}_\infty - \int\limits_0^T \widetilde{\Lambda}'(t) F_{\tau_{\iota_1}}(t)\mathrm{d}t \\ &= \widetilde{\Lambda}_\infty - \int\limits_0^T \widetilde{\Lambda}'(t)\left(F_{\tau_1}(t) + F_{\tau_2}(t) - \check{C}_2\left(F_{\tau_1}(t), F_{\tau_2}(t)\right)\right)\mathrm{d}t \end{aligned}$$

$$
\begin{aligned}
&= \widetilde{\Lambda}_\infty - \widetilde{\Lambda}(T) + \widetilde{\Lambda}(0)\\
&\qquad + \int\limits_0^T \widetilde{\Lambda}'(t)\left[1 - F_{\tau_1}(t) - F_{\tau_2}(t) + \check{C}_2\left(F_{\tau_1}(t), F_{\tau_2}(t)\right)\right]\mathrm{d}t\\
&\overset{(*)}{=} \widetilde{\Lambda}(0) + \int\limits_0^T \widetilde{\Lambda}'(t)\check{C}_2\left(1 - F_{\tau_1}(t), 1 - F_{\tau_2}(t)\right)\mathrm{d}t\\
&= \widehat{\Lambda}(0) - \int\limits_0^T \widehat{\Lambda}'(-t)\check{C}_2\left(\widehat{F}_{\tau_1}(-t), \widehat{F}_{\tau_2}(-t)\right)\mathrm{d}t\\
&= \widehat{\Lambda}(t)\widehat{F}_{\tau_{\iota_2}}(t)\Big|_{-\infty}^{0} - \int\limits_{-T}^{0} \widehat{\Lambda}'(t)\widehat{F}_{\tau_{\iota_2}}(t)\mathrm{d}t = \int\limits_{-\infty}^{0} \widehat{\Lambda}(t)\widehat{f}_{\tau_{\iota_2}}(t)\mathrm{d}t
\end{aligned}
$$

Consequently it is possible to evaluate an m–th to Default Swap with exactly the same effort as needed for the evaluation of a $(d-m+1)$–th to Default Swap. Thus, especially the evaluation of an m–th to Default Swap with small or huge m becomes very efficient. By using this modification we can reduce the complexity of $F_{\tau_{\iota_m}}$ from $\mathcal{O}(2^d)$ down to $\mathcal{O}(2^{d-1})$ in the worst case.

Although Remark 6.4 discloses a significant effort reduction we have to notice that it will not be applicable, if the underlying copula is not radial symmetric. In contrast to this, the next remark shows a modification, which is applicable for any copula.

Remark 6.5 (Simultaneous Pricing). Due to the special structure of $F_{\tau_{\iota_m}}$ we are able to price different contracts simultaneously. Within the calculation of V_m with respect to basket S, we can collect all required terms for the calculation of

- ▶ any k-th to Default Swaps with respect to S, $k \in \{m+1, \ldots, d\}$, and
- ▶ any k-th to Default Swaps with respect to $S' \subset S$, $k \in \{m+1, \ldots, d\}$.

Additionally, it is possible to derive an “mBDS – CDS – parity”, see the following lemma.

Lemma 6.2 (mBDS – CDS – parity)**.** *Let $S := \{S_1, \ldots, S_d\}$ be any basket of obligors and let $V_1, \ldots, V_d$ be the values of the particular 1–st, $\ldots$, d–th to Default Swaps. Furthermore, let $V_1^{CDS}, \ldots, V_d^{CDS}$ denote the values of the CDS contracts regarding the particular obligors. Then, equality*

$$\sum_{m=1}^{d} V_m = \sum_{i=1}^{d} V_i^{CDS} \tag{6.11}$$

will hold, if we assume a common payoff $\widetilde{\Lambda}$.

Proof. Consider the sum

$$\begin{aligned}\sum_{m=1}^{d} F_{\tau_{\iota_m}}(t) &= \sum_{m=1}^{d}\sum_{k=m}^{d}(-1)^{k-m}\binom{k-1}{m-1}\sum_{\substack{I\subseteq\{1,\dots,d\}\\|I|=k}} F_{\boldsymbol{\tau}_I}(\boldsymbol{t}_k)\\ &= \sum_{k=1}^{d}\sum_{m=1}^{k}(-1)^{k-m}\binom{k-1}{m-1}\sum_{\substack{I\subseteq\{1,\dots,d\}\\|I|=k}} F_{\boldsymbol{\tau}_I}(\boldsymbol{t}_k)\\ &= \sum_{k=1}^{d}\sum_{\substack{I\subseteq\{1,\dots,d\}\\|I|=k}} F_{\boldsymbol{\tau}_I}(\boldsymbol{t}_k)\sum_{m=1}^{k}(-1)^{k-m}\binom{k-1}{m-1}\end{aligned}$$

and exploit that

$$\sum_{m=1}^{k}(-1)^{k-m}\binom{k-1}{m-1} = \sum_{m=0}^{k-1}(-1)^{k-1-m}\binom{k-1}{m} = 0$$

holds for any $k > 1$. Thus, only the summand for $k = 1$ remains and equality

$$\sum_{m=1}^{d} F_{\tau_{\iota_m}}(t) = \sum_{\substack{I\subseteq\{1,\dots,d\}\\|I|=1}} F_{\boldsymbol{\tau}_I}(t) = \sum_{i=1}^{d} F_{\tau_i}(t),$$

holds for any $t \in \overline{\mathbb{R}}_{\geq 0}$. The common consideration of the equation above and Theorem 6.1 proves the assumption. □

Due to the special form of $F_{\tau_{\iota_m}}$ we can easily derive further parities as we can see in the following example.

Example 6.3 (Further parities)**.** Let $S = S_{123} := \{S_1, S_2, S_3\}$ be any basket of obligors and let $m = 1$ hold. Then, the value $V_1^{S_{123}}$ of the underlying First to Default Swap with respect to basket S_{123} is given by

$$\begin{aligned}V_1^{S_{123}} &= \int_0^\infty \widetilde{\Lambda}(t)\mathrm{d}F_{\tau_{\iota_1}}(t)\\ &= \int_0^\infty \widetilde{\Lambda}(t)\mathrm{d}\left[F_{\tau_1}(t) + F_{\tau_2}(t) + F_{\tau_3}(t) - F_{\tau_1,\tau_2}(t,t) - F_{\tau_1,\tau_3}(t,t) - F_{\tau_2,\tau_3}(t,t)\right.\\ &\qquad\qquad \left.+F_{\tau_1,\tau_2,\tau_3}(t,t,t)\right].\end{aligned}$$

This guarantees the validity of parity

$$V_1^{S_{123}} = \frac{1}{2}\left(V_1^{S_{12}} + V_1^{S_{13}} + V_1^{S_{23}} - V_2^{S_{123}}\right).$$

6.1.2. Special cases

In this subsection we regard special cases, which occur by applying Lemma 6.1 and Theorem 6.1. Each special case offers huge possibilities for effort reduction and the special case of independent default times will be of high importance later on.

Problem: Recovery rate and nominal value

First we discuss a problem, which has been neglected so far. In Section 2.1 we introduced some typical payments of mBDSs. Except its dependence on the m–th smallest default time t_{ι_m}, the default payment (cp. Definition 2.2) also depends on the recovery rate R_{ι_m} (and the nominal value N_{ι_m}) of the obligor, who causes the m–th default event (cp. equation (2.1)). Unfortunately, this dependence cannot be included into the pricing with respect to $F_{\tau_{\iota_m}}$ and payoff $\widetilde{\Lambda}$.

In practice we can neglect this problem, because the recovery rate represents an unknown parameter. This means that we have to estimate its value and due to this we can often observe homogeneous input data like $R := R_1 = \ldots = R_d$. Furthermore, products traded at markets are often regulated, i.e. they show homogeneous nominal values. In portfolios, which are not homogeneous, we can at least calculate bounds for its value by inserting the lowest and highest recovery rate respectively (analogical proceeding with respect to the nominal value).

In general we suggest a context depending application of the pricing via the m–th smallest default time. In very inhomogeneous baskets (regarding recovery rates and nominal values) we do not advise this technique.

Homogeneous hazard rates

If each MDF of random vector $\boldsymbol{\tau}$ is based on an equal hazard rate $h_1 = \ldots = h_d =: h$, the calculation of CDF $F_{\tau_{\iota_m}}(t)$ will be simplified enormously. Due to Lemma 6.1 $F_{\tau_{\iota_m}}(t)$ is given by

$$F_{\tau_{\iota_m}}(t) = \sum_{k=m}^{d} (-1)^{k-m} \binom{k-1}{m-1} \sum_{\substack{I \subseteq \{1,\ldots,d\} \\ |I|=k}} \check{C}_k^I \left(F_{\tau_{I_1}}(t), \ldots, F_{\tau_{I_k}}(t) \right).$$

If $h_1 = \ldots = h_d = h$ holds, we will be able to simplify $F_{\tau_1}(t) = \ldots = F_{\tau_d}(t) =: F(t)$ and get

$$\begin{aligned} F_{\tau_{\iota_m}}(t) &= \sum_{k=m}^{d} (-1)^{k-m} \binom{k-1}{m-1} \sum_{\substack{I \subseteq \{1,\ldots,d\} \\ |I|=k}} \check{C}_k(\underbrace{F(t), \ldots, F(t)}_{k \text{ elements}}) \\ &= \sum_{k=m}^{d} (-1)^{k-m} \binom{k-1}{m-1} \binom{d}{k} \check{C}_k \left(F(t), \ldots, F(t) \right) \\ &= \sum_{k=m}^{d} (-1)^{k-m} \frac{d!}{k(m-1)!(k-m)!(d-k)!} \check{C}_k \left(F(t), \ldots, F(t) \right), \end{aligned}$$

which represents an immense effort reduction.

Independent default times

In the following we consider d different obligors $S_1, \ldots, S_d$, which default independently of each other. Assuming this, we can simplify the calculation of $F_{\tau_{\iota_m}}$ significantly.

As we know through Lemma 6.1 and Example 6.2 the CDF $F_{\tau_{\iota_m}}$ will be given by

$$F_{\tau_{\iota_m}}(t) = \sum_{k=m}^{d} (-1)^{k-m} \binom{k-1}{m-1} \sum_{\substack{I \subseteq \{1,\ldots,d\} \\ |I|=k}} \prod_{l=1}^{k} F_{\tau_{I_l}}(t), \tag{6.12}$$

if we assume independent default times. Let us now assume that we are interested in the distribution of the first default time, i.e. $m = 1$ holds. Then, we can state the following lemma.

Lemma 6.3 (Nested calculation, $m = 1$)**.** *If we assume independent default times $\tau_1, \ldots, \tau_d$ and a triggering event $m = 1$, the simplification*

$$F_{\tau_{\iota_1}}(t) = \sum_{k=1}^{d} (-1)^{k-1} \sum_{\substack{I \subseteq \{1,\ldots,d\} \\ |I|=k}} \prod_{l=1}^{k} F_{\tau_{I_l}}(t) = \eta_d(\tau_1, \ldots, \tau_d)(t) \tag{6.13}$$

will hold for any $t \in \overline{\mathbb{R}}_{\geq 0}$, in which $\eta_d(\tau_1, \ldots, \tau_i)(t)$ is defined recursively by

$$\eta_d(\tau_1, \ldots, \tau_i)(t) := \begin{cases} (F_{\tau_i}(t) - 1)\, \eta_d(\tau_1, \ldots, \tau_{i-1})(t) + (-1)^{d-i} F_{\tau_i}(t), & \textit{if } 0 < i \leq d \\ 0, & \textit{else} \end{cases}$$

for each $i \in \{0, \ldots, d\}$.

Proof. The proof uses the method of induction with respect to $d \in \mathbb{N}_{\geq 2}$. In the base clause $d = 2$ we get

$$F_{\tau_{\iota_1}}(t) = F_{\tau_1}(t) + F_{\tau_2}(t) - F_{\tau_1}(t) F_{\tau_2}(t),$$

which is valid obviously. In the induction step we assume the validity of equation (6.13) for a fixed $d \in \mathbb{N}_{\geq 2}$. Then, we have to prove

$$F_{\tau_{\iota_1}}(t) = \sum_{k=1}^{d+1} (-1)^{k-1} \sum_{\substack{I \subseteq \{1,\ldots,d+1\} \\ |I|=k}} \prod_{l=1}^{k} F_{\tau_{I_l}}(t) \overset{!}{=} \eta_{d+1}(\tau_1, \ldots, \tau_{d+1})(t), \tag{6.14}$$

for any $t \in \overline{\mathbb{R}}_{\geq 0}$, in which $\eta_{d+1}(\tau_1, \ldots, \tau_{d+1})(t)$ is defined by

$$\begin{aligned} &\eta_{d+1}(\tau_1, \ldots, \tau_i)(t) \\ &\quad := \begin{cases} (F_{\tau_i}(t) - 1)\, \eta_{d+1}(\tau_1, \ldots, \tau_{i-1})(t) + (-1)^{d+1-i} F_{\tau_i}(t), & \text{if } 0 < i \leq d+1 \\ 0, & \text{else} \end{cases} \end{aligned} \tag{6.15}$$

for each $i \in \{0, \ldots, d+1\}$. For this, we decompose equation (6.14) and transform it to

$$\begin{aligned}
F_{\tau_{\iota_1}}(t) &= \sum_{k=1}^{d+1} (-1)^{k-1} \sum_{\substack{I \subseteq \{1,\ldots,d+1\} \\ |I|=k}} \prod_{l=1}^{k} F_{\tau_{I_l}}(t) \\
&= \sum_{k=1}^{d} (-1)^{k-1} \sum_{\substack{I \subseteq \{1,\ldots,d\} \\ |I|=k}} \prod_{l=1}^{k} F_{\tau_{I_l}}(t) + \sum_{k=1}^{d+1} (-1)^{k-1} \sum_{\substack{I \subseteq \{1,\ldots,d+1\} \\ |I|=k,\ d+1 \in I}} \prod_{l=1}^{k} F_{\tau_{I_l}}(t) \\
&= \Big(-F_{\tau_{d+1}} + 1\Big) \cdot \left[\sum_{k=1}^{d} (-1)^{k-1} \sum_{\substack{I \subseteq \{1,\ldots,d\} \\ |I|=k}} \prod_{l=1}^{k} F_{\tau_{I_l}}(t) \right] + F_{\tau_{d+1}} \\
&\overset{(*)}{=} \Big(F_{\tau_{d+1}}(t) - 1\Big) \cdot (-1) \cdot \eta_d\left(\tau_1, \ldots, \tau_d\right)(t) + F_{\tau_{d+1}}(t) \\
&= \Big(F_{\tau_{d+1}}(t) - 1\Big) \cdot \eta_{d+1}\left(\tau_1, \ldots, \tau_d\right)(t) + (-1)^{d+1-(d+1)} F_{\tau_{d+1}}(t) \\
&= \eta_{d+1}\left(\tau_1, \ldots, \tau_{d+1}\right)(t),
\end{aligned}$$

in which the induction hypothesis is used at equality $(*)$. □

By means of Lemma 6.3 we achieve an enormous effort reduction for calculating $F_{\tau_{\iota_1}}$. We can easily count that a plain evaluation of $F_{\tau_{\iota_1}}$ via equation (6.12) requires at least $\eta_{d,+}^{\text{plain}} := 2^d - 2$ additions and $\eta_{d,\cdot}^{\text{plain}} := 2^d - 1 - d$ multiplications. By applying Lemma 6.3 we are able to reduce the effort to $\eta_{d,+}^{\text{nested}} := 2(d-1)$ additions and $\eta_{d,\cdot}^{\text{nested}} := d - 1$ multiplications (cp. Figure 6.1), which makes this simplification essential. For a better illustration we give an example in the following.

Example 6.4 (Independent default times, $m = 1$)**.** Let $d = 4$, $m = 1$ hold and let $\tau_1, \ldots, \tau_4$ denote independent random variables. Applying the plain calculation of $F_{\tau_{\iota_1}}$ via equation (6.12) and neglecting the argument $t \in \overline{\mathbb{R}}_{\geq 0}$ we get

$$\begin{aligned}
F_{\tau_{\iota_1}} =& F_{\tau_1} + F_{\tau_2} + F_{\tau_3} + F_{\tau_4} \\
&- \left(F_{\tau_1}F_{\tau_2} + F_{\tau_1}F_{\tau_3} + F_{\tau_1}F_{\tau_4} + F_{\tau_2}F_{\tau_3} + F_{\tau_2}F_{\tau_4} + F_{\tau_3}F_{\tau_4}\right) \\
&+ F_{\tau_1}F_{\tau_2}F_{\tau_3} + F_{\tau_1}F_{\tau_2}F_{\tau_4} + F_{\tau_1}F_{\tau_3}F_{\tau_4} + F_{\tau_2}F_{\tau_3}F_{\tau_4} \\
&- F_{\tau_1}F_{\tau_2}F_{\tau_3}F_{\tau_4}.
\end{aligned}$$

Consequently, we have to execute 14 additions and 11 multiplications. In contrast to this, the application of Lemma 6.3 calculates $F_{\tau_{\iota_1}}(t)$ as

$$F_{\tau_{\iota_1}} = \left(F_{\tau_4} - 1\right) \left(\left(F_{\tau_3} - 1\right) \Big(\left(F_{\tau_2} - 1\right) \left(-F_{\tau_1}\right) + F_{\tau_2} \Big) - F_{\tau_3} \right) + F_{\tau_4},$$

which requires only 6 additions and 3 multiplications.

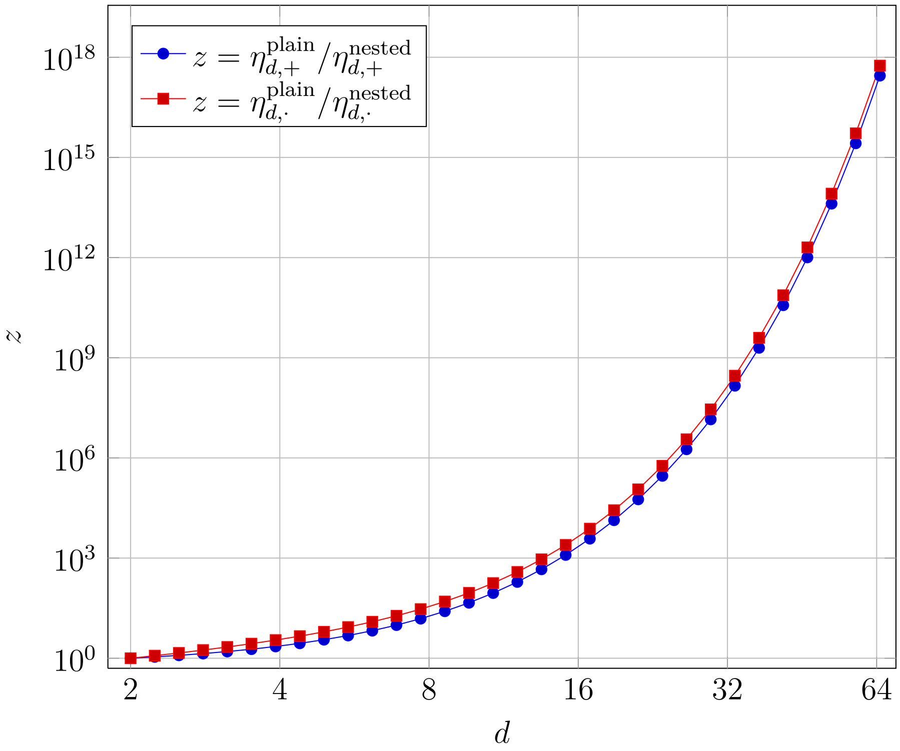


Figure 6.1.: Ratio of effort reduction due to Lemma 6.3.

Nevertheless, the simplification presented in Lemma 6.3 is not applicable for general $m \in \{1, \ldots, d\}$, because the equality of $\binom{m-1}{m-1} = \binom{l-1}{m-1}$ for each $l \in \{m, \ldots, d\}$ is essential for its validity. Because this equality is only valid in the special case $m = 1$, we have to look for another simplification, which is valid for each $m \in \{1, \ldots, d\}$.

Lemma 6.4 (Nested calculation, $m \in \{1, \ldots, d\}$)**.** *Let $\tau_1, \ldots, \tau_d$ be independent random variables and let $m \in \{1, \ldots, d\}$ be any number. Then, for any $t \in \overline{\mathbb{R}}_{\geq 0}$, we can simplify*

$$\begin{aligned} F_{\tau_{\iota_m}}(t) &= \sum_{k=m}^{d} (-1)^{k-m} \binom{k-1}{m-1} \sum_{\substack{I \subseteq \{1,\ldots,d\} \\ |I| = k}} \prod_{l=1}^{k} F_{\tau_{I_l}}(t) \\ &= \sum_{k=m}^{d} (-1)^{k-m} \binom{k-1}{m-1} \cdot \widetilde{\eta}(k, d-k+1)(t), \end{aligned} \tag{6.16}$$

in which $\widetilde{\eta}(k,l)(t)$ is defined recursively by

$$\widetilde{\eta}(k,l)(t) := \begin{cases} F_{\tau_1}(t), & \textit{if } k = l = 1 \\ \widetilde{\eta}(k, l-1)(t) + F_{\tau_l}(t), & \textit{if } k = 1 \textit{ and } l > 1 \\ \widetilde{\eta}(k-1, l)(t) \cdot F_{\tau_k}(t), & \textit{if } k > 1 \textit{ and } l = 1 \\ \widetilde{\eta}(k-1, l)(t) \cdot F_{\tau_{k+l-1}}(t) + \widetilde{\eta}(k, l-1)(t), & \textit{if } k > 1 \textit{ and } l > 1 \end{cases} \tag{6.17}$$

for each $k \in \{1, \ldots, d\}$ and $l \in \{1, \ldots, \min\{d-k+1, d-m+1\}\}$.

Proof. For proving the assumption we just have to show

$$\sum_{\substack{I\subseteq\{1,\dots,d\}\\|I|=k}} \prod_{l=1}^{k} F_{\tau_{I_l}}(t) = \widetilde{\eta}(k, d-k+1)(t), \tag{6.18}$$

which is done by applying a 2–dimensional induction regarding $d \in \mathbb{N}_{\geq 2}$ (outer induction) and $k \in \{1, \dots, d\}$ (inner induction). Clearly spoken equation (6.18) denotes the sum of all possible k–dimensional products with respect to any d–dimensional basic set. For the base clause in the outer induction we consider $d = 2$ and obviously get

$$\begin{cases} \sum\limits_{\substack{I\subseteq\{1,2\}\\|I|=k}} \prod\limits_{l=1}^{k} F_{\tau_{I_l}}(t) = F_{\tau_1}(t) + F_{\tau_2}(t) = \widetilde{\eta}(1,1)(t) + F_{\tau_2}(t) = \widetilde{\eta}(1,2)(t), & \text{for } k = 1 \\ \sum\limits_{\substack{I\subseteq\{1,2\}\\|I|=k}} \prod\limits_{l=1}^{k} F_{\tau_{I_l}}(t) = F_{\tau_1}(t) \cdot F_{\tau_2}(t) = \widetilde{\eta}(1,1)(t) \cdot F_{\tau_2}(t) = \widetilde{\eta}(2,1)(t), & \text{for } k = 2 \end{cases}.$$

We want to assume that equation (6.18) holds for a fixed $d \in \mathbb{N}_{>2}$ and each $k \in \{1, \dots, d\}$ in order to form the outer induction hypothesis. To justify this, we have to run an inner induction with respect to k. This induction starts with the base clause $k = 1$, which leads to

$$\sum_{\substack{I\subseteq\{1,\dots,d\}\\|I|=1}} \prod_{l=1}^{1} F_{\tau_{I_l}}(t) = \sum_{i=1}^{d} F_{\tau_i}(t) = \widetilde{\eta}(1,d)(t)$$

and is valid obviously. Now we assume that equation (6.18) holds for a fixed d and a fixed $1 < k < d$ (inner induction hypothesis) and have the show

$$\sum_{\substack{I\subseteq\{1,\dots,d\}\\|I|=k+1}} \prod_{l=1}^{k+1} F_{\tau_{I_l}}(t) = \widetilde{\eta}(k+1, d-k)(t).$$

Using some intelligent transformations we get

$$\begin{aligned}
&\sum_{\substack{I\subseteq\{1,\dots,d\}\\|I|=k+1}} \prod_{l=1}^{k+1} F_{\tau_{I_l}}(t) = F_{\tau_d}(t) \sum_{\substack{I\subseteq\{1,\dots,d-1\}\\|I|=k}} \prod_{l=1}^{k} F_{\tau_{I_l}}(t) + \sum_{\substack{I\subseteq\{1,\dots,d-1\}\\|I|=k+1}} \prod_{l=1}^{k+1} F_{\tau_{I_l}}(t) \\
&\overset{(*)}{=} F_{\tau_d}(t) \cdot \widetilde{\eta}(k, d-k)(t) + F_{\tau_{d-1}}(t) \sum_{\substack{I\subseteq\{1,\dots,d-2\}\\|I|=k}} \prod_{l=1}^{k} F_{\tau_{I_l}}(t) + \sum_{\substack{I\subseteq\{1,\dots,d-2\}\\|I|=k+1}} \prod_{l=1}^{k+1} F_{\tau_{I_l}}(t) \\
&\overset{(*)}{=} F_{\tau_d}(t) \cdot \widetilde{\eta}(k, d-k)(t) + F_{\tau_{d-1}} \cdot \widetilde{\eta}(k, d-k-1)(t) + \sum_{\substack{I\subseteq\{1,\dots,d-2\}\\|I|=k+1}} \prod_{l=1}^{k+1} F_{\tau_{I_l}}(t) \\
&= \ldots \overset{(*)}{=} \sum_{l=d}^{d-k+2} F_{\tau_l}(t) \cdot \widetilde{\eta}(k, l-k)(t) + \prod_{l=1}^{k+1} F_{\tau_l}(t) \\
&= \sum_{l=d}^{d-k+2} F_{\tau_l}(t) \cdot \widetilde{\eta}(k, l-k)(t) + F_{\tau_{k+1}} \widetilde{\eta}(k,1)(t)
\end{aligned}$$

$$= \widetilde{\eta}(k+1, d-k)(t),$$

in which the inner induction hypothesis was used at $(*)$. Now, we can assume the induction hypothesis for the outer induction, i.e. equation (6.18) holds for a fixed $d \in \mathbb{N}_{\geq 2}$ and for each $k \in \{1, \dots, d\}$. In order to complete the outer induction step we finally have to show

$$\forall k \in \{1, \dots, d+1\} : \sum_{\substack{I \subseteq \{1,\dots,d+1\} \\ |I|=k}} \prod_{l=1}^{k} F_{\tau_{I_l}}(t) = \widetilde{\eta}(k, d-k+2)(t). \tag{6.19}$$

At this, we do not have to execute an inner induction because we have already done this in the outer induction hypothesis. Thus, we can state for each $k \in \{1, \dots, d+1\}$

$$\begin{aligned}\sum_{\substack{I \subseteq \{1,\dots,d+1\} \\ |I|=k}} \prod_{l=1}^{k} F_{\tau_{I_l}}(t) &= F_{\tau_{d+1}}(t) \sum_{\substack{I \subseteq \{1,\dots,d\} \\ |I|=k-1}} \prod_{l=1}^{k-1} F_{\tau_{I_l}}(t) + \sum_{\substack{I \subseteq \{1,\dots,d\} \\ |I|=k}} \prod_{l=1}^{k} F_{\tau_{I_l}}(t) \\ &\overset{(*)}{=} F_{\tau_{d+1}}(t) \cdot \widetilde{\eta}(k-1, d-k+2)(t) + \widetilde{\eta}(k, d-k+1)(t) \\ &= \widetilde{\eta}(k, d-k+2),\end{aligned}$$

which uses the outer induction hypothesis at $(*)$ and finally proves the assumption (cp. equation (6.18)). □

In order to give a further illustration of equation (6.16) and recursion (6.17) we can easily verify the following statement: for each $l \leq d-k+1$ the term $\widetilde{\eta}(k,l)$ equals the sum of all possible k–dimensional products of the basic set $\left\{F_{\tau_1}, \dots, F_{\tau_{k+l-1}}\right\}$. Hence, for obtaining the sum of all possible k–dimensional products regarding the basic set $\{F_{\tau_1}, \dots, F_{\tau_d}\}$ we have to choose $l = d-k+1$.

Although the effort reduction of Lemma 6.4 is not as significant as in Lemma 6.3, we can still observe an immense reduction of complexity. Furthermore, by using Lemma 6.4 (instead of Lemma 6.3) we can simultaneously calculate the value of any m'–th to Default Swap, with $d \geq m' > m$, because we have already calculated each required summand $\widetilde{\eta}(k, d-k+1)$ for $d \geq k \geq m' > m$ by evaluating V_m (cp. also Remark 6.5).

Next, we count the number of arithmetic operations, which are required for the calculation of $F_{\tau_{\iota_m}}(t)$, in which $m \in \{1, \dots, d\}$ and $t \in \overline{\mathbb{R}}_{\geq 0}$ are fixed. Applying

Lemma 6.4 for the calculation of $F_{\tau_{\iota_m}}(t)$ we require the following function calls of $\widetilde{\eta}$:

$$\begin{array}{cccc}
\widetilde{\eta}(1,1)(t) & \cdots & \widetilde{\eta}(1,d-m)(t) & \widetilde{\eta}(1,d-m+1)(t) \\
\vdots & & & \vdots \\
\widetilde{\eta}(m,1)(t) & \cdots & \widetilde{\eta}(m,d-m)(t) & \widetilde{\eta}(m,d-m+1)(t) \\
\widetilde{\eta}(m+1,1)(t) & \cdots & \widetilde{\eta}(m+1,d-m)(t) & \\
\vdots & \iddots & & \\
\widetilde{\eta}(d,1)(t) & & &
\end{array}$$

Combined with the outer sum $\sum\limits_{k=m}^{d}$ we get $\widetilde{\eta}^{\text{nested}}_{d,m,+}$ additions and $\widetilde{\eta}^{\text{nested}}_{d,m,\cdot}$ multiplications, which are defined as

$$\begin{aligned}
\widetilde{\eta}^{\text{nested}}_{d,m,+} : &= \sum_{l=0}^{d-m-1} l + m(d-m) + (d-m) \\
&= \frac{(d-m)(d-m-1)}{2} + (d-m)(m+1) \\
&= (d-m)\left(\frac{d+m+1}{2}\right) \\
\widetilde{\eta}^{\text{nested}}_{d,m,\cdot} : &= \sum_{l=0}^{d-m} l + (m-1)(d-m+1) + (d-m) \\
&= \frac{(d-m)(d-m+1)}{2} + (m-1)(d-m+1) + (d-m) \\
&= (d-m+1)\frac{d+m}{2} - 1.
\end{aligned}$$

At this, the summand $(d-m)$ denotes the additional arithmetic operations, which are necessary to form $F_{\tau_{\iota_m}}(t)$ out of the different function calls $\widetilde{\eta}(m, d-m+1)(t), \ldots, \widetilde{\eta}(d,1)(t)$.

If we use the plain calculation of $F_{\tau_{\iota_m}}(t)$ according to the principle of inclusion and exclusion (cp. Lemma 6.1) and store each calculated product in an intelligent way, we will need

$$\widetilde{\eta}^{\text{plain}}_{d,m,+} := \sum_{l=m}^{d} \binom{d}{l} - 1$$

additions and

$$\widetilde{\eta}^{\text{plain}}_{d,m,\cdot} := \sum_{l=2}^{m} \binom{d-m+l}{l} + \sum_{l=m+1}^{d} \binom{d}{l} + (d-m)$$

multiplications. For a better illustration we refer to the following example.

Example 6.5 (Independent default times, $m \in \{1, \dots, d\}$)**.** Let $d = 5$ and $m = 3$ hold and let $\tau_1, \dots, \tau_5$ denote independent random variables. Applying the plain calculation of $F_{\tau_{\iota_3}}$ as in equation (6.12) and neglecting the argument $t \in \overline{\mathbb{R}}_{\geq 0}$ we get

$$\begin{aligned} F_{\tau_{\iota_3}} =& F_{\tau_1}F_{\tau_2}F_{\tau_3} + F_{\tau_1}F_{\tau_2}F_{\tau_4} + F_{\tau_1}F_{\tau_2}F_{\tau_5} + F_{\tau_1}F_{\tau_3}F_{\tau_4} + F_{\tau_1}F_{\tau_3}F_{\tau_5} \\ &+ F_{\tau_1}F_{\tau_4}F_{\tau_5} + F_{\tau_2}F_{\tau_3}F_{\tau_4} + F_{\tau_2}F_{\tau_3}F_{\tau_5} + F_{\tau_2}F_{\tau_4}F_{\tau_5} + F_{\tau_3}F_{\tau_4}F_{\tau_5} \\ &- 3\left[F_{\tau_1}F_{\tau_2}F_{\tau_3}F_{\tau_4} + F_{\tau_1}F_{\tau_2}F_{\tau_3}F_{\tau_5} + F_{\tau_1}F_{\tau_2}F_{\tau_4}F_{\tau_5} + F_{\tau_1}F_{\tau_3}F_{\tau_4}F_{\tau_5}\right. \\ &\left.+ F_{\tau_2}F_{\tau_3}F_{\tau_4}F_{\tau_5}\right] + 6F_{\tau_1}F_{\tau_2}F_{\tau_3}F_{\tau_4}F_{\tau_5}. \end{aligned}$$

This calculation requires 15 additions and 24 multiplications. In contrast, Lemma 6.4 calculates $F_{\tau_{\iota_3}}$ as

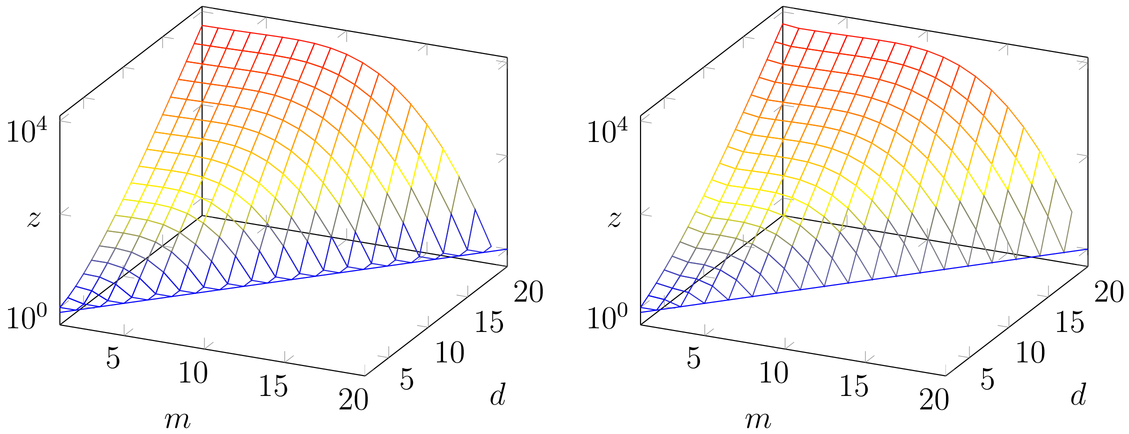
$$\begin{aligned} F_{\tau_{\iota_3}} =& \widetilde{\eta}(3,3) - 3\widetilde{\eta}(4,2) + 6\widetilde{\eta}(5,1) \\ =& \widetilde{\eta}(2,3)F_{\tau_5} + \widetilde{\eta}(3,2) - 3\left[\widetilde{\eta}(3,2)F_{\tau_5} + \widetilde{\eta}(4,1)\right] + 6\left[\widetilde{\eta}(4,1)F_{\tau_5}\right] \\ =& \left(\widetilde{\eta}(1,3)F_{\tau_4} + \widetilde{\eta}(2,2)\right)F_{\tau_5} + \left(\widetilde{\eta}(2,2)F_{\tau_4} + \widetilde{\eta}(3,1)\right) \\ &- 3\left[\left(\widetilde{\eta}(2,2)F_{\tau_4} + \widetilde{\eta}(3,1)\right)F_{\tau_5} + \left(\widetilde{\eta}(3,1)F_{\tau_4}\right)\right] \\ &+ 6\left[\widetilde{\eta}(3,1)F_{\tau_4}F_{\tau_5}\right] \\ =& \left(\left(\widetilde{\eta}(1,2) + F_{\tau_3}\right)F_{\tau_4} + \left(\widetilde{\eta}(1,2)F_{\tau_3} + \widetilde{\eta}(2,1)\right)\right)F_{\tau_5} \\ &\qquad + \left(\left(\widetilde{\eta}(1,2)F_{\tau_3} + \widetilde{\eta}(2,1)\right)F_{\tau_4} + \left(\widetilde{\eta}(2,1)F_{\tau_3}\right)\right) \\ &- 3\left[\left(\left(\widetilde{\eta}(1,2)F_{\tau_3} + \widetilde{\eta}(2,1)\right)F_{\tau_4} + \left(\widetilde{\eta}(2,1)F_{\tau_3}\right)\right)F_{\tau_5} + \left(\left(\widetilde{\eta}(2,1)F_{\tau_3}\right)F_{\tau_4}\right)\right] \\ &+ 6\left[\widetilde{\eta}(2,1)F_{\tau_3}F_{\tau_4}F_{\tau_5}\right] \\ =& \left(\left(\left(F_{\tau_1} + F_{\tau_2}\right) + F_{\tau_3}\right)F_{\tau_4} + \left(\left(F_{\tau_1} + F_{\tau_2}\right)F_{\tau_3} + \left(F_{\tau_1}F_{\tau_2}\right)\right)\right)F_{\tau_5} \\ &\qquad + \left(\left(\left(F_{\tau_1} + F_{\tau_2}\right)F_{\tau_3} + \left(F_{\tau_1}F_{\tau_2}\right)\right)F_{\tau_4} + \left(\left(F_{\tau_1}F_{\tau_2}\right)F_{\tau_3}\right)\right) \\ &- 3\left[\left(\left(\left(F_{\tau_1} + F_{\tau_2}\right)F_{\tau_3} + \left(F_{\tau_1}F_{\tau_2}\right)\right)F_{\tau_4} + \left(\left(F_{\tau_1}F_{\tau_2}\right)F_{\tau_3}\right)\right)F_{\tau_5}\right. \\ &\qquad \left.+ \left(\left(\left(F_{\tau_1}F_{\tau_2}\right)F_{\tau_3}\right)F_{\tau_4}\right)\right] \\ &+ 6\left[F_{\tau_1}F_{\tau_2}F_{\tau_3}F_{\tau_4}F_{\tau_5}\right]. \end{aligned}$$

and only requires 9 additions and 11 multiplications.

In Figure 6.2 we show the ratio of effort reduction by applying Lemma 6.4 and arbitrary $d \in \mathbb{N}_{\geq 2}$ and $m \in \{1, \dots, d\}$. Obviously, the calculation with the help of Lemma 6.4 is at worst as complex as the plain method ($m = d$). For almost any choice of d and m the usage of Lemma 6.4 offers a huge effort reduction, though.

To sum up, we highly suggest the usage of Lemmata 6.3 and 6.4 for calculating $F_{\tau_{\iota_m}}(t)$ in case of independent random variables. These modifications offer great possibilities for reducing the complexity of the underlying calculation for any choice of d and m.

After considering these special cases we apply certain copulae in the following.



(a) $z = \widetilde{\eta}^{\text{plain}}_{d,m,+} / \widetilde{\eta}^{\text{nested}}_{d,m,+}$

(b) $z = \widetilde{\eta}^{\text{plain}}_{d,m,\cdot} / \widetilde{\eta}^{\text{nested}}_{d,m,\cdot}$

Figure 6.2.: Ratio of effort reduction due to Lemma 6.4.

6.1.3. Application of the Gaussian copula

In this subsection the CDF of the m–th smallest default time is modeled by means of the Gaussian copula. For this, let Σ denote the correlation matrix of the random vector $(\tau_1, \ldots, \tau_d)^{\text{tr}}$. Then, for each $I = \{I_1, \ldots, I_k\} \subseteq \{1, \ldots, d\}$ and for any $\boldsymbol{t}_I \in \overline{\mathbb{R}}^k_{\geq 0}$ the joint CDF of random vector $\boldsymbol{\tau}_I = (\tau_{I_1}, \ldots, \tau_{I_k})^{\text{tr}}$ is modeled by

$$\begin{aligned} F_{\boldsymbol{\tau}_I}(t_{I_1}, \ldots, t_{I_k}) &= \check{C}^I_k \left(F_{\tau_{I_1}}(t_{I_1}), \ldots, F_{\tau_{I_k}}(t_{I_k}) \right) \\ &= C^{I,\text{Gau}}_{k,\Sigma^I_k} \left(F_{\tau_{I_1}}(t_{I_1}), \ldots, F_{\tau_{I_k}}(t_{I_k}) \right), \end{aligned}$$

in which Σ^I_k denotes the correlation matrix of the random vector $\boldsymbol{\tau}_I$. The aim of this subsection is the efficient evaluation of any mBDS via

$$V_m = \int\limits_0^\infty \widetilde{\Lambda}(t) \cdot f_{\tau_{\iota_m}}(t) dt = \widetilde{\Lambda}_\infty - \int\limits_0^T \widetilde{\Lambda}'(t) \cdot F_{\tau_{\iota_m}}(t) \mathrm{d}t, \tag{6.20}$$

in which $F_{\tau_{\iota_m}}(t)$ is given by

$$F_{\tau_{\iota_m}}(t) = \sum_{k=m}^{d} (-1)^{k-m} \binom{k-1}{m-1} \sum_{\substack{I \subseteq \{1,\ldots,d\} \\ |I|=k}} C^{I,\text{Gau}}_{k,\Sigma^I_k} \left(F_{\tau_{I_1}}(t), \ldots, F_{\tau_{I_k}}(t) \right). \tag{6.21}$$

for any $t \in \overline{\mathbb{R}}_{\geq 0}$. In order to evaluate equation (6.20) efficiently we have to develop a method, which can evaluate the term

$$C^{I,\text{Gau}}_{k,\Sigma^I_k} \left(F_{\tau_{I_1}}(t), \ldots, F_{\tau_{I_k}}(t) \right) \tag{6.22}$$

efficiently. The term (6.22) denotes the k–dimensional standard Gaussian joint CDF (cp. Definitions 4.2 and 4.3) and equals a k–dimensional integral. Applying a k–dimensional quadrature rule would destroy the dimension reduction we have achieved so far. Instead we assume that Σ has the one–factor structure (cp. Subsection 3.1.1)

$$\Sigma = \Sigma_1^{\mathrm{F}} = (\psi_1, \ldots, \psi_d)^{\mathrm{tr}} (\psi_1, \ldots, \psi_d) + \mathrm{diag}_d \left(1 - \psi_1^2, \ldots, 1 - \psi_d^2\right).$$

We could also use a $k^{'}$–factor structure, $k^{'} \in \{2, \ldots, k-1\}$, but for simplicity we choose a one–dimensional model. With the help of Theorem 4.1, we can simplify (6.22) to the one–dimensional integral

$$C_{k,\Sigma_k^I}^{I,\mathrm{Gau}} \left(F_{\tau_{I_1}}(t), \ldots, F_{\tau_{I_k}}(t)\right) = \int\limits_{\mathbb{R}} \phi(y) \prod_{i=1}^{k} \Phi \left(\frac{\Phi^{-1}\left(F_{\tau_{I_i}}(t)\right) - \psi_{I_i} \cdot y}{\sqrt{1 - \psi_{I_i}^2}} \right) \mathrm{d}y. \tag{6.23}$$

Next, we discuss the efficient evaluation of this integral.

Evaluating the multivariate standard Gaussian distribution

For evaluating the multivariate standard Gaussian joint CDF we use a Gauss–Hermite quadrature (abbr.: GHQ) in the following (cp. [FH07, SK09, SH12a]). This quadrature approximates the integral from equation (6.23) by the weighted sum

$$\begin{aligned} &\int\limits_{\mathbb{R}} \phi(y) \prod_{i=1}^{k} \Phi \left(\frac{\Phi^{-1}\left(F_{\tau_{I_i}}(t)\right) - \psi_{I_i} \cdot y}{\sqrt{1 - \psi_{I_i}^2}} \right) \mathrm{d}y \\ &\qquad \approx \sum_{j=1}^{N_{\mathrm{H}}} w_{j,\mathrm{H}} \exp\left(y_{j,\mathrm{H}}^2\right) \phi\left(y_{j,\mathrm{H}}\right) \prod_{i=1}^{k} \Phi \left(\frac{\Phi^{-1}\left(F_{\tau_{I_i}}(t)\right) - \psi_{I_i} \cdot y_{j,\mathrm{H}}}{\sqrt{1 - \psi_{I_i}^2}} \right) \\ &\qquad =: \hat{\mathcal{N}}_{\mathrm{H}} \left(N_{\mathrm{H}}, k, \underbrace{\left(F_{\tau_{I_1}}(t), \ldots, F_{\tau_{I_k}}(t)\right)^{\mathrm{tr}}}_{=:\boldsymbol{u}}, \boldsymbol{\psi} \right), \end{aligned} \tag{6.24}$$

in which $y_{j,\mathrm{H}}$ and $w_{j,\mathrm{H}}$ denote N_{H} different nodes and weights of the GHQ (cp. [AS70, PdDKÜK83]).

For quantifying the quality of this approach we apply different values for k, N_{H}, $\boldsymbol{u}$ and $\boldsymbol{\psi}$ and compare the resulting approximations $\hat{\mathcal{N}}_{\mathrm{H}}(N_{\mathrm{H}}, k, \boldsymbol{u}, \boldsymbol{\psi})$ to each other. For this, we use the relative error estimator

$$\epsilon^{'}(N_{\mathrm{H}}, k, \boldsymbol{u}, \boldsymbol{\psi}) := \frac{\left|\hat{\mathcal{N}}_{\mathrm{H}}(N_{\mathrm{H}}, k, \boldsymbol{u}, \boldsymbol{\psi}) - \hat{\mathcal{N}}_{\mathrm{H}}(2N_{\mathrm{H}}, k, \boldsymbol{u}, \boldsymbol{\psi})\right|}{\hat{\mathcal{N}}_{\mathrm{H}}(2N_{\mathrm{H}}, k, \boldsymbol{u}, \boldsymbol{\psi})} \tag{6.25}$$

of approximation $\hat{\mathcal{N}}_{\mathrm{H}}(N_{\mathrm{H}}, k, \boldsymbol{u}, \boldsymbol{\psi})$. Obviously, error estimator $\epsilon^{'}(N_{\mathrm{H}}, k, \boldsymbol{u}, \boldsymbol{\psi})$ also offers an error bound for approximation $\hat{\mathcal{N}}_{\mathrm{H}}(2N_{\mathrm{H}}, k, \boldsymbol{u}, \boldsymbol{\psi})$.

In the following analysis we draw $N_{\boldsymbol{u},\boldsymbol{\psi}} = 10^5$ independently and uniformly distributed tuples $\boldsymbol{u}^{(i)} \in [0,1]^k$ and $\boldsymbol{\psi}^{(i)} \in [-1,1]^k$ and calculate the approximation

$\hat{\mathcal{N}}_{\mathrm{H}}\left(N_{\mathrm{H}}, k, \boldsymbol{u}, \boldsymbol{\psi}\right)$ afterwards. At this, we choose $k \in \{5, 10, 20, 30\}$ and $N_{\mathrm{H}} \in \{4, 8, 16, 32, 64, 128, 256\}$. The value $\hat{\mathcal{N}}_{\mathrm{H}}\left(256, k, \boldsymbol{u}, \boldsymbol{\psi}\right)$ is assumed to be "exact" in any approximation and consequently the relative error of any parameter tuple $\left(N_{\mathrm{H}} \neq 256, k, \boldsymbol{u}, \boldsymbol{\psi}\right)$ is given by

$$\epsilon\left(N_{\mathrm{H}}, k, \boldsymbol{u}, \boldsymbol{\psi}\right) := \frac{\left|\hat{\mathcal{N}}_{\mathrm{H}}\left(N_{\mathrm{H}}, k, \boldsymbol{u}, \boldsymbol{\psi}\right) - \hat{\mathcal{N}}_{\mathrm{H}}\left(256, k, \boldsymbol{u}, \boldsymbol{\psi}\right)\right|}{\hat{\mathcal{N}}_{\mathrm{H}}\left(256, k, \boldsymbol{u}, \boldsymbol{\psi}\right)}. \tag{6.26}$$

Additionally, the relative error of $\hat{\mathcal{N}}_{\mathrm{H}}\left(256, k, \boldsymbol{u}, \boldsymbol{\psi}\right)$ is bounded by $\epsilon^{'}\left(128, k, \boldsymbol{u}, \boldsymbol{\psi}\right) = \epsilon\left(128, k, \boldsymbol{u}, \boldsymbol{\psi}\right)$.

Alternatively, we can substitute $\hat{\mathcal{N}}_{\mathrm{H}}\left(256, k, \boldsymbol{u}, \boldsymbol{\psi}\right)$ in equation (6.26) by an "evidenced exact" value, which was calculated by applying an adaptive Simpson quadrature (cp. [CK07]) with error bound 10^{-15}. However, this proceeding offers similar results by causing a considerably higher effort. For example, choosing $k = 5$, an adaptive Simpson quadrature with error bound 10^{-15} requires about 4000 nodes and is therefore neglected in this analysis.

The results of the analysis above are shown in Table 6.1 for $k = 5$ (different values for k are shown in Tables C.1 – C.3). At this

$$\overline{\epsilon}_{N_{\mathrm{H}},k} := \frac{1}{N_{\boldsymbol{u},\boldsymbol{\psi}}} \sum_{l=1}^{N_{\boldsymbol{u},\boldsymbol{\psi}}} \epsilon\left(N_{\mathrm{H}}, k, \boldsymbol{u}^{(l)}, \boldsymbol{\psi}^{(l)}\right)$$

denotes the estimated mean of $\epsilon\left(N_{\mathrm{H}}, k, \boldsymbol{u}, \boldsymbol{\psi}\right)$, $\overline{\sigma}_{N_{\mathrm{H}},k}$ denotes its estimated standard deviation and $\overline{\mathcal{M}}_{N_{\mathrm{H}},k}$ denotes its estimated median. The results are independent of input vector $\boldsymbol{u}$. Furthermore, it is obvious that an increasing N_{H} and a decreasing k improves the approximation. However, we can also observe a dependence on $\boldsymbol{\psi}$. This is explained as follows:

The quality of a Gauss quadrature highly depends on the smoothness of the underlying function f. The smoother f is the better it is approximated by applying a Gauss quadrature. Let us consider a Gauss quadrature using N different nodes on the interval $[a, b] \subset \mathbb{R}$. Then, the approximation error has the structure $c(N, b, a) \cdot f^{(2N)}\left(\xi\right)/(2N!)$ (cp. [FH07, Theorem (3.6.24)]), in which $c(N, b, a)$ is constant for fixed N, b, a and ξ is located in $[a, b]$.

$\lvert\psi_j\rvert \downarrow$	$N_{\mathrm{H}} \rightarrow$	4	8	16	32	64	128
≤ 1	$\overline{\epsilon}_{N_{\mathrm{H}},k}$	8.77e-02	2.98e-02	1.87e-02	4.67e-03	1.86e-03	6.94e-04
	$\overline{\sigma}_{N_{\mathrm{H}},k}$	2.17e-01	1.24e-01	2.37e+00	6.63e-02	4.35e-02	2.37e-02
	$\overline{\mathcal{M}}_{N_{\mathrm{H}},k}$	1.74e-02	4.08e-04	3.21e-07	9.15e-13	2.40e-15	9.04e-16
≤ 0.95	$\overline{\epsilon}_{N_{\mathrm{H}},k}$	4.01e-02	3.25e-03	1.34e-04	1.29e-06	8.40e-10	8.38e-15
	$\overline{\sigma}_{N_{\mathrm{H}},k}$	8.87e-02	1.43e-02	1.26e-03	3.32e-05	5.29e-08	1.28e-12
	$\overline{\mathcal{M}}_{N_{\mathrm{H}},k}$	1.00e-02	1.38e-04	3.67e-08	5.14e-15	1.86e-15	6.94e-16
≤ 0.9	$\overline{\epsilon}_{N_{\mathrm{H}},k}$	2.65e-02	7.63e-04	4.18e-06	3.50e-09	5.96e-14	1.10e-15
	$\overline{\sigma}_{N_{\mathrm{H}},k}$	6.66e-02	4.77e-03	6.26e-05	2.07e-07	1.21e-11	1.19e-15
	$\overline{\mathcal{M}}_{N_{\mathrm{H}},k}$	6.24e-03	4.92e-05	5.15e-09	2.30e-15	1.67e-15	6.73e-16
≤ 0.85	$\overline{\epsilon}_{N_{\mathrm{H}},k}$	2.21e-02	4.45e-04	2.79e-07	1.42e-11	1.75e-15	1.07e-15
	$\overline{\sigma}_{N_{\mathrm{H}},k}$	5.88e-02	3.96e-03	6.45e-06	8.96e-10	9.83e-16	1.14e-15
	$\overline{\mathcal{M}}_{N_{\mathrm{H}},k}$	4.33e-03	1.98e-05	7.67e-10	1.63e-15	1.65e-15	6.60e-16
≤ 0.8	$\overline{\epsilon}_{N_{\mathrm{H}},k}$	2.04e-02	3.82e-04	1.02e-07	6.87e-14	1.72e-15	1.04e-15
	$\overline{\sigma}_{N_{\mathrm{H}},k}$	5.43e-02	2.81e-03	3.34e-06	4.61e-12	9.11e-16	1.10e-15
	$\overline{\mathcal{M}}_{N_{\mathrm{H}},k}$	3.51e-03	1.05e-05	1.66e-10	1.47e-15	1.65e-15	6.57e-16
≤ 0.75	$\overline{\epsilon}_{N_{\mathrm{H}},k}$	2.01e-02	3.95e-04	1.17e-07	5.72e-15	1.70e-15	1.00e-15
	$\overline{\sigma}_{N_{\mathrm{H}},k}$	5.07e-02	3.23e-03	4.33e-06	1.77e-13	8.78e-16	1.05e-15
	$\overline{\mathcal{M}}_{N_{\mathrm{H}},k}$	3.64e-03	9.84e-06	9.04e-11	1.46e-15	1.66e-15	6.44e-16
≤ 0.7	$\overline{\epsilon}_{N_{\mathrm{H}},k}$	1.98e-02	3.92e-04	1.11e-07	5.81e-15	1.68e-15	9.76e-16
	$\overline{\sigma}_{N_{\mathrm{H}},k}$	4.68e-02	2.45e-03	2.13e-06	1.45e-13	8.50e-16	1.01e-15
	$\overline{\mathcal{M}}_{N_{\mathrm{H}},k}$	4.48e-03	1.45e-05	1.51e-10	1.50e-15	1.66e-15	6.36e-16

Table 6.1.: Different statistics of error estimator ϵ using $k = 5$ and $N_{\boldsymbol{u},\boldsymbol{\psi}} = 10^5$.

To analyze the smoothness of a multidimensional standard Gaussian joint CDF we define

$$
\begin{aligned}
f(y) :=& \underbrace{\phi(y)}_{=:g_0(y)} \prod_{j=1}^{k} \underbrace{\Phi\left(\frac{u_j - \psi_j \cdot y}{\sqrt{1-\psi_j^2}}\right)}_{=:g_j(y)} \\
=& \prod_{j=0}^{k} g_j(y).
\end{aligned}
$$

Obviously, the smoothness of f depends on the smoothness of $g_0, \ldots, g_k$. In general, these functions are numerically well integrable, but within functions $g_1, \ldots, g_k$ we have to pay attention to the argument

$$
h_j(y) = \frac{u_j - \psi_j \cdot y}{\sqrt{1-\psi_j^2}}.
$$

If function $h_j(y)$ shows a high gradient $-\psi_j/\sqrt{1-\psi_j^2}$, then $g_j(y)$ as well as $f(y)$ will also show a high gradient. This behavior is passed on to any further derivative of f, because the n–th derivative of f is given by

$$f^{(n)}(y) = \sum_{\boldsymbol{i}\in\mathcal{I}} \underbrace{\binom{n}{i_0,\ldots,i_k} \prod_{j=0}^{k} \frac{\partial^{i_j} g_j(y)}{\partial y^{i_j}}}_{=:s_{\boldsymbol{i}}(y)},$$

in which $\mathcal{I}$ denotes the set

$$\mathcal{I} := \left\{ \boldsymbol{i} = \{i_0,\ldots,i_k\} \in \{0,1,\ldots,k\}^{k+1} \,\middle|\, \sum_{j=0}^{k} i_j = n \right\}$$

and $\binom{n}{i_0,\ldots,i_k}$ denotes the multinomialcoefficient (cp. [Maz63])

$$\binom{n}{i_0,\ldots,i_k} = \frac{n!}{i_0!\cdot\ldots\cdot i_k!}.$$

In many summands $s_{\boldsymbol{i}}(y)$ we consequently multiply factors like $-\psi_j/\sqrt{1-\psi_j^2}$, which causes extreme gradients of f, especially for values $|\psi_j| \approx 1$. Hence, it makes sense to restrict the domain of $\boldsymbol{\psi}$ to an interval $\mathcal{I}_{\psi} \subset [-1,1]^k$.

All in all we can sum up that in most cases a GHQ with a moderate number of nodes N_{H} is well suited for approximating the multidimensional standard Gaussian joint CDF with one–factor correlation. Only in extreme cases ($|\psi_j| \approx 1$ or k large) we should enlarge the number of nodes or should apply other approximation methods.

Remark 6.6. Because of the high dependence on parameters k and $\boldsymbol{\psi}$, it makes sense to choose the number of nodes N_{H} dynamically with respect to k and $\boldsymbol{\psi}$. Therefore, in the following we choose the relation shown in Table 6.2. Choosing these numbers of nodes we can reach an average error bound of 10^{-14} (excepted $N_{\mathrm{H}} = 256$). If $k > 30$ holds, we will choose $N_{\mathrm{H}} = 256$ without restriction.

In summary, by means of a GHQ we can efficiently evaluate the multivariate standard Gaussian joint CDF, which implies an efficient evaluation of $F_{\tau_{\iota_m}}$. For further analyses we refer to [SH12a].

Let us now return to the evaluation of an mBDS. Using a Gaussian copula we can state its value as

$$V_m = \int_0^\infty \widetilde{\Lambda}(t)\cdot f_{\tau_{\iota_m}}(t)dt = \widetilde{\Lambda}_\infty - \int_0^T \widetilde{\Lambda}'(t)\cdot F_{\tau_{\iota_m}}(t)\mathrm{d}t, \qquad (6.27)$$

in which $F_{\tau_{\iota_m}}$ is given by equation (6.21) and T denotes the maturity of the underlying contract.

$k \downarrow$	$\lvert\psi_j\rvert \leq 1$	$\lvert\psi_j\rvert \leq 0.95$	$\lvert\psi_j\rvert \leq 0.9$	$\lvert\psi_j\rvert \leq 0.85$	$\lvert\psi_j\rvert \leq 0.8$	$\lvert\psi_j\rvert \leq 0.75$	$\lvert\psi_j\rvert \leq 0.7$
≤ 5	256	128	64	64	32	32	32
≤ 10	256	256	128	64	64	64	32
≤ 20	256	256	128	128	128	64	64
≤ 30	256	256	256	128	128	128	64

Table 6.2.: Suggested number of nodes N_{H} within a GHQ, in which $\boldsymbol{\psi}$ and k are known.

This (outer) integral is approximated by applying a Gauss–Kronrod quadrature (abbr.: GKQ). Except the integral approximation, this quadrature is able to state an error estimator simultaneously.

The GKQ is an extension of the Gauss–Legendre quadrature (abbr.: GLQ). Let $x_1^{(\mathrm{L})} < \ldots < x_{N_{\mathrm{L}}}^{(\mathrm{L})}$ denote the N_{L} different nodes of a GLQ. Then, a GKQ consists of the N_{K} different nodes $x_1^{(\mathrm{K})} < \ldots < x_{N_{\mathrm{K}}}^{(\mathrm{K})}$ with $N_{\mathrm{K}} = 2N_{\mathrm{L}} + 1$ and $x_j^{(\mathrm{K})} = x_{j/2}^{(\mathrm{L})}$ for each $j \in \{2, 4, \ldots, 2N_{\mathrm{L}}\}$. Although the resulting quadrature is of lower order than a GLQ using N_{K} different nodes, it is able to state an error estimator based on $x_1^{(\mathrm{L})} < \ldots < x_{N_{\mathrm{L}}}^{(\mathrm{L})}$ simultaneously.

Let V_m be the value of any mBDS, which is defined according to equation (6.27). Then, the two–dimensional integral within V_m is approximated by using a Gauss–Kronrod quadrature with a nested Gauss–Hermite quadrature (abbr.: GKHQ) as

$$\begin{aligned}
\widetilde{\Lambda}_\infty - V_m &= \int_0^T \widetilde{\Lambda}'(t) \cdot F_{\tau_{\iota_m}}(t) \mathrm{d}t \\
&= \int_0^T \widetilde{\Lambda}'(t) \cdot \sum_{k=m}^{d} \underbrace{(-1)^{k-m} \binom{k-1}{m-1}}_{=: c_{k,m}} \sum_{\substack{I \subseteq \{1,\ldots,d\} \\ |I| = k}} C_{k,\Sigma_k^I}^{I,\mathrm{Gau}} \left(F_{\tau_{I_1}}(t), \ldots, F_{\tau_{I_k}}(t) \right) \mathrm{d}t \\
&= \int_0^T \widetilde{\Lambda}'(t) \cdot \sum_{k=m}^{d} c_{k,m} \sum_{\substack{I \subseteq \{1,\ldots,d\} \\ |I| = k}} \int_{\mathbb{R}} \phi(y) \prod_{i=1}^{k} \underbrace{\Phi \left(\frac{\Phi^{-1}\left(F_{\tau_{I_i}}(t)\right) - \psi_{I_i} \cdot y}{\sqrt{1 - \psi_{I_i}^2}} \right)}_{=: \Phi_{I_i}(t,y)} \mathrm{d}y \mathrm{d}t \qquad (6.28) \\
&\approx \frac{T}{2} \sum_{\beta=1}^{N_{\mathrm{K}}} w_\beta^{(\mathrm{K})} \widetilde{\Lambda}' \left(t_\beta^{(\mathrm{K})} \right) \sum_{k=m}^{d} c_{k,m} \sum_{\substack{I \subseteq \{1,\ldots,d\} \\ |I| = k}} \sum_{\alpha=1}^{N_{\mathrm{H}}} \frac{w_\alpha^{(\mathrm{H})} \cdot \phi(x_\alpha^{(\mathrm{H})})}{\exp\left(- \left(x_\alpha^{(\mathrm{H})} \right)^2 \right)} \prod_{i=1}^{k} \Phi_{I_i} \left(t_\beta^{(\mathrm{K})}, x_\alpha^{(\mathrm{H})} \right) \\
&\qquad (6.29) \\
&=: \widetilde{\Lambda}_\infty - V_{m,\mathrm{GKHQ,plain}}^{(N_{\mathrm{K}},N_{\mathrm{H}})}.
\end{aligned}$$

At this, N_{K} denotes the odd number of nodes within the GKQ and N_{H} denotes the number of nodes within the GHQ. Furthermore, $x_1^{(\mathrm{H})}, \ldots, x_{N_{\mathrm{H}}}^{(\mathrm{H})}$ and $w_1^{(\mathrm{H})}, \ldots, w_{N_{\mathrm{H}}}^{(\mathrm{H})}$ denote the nodes and weights of the GHQ and $w_1^{(\mathrm{K})}, \ldots, w_{N_{\mathrm{K}}}^{(\mathrm{K})}$ denote the weights of the GKQ. In addition, $t_1^{(\mathrm{K})}, \ldots, t_{N_{\mathrm{K}}}^{(\mathrm{K})} \in [0,T]^{N_{\mathrm{K}}}$ denote the transformed nodes of the GKQ, which are calculated by $t_i^{(\mathrm{K})} = \left(x_i^{(\mathrm{K})} + 1\right) \cdot T/2$ for each $i \in \{1, \ldots, N_{\mathrm{K}}\}$.

As already mentioned, within the calculation of $V_{m,\mathrm{GKHQ,plain}}^{(N_{\mathrm{K}},N_{\mathrm{H}})}$ we can easily create an approximation $V_{m,\mathrm{GLHQ,plain}}^{(N_{\mathrm{L}},N_{\mathrm{H}})}$ based on a Gauss–Legendre quadrature with a nested Gauss–Hermite quadrature (abbr.: GLHQ). Of course this approximation has a lower order than $V_{m,\mathrm{GKHQ,plain}}^{(N_{\mathrm{K}},N_{\mathrm{H}})}$, but it is very useful for calculating an error estimator. For this, let $w_1^{(\mathrm{L})}, \ldots, w_{N_{\mathrm{L}}}^{(\mathrm{L})}$ denote the weights of the GLQ. Then, $V_{m,\mathrm{GLHQ,plain}}^{(N_{\mathrm{L}},N_{\mathrm{H}})}$ is given by

$$\widetilde{\Lambda}_\infty - V_{m,\mathrm{GLHQ,plain}}^{(N_{\mathrm{L}},N_{\mathrm{H}})} :=$$
$$\frac{T}{2} \sum_{\beta'=1}^{N_{\mathrm{L}}} w_{\beta'}^{(\mathrm{L})} \widetilde{\Lambda}' \left(t_{2\beta'}^{(\mathrm{K})}\right) \sum_{k=m}^{d} c_{k,m} \sum_{\substack{I \subseteq \{1,\ldots,d\} \\ |I|=k}} \sum_{\alpha=1}^{N_{\mathrm{H}}} \frac{w_\alpha^{(\mathrm{H})} \cdot \phi(x_\alpha^{(\mathrm{H})})}{\exp\left(-\left(x_\alpha^{(\mathrm{H})}\right)^2\right)} \prod_{i=1}^{k} \Phi_{I_i}\left(t_{2\beta'}^{(\mathrm{K})}, x_\alpha^{(\mathrm{H})}\right) \quad (6.30)$$

and is calculated simultaneously to $V_{m,\mathrm{GKHQ,plain}}^{(N_{\mathrm{K}},N_{\mathrm{H}})}$. Using these two approximations we can easily calculate a corresponding error estimator $\epsilon_{m,\mathrm{plain}}^{(N_{\mathrm{K}},N_{\mathrm{H}})}$ as (cp. [Sha08])

$$\epsilon_{m,\mathrm{plain}}^{(N_{\mathrm{K}},N_{\mathrm{H}})} := \left| \frac{V_{m,\mathrm{GLHQ,plain}}^{(N_{\mathrm{L}},N_{\mathrm{H}})} - V_{m,\mathrm{GKHQ,plain}}^{(N_{\mathrm{K}},N_{\mathrm{H}})}}{V_{m,\mathrm{GKHQ,plain}}^{(N_{\mathrm{K}},N_{\mathrm{H}})}} \right| . \quad (6.31)$$

This error estimator is widely used, for example within the quadrature routine `quadgk` of the software package MATLAB® (cp. [TM11]). For a detailed introduction to GKQ, GLQ and GHQ we refer to [Kro65, PdDKÜK83, KMN89, FH07, SK09] and for an efficient calculation of $x_1^{(\mathrm{K})} < \ldots < x_{N_{\mathrm{K}}}^{(\mathrm{K})}$ compare [Lau97, Gau04].

Complexity and implementation

Let us analyze the complexity of $V_{m,\mathrm{GKHQ,plain}}^{(N_{\mathrm{K}},N_{\mathrm{H}})}$ and discuss its implementation. For this, we reform (6.29) to

$$\widetilde{\Lambda}_\infty - V_{m,\mathrm{GKHQ,plain}}^{(N_{\mathrm{K}},N_{\mathrm{H}})}$$
$$= \frac{T}{2} \sum_{\beta=1}^{N_{\mathrm{K}}} \underbrace{w_\beta^{(\mathrm{K})} \widetilde{\Lambda}' \left(t_\beta^{(\mathrm{K})}\right)}_{\widetilde{w}_\beta^{(\mathrm{K})}} \sum_{\alpha=1}^{N_{\mathrm{H}}} \underbrace{\frac{w_\alpha^{(\mathrm{H})} \cdot \phi(x_\alpha^{(\mathrm{H})})}{\exp\left(-\left(x_\alpha^{(\mathrm{H})}\right)^2\right)}}_{\widetilde{w}_\alpha^{(\mathrm{H})}} \sum_{k=m}^{d} c_{k,m} \sum_{\substack{I \subseteq \{1,\ldots,d\} \\ |I|=k}} \prod_{i=1}^{k} \Phi_{I_i}\left(t_\beta^{(\mathrm{K})}, x_\alpha^{(\mathrm{H})}\right) . \quad (6.32)$$

Obviously, we can now apply the same simplifications as we applied in case of independent default times (cp. Lemmata 6.3 and 6.4). For this let $\widetilde{\eta}_\Phi(i,j)(t,x)$ be defined analogically to $\widetilde{\eta}(i,j)(t)$ in equation (6.17), with the difference that $F_{\tau_i}(t)$ is replaced by $\Phi_i\left(t,x\right)$. Then, we can state

$$\begin{aligned} V^{(N_{\mathrm{K}},N_{\mathrm{H}})}_{m,\mathrm{GKHQ,nested}} &:= \widetilde{\Lambda}_\infty - \frac{T}{2}\sum_{\beta=1}^{N_{\mathrm{K}}}\widetilde{w}^{(\mathrm{K})}_\beta \sum_{\alpha=1}^{N_{\mathrm{H}}}\widetilde{w}^{(\mathrm{H})}_\alpha \sum_{k=m}^{d} c_{k,m}\widetilde{\eta}_\Phi(k,d-k+1)\left(t^{(\mathrm{K})}_\beta, x^{(\mathrm{H})}_\alpha\right) \qquad (6.33)\\ &= V^{(N_{\mathrm{K}},N_{\mathrm{H}})}_{m,\mathrm{GKHQ,plain}} \end{aligned}$$

in which $\widetilde{\eta}_\Phi(k,l)(t,x)$ is defined recursively by

$$\widetilde{\eta}_\Phi(k,l)(t,x) := \begin{cases} \Phi_1\left(t,x\right), & \text{if } k=l=1\\ \widetilde{\eta}_\Phi(k,l-1)(t,x)+\Phi_l(t,x), & \text{if } k=1,\ l>1\\ \widetilde{\eta}_\Phi(k-1,l)(t,x)\cdot\Phi_k(t,x), & \text{if } k>1,\ l=1\\ \widetilde{\eta}_\Phi(k-1,l)(t,x)\cdot\Phi_{k+l-1}(t,x)+\widetilde{\eta}_\Phi(k,l-1)(t,x), & \text{if } k>1,\ l>1 \end{cases} \qquad (6.34)$$

for any $(t,x)\in\overline{\mathbb{R}}_{\geq 0}\times\mathbb{R}$ and for each $k\in\{1,\dots,d\}$ and $l\in\{1,\dots,\min\{d-k+1,d-m+1\}\}$.

By using the simplification above the calculation of $V^{(N_{\mathrm{K}},N_{\mathrm{H}})}_{m,\mathrm{GKHQ,nested}}$ requires a number of arithmetic operations, which is estimated by $\mathcal{O}\left(N_{\mathrm{K}}\cdot N_{\mathrm{H}}\cdot d^2\right)$. In the special case $m=1$ they can even be estimated by $\mathcal{O}\left(N_{\mathrm{K}}\cdot N_{\mathrm{H}}\cdot d\right)$ (cp. Lemma 6.3). If we calculate $V^{(N_{\mathrm{K}},N_{\mathrm{H}})}_{m,\mathrm{GKHQ,plain}}$ (cp. equation (6.32)) without using the simplifications from Lemmata 6.3 and 6.4, we will get an overall complexity of $\mathcal{O}\left(N_{\mathrm{K}}\cdot N_{\mathrm{H}}\cdot 2^d\right)$. If we even neglect the usage of $F_{\tau_{\iota_m}}$ and apply a d–dimensional quadrature, we will get an overall complexity of $\mathcal{O}\left(N^d\right)$, in which N denotes the number of nodes in each direction. This tremendous effort reduction is further illustrated in Subsection 6.1.5, in which a real credit derivative is evaluated.

Next, we discuss the implementation of equation (6.33). For reducing the CPU time significantly we have to attend the storage of values, which are required frequently. For this, we initialize a matrix $\mathbb{X}\in\mathbb{R}^{d\times N_{\mathrm{K}}\times N_{\mathrm{H}}}$ as

$$\mathbb{X}=\left(\mathbb{X}\right)_{i,j,k} := \Phi_i\left(t^{(\mathrm{K})}_j, x^{(\mathrm{H})}_k\right). \qquad (6.35)$$

before running the underlying algorithm. Additionally, we initialize vectors

$$\begin{aligned} &\widetilde{\boldsymbol{w}}^{(\mathrm{K})} = \left(w^{(\mathrm{K})}_1,\dots,w^{(\mathrm{K})}_{N_{\mathrm{K}}}\right)^{\mathrm{tr}}\in\mathbb{R}^{N_{\mathrm{K}}}\\ \text{and}\quad &\widetilde{\boldsymbol{w}}^{(\mathrm{H})} = \left(w^{(\mathrm{H})}_1,\dots,w^{(\mathrm{H})}_{N_{\mathrm{H}}}\right)^{\mathrm{tr}}\in\mathbb{R}^{N_{\mathrm{H}}} \end{aligned}$$

according to equation (6.32).

The detailed implementation of calculating $V^{(N_\mathrm{K},N_\mathrm{H})}_{m,\mathrm{GKHQ,nested}}$ is given in Algorithm 6.1. The implementation of $V^{(N_\mathrm{K},N_\mathrm{H})}_{m,\mathrm{GKHQ,plain}}$ is neglected here, because it has a higher complexity than the calculation of $V^{(N_\mathrm{K},N_\mathrm{H})}_{m,\mathrm{GKHQ,nested}}$ in any possible case. For the general approach of implementing $V^{(N_\mathrm{K},N_\mathrm{H})}_{m,\mathrm{GKHQ,plain}}$ we refer to the implementation of $V^{(N_\mathrm{K},\mathrm{Arc})}_{m,\mathrm{GKQ,plain}}$ (cp. Subsection 6.1.4), which applies an analogical method.

Algorithm 6.1: Calculation of $V^{(N_\mathrm{K},N_\mathrm{H})}_{m,\mathrm{GKHQ,nested}}$

```
Input  : $\mathbb{X}$, $\widetilde{\boldsymbol{w}}^{(\mathrm{K})}$, $\widetilde{\boldsymbol{w}}^{(\mathrm{H})}$
Output: $V^{(N_\mathrm{K},N_\mathrm{H})}_{m,\mathrm{GKHQ,nested}}$
Global : $d$, $m$, $N_\mathrm{K}$, $N_\mathrm{H}$

Initialize floating–point numbers $E_1 = E_2 = E_3 \leftarrow 0$ and $V^{(N_\mathrm{K},N_\mathrm{H})}_{m,\mathrm{GKHQ,nested}} \leftarrow \widetilde{\Lambda}_\infty$
for $i \leftarrow 1$ to $N_K$ do
    $E_2 \leftarrow 0$
    for $j \leftarrow 1$ to $N_H$ do
        if $m = 1$ then
            $E_3 \leftarrow$ CalculateProduct $(\mathbb{X}, i, j)$ // cp.  Algorithm 6.2
        else
            $E_3 \leftarrow$ CalculateSum $(m, \mathbb{X}, i, j)$ // cp.  Algorithm 6.3
        $E_2 \leftarrow E_2 + \widetilde{w}^{(\mathrm{H})}_j \cdot E_3$
    $E_1 \leftarrow E_1 + \widetilde{w}^{(\mathrm{K})}_i \cdot E_2$
$V^{(N_\mathrm{K},N_\mathrm{H})}_{m,\mathrm{GKHQ,nested}} \leftarrow V^{(N_\mathrm{K},N_\mathrm{H})}_{m,\mathrm{GKHQ,nested}} - \frac{T}{2} \cdot E_1$
```

In Algorithm 6.1 we first test whether $m = 1$ holds or not. If it holds, we will calculate

$$\sum_{k=m}^{d} c_{k,m} \sum_{\substack{I\subseteq\{1,\dots,d\}\\|I|=k}} \prod_{i=1}^{k} \Phi_{I_i}(t,x)$$

in function `CalculateProduct` (cp. Algorithm 6.2) according to Lemma 6.3. If $m = 1$ does not hold, we will have to calculate this term via function `CalculateSum` in Algorithm 6.3 (according to Lemma 6.4).

Considering Algorithms 6.1 and 6.3 in detail, it is obvious that already small modifications offer the opportunity to evaluate any $m^{'}$–th to Default Swap with $m^{'} > m$. In Algorithm 6.4 we show these modifications and present an algorithm, which is able to price each m–th to Default Swap simultaneously, with $m \in \{1,\dots,d\}$.

Algorithm 6.2: `CalculateProduct(`$\mathbb{X}$`, `i`, `j`)`

```
Input  : $\mathbb{X}$, $i$, $j$
Output: $\mathcal{X} \in \mathbb{R}$
Global : $d$, $N_{\mathrm{K}}$, $N_{\mathrm{H}}$
Initialize $\mathcal{X} \leftarrow (-1)^{d-1}\mathbb{X}_{1,i,j}$
for $k = 2$ to $k = d$ do
    $\mathcal{X} \leftarrow (\mathbb{X}_{k,i,j} - 1) \cdot \mathcal{X} + (-1)^{d-k} \cdot \mathbb{X}_{k,i,j}$
```

Algorithm 6.3: `CalculateSum(`m`, `$\mathbb{X}$`, `i`, `j`)`

```
Input  : $m$, $\mathbb{X}$, $i$, $j$
Output: $\mathcal{X} \in \mathbb{R}$
Global : $d$, $N_{\mathrm{K}}$, $N_{\mathrm{H}}$
Initialize $\mathcal{X} \leftarrow 0$ and $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{d-m+1}$
for $k = 1$ to $k = d$ do
    for $l = 1$ to $l = \min\{d-m+1, d-k+1\}$ do
        if $(k > 1$ and $l > 1)$ then
            $\mathcal{Y}_l \leftarrow \mathcal{Y}_l \cdot \mathbb{X}_{k+l-1,i,j} + \mathcal{Y}_{l-1}$
        else
            if $(k > 1$ and $l = 1)$ then
                $\mathcal{Y}_l \leftarrow \mathcal{Y}_l \cdot \mathbb{X}_{k,i,j}$
            else
                if $(k = 1$ and $l > 1)$ then
                    $\mathcal{Y}_l \leftarrow \mathcal{Y}_{l-1} + \mathbb{X}_{l,i,j}$
                else
                    $\mathcal{Y}_l \leftarrow \mathbb{X}_{1,i,j}$
    if $k >= m$ then
        $\mathcal{X} \leftarrow \mathcal{X} + c_{k,m}\mathcal{Y}_{d-k+1}$
```

For this, let $\widetilde{\boldsymbol{c}}_k \in \mathbb{Z}^d$ be defined as

$$\widetilde{\boldsymbol{c}}_k = \widetilde{c}_{k_i} := \begin{cases} c_{k,i}, & \text{if } k \geq i \\ 0, & \text{else} \end{cases} . \tag{6.36}$$

Then, we can easily modify Algorithms 6.1 and 6.3 into vector algorithms. Finally, Algorithm 6.4 returns the vector $\boldsymbol{V}^{(N_\mathrm{K},N_\mathrm{H})}_{\mathrm{GKHQ,nested}} = \left(V^{(N_\mathrm{K},N_\mathrm{H})}_{1,\mathrm{GKHQ,nested}}, \dots, V^{(N_\mathrm{K},N_\mathrm{H})}_{d,\mathrm{GKHQ,nested}}\right)^{\mathrm{tr}}$ (cp. Algorithms 6.4 and 6.5).

Algorithm 6.4: Calculation of $\boldsymbol{V}^{(N_\mathrm{K},N_\mathrm{H})}_{\mathrm{GKHQ,nested}}$

```
Input  : $\mathbb{X}$, $\widetilde{\boldsymbol{w}}^{(\mathrm{K})}$, $\widetilde{\boldsymbol{w}}^{(\mathrm{H})}$
Output: $\boldsymbol{V}^{(N_\mathrm{K},N_\mathrm{H})}_{\mathrm{GKHQ,nested}}$
Global : $d$, $N_\mathrm{K}$, $N_\mathrm{H}$

Initialize floating–point vectors $\boldsymbol{E}_1 = \boldsymbol{E}_2 = \boldsymbol{E}_3 \leftarrow \boldsymbol{0} \in \mathbb{R}^d$ and
$\boldsymbol{V}^{(N_\mathrm{K},N_\mathrm{H})}_{\mathrm{GKHQ,nested}} \leftarrow \left(\widetilde{\Lambda}_\infty, \dots, \widetilde{\Lambda}_\infty\right)^{\mathrm{tr}} \in \mathbb{R}^d$
for $i \leftarrow 1$ to $N_K$ do
    $\boldsymbol{E}_2 \leftarrow \boldsymbol{0}$
    for $j \leftarrow 1$ to $N_H$ do
        $\boldsymbol{E}_3 \leftarrow$ CalculateSumVec $(\mathbb{X}, i, j)$
        $\boldsymbol{E}_2 \leftarrow \boldsymbol{E}_2 + \widetilde{w}^{(\mathrm{H})}_j \cdot \boldsymbol{E}_3$
    $\boldsymbol{E}_1 \leftarrow \boldsymbol{E}_1 + \widetilde{w}^{(\mathrm{K})}_i \cdot \boldsymbol{E}_2$
$\boldsymbol{V}^{(N_\mathrm{K},N_\mathrm{H})}_{\mathrm{GKHQ,nested}} \leftarrow \boldsymbol{V}^{(N_\mathrm{K},N_\mathrm{H})}_{\mathrm{GKHQ,nested}} - \frac{T}{2} \cdot \boldsymbol{E}_1$
```

In summary we remark that we are able to evaluate any mBDS with respect to the Gaussian copula efficiently by means of Algorithms 6.1 to 6.5. Furthermore, the usage of a GKQ offers the opportunity to calculate an error estimator besides. For this, we simply have to modify Algorithms 6.1 and 6.4 by adding another variable, which calculates $V^{(N_\mathrm{L},N_\mathrm{H})}_{m,\mathrm{GLHQ,nested}}$ (cp. equation (6.30)) via a GLHQ. Afterwards we can calculate the error estimator according to equation (6.31). Again, we highlight that the complexity of above evaluations is bounded by $\mathcal{O}\left(N_\mathrm{K} N_\mathrm{H} d^2\right)$ in the worst case.

6.1.4. Application of Archimedean copulae

Applying Archimedean copulae in order to evaluate any mBDS induces much less difficulties than applying the Gaussian copula. Usually, it is possible to evaluate Archimedean copulae as well as their generators in a straight forward fashion. They do not have to be approximated.

Algorithm 6.5: `CalculateSumVec(`$\mathbb{X}$`, `i`, `j`)`

```
Input  : $\mathbb{X}$, $i$, $j$
Output: $\boldsymbol{\mathcal{X}} \in \mathbb{R}^m$
Global : $d$, $N_\mathrm{K}$, $N_\mathrm{H}$
Initialize floating–point vectors $\boldsymbol{\mathcal{X}} \leftarrow \mathbf{0}$ and $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{d-m+1}$
for $k = 1$ to $k = d$ do
    for $l = 1$ to $l = d - k + 1$ do
        if $(k > 1$ and $l > 1)$ then
            $\mathcal{Y}_l \leftarrow \mathcal{Y}_l \cdot \mathbb{X}_{k+l-1,i,j} + \mathcal{Y}_{l-1}$
        else
            if $(k > 1$ and $l = 1)$ then
                $\mathcal{Y}_l \leftarrow \mathcal{Y}_l \cdot \mathbb{X}_{k,i,j}$
            else
                if $(k = 1$ and $l > 1)$ then
                    $\mathcal{Y}_l \leftarrow \mathcal{Y}_{l-1} + \mathbb{X}_{l,i,j}$
                else
                    $\mathcal{Y}_l \leftarrow \mathbb{X}_{1,i,j}$
    $\boldsymbol{\mathcal{X}} \leftarrow \boldsymbol{\mathcal{X}} + \widetilde{\boldsymbol{c}}_k \cdot \mathcal{Y}_{d-k+1}$
```

In the following we assume that the joint CDF $F_{\boldsymbol{\tau}_I}$ of random vector $\boldsymbol{\tau}_I = \left(\tau_{I_1}, \ldots, \tau_{I_k}\right)^{\mathrm{tr}}$ is given by

$$\begin{aligned} F_{\boldsymbol{\tau}_I}\left(t_{I_1}, \ldots, t_{I_k}\right) &= \check{C}_k^I\left(F_{\tau_{I_1}}\left(t_{I_1}\right), \ldots, F_{\tau_{I_k}}\left(t_{I_k}\right)\right) \\ &= C_{k,\phi_\theta}^{I,\mathrm{Arc}}\left(F_{\tau_{I_1}}\left(t_{I_1}\right), \ldots, F_{\tau_{I_k}}\left(t_{I_k}\right)\right) \\ &= \phi_\theta^{-1}\left[\phi_\theta\left(F_{\tau_{I_1}}\left(t_{I_1}\right)\right) + \ldots + \phi_\theta\left(F_{\tau_{I_k}}\left(t_{I_k}\right)\right)\right] \end{aligned} \tag{6.37}$$

for any $\boldsymbol{t}_I \in \overline{\mathbb{R}}_{\geq 0}^k$ and for each $I = \{I_1, \ldots, I_k\} \subseteq \{1, \ldots, d\}$. The aim of this subsection is the efficient calculation of V_m. Here, V_m denotes the value of an mBDS with maturity T and is given by

$$V_m = \int\limits_0^\infty \widetilde{\Lambda}(t) \cdot f_{\tau_{\iota_m}}(t)dt = \widetilde{\Lambda}_\infty - \int\limits_0^T \widetilde{\Lambda}'(t) \cdot F_{\tau_{\iota_m}}(t)\mathrm{d}t,$$

with

$$F_{\tau_{\iota_m}}(t) = \sum_{k=m}^d (-1)^{k-m}\binom{k-1}{m-1} \sum_{\substack{I\subseteq\{1,\ldots,d\}\\|I|=k}} C_{k,\phi_\theta}^{I,\mathrm{Arc}}\left(F_{\tau_{I_1}}(t), \ldots, F_{\tau_{I_k}}(t)\right) \tag{6.38}$$

for any $t \in \overline{\mathbb{R}}_{\geq 0}$. Due to the usage of an Archimedean copula, we can approximate V_m by applying an one–dimensional Gauss–Kronrod quadrature. Evaluating the multivariate standard Gaussian joint CDF (cp. Subsection 6.1.3) is no challenge anymore.

Let N_{K} denote the number of nodes within the GKQ. Then, V_m is approximated by

$$\begin{aligned} &\widetilde{\Lambda}_\infty - V_m = \int\limits_0^T \widetilde{\Lambda}'(t) \cdot F_{\tau_{\iota_m}}(t)\mathrm{d}t \\ &= \int\limits_0^T \widetilde{\Lambda}'(t) \cdot \sum_{k=m}^d \underbrace{(-1)^{k-m}\binom{k-1}{m-1}}_{=:c_{k,m}} \sum_{\substack{I\subseteq\{1,\ldots,d\}\\|I|=k}} C_{k,\phi_\theta}^{I,\mathrm{Arc}}\left(F_{\tau_{I_1}}(t), \ldots, F_{\tau_{I_k}}(t)\right)\mathrm{d}t \\ &= \int\limits_0^T \widetilde{\Lambda}'(t) \cdot \sum_{k=m}^d c_{k,m} \sum_{\substack{I\subseteq\{1,\ldots,d\}\\|I|=k}} \phi_\theta^{-1}\left[\phi_\theta\left(F_{\tau_{I_1}}(t)\right) + \ldots + \phi_\theta\left(F_{\tau_{I_k}}(t)\right)\right]\mathrm{d}t \qquad (6.39) \\ &\approx \frac{T}{2}\sum_{\alpha=1}^{N_{\mathrm{K}}} w_\alpha^{\mathrm{K}}\widetilde{\Lambda}'\left(t_\alpha^{(\mathrm{K})}\right) \cdot \sum_{k=m}^d c_{k,m} \sum_{\substack{I\subseteq\{1,\ldots,d\}\\|I|=k}} \phi_\theta^{-1}\left[\phi_\theta\left(F_{\tau_{I_1}}\left(t_\alpha^{(\mathrm{K})}\right)\right) + \ldots + \phi_\theta\left(F_{\tau_{I_k}}\left(t_\alpha^{(\mathrm{K})}\right)\right)\right] \\ &\qquad (6.40) \\ &=: \widetilde{\Lambda}_\infty - V_{m,\mathrm{GKQ,plain}}^{(N_{\mathrm{K}},\mathrm{Arc})}. \end{aligned}$$

At this, $x_1^{(\mathrm{K})}, \ldots, x_{N_{\mathrm{K}}}^{(\mathrm{K})}$ and $w_1^{(\mathrm{K})}, \ldots, w_{N_{\mathrm{K}}}^{(\mathrm{K})}$ denote the nodes and weights within the GKQ and $t_i^{(\mathrm{K})} = \left(x_i^{(\mathrm{K})} + 1\right) \cdot T/2$ holds for each $i \in \{1, \ldots, N_{\mathrm{K}}\}$. In the following we choose Arc $\in$ {Cla,Fra,Gum} without restriction (cp. Table 4.1).

Analogically to Subsection 6.1.3, the GKQ offers an easy possibility for calculating an error estimator simultaneously to the approximation $V_{m,\mathrm{GKQ,plain}}^{(N_{\mathrm{K}},\mathrm{Arc})}$. For this, we have to calculate an approximation $V_{m,\mathrm{GLQ,plain}}^{(N_{\mathrm{L}},\mathrm{Arc})}$, which is based on a $N_{\mathrm{L}} = (N_{\mathrm{K}} - 1)/2$ –element GLQ with nodes $x_1^{(\mathrm{L})} = x_2^{(\mathrm{K})}, \ldots, x_{N_{\mathrm{L}}}^{(\mathrm{L})} = x_{2N_{\mathrm{L}}}^{(\mathrm{K})}$. Then, $V_{m,\mathrm{GLQ,plain}}^{(N_{\mathrm{L}},\mathrm{Arc})}$ is given by

$$\widetilde{\Lambda}_\infty - V_{m,\mathrm{GLQ,plain}}^{(N_{\mathrm{L}},\mathrm{Arc})} : =$$

$$\frac{T}{2}\sum_{\alpha'=1}^{N_{\mathrm{L}}} w_{\alpha'}^{(\mathrm{L})}\widetilde{\Lambda}'\left(t_{2\alpha'}^{(\mathrm{K})}\right)\sum_{k=m}^{d} c_{k,m}\sum_{\substack{I\subseteq\{1,\ldots,d\}\\|I|=k}}\phi_\theta^{-1}\left[\phi_\theta\left(F_{\tau_{I_1}}\left(t_{2\alpha'}^{(\mathrm{K})}\right)\right)+\ldots+\phi_\theta\left(F_{\tau_{I_k}}\left(t_{2\alpha'}^{(\mathrm{K})}\right)\right)\right],$$

in which $w_1^{(\mathrm{L})}, \ldots, w_{N_{\mathrm{L}}}^{(\mathrm{L})}$ denote the weights of the GLQ. Finally, the error estimator $\epsilon_{m,\mathrm{plain}}^{(N_{\mathrm{K}},\mathrm{Arc})}$ is calculated by means of

$$\epsilon_{m,\mathrm{plain}}^{(N_{\mathrm{K}},\mathrm{Arc})} := \left|\frac{V_{m,\mathrm{GKQ,plain}}^{(N_{\mathrm{K}},\mathrm{Arc})} - V_{m,\mathrm{GLQ,plain}}^{(N_{\mathrm{L}},\mathrm{Arc})}}{V_{m,\mathrm{GKQ,plain}}^{(N_{\mathrm{K}},\mathrm{Arc})}}\right|. \tag{6.41}$$

Complexity and implementation

If we apply an Archimedean copula for modeling the joint CDF of random vector $\boldsymbol{\tau}$, we will easily be able to estimate the overall complexity of approximation $V_{m,\mathrm{GKQ,plain}}^{(N_{\mathrm{K}},\mathrm{Arc})} \approx V_m$. Let N_{K} denote the number of nodes in the underlying GKQ. Then, the overall complexity is bounded by $\mathcal{O}\left(N_{\mathrm{K}} \cdot 2^d \cdot d/2\right)$, with factor $d/2$ denoting the average cardinality of any subset I.

For the implementation of this method we first have to regard equation (6.40) in detail. It becomes obvious that term $\phi_\theta\left(F_{\tau_i}\left(t_\alpha^{\mathrm{K}}\right)\right)$ is used up to 2^{d-1} times within the calculation of $V_{m,\mathrm{GKQ,plain}}^{(N_{\mathrm{K}},\mathrm{Arc})}$. Hence, we initialize a matrix $\mathbb{Y} \in \mathbb{R}^{d\times N_{\mathrm{K}}}$ as

$$\forall i \in \{1,\ldots,d\},\ j \in \{1,\ldots,N_{\mathrm{K}}\} : \mathbb{Y} = (\mathbb{Y})_{i,j} := \phi_\theta\left(F_{\tau_i}\left(t_j^{\mathrm{K}}\right)\right)$$

before we start with the underlying algorithm. Inserted in equation (6.40) we get

$$V_{m,\mathrm{GKQ,plain}}^{(N_{\mathrm{K}},\mathrm{Arc})} = \widetilde{\Lambda}_\infty - \frac{T}{2}\sum_{\alpha=1}^{N_{\mathrm{K}}}\underbrace{w_\alpha^{(\mathrm{K})}\widetilde{\Lambda}'\left(t_\alpha^{\mathrm{K}}\right)}_{\widetilde{w}_\alpha^{(\mathrm{K})}}\cdot\sum_{k=m}^{d} c_{k,m}\sum_{\substack{I\subseteq\{1,\ldots,d\}\\|I|=k}}\phi_\theta^{-1}\left[\mathbb{Y}_{I_1,\alpha}+\ldots+\mathbb{Y}_{I_k,\alpha}\right], \tag{6.42}$$

in which $\widetilde{\boldsymbol{w}}^{(\mathrm{K})} = \left(\widetilde{w}_1^{(\mathrm{K})}, \ldots, \widetilde{w}_{N_{\mathrm{K}}}^{(\mathrm{K})}\right)^{\mathrm{tr}} \in \mathbb{R}^{N_{\mathrm{K}}}$ holds. The detailed implementation of the resulting equation (6.42) is shown in Algorithm 6.6.

Additionally to Algorithm 6.6, we need two further algorithms for creating and updating certain subsets. At this, we have to remark that an intelligent storage

Algorithm 6.6: Calculation of $V^{(N_{\mathrm{K}},\mathrm{Arc})}_{m,\mathrm{GKQ,plain}}$

```
Input  : $\mathbb{Y}$, $\widetilde{\boldsymbol{w}}^{(\mathrm{K})}$
Output: $V^{(N_{\mathrm{K}},\mathrm{Arc})}_{m,\mathrm{GKQ,plain}}$
Global : $d$, $m$, $N_{\mathrm{K}}$, $\phi_\theta$, $\phi_\theta^{-1}$ and $\theta$
Initialize $\delta \leftarrow \binom{d}{\lfloor d/2 \rfloor}$ and $V^{(N_{\mathrm{K}},\mathrm{Arc})}_{m,\mathrm{GKQ,plain}} \leftarrow \widetilde{\Lambda}_\infty$
Initialize matrix $\boldsymbol{\mathcal{T}} \leftarrow (\emptyset, \dots, \emptyset)^{\mathrm{tr}} \in \{1, \dots, d\}^\delta$ and vector $\boldsymbol{U} \leftarrow \boldsymbol{0} \in \mathbb{R}^\delta$
Initialize floating–point numbers $E_1 = E_2 \leftarrow 0$
for $i \leftarrow 1$ to $N_K$ do
    InitializeSubsets ($m$, $\boldsymbol{\mathcal{T}}$, $\boldsymbol{U}$, $\mathbb{Y}$, $i$) // cp.  Algorithm 6.7
    $E_2 \leftarrow c_{m,m} \cdot \sum_{l=1}^{\binom{d}{m}} \phi_\theta^{-1}(U_l)$
    for $k \leftarrow m + 1$ to $d$ do
        UpdateSubsets ($k$, $\boldsymbol{\mathcal{T}}$, $\boldsymbol{U}$, $\mathbb{Y}$, $i$) // cp.  Algorithm 6.8
        $E_2 \leftarrow E_2 + c_{k,m} \cdot \sum_{l=1}^{\binom{d}{k}} \phi_\theta^{-1}(U_l)$
    $E_1 \leftarrow E_1 + \widetilde{w}_i^{(\mathrm{K})} \cdot E_2$
$V^{(N_{\mathrm{K}},\mathrm{Arc})}_{m,\mathrm{GKQ,plain}} \leftarrow V^{(N_{\mathrm{K}},\mathrm{Arc})}_{m,\mathrm{GKQ,plain}} - \frac{T}{2} \cdot E_1$
```

of particular nested sums eliminates the factor $d/2$ within the overall complexity. The detailed implementations of the required functions `InitializeSubsets` and `UpdateSubsets` are shown in Algorithms 6.7 and 6.8.

Analogically to Subsection 6.1.3, we can simply implement a vectorization of Algorithm 6.6. This modification is shown in Algorithm 6.9 and results in a vector $\boldsymbol{V}^{(N_{\mathrm{K}},\mathrm{Arc})}_{\mathrm{GKQ,plain}} = \left(V^{(N_{\mathrm{K}},\mathrm{Arc})}_{1,\mathrm{GKQ,plain}}, \dots, V^{(N_{\mathrm{K}},\mathrm{Arc})}_{d,\mathrm{GKQ,plain}}\right)^{\mathrm{tr}}$. Vector $\widetilde{\boldsymbol{c}}_k \in \mathbb{Z}^d$, which is used in Algorithm 6.9, is defined according to equation (6.36).

Besides the plain implementation in Algorithms 6.6 to 6.9, we can develop a second technique of approximating V_m by using Archimedean copulae. Analogically to the usage of a Gaussian copulae on pages 71ff., this method uses the factor structure, which is offered by any Archimedean copula by definition.

In Subsection 4.2.1 we have already mentioned that any Archimedean copula is determined uniquely by the Laplace–Stieltjes transformation of its inverse generator ϕ_θ^{-1}. This Laplace–Stieltjes transformation equals the CDF F^{Arc} of a factor V^{Arc} (cp. Table 4.3). Hence, default times $\tau_1, \dots, \tau_d$ are conditionally independent with respect to this factor V^{Arc} (cp. also Algorithm 4.3).

Algorithm 6.7: `InitializeSubsets`(k, $\boldsymbol{\mathcal{T}}$, $\boldsymbol{U}$, $\mathbb{Y}$, i)

```
Input   : $k$, $\boldsymbol{\mathcal{T}}$, $\boldsymbol{U}$, $\mathbb{Y}$, $i$
Output: $\boldsymbol{\mathcal{T}}$, $\boldsymbol{U}$
Global : $d$, $N_{\mathrm{K}}$, $N_{\mathrm{H}}$, $\phi_\theta$, $\phi_\theta^{-1}$ and $\theta$

Initialize $(I_1, \dots, I_k) \leftarrow (1, \dots, k)$, $z, h, g \in \mathbb{N}$ and $z \leftarrow 2$
Initialize $\mathcal{T}_1 \leftarrow \{I_1, \dots, I_k\}$ and $U_1 \leftarrow \sum_{l=1}^{k} \mathbb{Y}_{I_l,i}$
while $z \leq \binom{d}{k}$ do // z counting the number of subsets
    $g \leftarrow k$ and $h \leftarrow z$
    while $z = h$ do
        if $I_g + 1 \leq d - (k - g)$ then
            $I_g + +$
            for $l \leftarrow g + 1$ to $k$ do
                $I_l \leftarrow I_g + (l - g)$
            $\mathcal{T}_z \leftarrow \{I_1, \dots, I_k\}$ and $U_z \leftarrow \sum_{l=1}^{k} \mathbb{Y}_{I_l,i}$
            $z + +$ and $g \leftarrow k$
        else
            $g - -$
```

Algorithm 6.8: `UpdateSubsets`(k, $\boldsymbol{\mathcal{T}}$, $\boldsymbol{U}$, $\mathbb{Y}$, i)

```
Input   : $k$, $\boldsymbol{\mathcal{T}}$, $\boldsymbol{U}$, $\mathbb{Y}$, $i$
Output: $\boldsymbol{\mathcal{T}}$, $\boldsymbol{U}$
Global : $d$, $N_{\mathrm{K}}$, $N_{\mathrm{H}}$, $\phi_\theta$, $\phi_\theta^{-1}$ and $\theta$

Initialize $\boldsymbol{\mathcal{T}}^{\mathrm{new}} \leftarrow \boldsymbol{\mathcal{T}}$, $\boldsymbol{U}^{\mathrm{new}} \leftarrow \boldsymbol{U}$, $z \leftarrow 1$ and $g \leftarrow 1$
Initialize $(I_1, \dots, I_{k-1}) \leftarrow (\mathcal{T}_{1_1}, \dots, \mathcal{T}_{1_{k-1}})$
while $z \leq \binom{d}{k}$ do // z counting the number of subsets
    for $l = \mathcal{T}_{g_{k-1}} + 1$ to $l = d$ do
        $\mathcal{T}_z^{\mathrm{new}} \leftarrow \mathcal{T}_g \cup \{l\}$
        $U_z^{\mathrm{new}} \leftarrow U_g + \mathbb{Y}_{l,i}$
        $z + +$
    $g + +$
$\boldsymbol{\mathcal{T}} \leftarrow \boldsymbol{\mathcal{T}}^{\mathrm{new}}$ and $\boldsymbol{U} \leftarrow \boldsymbol{U}^{\mathrm{new}}$
```

Algorithm 6.9: Calculation of $\boldsymbol{V}^{(N_{\text{K}},\text{Arc})}_{\text{GKQ,plain}}$

```
Input  : $\mathbb{Y}$, $\widetilde{\boldsymbol{w}}^{(\text{K})}$
Output: $\boldsymbol{V}^{(N_{\text{K}},\text{Arc})}_{\text{GKQ,plain}}$
Global : $d$, $N_{\text{K}}$, $\theta$, $\phi_\theta$, $\phi_\theta^{-1}$
Initialize $\delta \leftarrow \binom{d}{\lfloor d/2 \rfloor}$ and $\boldsymbol{V}^{(N_{\text{K}},\text{Arc})}_{\text{GKQ,plain}} \leftarrow \widetilde{\Lambda}_\infty$
Initialize matrix $\boldsymbol{\mathcal{T}} = (\emptyset, \dots, \emptyset)^{\text{tr}} \in \mathbb{N}^{\delta \times d}$ and vector $\boldsymbol{U} = \boldsymbol{0} \in \mathbb{R}^{\delta}$
Initialize floating–point vectors $\boldsymbol{E}_1 = \boldsymbol{E}_2 = \boldsymbol{0} \in \mathbb{R}^d$
for $i \leftarrow 1$ to $N_K$ do
    InitializeSubsets ($1$, $\boldsymbol{\mathcal{T}}$, $\boldsymbol{U}$, $\mathbb{Y}$, $i$) // cp.  Algorithm 6.7
    $E_{2_1} \leftarrow \sum_{l=1}^{d} \phi_\theta^{-1}(U_l)$
    for $k \leftarrow 2$ to $d$ do
        UpdateSubsets ($k$, $\boldsymbol{\mathcal{T}}$, $\boldsymbol{U}$, $\mathbb{Y}$, $i$) // cp.  Algorithm 6.8
        $\boldsymbol{E}_2 \leftarrow \boldsymbol{E}_2 + \widetilde{\boldsymbol{c}}_k \cdot \sum_{l=1}^{\binom{d}{k}} \phi_\theta^{-1}(U_l)$
    $\boldsymbol{E}_1 \leftarrow \boldsymbol{E}_1 + \widetilde{w}_i^{(\text{K})} \cdot \boldsymbol{E}_2$
$\boldsymbol{V}^{(N_{\text{K}},\text{Arc})}_{\text{GKQ,plain}} \leftarrow \boldsymbol{V}^{(N_{\text{K}},\text{Arc})}_{\text{GKQ,plain}} - \frac{T}{2} \cdot \boldsymbol{E}_1$
```

We can state for any $\boldsymbol{t} \in \overline{\mathbb{R}}_{\geq 0}$

$$
\begin{aligned}
F_{\boldsymbol{\tau}}(t_1, \dots, t_d) &= C^{\text{Arc}}_{d,\phi_\theta}\left(F_{\tau_1}(t_1), \dots, F_{\tau_d}(t_d)\right) = \phi_\theta^{-1}\left(\sum_{i=1}^{d} \phi_\theta\left(F_{\tau_i}(t_i)\right)\right) \\
&= \phi_\theta^{-1}\left(-\sum_{i=1}^{d} \ln\left(G_{\tau_i}(t_i)\right)\right) = \int\limits_0^\infty f^{\text{Arc}}(v) \prod_{i=1}^{d} \left(G_{\tau_i}(t_i)\right)^v \mathrm{d}v, \qquad (6.43)
\end{aligned}
$$

in which G_{τ_i} denotes the CDF $G_{\tau_i}(t) = \exp\left(-\phi_\theta\left(F_{\tau_i}(t_i)\right)\right)$ and f^{Arc} denotes the PDF of the corresponding CDF F^{Arc}. For a detailed view on the derivation above we refer to [LG05, Section 2.3].

With the help of the transformation above it is possible to apply the same recursion techniques as in case of independent default times (cp. Lemma 6.3, Lemma 6.4 and also Subsection 6.1.3). For this, let $x_1^{(\text{K})}, \dots, x_{N_{\text{K}}}^{(\text{K})}$ and $w_1^{(\text{K})}, \dots, w_{N_{\text{K}}}^{(\text{K})}$ denote the nodes and weight of a N_{K}–element GKQ again. Furthermore, let $x_1^{(\text{LG})}, \dots, x_{N_{\text{LG}}}^{(\text{LG})}$ and $w_1^{(\text{LG})}, \dots, w_{N_{\text{LG}}}^{(\text{LG})}$ denote the nodes and weight of a Gauss–Laguerre quadrature (abbr.: GLaQ, cp. [AS70, FH07, SK09]). Then, the value V_m of an mBDS with payoff $\widetilde{\Lambda}$ and maturity T is approximated by using a GKQ with a nested GLaQ (abbr.: GKLaQ) as

$$
\begin{aligned}
\widetilde{\Lambda}_\infty - V_m &= \int\limits_0^T \widetilde{\Lambda}'(t) \cdot F_{\tau_{\iota_m}}(t) \mathrm{d}t \\
&= \int\limits_0^T \widetilde{\Lambda}'(t) \cdot \sum_{k=m}^{d} c_{k,m} \sum_{\substack{I \subseteq \{1,\ldots,d\} \\ |I|=k}} C^{I,\mathrm{Arc}}_{k,\phi_\theta} \left(F_{\tau_{I_1}}(t), \ldots, F_{\tau_{I_k}}(t) \right) \mathrm{d}t \\
&= \int\limits_0^T \widetilde{\Lambda}'(t) \cdot \sum_{k=m}^{d} c_{k,m} \sum_{\substack{I \subseteq \{1,\ldots,d\} \\ |I|=k}} \int\limits_0^\infty f^{\mathrm{Arc}}(v) \prod_{i=1}^{k} \left(G_{\tau_{I_i}}(t) \right)^v \mathrm{d}v \mathrm{d}t \qquad (6.44) \\
&\approx \frac{T}{2} \sum_{\alpha=1}^{N_{\mathrm{K}}} \underbrace{w_\alpha^{(\mathrm{K})} \widetilde{\Lambda}' \left(t_\alpha^{(\mathrm{K})} \right)}_{=:\widetilde{w}_\alpha^{(\mathrm{K})}} \\
&\quad \cdot \sum_{k=m}^{d} c_{k,m} \sum_{\substack{I \subseteq \{1,\ldots,d\} \\ |I|=k}} \sum_{\beta=1}^{N_{\mathrm{LG}}} \underbrace{w_\beta^{(\mathrm{LG})} \exp\left(x_\beta^{(\mathrm{LG})} \right) f^{\mathrm{Arc}} \left(x_\beta^{(\mathrm{LG})} \right)}_{=:\widetilde{w}_\beta^{(\mathrm{LG})}} \prod_{i=1}^{k} \underbrace{\left(G_{\tau_{I_i}} \left(t_\alpha^{(\mathrm{K})} \right) \right)^{x_\beta^{(\mathrm{LG})}}}_{=:\mathcal{G}_{I_i}\left(t_\alpha^{(\mathrm{K})}, x_\beta^{(\mathrm{LG})} \right)} \\
&= \frac{T}{2} \sum_{\alpha=1}^{N_{\mathrm{K}}} \widetilde{w}_\alpha^{(\mathrm{K})} \sum_{\beta=1}^{N_{\mathrm{LG}}} \widetilde{w}_\beta^{(\mathrm{LG})} \cdot \sum_{k=m}^{d} c_{k,m} \sum_{\substack{I \subseteq \{1,\ldots,d\} \\ |I|=k}} \prod_{i=1}^{k} \mathcal{G}_{I_i} \left(t_\alpha^{(\mathrm{K})}, x_\beta^{(\mathrm{LG})} \right) \qquad (6.45) \\
&=: \widetilde{\Lambda}_\infty - V^{(N_{\mathrm{K}},N_{\mathrm{LG}},\mathrm{Arc})}_{m,\mathrm{GKLaQ,nested}}.
\end{aligned}
$$

Obviously, equation (6.45) has the same structure as equation (6.32). Consequently, we can directly apply recursions like equation (6.34), (6.15) or (6.17) and calculate $V^{(N_{\mathrm{K}},N_{\mathrm{LG}},\mathrm{Arc})}_{m,\mathrm{GKLaQ,nested}}$ efficiently by

$$
\begin{aligned}
&V^{(N_{\mathrm{K}},N_{\mathrm{LG}},\mathrm{Arc})}_{m,\mathrm{GKLaQ,nested}} = \\
&\qquad \widetilde{\Lambda}_\infty - \frac{T}{2} \sum_{\alpha=1}^{N_{\mathrm{K}}} \widetilde{w}_\alpha^{(\mathrm{K})} \sum_{\beta=1}^{N_{\mathrm{LG}}} \widetilde{w}_\beta^{(\mathrm{LG})} \sum_{k=m}^{d} c_{k,m} \cdot \widetilde{\eta}_{\mathcal{G}}(k, d-k+1) \left(t_\alpha^{(\mathrm{K})}, x_\beta^{(\mathrm{LG})} \right), \qquad (6.46)
\end{aligned}
$$

in which $\widetilde{\eta}_{\mathcal{G}}(i,j)(t,x)$ is defined recursively by

$$
\widetilde{\eta}_{\mathcal{G}}(k,l)(t,x) := \begin{cases} \mathcal{G}_1(t,x), & \text{if } k=l=1 \\ \widetilde{\eta}_{\mathcal{G}}(k,l-1)(t,x) + \mathcal{G}_l(t,x), & \text{if } k=1,\ l>1 \\ \widetilde{\eta}_{\mathcal{G}}(k-1,l)(t,x) \cdot \mathcal{G}_k(t,x), & \text{if } k>1,\ l=1 \\ \widetilde{\eta}_{\mathcal{G}}(k-1,l)(t,x) \cdot \mathcal{G}_{k+l-1}(t,x) + \widetilde{\eta}_{\mathcal{G}}(k,l-1)(t,x), & \text{if } k>1,\ l>1 \end{cases}
$$

for any $(t,x) \in \overline{\mathbb{R}}_{\geq 0} \times \mathbb{R}_{\geq 0}$ and for each $k \in \{1,\ldots,d\}$ and $l \in \{1,\ldots,\min\{d-k+1, d-m+1\}\}$. The detailed implementation of above technique resembles Algorithms 6.1 – 6.5 and is neglected here.

Using this modification we are able to eliminate the factor 2^d within the previous complexity estimation of $V^{(N_{\mathrm{K}},\mathrm{Arc})}_{m,\mathrm{GKQ,plain}}$. However, not in any case the calculation of $V^{(N_{\mathrm{K}},N_{\mathrm{LG}},\mathrm{Arc})}_{m,\mathrm{GKLaQ,nested}}$ is more efficient than the calculation of $V^{(N_{\mathrm{K}},\mathrm{Arc})}_{m,\mathrm{GKQ,plain}}$, because we have

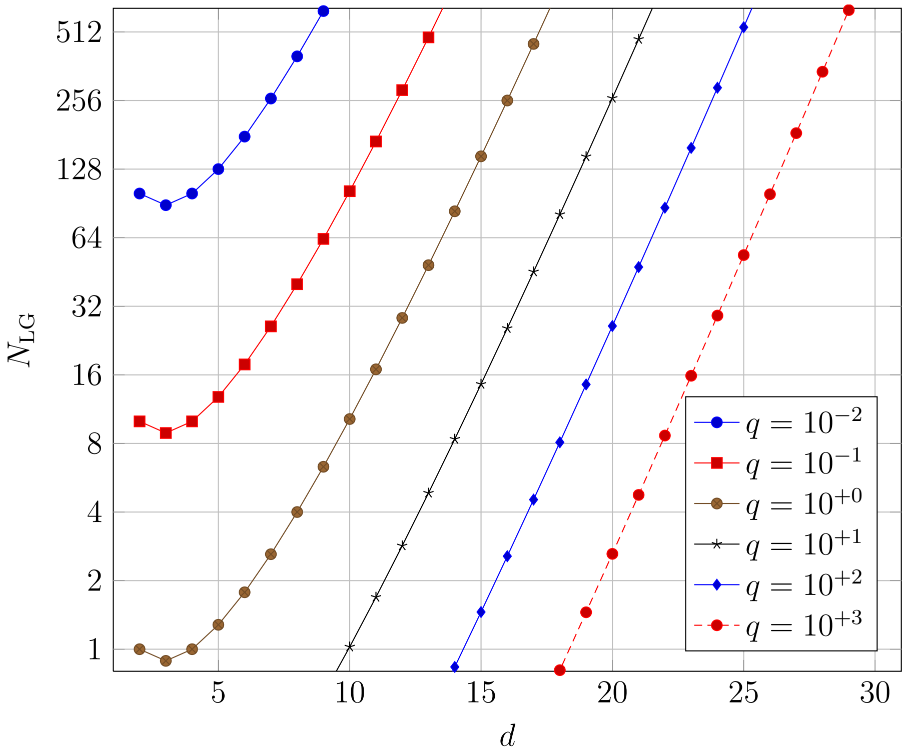


Figure 6.3.: Different levels of ratio $q = \frac{2^d}{N_{\mathrm{LG}} \cdot d^2}$, resulting from an effort comparison between $V_{m,\mathrm{GKLaQ,nested}}^{(N_{\mathrm{K}},N_{\mathrm{LG}},\mathrm{Arc})}$ and $V_{m,\mathrm{GKQ,plain}}^{(N_{\mathrm{K}},\mathrm{Arc})}$.

to consider an additional factor N_{LG}. Overall, the calculation of $V_{m,\mathrm{GKLaQ,nested}}^{(N_{\mathrm{K}},N_{\mathrm{LG}},\mathrm{Arc})}$ has a complexity of

$$\mathcal{O}\left(N_{\mathrm{K}} \cdot N_{\mathrm{LG}} \cdot d^2\right).$$

Furthermore, we have to remark that any Archimedean copula implies a different CDF F^{Arc}, i.e. the quality of $V_{m,\mathrm{GKLaQ,nested}}^{(N_{\mathrm{K}},N_{\mathrm{LG}},\mathrm{Arc})}$ as well as its detailed complexity (evaluating f^{Arc}) depend highly on the chosen Archimedean copula. In Figure 6.3 we compare complexities $\mathcal{O}\left(N_{\mathrm{K}} \cdot N_{\mathrm{LG}} \cdot d^2\right)$ and $\mathcal{O}\left(N_{\mathrm{K}} \cdot 2^d\right)$ exemplary.

6.1.5. Application in practice: SWN Synthia 2009

In order to illustrate the relevance of the methods above in practice we price the “Credit Linked Note” *SWN Synthia 2009* (ISIN: DE000LBW47W9) of the Landesbank Baden–Württemberg in this subsection (cp. [LBW09a, LBW09b, LBW09c]).

This financial instrument represents a bond, whose interest and back payments are depending on certain credit events. These credit events refer to corresponding obligors. Here, these obligors are the corporations BAS(10) (abbr.: B), DAI(11)

(10) BASF SE, Carl-Bosch-Str. 38, 67056 Ludwigshafen, Germany
(11) Daimler AG, Mercedesstraße 137, 70327 Stuttgart, Germany

(abbr.: D) and TKA[(12)] (abbr.: T). The term credit event denotes their insolvency, a delay of payment in corresponding bonds or a restructuring of these bonds (cp. [LBW09a, §5]). The corresponding bonds are corporate bonds with respect to each obligor and their prices are denoted by $B_{\mathrm{B}}, B_{\mathrm{D}}, B_{\mathrm{T}}$.

Furthermore, N denotes the nominal value of this Credit Linked Note and r_f denotes its referenced interest rate, which is fixed to $r_f = 5.55\%$ (cp. [LBW09b]). The interest rate, which is observed at market, is denoted by r_t. The date of issue t_{iss} equals 24/06/2009, maturity date T equals 20/06/2012 and the dates of interest payments $t_{z_1}, t_{z_2}, t_{z_3}$ are 20/06/2010, 20/06/2011 and 20/06/2012.

The payoff $\Lambda_{\bar{t}}^{\mathrm{Syn}}$ depends on the occurrence dates $t_{\mathrm{B}}, t_{\mathrm{D}}$ and t_{T} of the corresponding credit events. If $t_{\min} := \min\{t_{\mathrm{B}}, t_{\mathrm{D}}, t_{\mathrm{T}}\} > T$ holds, then each interest payment $r_f \cdot N$ as well as the back payment N is paid. However, if $t_{\mathrm{id}} =: t_{\min} \le T$, id $\in$ {B, D, T}, holds, then $B_{\mathrm{id}} \cdot N$ is paid immediately at $t_{\min}$ and any further payment will fail to appear. Even the accrued interest is not paid.

Let $\bar{t} \in (t_{\mathrm{iss}}, T)$ denote any date between issue date and maturity date. Furthermore, let $\tau_{\mathrm{B}}, \tau_{\mathrm{D}}, \tau_{\mathrm{T}}$ denote the random variables, which represent the occurrence dates of the corresponding credit events and let $\tau_{\min}$ denote its minimum. The corresponding CDFs are denoted by $F_{\tau_{\mathrm{B}}}, F_{\tau_{\mathrm{D}}}, F_{\tau_{\mathrm{T}}}$ and $F_{\tau_{\min}}$. The bond price, which corresponds to the obligor, who causes the first default event, is denoted by $B_{\min}$. Then, the value of the underlying contract at date $t = \bar{t} \le t_{\min}$ is given by

$$V_{\bar{t}}^{\mathrm{Syn}} := \mathbb{E}_{\mathbb{P}}\left[\Lambda_{\bar{t}}^{\mathrm{Syn}}\left(\tau_{\mathrm{B}} - \bar{t}, \tau_{\mathrm{D}} - \bar{t}, \tau_{\mathrm{T}} - \bar{t}\right)\right],$$

in which $\Lambda_{\bar{t}}^{\mathrm{Syn}}$ and $\widetilde{\Lambda}_{\bar{t}}^{\mathrm{Syn}}$ are defined as

$$\begin{aligned}
&\Lambda_{\bar{t}}^{\mathrm{Syn}}\left(\Delta t_{\mathrm{B}}, \Delta t_{\mathrm{D}}, \Delta t_{\mathrm{T}}\right) := \widetilde{\Lambda}_{\bar{t}}^{\mathrm{Syn}}\left(\Delta t_{\min}\right) \\
&:= \begin{cases}
\exp\left(-r_{\Delta t_{\min}} \Delta t_{\min}\right) B_{\min} \cdot N, & \text{if } t_{\mathrm{iss}} < t_{\min} \le t_{z_1} \\
\sum\limits_{i=1}^{j} \exp\left(-r_{\Delta t_{z_i}} \Delta t_{z_i}\right) r_f N \cdot \mathbf{1}_{\{t_{z_i} > \bar{t}\}} & \\
\qquad\qquad + \exp\left(-r_{\Delta t_{\min}} \Delta t_{\min}\right) B_{\min} \cdot N, & \text{if } t_{z_j} < t_{\min} \le t_{z_{j+1}},\ j \in \{1, 2\} \\
\sum\limits_{i=1}^{3} \exp\left(-r_{\Delta t_{z_i}} \Delta t_{z_i}\right) r_f N \cdot \mathbf{1}_{\{t_{z_i} > \bar{t}\}} & \\
\qquad\qquad + \exp\left(-r_{\Delta T} \Delta T\right) N, & \text{else}
\end{cases}
\end{aligned}$$

for any $t_{\min} \in [\bar{t}, T]$. Additionally, notation $\Delta t := t - \bar{t}$ holds for $t \ge \bar{t}$ and $\mathbb{P}$ denotes the product probability measure of random vector $\left(\tau_{\mathrm{B}}, \tau_{\mathrm{D}}, \tau_{\mathrm{T}}\right)^{\mathrm{tr}}$.

In the following we apply an evaluation using the distribution of the first default event. The only problem remaining is the dependence of $\widetilde{\Lambda}_{\bar{t}}^{\mathrm{Syn}}$ on $B_{\min}$. Usually, $B_{\min}$

[(12)] ThyssenKrupp AG, ThyssenKrupp Allee 1, 45143 Essen, Germany depends on the question, which obligor causes the first default event. Therefore, B_{min} has a stochastic character. To solve this problem we assume $B_{\text{min}} = 0.4$ to be constant, which is often assumed in practice. For a more detailed view we refer to Remark 6.7.

Let us assume $\bar{t} = 20/08/2010$ in the following. On this date *SWN Synthia 2009* denoted at a price of 101.41 EUR (assuming a nominal value of $N = 100$ EUR) on the Stuttgart stock exchange.

In order to ensure a practical evaluation we first have to model the hazard rates $h_{\text{id}}(t)$, id $\in \{\text{B}, \text{D}, \text{T}\}$ and $t \in \mathbb{R}_{\geq 0}$, of the corresponding obligors. This is done by calculating discrete hazard rates $H^{\text{id}}_{\Delta t_i}$, which depend on the corresponding CDS premiums $s^{\text{id}}_{\Delta t_i}$, via a bootstrap–technique, $i \in \{0, \ldots, n\}$ (cp. [MRW06, 4.2.2]). At this, $\{\Delta t_0 = 0, \Delta t_1, \ldots, \Delta t_n\}$ denotes any set of dates in years. Typically, $\{0, 1, 2, 3, 4, 5, 7, 10\}$ is chosen, because CDS contracts with such maturities are traded liquidly. By means of these discrete hazard rates the proper hazard rate $h_{\text{id}}(t)$ is created by $h_{\text{id}}(t) = H^{\text{id}}_{\Delta t_i}$, in which $\Delta t_{i-1} \leq t < \Delta t_i$ holds for each $i \in \{1, \ldots, n\}$. The CDS premiums with respect to date set $\{1, 2, 3, 4, 5, 7, 10\}$ and the corresponding discrete hazard rates are given in Table 6.3 (source: Bloomberg Terminal access). The resulting CDFs $F_{\tau_{\text{id}}}(t)$ are shown in Figure 6.4.

id ↓	s^{id}_1	s^{id}_2	s^{id}_3	s^{id}_4	s^{id}_5	s^{id}_7	s^{id}_{10}
B	47.6025	57.0635	62.386	68.892	74.9885	78.2355	81.496
D	54.5525	72.5275	83.5015	94.385	102.1635	109.2245	115.349
T	237.1435	318.917	348.7925	372.7025	390.389	392.3285	397.413

	$H^{\text{id}}_1 \cdot 10^2$	$H^{\text{id}}_2 \cdot 10^2$	$H^{\text{id}}_3 \cdot 10^2$	$H^{\text{id}}_4 \cdot 10^2$	$H^{\text{id}}_5 \cdot 10^2$	$H^{\text{id}}_7 \cdot 10^2$	$H^{\text{id}}_{10} \cdot 10^2$
B	0.7934	1.1131	1.2251	1.4948	1.6952	1.4636	1.5237
D	0.9092	1.5178	1.7767	2.1596	2.2793	2.1767	2.2457
T	3.9524	6.7942	6.9438	7.6415	8.0069	6.6576	6.9765

Table 6.3.: CDS premiums of different maturities (in basis points, bp) and corresponding discrete hazard rates.

In addition to the hazard rates, we still have to model the correlation and the interest rate r_t. The correlation matrix Σ was calibrated historically by considering data from the last three years. It is given by

$$\Sigma = \begin{pmatrix} 1.0 & 0.675 & 0.702 \\ 0.675 & 1.0 & 0.669 \\ 0.702 & 0.669 & 1.0 \end{pmatrix} = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \end{pmatrix} \cdot (\psi_1, \psi_2, \psi_3) + \begin{pmatrix} 1 - \psi_1^2 & 0 & 0 \\ 0 & 1 - \psi_1^2 & 0 \\ 0 & 0 & 1 - \psi_1^2 \end{pmatrix},$$

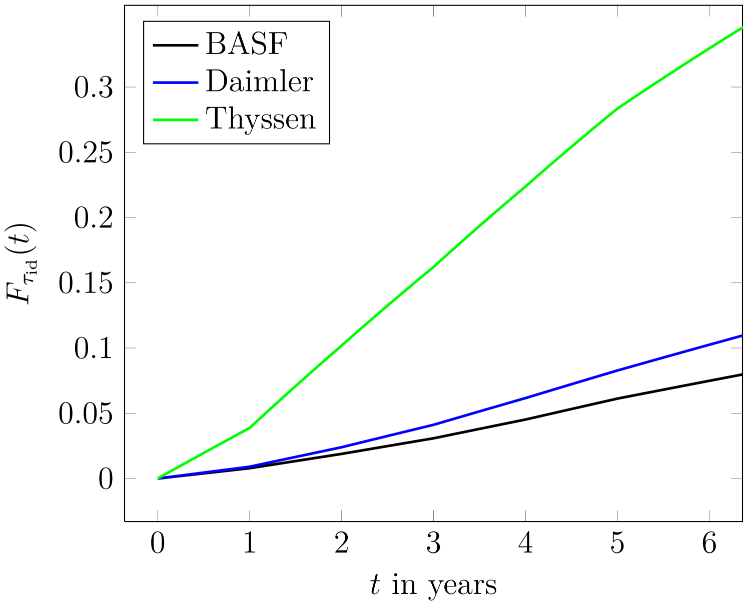


Figure 6.4.: CDFs $F_{\tau_{\text{id}}}(t)$ for id $\in$ {B, D, T}.

in which $\psi_1 = \frac{0.675}{\psi_2} \approx 0.8416032$, $\psi_2 = \frac{0.669}{\psi_3} \approx 0.8020407$ and $\psi_3 = \sqrt{\frac{0.702 \cdot 0.669}{0.675}} \approx 0.8341223$ hold. Obviously, applying factor correlation is not a restriction in case of $d = 3$. The risk free interest rate at date $\bar{t}$ is shown in Table 6.4 and is based on the corresponding LIBOR® (London Interbank Offered Rate).

t	1 W	2 W	1 M	2 M	3 M	4 M	5 M	6 M
r_t	0.59	0.62	0.70	0.79	0.96	1.03	1.11	1.21
t	7 M	8 M	9 M	10 M	11 M	1 Y	2 Y	3 Y
r_t	1.26	1.31	1.36	1.40	1.44	1.49	1.55	1.64

Table 6.4.: Risk free interest rate r_t in % for t in Weeks, Months and Years.

Considering all these information we can calculate the value $V_{\bar{t}}^{\text{Syn}}$ as

$$\begin{aligned} V_{\bar{t}}^{\text{Syn}} &= \int_0^\infty \widetilde{\Lambda}_{\bar{t}}^{\text{Syn}}(t) f_{\tau_{\iota_1}}(t)\mathrm{d}t \\ &= \int\limits_0^{\Delta t_{z_2}} \widetilde{\Lambda}_{\bar{t}}^{\text{Syn}}(t) f_{\tau_{\iota_1}}(t)\mathrm{d}t + \int\limits_{\Delta t_{z_2}}^{\Delta T} \widetilde{\Lambda}_{\bar{t}}^{\text{Syn}}(t) f_{\tau_{\iota_1}}\mathrm{d}t + \int\limits_{\Delta T}^{\infty} \widetilde{\Lambda}_{\bar{t}}^{\text{Syn}}(t) f_{\tau_{\iota_1}}\mathrm{d}t \end{aligned}$$

$$
\begin{aligned}
&= \lim_{s\nearrow \Delta t_{z_2}} \left(\widetilde{\Lambda}^{\mathrm{Syn}}_{\bar{t}}(s)F_{\tau_{\iota_1}}(s)\right) - \int\limits_0^{\Delta t_{z_2}} \widetilde{\Lambda}^{\mathrm{Syn}'}_{\bar{t}}(t)F_{\tau_{\iota_1}}(t)\mathrm{d}t \\
&\quad + \lim_{s\nearrow \Delta T} \left(\widetilde{\Lambda}^{\mathrm{Syn}}_{\bar{t}}(s)F_{\tau_{\iota_1}}(s)\right) - \lim_{q\searrow \Delta t_{z_2}} \left(\widetilde{\Lambda}^{\mathrm{Syn}}_{\bar{t}}(q)F_{\tau_{\iota_1}}(q)\right) - \int\limits_{\Delta t_{z_2}}^{\Delta T} \widetilde{\Lambda}^{\mathrm{Syn}'}_{\bar{t}}(t)F_{\tau_{\iota_1}}(t)\mathrm{d}t \\
&\quad + \lim_{s\nearrow \infty} \left(\widetilde{\Lambda}^{\mathrm{Syn}}_{\bar{t}}(s)F_{\tau_{\iota_1}}(s)\right) - \lim_{q\searrow \Delta T} \left(\widetilde{\Lambda}^{\mathrm{Syn}}_{\bar{t}}(q)F_{\tau_{\iota_1}}(q)\right) - \underbrace{\int\limits_{\Delta T}^{\infty} \widetilde{\Lambda}^{\mathrm{Syn}'}_{\bar{t}}(t)F_{\tau_{\iota_1}}(t)\mathrm{d}t}_{=0} \\
&= \left(\lim_{s\nearrow \Delta t_{z_2}} \widetilde{\Lambda}^{\mathrm{Syn}}_{\bar{t}}(s) - \lim_{q\searrow \Delta t_{z_2}} \widetilde{\Lambda}^{\mathrm{Syn}}_{\bar{t}}(q)\right) F_{\tau_{\iota_1}}(\Delta t_{z_2}) - \int\limits_0^{\Delta t_{z_2}} \widetilde{\Lambda}^{\mathrm{Syn}'}_{\bar{t}}(t)F_{\tau_{\iota_1}}(t)\mathrm{d}t \\
&\quad + \left(\lim_{s\nearrow \Delta T} \widetilde{\Lambda}^{\mathrm{Syn}}_{\bar{t}}(s) - \lim_{q\searrow \Delta T} \widetilde{\Lambda}^{\mathrm{Syn}}_{\bar{t}}(q)\right) F_{\tau_{\iota_1}}(\Delta T) - \int\limits_{\Delta t_{z_2}}^{\Delta T} \widetilde{\Lambda}^{\mathrm{Syn}'}_{\bar{t}}(t)F_{\tau_{\iota_1}}(t)\mathrm{d}t \\
&\quad + \underbrace{\lim_{s\nearrow \infty} \widetilde{\Lambda}^{\mathrm{Syn}}_{\bar{t}}(s)}_{=:\widetilde{\Lambda}^{\mathrm{Syn}}_{\infty}=\widetilde{\Lambda}^{\mathrm{Syn}}_{\bar{t}}(t),\ t>T} \quad . \qquad (6.47)
\end{aligned}
$$

For approximating this value we use the Gaussian copula as well as the Clayton Copula in the following. At this, we differentiate between four approximations: Let $V^{\mathrm{Syn},(N_{\mathrm{K}},N_{\mathrm{H}})}_{\bar{t},\mathrm{GKHQ,plain}}$ and $V^{\mathrm{Syn},(N_{\mathrm{K}},\mathrm{Cla})}_{\bar{t},\mathrm{GKQ,plain}}$ denote the approaches, which use Lemma 6.1 in its plain form and let $V^{\mathrm{Syn},(N_{\mathrm{K}},N_{\mathrm{H}})}_{\bar{t},\mathrm{GKHQ,nested}}$ and $V^{\mathrm{Syn},(N_{\mathrm{K}},N_{\mathrm{La}},\mathrm{Cla})}_{\bar{t},\mathrm{GKLaQ,nested}}$ denote the methods, which use the simplifications introduced by Lemmata 6.3 and 6.4. For calibrating the Clayton copula we choose a correlation, which is induced by $\theta = 1.83$.

We expect that the nested method outperforms the plain method regarding the Gaussian copula, because we proved its dominance in any case. We further expect that the plain method dominates with respect to the Clayton copula, because the dimension $d = 3$ is moderate.

In Tables 6.5 and 6.6 the performance of the approximations $V^{\mathrm{Syn},(N_{\mathrm{K}},N_{\mathrm{H}})}_{\bar{t},\mathrm{GKHQ,nested}}$ and $V^{\mathrm{Syn},(N_{\mathrm{K}},\mathrm{Cla})}_{\bar{t},\mathrm{GKQ,plain}}$ is shown. At this, we compare the error estimators $\epsilon^{(N_{\mathrm{K}},N_{\mathrm{H}})}_{m,\mathrm{nested}}$ and $\epsilon^{(N_{\mathrm{K}},\mathrm{Cla})}_{m,\mathrm{plain}}$ of the underlying GKQ, the “exact” relative errors ϵ, which are calculated by using “exact” approximations ($N_{\mathrm{K}} = 513$, $N_{\mathrm{H}} = 256$), and the corresponding CPU times in seconds. The exact approximations are given by $V^{\mathrm{Syn},(513,256)}_{\bar{t},\mathrm{GKHQ,nested}} = 101.53416$ EUR and $V^{\mathrm{Syn},(513,\mathrm{Cla})}_{\bar{t},\mathrm{GKQ,plain}} = 102.235$ EUR. Of course, these values are not approximately equal to the market price of 101.41 EUR, but considering the plenty of parameters, which are to be fitted, they represent very good approximations.

For choosing parameter N_{H} we refer to Table 6.2, which suggests the choice $N_{\mathrm{H}} = 64$ ($k = 3$, $\max_i |\psi_i| \approx 0.84$). In deed we can observe that neither the error estimator $\hat{\epsilon}$ nor the exact error ϵ changes significantly for $N_{\mathrm{H}} > 64$ (changes $< 10^{-16}$). Addi-

$N_{\mathrm{H}} \downarrow$	$N_{\mathrm{K}} \rightarrow$	33	65	129	257
32	$\hat{\epsilon}$	1.49273e-08	4.12052e-08	9.41537e-09	1.10695e-09
	ϵ	3.86214e-08	9.35463e-09	1.56019e-09	2.77648e-11
	sec	6.0e-03	1.2e-02	2.3e-02	4.6e-02
64	$\hat{\epsilon}$	1.49273e-08	4.12052e-08	9.41537e-09	1.10695e-09
	ϵ	3.86207e-08	9.35392e-09	1.55948e-09	2.70549e-11
	sec	1.1e-02	2.3e-02	4.5e-02	8.8e-02
128	$\hat{\epsilon}$	1.49273e-08	4.12052e-08	9.41536e-09	1.10695e-09
	ϵ	3.86207e-08	9.35392e-09	1.55948e-09	2.70549e-11
	sec	2.2e-02	4.3e-02	8.4e-02	1.7e-01

Table 6.5.: Error estimator $\hat{\epsilon} := \epsilon_{m,\mathrm{nested}}^{(N_{\mathrm{K}},N_{\mathrm{H}})}$, “exact” error ϵ and CPU time (in sec.) of the approximation $V_{\bar{t},\mathrm{GKHQ,nested}}^{\mathrm{Syn},(N_{\mathrm{K}},N_{\mathrm{H}})}$.

$N_{\mathrm{K}} \rightarrow$	33	65	129	257
$\hat{\epsilon}$	3.72e-08	3.92e-08	8.65e-09	8.77e-10
ϵ	3.86e-08	8.66e-09	1.19e-09	6.14e-11
sec	1.10e-04	2.00e-04	3.80e-04	7.40e-04

Table 6.6.: Error estimator $\hat{\epsilon} := \epsilon_{m,\mathrm{plain}}^{(N_{\mathrm{K}},\mathrm{Cla})}$, “exact” error ϵ and CPU time (in sec.) of the approximation $V_{\bar{t},\mathrm{GKQ,plain}}^{\mathrm{Syn},(N_{\mathrm{K}},\mathrm{Cla})}$.

tionally, we can remark that $\hat{\epsilon}$ is a good and conservative estimation of the exact error in any case (cp. Tables 6.5 and 6.6).

Finally, in Figure 6.5 a comparison of all possible evaluation methods is shown. As we expected, approximation $V_{\bar{t},\mathrm{GKHQ,nested}}^{\mathrm{Syn},(N_{\mathrm{K}},N_{\mathrm{H}})}$ (blue line) dominates $V_{\bar{t},\mathrm{GKHQ,plain}}^{\mathrm{Syn},(N_{\mathrm{K}},N_{\mathrm{H}})}$ (red line) lightly and approximation $V_{\bar{t},\mathrm{GKQ,plain}}^{\mathrm{Syn},(N_{\mathrm{K}},\mathrm{Cla})}$ (black line) dominates $V_{\bar{t},\mathrm{GKLaQ,nested}}^{\mathrm{Syn},(N_{\mathrm{K}},N_{\mathrm{La}},\mathrm{Cla})}$ (green line) clearly, because of the moderate dimension $d = 3$.

Remark 6.7 (Recovery rate). In the beginning of this subsection we assumed $B_{\min} = 0.4$, which represents a weak point within this evaluation. Considering equation (6.47) in detail we can see that $V_{\bar{t}}^{\mathrm{Syn}}$ depends linear on $B_{\min}$, though. Consequently, we can simply calculate approximations with respect to any $B_{\min} \in [0, 1]$. For illustration we show the dependence of $V_{\bar{t},\mathrm{GKHQ,nested}}^{\mathrm{Syn},(513,256)}$ on $B_{\min} \in [0.2, 0.5]$ in Figure 6.6.

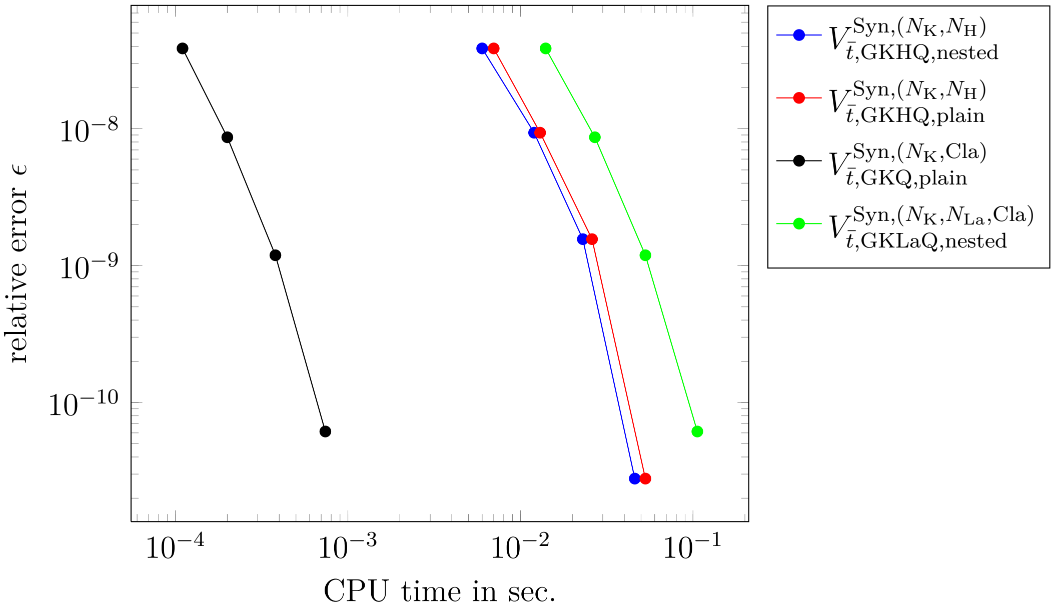


Figure 6.5.: Comparison of different approximation methods for $V_{\bar{t}}^{\mathrm{Syn}}$ with respect to their relative errors ϵ and their CPU times (in sec.).

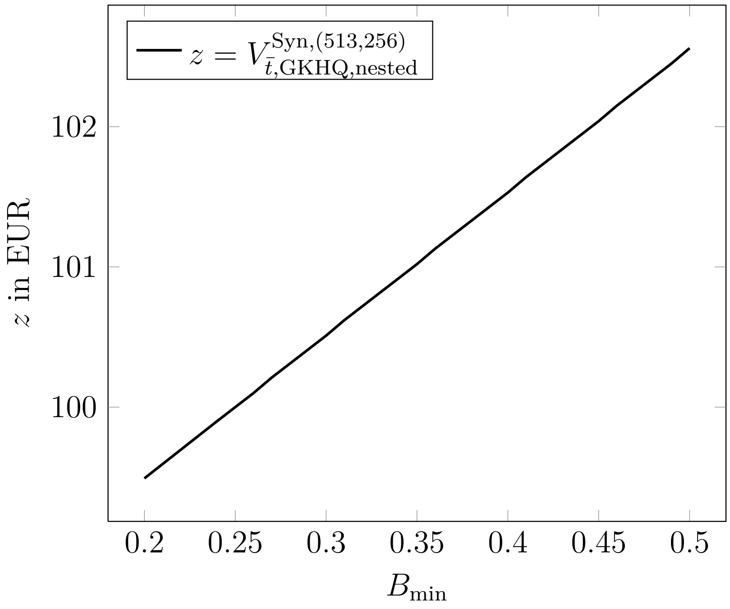


Figure 6.6.: Linear dependence of $V_{\bar{t},\mathrm{GKHQ,nested}}^{\mathrm{Syn},(513,256)}$ and $B_{\min}$.

6.2. Monte-Carlo methods

Alternatively to the evaluation with respect to the m–th smallest default time, we discuss the evaluation by means of Monte–Carlo methods in this section. They will be used if dimension d is very large or the underlying portfolio is inhomogeneous, i.e. we cannot assume homogeneous recovery rates and nominal values.

The general approach for evaluating an mBDS by means of Monte–Carlo methods is rather easy. Let Λ_m denote the payoff of an mBDS and let $\tau_1, \ldots, \tau_d$ denote the random variables, which represent the default times of obligors $S_1, \ldots, S_d$. In addition, their CDFs are given by $F_{\tau_1}(t_1), \ldots, F_{\tau_d}(t_d)$ (cp. equation (6.1)) and their joint CDF is given by $F_{\boldsymbol{\tau}}(t_1 \ldots, t_d) = \check{C}_d(F_{\tau_1}(t_1), \ldots, F_{\tau_d}(t_d))$ for any $\boldsymbol{t} \in \overline{\mathbb{R}}^d_{\geq 0}$ (cp. equation (6.2)). Then, the value V_m of the underlying contract is given by (cp. equation (6.3))

$$V_m := \int_0^\infty \cdots \int_0^\infty \Lambda_m(\boldsymbol{t})\, \mathrm{d}F_{\boldsymbol{\tau}}(\boldsymbol{t}) = \int_0^\infty \cdots \int_0^\infty \Lambda_m(\boldsymbol{t})\, f_{\boldsymbol{\tau}}(\boldsymbol{t})\, \mathrm{d}\boldsymbol{t}, \tag{6.48}$$

in which $f_{\boldsymbol{\tau}}(\boldsymbol{t})$ denotes the joint PDF of $F_{\boldsymbol{\tau}}(\boldsymbol{t})$.

Consequently, a plain Monte–Carlo integration performs as follows:

1. Draw M random tuples $\boldsymbol{v}^{(j)} = \left(v_1^{(j)}, \ldots, v_d^{(j)}\right)^{\mathrm{tr}} \sim \check{C}_d,\ j \in \{1, \ldots, M\}$, which are distributed according to copula $\check{C}_d$ (cp. Algorithms 4.1 – 4.3).
2. Calculate $\hat{\boldsymbol{t}}^{(j)} = \left(\hat{t}_1^{(j)}, \ldots, \hat{t}_d^{(j)}\right)^{\mathrm{tr}},\ j \in \{1, \ldots, M\}$ via $\hat{t}_i^{(j)} = F_{\tau_i}^{-1}\left(v_i^{(j)}\right),\ i \in \{1, \ldots, d\}$.
3. Approximate $V_m \approx V_{m,\mathrm{MC}}^{M,\mathrm{plain}} := \frac{1}{M} \cdot \sum\limits_{j=1}^{M} \Lambda_m\left(\hat{t}_1^{(j)}, \ldots, \hat{t}_d^{(j)}\right)$.

Analyzing this approach in detail we have to mention a big disadvantage. The smaller we choose maturity T and hazard rates h_i the more improbable is the occurrence of at least m defaults within a time horizon $[0, T]$. Hence, most of the random tuples $\hat{\boldsymbol{t}}^{(j)}$ meet the condition $\hat{t}_{\iota_m}^{(j)} > T$ and result in the constant mBDS payoff $\Lambda_{m,\infty} := \Lambda_m(\boldsymbol{t}),\ \boldsymbol{t} \in (T, \infty]^d$ (cp. Definition 6.1). Hence, we get an approximation $V_{m,\mathrm{MC}}^{M,\mathrm{plain}}$, which shows a huge variance $\mathrm{Var}\left(V_{m,\mathrm{MC}}^{M,\mathrm{plain}}\right)$ (cp. Example 6.6).

Example 6.6 (Need of variance reduction)**.** In order to illustrate the effect above we show the payments $\Lambda_{\bar{t}}^{\mathrm{Syn}}\left(t_{\mathrm{B}}^{(j)}, t_{\mathrm{D}}^{(j)}, t_{\mathrm{T}}^{(j)}\right)$ for 200 random tuples $\left(t_{\mathrm{B}}^{(j)}, t_{\mathrm{D}}^{(j)}, t_{\mathrm{T}}^{(j)}\right)^{\mathrm{tr}}$, $j \in \{1, \ldots, 200\}$, in Figure 6.7 (cp. Subsection 6.1.5 for the underlying notation). Random tuples $\left(t_{\mathrm{B}}^{(j)}, t_{\mathrm{D}}^{(j)}, t_{\mathrm{T}}^{(j)}\right)^{\mathrm{tr}}$ were drawn with respect to the Gaussian copula (cp. Algorithm 4.1 or 4.2).

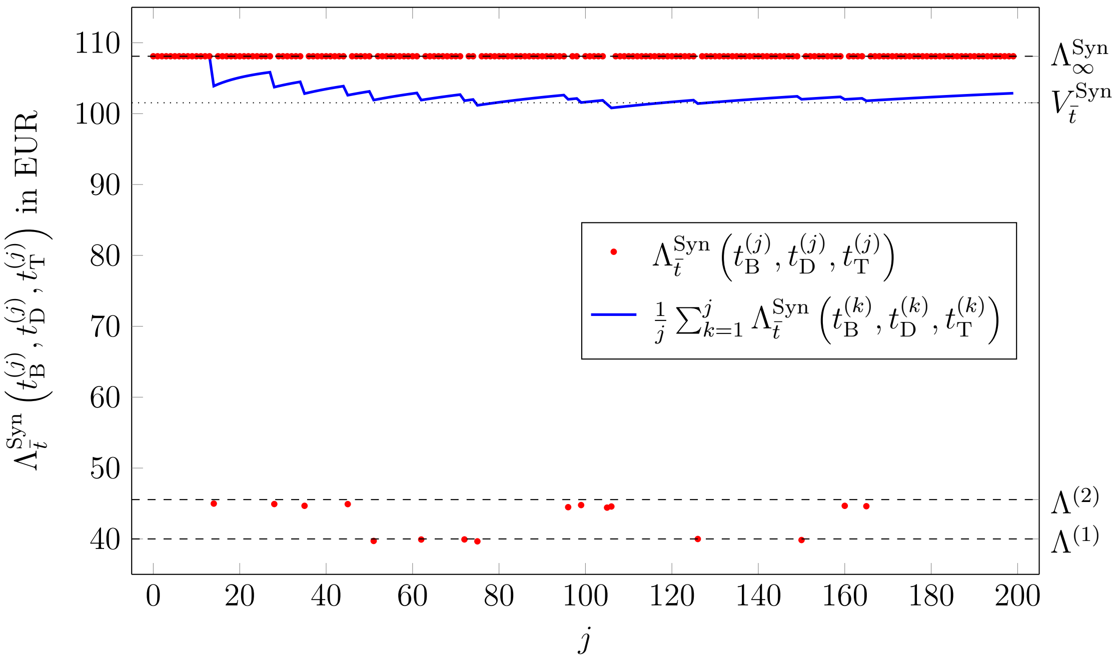


Figure 6.7.: Payments $\Lambda_{\bar{t}}^{\mathrm{Syn}}\left(t_{\mathrm{B}}^{(j)}, t_{\mathrm{D}}^{(j)}, t_{\mathrm{T}}^{(j)}\right)$ of 200 random tuples $\left(t_{\mathrm{B}}^{(j)}, t_{\mathrm{D}}^{(j)}, t_{\mathrm{T}}^{(j)}\right)$, which were drawn with respect to the Gaussian copula.

Considering Figure 6.7 in detail it becomes obvious that each resulting payment $\Lambda_{\bar{t}}^{\mathrm{Syn}}\left(t_{\mathrm{B}}^{(j)}, t_{\mathrm{D}}^{(j)}, t_{\mathrm{T}}^{(j)}\right)$ is located close to or exactly on one of the dashed lines, which increases the resulting variance significantly (detailed figures concerning the variance are given later on). At this, notations $\Lambda_{\infty}^{\mathrm{Syn}} := \Lambda_{\bar{t}}^{\mathrm{Syn}}(T+1, T+1, T+1)$, $\Lambda^{(2)} := \exp\left(-r_{5/6}5/6\right)(B_{\min} + r_f) \cdot N$ and $\Lambda^{(1)} := B_{\min} \cdot N$ hold and $V_{\bar{t}}^{\mathrm{Syn}}$ denotes the exact value of the underlying contract (Credit linked note *SWN Synthia 2009*, cp. Subsection 6.1.5).

In general it is desirable that as many tuples as possible result in a payment $\Lambda_m(\boldsymbol{t}) \neq \Lambda_{m,\infty}$. Hence, we have to sample tuples $\boldsymbol{t}$, which meet the condition $t_{\iota_m} \leq T$. The corresponding region for $\boldsymbol{t}$ is called the *importance region*. However, we have to ensure that the expectation of the underlying Monte–Carlo simulation remains constant. Such methods reduce the variance of the underlying simulation significantly and are known as *importance sampling* methods. In the next subsections we explain these methods with respect to Gaussian and Archimedean copulae.

6.2.1. Importance sampling with respect to a Gaussian copula

Variance reduction by means of importance sampling regarding the Gaussian copula has already been discussed in [JK04, CG08]. More precisely, the algorithm, which is presented in the following, was developed in [JK04] and improved in [CG08].

Let $F_{\boldsymbol{\tau}}(t_1,\dots,t_d) = C^{\mathrm{Gau}}_{d,\Sigma}(F_{\tau_1}(t_1),\dots,F_{\tau_d}(t_d))$ denote the joint CDF of random vector $\boldsymbol{\tau}$ for any $\boldsymbol{t} \in \overline{\mathbb{R}}^d_{\geq 0}$ and let V_m denote the value of any mBDS with payoff Λ_m and maturity T. Then, V_m is approximated by

$$\begin{aligned} V_m &= \int\limits_0^\infty \cdots \int\limits_0^\infty \Lambda_m(\boldsymbol{t})\,\mathrm{d}F_{\boldsymbol{\tau}}(\boldsymbol{t}) \\ &\approx V^{M,\mathrm{IS}}_{m,\mathrm{MC,Gau}} := \frac{1}{M} \cdot \sum_{j=1}^M \widehat{\Lambda}_m\left(\hat{t}^{(j)}_1,\dots,\hat{t}^{(j)}_d\right), \end{aligned} \tag{6.49}$$

in which tuples $\left(\hat{t}^{(j)}_1,\dots,\hat{t}^{(j)}_d\right)^{\mathrm{tr}}$ are drawn by using importance sampling and $\widehat{\Lambda}_m$ still has to be specified. The aim of this subsection is the development of an algorithm, which draws tuples $\hat{\boldsymbol{t}}$ meeting $\hat{t}_{\iota_m} \leq T$ and the development of a corresponding payoff $\widehat{\Lambda}_m$, which ensures an unbiased approximation.

For this, we transform Algorithm 4.1, so that its output $\boldsymbol{v}$ meets the condition $\hat{t}_{\iota_m} \leq T$ for $\hat{t}_i = F^{-1}_{\tau_i}(v_i)$, $i \in \{1,\dots,d\}$. For a better readability let us briefly summarize Algorithm 4.1 again. Let $u_1,\dots,u_d$ be independently and uniformly distributed on $[0,1]^d$. Then, the following program sequence produces a tuple $\boldsymbol{v} = (v_1,\dots,v_d)^{\mathrm{tr}}$, which is distributed according to copula $C^{\mathrm{Gau}}_{d,\Sigma}$.

1. $z_i \leftarrow \Phi^{-1}(u_i)$, i.e. $z_i \sim \Phi,\ i \in \{1,\dots,d\}$
2. $A \leftarrow \texttt{Cholesky}(d,\Sigma)$, i.e. $AA^{\mathrm{tr}} = \Sigma$
3. $\boldsymbol{w} \leftarrow A\boldsymbol{z}$, i.e. $\boldsymbol{w} \sim \Phi^d_\Sigma$
4. $v_i \leftarrow \Phi(w_i),\ i \in \{1,\dots,d\}$

Let $\forall \boldsymbol{x} \in \overline{\mathbb{R}}^d : f_{\boldsymbol{u}}(\boldsymbol{x}) = \prod_{i=1}^d f_i(x_i) = \prod_{i=1}^d \mathbf{1}_{\{x_i \in [0,1]\}}$ denote the joint PDF of $\boldsymbol{u}$ and let $\Xi(\boldsymbol{u}) = \hat{\boldsymbol{t}}$ denote the mapping, which includes the enumeration above (input: $\boldsymbol{u}$ uniformly and independently distributed on $[0,1]^d$; output: $\hat{\boldsymbol{t}} \sim F_{\boldsymbol{\tau}}$). In the following we develop an importance sampling approach of the form

$$\begin{aligned} V_m &= \int\limits_{[0,1]^d} \Lambda_m(\Xi(\boldsymbol{x})) \cdot f_{\boldsymbol{u}}(\boldsymbol{x})\mathrm{d}x_1\cdots\mathrm{d}x_d \\ &= \int\limits_{[0,1]^d} \Lambda_m(\Xi(\boldsymbol{x})) \cdot \prod_{i=1}^d f_i(x_i)\,\mathrm{d}x_1\cdots\mathrm{d}x_d \\ &= \int\limits_{[0,1]^d} \underbrace{\Lambda_m(\Xi(\boldsymbol{x}))\,\frac{\prod\limits_{i=1}^d f_i(x_i)}{\prod\limits_{i=1}^d f^{\mathrm{new}}_i(x_i)}}_{:=\widehat{\Lambda}_m(\hat{t})} \cdot \prod_{i=1}^d f^{\mathrm{new}}_i(x_i)\,\mathrm{d}x_1\cdots\mathrm{d}x_d. \end{aligned}$$

In order to create suitable density functions f^{new}_i, which cause at least m defaults, we proceed as follows. Let $u_1, \ldots, u_{i-1}$ already be drawn or rather let $\hat{t}_1, \ldots, \hat{t}_{i-1}$ be known. Then, we can verify the following relation (cp. [CG08, Lemma 1])

$$p_i := \mathbb{P}\left(\tau_i \le T \,|\tau_1, \ldots, \tau_{i-1}\right) = \Phi\left(\frac{\Phi^{-1}\left(F_{\tau_i}(T)\right) - \sum\limits_{j=1}^{i-1} A_{i,j} z_j}{A_{i,i}}\right),$$

which is easily calculated by considering Algorithm 4.1 or the enumeration above in detail. Furthermore, we assume that within the first $i-1$ default times $\hat{t}_1, \ldots, \hat{t}_{i-1}$ the condition $\hat{t}_j \le T,\ j \in \{1, \ldots, i-1\}$ was met $k < m$ times. Then, we artificially transform the default probability p_i according to

$$\widetilde{p}_i^{(k)} := \begin{cases} \max\left(\frac{m-k}{d-i+1}, p_i\right), & \text{if } k < m \\ p_i, & \text{else} \end{cases},$$

which obviously ensures the occurrence of at least m defaults. In order to meet this default probability we have to transform the input random number u_i. If u_i is smaller than default probability $\widetilde{p}_i^{(k)}$, we will artificially shift u_i into the "default area" by calculating a new input random number u_i^{new}. This new random number u_i^{new} meets

$$u_i^{\text{new}} := \begin{cases} \frac{p_i u_i}{\widetilde{p}_i^{(k)}}, & \text{if } u_i \le \widetilde{p}_i^{(k)} \\ p_i + \frac{(1-p_i)\left(u_i - \widetilde{p}_i^{(k)}\right)}{\left(1-\widetilde{p}_i^{(k)}\right)}, & \text{else} \end{cases}. \tag{6.50}$$

By using this approach we can guarantee that each generated tuple $\hat{\boldsymbol{t}}$ meets the condition $\hat{t}_{\iota_m} \le T$ (cp. [CG08, Lemma 2]). However, in order to ensure an unbiased approximation, we have to multiply each evaluated payment by a factor G (cp. Algorithm 6.10 or [JK04, CG08]). This weight factor G results from the transformation of u_i to u_i^{new}. Hereby, we transform the density function $f_i\left(x_i\right) = \mathbf{1}_{\{x \in [0,1]\}}$ to an artificial density function

$$\forall x \in [0,1] : f_i^{\text{new}}\left(x_i\right) := \begin{cases} \begin{cases} \frac{\widetilde{p}_i^{(k)}}{p_i}, & \text{if } x_i < p_i \\ \frac{1-\widetilde{p}_i^{(k)}}{1-p_i}, & \text{else} \end{cases}, & \text{if } \widetilde{p}_i^{(k)} = \frac{m-k}{d-i+1} \\ 1, & \text{else} \end{cases}$$

of u_i^{new}. Thus, the weight factor G equals the product

$$G := \prod_{i=1}^{d} \frac{f_i\left(u_i\right)}{f_i^{\text{new}}\left(u_i^{\text{new}}\right)} = \prod_{i=1}^{d} \frac{1}{f_i^{\text{new}}\left(u_i^{\text{new}}\right)}$$

and guarantees an unbiased evaluation (cp. [CG08, Theorem 1]) via

$$\widehat{\Lambda}_m\left(\hat{t}_1^{(j)}, \ldots, \hat{t}_d^{(j)}\right) = \Lambda_m\left(\hat{t}_1^{(j)}, \ldots, \hat{t}_d^{(j)}\right) \cdot G$$

for any $\hat{\boldsymbol{t}} \in \mathbb{R}^d_{\ge 0}$.

Remark 6.8 (Payoff transformation). The proof of [CG08, Theorem 1] assumes that $\Lambda_{m,\infty} = \Lambda_m(t_1,\dots,t_d) = 0$ holds in case of $t_{\iota_m} > T$, which is not valid for any mBDS. Nevertheless, we know that at least equality

$$\exists c \in \mathbb{R} : \ \forall \boldsymbol{t} \in \overline{\mathbb{R}}^d_{\geq 0} \text{ with } t_{\iota_m} > T : \ \Lambda_m(t_1,\dots,t_d) = \Lambda_{m,\infty} = c$$

holds (cp. Definition 6.1). Hence, we can easily transform equation (6.48) according to

$$\begin{aligned} V_m &= \int\limits_0^\infty \cdots \int\limits_0^\infty \Lambda_m(\boldsymbol{t}) f_{\boldsymbol{\tau}}(\boldsymbol{t}) \,\mathrm{d}\boldsymbol{t} \\ &= \int\limits_0^\infty \cdots \int\limits_0^\infty (\Lambda_m(\boldsymbol{t}) - c + c) f_{\boldsymbol{\tau}}(\boldsymbol{t}) \,\mathrm{d}\boldsymbol{t} \\ &= \int\limits_0^\infty \cdots \int\limits_0^\infty \underbrace{(\Lambda_m(\boldsymbol{t}) - c)}_{=:\bar{\Lambda}_m(\boldsymbol{t})} f_{\boldsymbol{\tau}}(\boldsymbol{t}) \,\mathrm{d}\boldsymbol{t} + c \underbrace{\int\limits_0^\infty \cdots \int\limits_0^\infty f_{\boldsymbol{\tau}}(\boldsymbol{t}) \,\mathrm{d}\boldsymbol{t}}_{=1} \end{aligned}$$

and apply importance sampling techniques regarding payoff function $\bar{\Lambda}_m$. This result does not depend on the underlying copula.

The implementation of the importance sampling technique above is shown in Algorithm 6.10. For a more detailed description we refer to [JK04, CG08].

In case of small hazard rates h_i and a short maturity T Algorithm 6.10 obviously reduces the variance of a Monte–Carlo simulation. However, if these conditions are not met, we will not be able to ensure that the resulting variance is smaller or equal to the variance of a plain Monte–Carlo simulation (cp. [CG08, Figure 1]). Furthermore, the artificial default probabilities $\widetilde{p}_i^{(k)}$ are somewhat arbitrary, which impairs the performance of the underlying algorithm.

Hence, we develop artificial probabilities q_i in the following, which replace probabilities $\widetilde{p}_i^{(k)}$ in Algorithm 6.10 and claim to be closer to reality. For the development of these probabilities we assume the one–factor correlation

$$\begin{aligned} \Sigma = \Sigma_1^{\mathrm{F}} &= (\psi_1,\dots,\psi_d)^{\mathrm{tr}} (\psi_1,\dots,\psi_d) + \operatorname{diag}_d\left(1-\psi_1^2,\dots,1-\psi_d^2\right) \\ &= \boldsymbol{\psi} \cdot \boldsymbol{\psi}^{\mathrm{tr}} + D \end{aligned}$$

of Σ again (we could assume a k–factor structure as well, cp. equation (3.5)). Tuple $(w_1,\dots,w_d)^{\mathrm{tr}}$ is consequently modeled by the linear one–factor model (cp. Subsection 3.1.1)

$$(w_1,\dots,w_d)^{\mathrm{tr}} = \boldsymbol{\psi} \cdot \mathscr{Z} + D \cdot (\epsilon_1,\dots,\epsilon_d)^{\mathrm{tr}}, \tag{6.51}$$

Algorithm 6.10: Calculation of $V^{M,\text{IS}}_{m,\text{MC,Gau}}$

```
Global : $d,\ m,\ M,\ \Sigma,\ \Lambda_m,\ F_{\tau_1},\dots,F_{\tau_d}$
Output : $V^{M,\text{IS}}_{m,\text{MC,Gau}}$
Assume: $\Lambda_{m,\infty} = 0$, otherwise consider Remark 6.8

$V^{M,\text{IS}}_{m,\text{MC,Gau}} \leftarrow 0$
$A \leftarrow$Cholesky($d,\Sigma$) // $AA^{\texttt{tr}} = \Sigma$
for $l \leftarrow 1$ to $M$ do
    Sample $\boldsymbol{u} = (u_1,\dots,u_d)^{\text{tr}} \sim \mathcal{U}[0,1]^d,\ G \leftarrow 1,\ k \leftarrow 0$
    for $i \leftarrow 1$ to $d$ do
        $p_i \leftarrow \Phi\left(\left(\Phi^{-1}\left(F_{\tau_i}(T)\right) - \sum_{j=1}^{i-1} A_{i,j} z_i\right) / A_{i,i}\right)$
        if $k < m$ then
            $\widetilde{p}_i^{(k)} \leftarrow \max\left(\frac{m-k}{d-i+1}, p_i\right)$
        else
            $\widetilde{p}_i^{(k)} \leftarrow p_i$
        if $u_i \le \widetilde{p}_i^{(k)}$ then
            $u_i^{\text{new}} \leftarrow p_i \cdot u_i / \widetilde{p}_i^{(k)},\ G \leftarrow G \cdot p_i / \widetilde{p}_i^{(k)},\ k{+}{+}$
        else
            $u_i^{\text{new}} \leftarrow p_i + (1-p_i) \cdot \left(u_i - \widetilde{p}_i^{(k)}\right) / \left(1 - \widetilde{p}_i^{(k)}\right)$,
            $G \leftarrow G \cdot (1-p_i) / \left(1 - \widetilde{p}_i^{(k)}\right)$
        $z_i \leftarrow \Phi^{-1}\left(u_i^{\text{new}}\right)$ // $z_i \sim \Phi$
        $w_i \leftarrow \sum_{j=1}^{i} A_{i,j} z_i$
        $v_i \leftarrow \Phi(w_i)$
        $\hat{t}_i \leftarrow F_{\tau_i}^{-1}(v_i)$
    $V^{M,\text{IS}}_{m,\text{MC,Gau}} \leftarrow V^{M,\text{IS}}_{m,\text{MC,Gau}} + \Lambda_m\left(\hat{t}_1,\dots,\hat{t}_d\right) \cdot G$
$V^{M,\text{IS}}_{m,\text{MC,Gau}} \leftarrow V^{M,\text{IS}}_{m,\text{MC,Gau}} / M$
```

in which random variables $\mathcal{Z}, \epsilon_1, \ldots, \epsilon_d \sim \Phi$ are distributed independently according to Φ.

In the following $u_1, \ldots, u_{i-1}$ or rather the first $i-1$ default times $\hat{t}_1, \ldots, \hat{t}_{i-1}$ are assumed to be known. Furthermore, let χ_{i-1} denote the number of defaults within the first $i-1$ obligors, i.e. for exactly χ_{i-1} default times out of the set $\{\hat{t}_1, \ldots, \hat{t}_{i-1}\}$ the condition $\hat{t}_j \leq T$ holds, $j \in \{1, \ldots, i-1\}$. Then, the "realistic" artificial default probability q_i is given by

$$q_i^{(k)} := \mathbb{P}\left(\tau_i \leq T \,|\tau_1, \ldots, \tau_{i-1},\ \chi_{i-1} = k,\ \chi_d \geq m\right).$$

For calculating $q_i^{(k)}$ we first calculate a probability $\mathcal{P}_i^{(k)}$, which denotes the probability of observing at least m defaults within all obligors, assuming that k defaults have already occurred within the first i obligors. Thus, we define

$$\mathcal{P}_i^{(k)} := \mathbb{P}\left(\chi_d \geq m \,|\chi_i = k\right).$$

This probability can easily be calculated recursively by

$$\mathcal{P}_i^{(k)} = p_{i+1} \cdot \mathcal{P}_{i+1}^{(k+1)} + (1 - p_{i+1}) \cdot \mathcal{P}_{i+1}^{(k)}$$

with the terminal condition

$$\mathcal{P}_d^{(k)} = \begin{cases} 0, & \text{if } k \in \{0, \ldots, m-1\} \\ 1, & \text{else} \end{cases}.$$

Using probabilities $\mathcal{P}_i^{(k)}$ we can express the artificial probabilities $q_i^{(k)}$ as (cp. [CG08, Section 4.1])

$$q_i^{(k)} = \frac{p_i \cdot \mathcal{P}_i^{(k+1)}}{\mathcal{P}_{i-1}^{(k)}}.$$

Replacing $\widetilde{p}_i$ by $q_i^{(k)}$ in Algorithm 6.10 would finalize the desired modification at first sight.

Considering this modification in detail, though, it gets clear that it is not able to deal with completely dependent assets. Let us assume that $u_1^{\text{new}} \neq u_1, \ldots, u_{i-1}^{\text{new}} \neq u_{i-1}$ have been transformed according to $q_1^{(k_1)}, \ldots, q_{i-1}^{(k_{i-1})}$. Then, the value p_i, which has to be calculated next, does not equal the value p_i, which was used to calculate $\mathcal{P}_j^{(k)}$ and $q_{i-1}^{(k)}$ in the previous step. Thus, an essential condition for the application of this method is the independence of input random numbers u_j and probabilities p_i for $j \neq i$. This independence will only be guaranteed if and only if random variables $\tau_1, \ldots, \tau_d$ are at least conditional independent, which justifies the application of a factor correlation. Consequently, we also have to transform probabilities p_i to the new default probabilities

$$\widehat{p}_i := \Phi\left(\left(\Phi^{-1}\left(F_{\tau_i}(T)\right) - \psi_i \mathcal{Z}\right)/D_{i,i}\right),$$

which result from the usage of a one–factor model (cp. equation (6.51)).

The implementation of this method is shown in Algorithm 6.11. For a detailed derivation we refer to [CG08] again, in which the authors show that the method above reduces the variance of a Monte–Carlo simulation in any case (cp. [CG08, Proposition 7]). The resulting approximation, which uses M different simulations, is denoted by $V_{m,\mathrm{MC,Gau}}^{M,\mathrm{IS+FS}}$.

6.2.2. Importance sampling with respect to an Archimedean copula

Analogically to the usage of a Gaussian copula in the previous subsection, we develop an importance sampling method regarding any Archimedean copula in this subsection. For this, let $F_{\boldsymbol{\tau}}\left(t_1 \ldots, t_d\right) = C_{d,\phi_\theta}^{\mathrm{Arc}}\left(F_{\tau_1}\left(t_1\right), \ldots, F_{\tau_d}\left(t_d\right)\right)$ denote the joint CDF of random vector $\boldsymbol{\tau}$, for any $\boldsymbol{t} \in \overline{\mathbb{R}}_{\geq 0}^d$, and let V_m denote the value of an mBDS with payoff Λ_m and maturity T. Then, value V_m is approximated by

$$V_m \approx V_{m,\mathrm{MC,Arc}}^{M,\mathrm{IS}} := \frac{1}{M} \cdot \sum_{j=1}^{M} \widehat{\Lambda}_m\left(\hat{t}_1^{(j)}, \ldots, \hat{t}_d^{(j)}\right),$$

in which tuple $\left(\hat{t}_1^{(j)}, \ldots, \hat{t}_d^{(j)}\right)^{\mathrm{tr}}$ is drawn by using importance sampling and payoff $\widehat{\Lambda}_m$ still has to be specified.

A self–evident approach is the application of the importance sampling technique regarding the Gaussian copula (cp. Algorithm 6.10) including a corresponding density quotient weighting. For this, let

$$\begin{aligned} f_{\boldsymbol{\tau}}^{\mathrm{Arc}}\left(t_1, \ldots, t_d\right) : &= \frac{\partial^d}{\partial t_1 \cdots \partial t_d} F_{\boldsymbol{\tau}}\left(t_1 \ldots, t_d\right) \\ &= \frac{\partial^d}{\partial t_1 \cdots \partial t_d} C_{d,\phi_\theta}^{\mathrm{Arc}}\left(F_{\tau_1}\left(t_1\right), \ldots, F_{\tau_d}\left(t_d\right)\right) \end{aligned}$$

be the joint PDF of $\boldsymbol{\tau}$ with respect to any Archimedean copula and let

$$\begin{aligned} f_{\boldsymbol{\tau}}^{\mathrm{Gau}}\left(t_1, \ldots, t_d\right) : &= \frac{\partial^d}{\partial t_1 \cdots \partial t_d} \widetilde{F}_{\boldsymbol{\tau}}\left(t_1 \ldots, t_d\right) \\ : &= \frac{\partial^d}{\partial t_1 \cdots \partial t_d} C_{d,\Sigma}^{\mathrm{Gau}}\left(F_{\tau_1}\left(t_1\right), \ldots, F_{\tau_d}\left(t_d\right)\right) \end{aligned}$$

Algorithm 6.11: Calculation of $V^{M,\text{IS+FS}}_{m,\text{MC,Gau}}$

```
Global : $d,\ m,\ M,\ \boldsymbol{\psi},\ \Lambda_m,\ F_{\tau_1},\dots,F_{\tau_d}$
Output : $V^{M,\text{IS+FS}}_{m,\text{MC,Gau}}$
Assume: $\Lambda_{m,\infty} = 0$, otherwise consider Remark 6.8
$V^{M,\text{IS+FS}}_{m,\text{MC,Gau}} \leftarrow 0$
for $i \leftarrow 1$ to $d$ do
    if $i < m$ then
        $\mathcal{P}_d^{(i)} \leftarrow 0$
    else
        $\mathcal{P}_d^{(i)} \leftarrow 1$
for $l \leftarrow 1$ to $M$ do
    Sample $\boldsymbol{u} = (u_1,\dots,u_{d+1})^{\text{tr}} \sim \mathcal{U}[0,1]^{d+1},\ G \leftarrow 1,\ k \leftarrow 0$
    $\mathcal{Z} \leftarrow \Phi^{-1}(u_{d+1})$
    for $i \leftarrow 1$ to $d$ do
        $\widehat{p}_i \leftarrow \Phi\left(\left(\Phi^{-1}\left(F_{\tau_i}(T)\right) - \psi_i \mathcal{Z}\right)/D_{i,i}\right)$
    for $\alpha \leftarrow d-1$ to $0$ do
        for $\beta \leftarrow 0$ to $\alpha$ do
            $\mathcal{P}_\alpha^{(\beta)} \leftarrow \widehat{p}_{\alpha+1}\mathcal{P}_{\alpha+1}^{(\beta+1)} + (1-\widehat{p}_{\alpha+1})\,\mathcal{P}_{\alpha+1}^{(\beta)}$
    for $i \leftarrow 1$ to $d$ do
        $q_i^{(k)} \leftarrow \widehat{p}_i \cdot \mathcal{P}_i^{(k+1)}/\mathcal{P}_{i-1}^{(k)}$
        if $u_i \le q_i^{(k)}$ then
            $u_i^{\text{new}} \leftarrow \widehat{p}_i \cdot u_i/q_i^{(k)},\ G \leftarrow G \cdot \widehat{p}_i/q_i^{(k)},\ k++$
        else
            $u_i^{\text{new}} \leftarrow \widehat{p}_i + (1-\widehat{p}_i)\cdot\left(u_i - q_i^{(k)}\right)/\left(1-q_i^{(k)}\right)$
            $G \leftarrow G \cdot (1-\widehat{p}_i)/\left(1-q_i^{(k)}\right)$
        $z_i \leftarrow \Phi^{-1}\left(u_i^{\text{new}}\right)$ // $z_i \sim \Phi$
        $w_i \leftarrow \psi_i \mathcal{Z} + D_{i,i} z_i$
        $v_i \leftarrow \Phi(w_i)$
        $\hat{t}_i \leftarrow F_{\tau_i}^{-1}(v_i)$
    $V^{M,\text{IS+FS}}_{m,\text{MC,Gau}} \leftarrow V^{M,\text{IS+FS}}_{m,\text{MC,Gau}} + \Lambda_m\left(\hat{t}_1,\dots,\hat{t}_d\right)\cdot G$
$V^{M,\text{IS+FS}}_{m,\text{MC,Gau}} \leftarrow V^{M,\text{IS+FS}}_{m,\text{MC,Gau}}/M$
```

be the joint PDF of $\boldsymbol{\tau}$ with respect to the Gaussian copula, $\boldsymbol{t} \in \overline{\mathbb{R}}^d_{\geq 0}$. Then, V_m is calculated by

$$\begin{aligned} V_m &= \int\limits_0^\infty \cdots \int\limits_0^\infty \Lambda_m(\boldsymbol{t}) \cdot f^{\text{Arc}}_{\boldsymbol{\tau}}(\boldsymbol{t})\, \mathrm{d}\boldsymbol{t} \\ &= \int\limits_0^\infty \cdots \int\limits_0^\infty \underbrace{\Lambda_m(\boldsymbol{t}) \cdot \frac{f^{\text{Arc}}_{\boldsymbol{\tau}}(\boldsymbol{t})}{f^{\text{Gau}}_{\boldsymbol{\tau}}(\boldsymbol{t})}}_{=:\Lambda^{\text{IS}}_m(\boldsymbol{t})} \cdot f^{\text{Gau}}_{\boldsymbol{\tau}}(\boldsymbol{t})\, \mathrm{d}\boldsymbol{t} \\ &= \int\limits_0^\infty \cdots \int\limits_0^\infty \Lambda^{\text{IS}}_m(\boldsymbol{t}) \cdot f^{\text{Gau}}_{\boldsymbol{\tau}}(\boldsymbol{t})\, \mathrm{d}\boldsymbol{t}. \qquad (6.52) \end{aligned}$$

Obviously, equation (6.52) has the same structure like equation (6.49) and is solved efficiently by using Algorithm 6.10. This technique is working well in many cases and is presented in detail in [SH12b]. The resulting approximation is denoted by $V^{M,\text{IS}}_{m,\text{MC},\text{Arc}}$. However, depending on the underlying copula the evaluation of the joint PDF $f^{\text{Arc}}_{\boldsymbol{\tau}}$ is very complex. Additionally, the joint PDF $f^{\text{Gau}}_{\boldsymbol{\tau}}$ must be chosen carefully and depending on $f^{\text{Arc}}_{\boldsymbol{\tau}}$. Finally, a variance reducing effect will only be observed if the quotient $f^{\text{Arc}}_{\boldsymbol{\tau}}\left(\hat{\boldsymbol{t}}\right) / f^{\text{Gau}}_{\boldsymbol{\tau}}\left(\hat{\boldsymbol{t}}\right)$ shows moderate values for any $\hat{\boldsymbol{t}} \in \mathbb{R}^d_{\geq 0}$.

In order to cope with these disadvantages, we are looking for another importance sampling method with respect to any Archimedean copula. For this, we transform Algorithm 6.11, which uses the factor structure of the underlying correlation matrix.

Due to [LG05, Section 2.3] we know that any Archimedean copula is uniquely defined by the inverse Laplace–Stieltjes transformation of its inverse generator ϕ_θ^{-1}. Furthermore, this transformation represents a proper CDF, which uniquely defines a nonlinear one–factor model (cp. also Subsection 4.2.1 and pages 81f.). Then, with the help of Algorithm 4.3 (line 3), which already shows a nonlinear one–factor model for copula output $v_1, \dots, v_d$, we can directly transform the input parameters of Algorithm 6.11 to the usage of any Archimedean copula.

Let V^{Arc} denote the random variable, which represents the factor in the one–factor model above. Then, default probabilities $\widehat{p}_i$, $i \in \{1, \dots, d\}$, are given by

$$\begin{aligned} \widehat{p}_i :&= \mathbb{P}\left(\tau_i \leq T \,\middle|\, V^{\text{Arc}}\right) = \mathbb{P}\left(F^{-1}_{\tau_i}(v_i) \leq T \,\middle|\, V^{\text{Arc}}\right) \\ &= \mathbb{P}\left(\phi_\theta^{-1}\left(\frac{-\ln(\epsilon_i)}{V^{\text{Arc}}}\right) \leq F_{\tau_i}(T)\right) = \mathbb{P}\left(\frac{-\ln(\epsilon_i)}{V^{\text{Arc}}} \leq \phi_\theta\left(F_{\tau_i}(T)\right)\right) \\ &= \mathbb{P}\left(\epsilon_i \leq \exp\left(-V^{\text{Arc}} \cdot \phi_\theta\left(F_{\tau_i}(T)\right)\right)\right) = \exp\left(-V^{\text{Arc}} \cdot \phi_\theta\left(F_{\tau_i}(T)\right)\right). \end{aligned}$$

Additionally, the random variables $v_1, \dots, v_d$ satisfy the nonlinear one–factor model

$$v_i = \phi_\theta^{-1}\left(\frac{-\ln(\epsilon_i)}{V^{\text{Arc}}}\right),$$

in which $\epsilon_1, \ldots, \epsilon_d \sim \mathcal{U}[0,1]$ are distributed uniformly and independently and factor V^{Arc} is distributed according to distribution F^{Arc} from Table 4.3.

Using this information we can transform Algorithm 6.11 completely to the usage of any Archimedean copula, see Algorithm 6.12. The approximation, which is calculated by using this technique, is denoted by $V_{m,\text{MC},\text{Arc}}^{M,\text{IS+FS}}$.

6.2.3. Academic example: Exhibit 6 and Illustration 3 from [Li00]

In this subsection we compare the different importance sampling techniques from Sections 6.2.1 and 6.2.2 regarding their performance. Afterwards we also compare these techniques to different quadrature methods from Section 6.1.

Let us consider the example from [Li00, Illustration 3] in the following. In this example a simplified mBDS with respect to $d = 5$ obligors is described. Their hazard rates are assumed to be constant $h_i \equiv 0.1$ for each $i \in \{1, \ldots, 5\}$ and their correlation is assumed to be homogeneous $\rho := \rho_{i,j} \in [0,1]$, $i, j \in \{1, \ldots, 5\}$, $i \neq j$. Furthermore, a risk free interest rate of $r_t \equiv 0.1$, a maturity of $T = 2$ and a triggering obligor $m = 1$ are given. Nevertheless, in the following we change these parameters for creating different scenarios. Only the payoff remains constant.

The payoff of the underlying contract from [Li00, Illustration 3] consists of only one default payment (cp. Section 2.1). Within this default payment we assume a constant recovery rate $R := R_i \equiv 0$ for each obligor $i \in \{1, \ldots, d\}$ and a nominal value of $N = \sum_{i=1}^{d} N_i = 1$. Then, the resulting payoff is given by

$$\Lambda_m^{\text{Li}}(t_1, \ldots, t_d) := \exp(-r t_{\iota_m}) \cdot \mathbf{1}_{\{t_{\iota_m} \leq T\}} =: \widetilde{\Lambda}^{\text{Li}}(t_{\iota_m})$$

for any $\boldsymbol{t} \in \overline{\mathbb{R}}_{\geq 0}^{d}$ and the value V_m^{Li} of the underlying contract is given by (cp. Theorem 6.1 and Proposition 6.1)

$$\begin{aligned} V_m^{\text{Li}} &= \int\limits_{\mathbb{R}_{\geq 0}^d} \Lambda_m^{\text{Li}}(\boldsymbol{t})\, f_{\boldsymbol{\tau}}(\boldsymbol{t})\, \mathrm{d}\boldsymbol{t} \\ &= \underbrace{\lim_{t_u \nearrow T} \widetilde{\Lambda}^{\text{Li}}(t_u)}_{=\widetilde{\Lambda}^{\text{Li}}(T)} F_{\tau_{\iota_m}}(T) - \int\limits_0^T \widetilde{\Lambda}^{\text{Li}'}(t)\, F_{\tau_{\iota_m}}(t)\, \mathrm{d}t. \end{aligned} \tag{6.53}$$

We compare the different importance sampling techniques from Subsections 6.2.1 and 6.2.2 in the following. For this, we choose the input parameters $d = 10$, $m = 5$

Algorithm 6.12: Calculation of $V^{M,\text{IS+FS}}_{m,\text{MC,Arc}}$

```
Global : $d,\ m,\ M,\ \phi_\theta,\ \Lambda_m,\ F_{\tau_1},\dots,F_{\tau_d}$
Output : $V^{M,\text{IS+FS}}_{m,\text{MC,Arc}}$
Assume: $\Lambda_{m,\infty} = 0$, otherwise consider Remark 6.8

$V^{M,\text{IS+FS}}_{m,\text{MC,Arc}} \leftarrow 0$
for $i \leftarrow 1$ to $d$ do
    if $i < m$ then
        $\mathcal{P}^{(i)}_d \leftarrow 0$
    else
        $\mathcal{P}^{(i)}_d \leftarrow 1$
for $l \leftarrow 1$ to $M$ do
    Sample $\boldsymbol{u} = (u_1,\dots,u_d)^{\text{tr}} \sim \mathcal{U}[0,1]^d$ and $V^{\text{Arc}} \sim F^{\text{Arc}}$
    $G \leftarrow 1,\ k \leftarrow 0$
    for $i \leftarrow 1$ to $d$ do
        $\widehat{p}_i \leftarrow \exp\left(-V^{\text{Arc}} \cdot \phi_\theta\left(F_{\tau_i}(T)\right)\right)$
    for $\alpha \leftarrow d-1$ to $0$ do
        for $\beta \leftarrow 0$ to $\alpha$ do
            $\mathcal{P}^{(\beta)}_\alpha \leftarrow \widehat{p}_{\alpha+1}\mathcal{P}^{(\beta+1)}_{\alpha+1} + \left(1-\widehat{p}_{\alpha+1}\right)\mathcal{P}^{(\beta)}_{\alpha+1}$
    for $i \leftarrow 1$ to $d$ do
        $q^{(k)}_i \leftarrow \widehat{p}_i \cdot \mathcal{P}^{(k+1)}_i / \mathcal{P}^{(k)}_{i-1}$
        if $u_i \le q^{(k)}_i$ then
            $u^{\text{new}}_i \leftarrow \widehat{p}_i \cdot u_i / q^{(k)}_i,\ G \leftarrow G \cdot \widehat{p}_i / q^{(k)}_i,\ k++$
        else
            $u^{\text{new}}_i \leftarrow \widehat{p}_i + (1-\widehat{p}_i) \cdot \left(u_i - q^{(k)}_i\right) / \left(1 - q^{(k)}_i\right)$
            $G \leftarrow G \cdot (1-\widehat{p}_i) / \left(1 - q^{(k)}_i\right)$
        $v_i \leftarrow \phi_\theta^{-1}\left(-\ln\left(u^{\text{new}}_i\right) / V^{\text{Arc}}\right)$
        $\hat{t}_i \leftarrow F^{-1}_{\tau_i}\left(v_i\right)$
    $V^{M,\text{IS+FS}}_{m,\text{MC,Arc}} \leftarrow V^{M,\text{IS+FS}}_{m,\text{MC,Arc}} + \Lambda_m\left(\hat{t}_1,\dots,\hat{t}_d\right) \cdot G$
$V^{M,\text{IS+FS}}_{m,\text{MC,Arc}} \leftarrow V^{M,\text{IS+FS}}_{m,\text{MC,Arc}} / M$
```

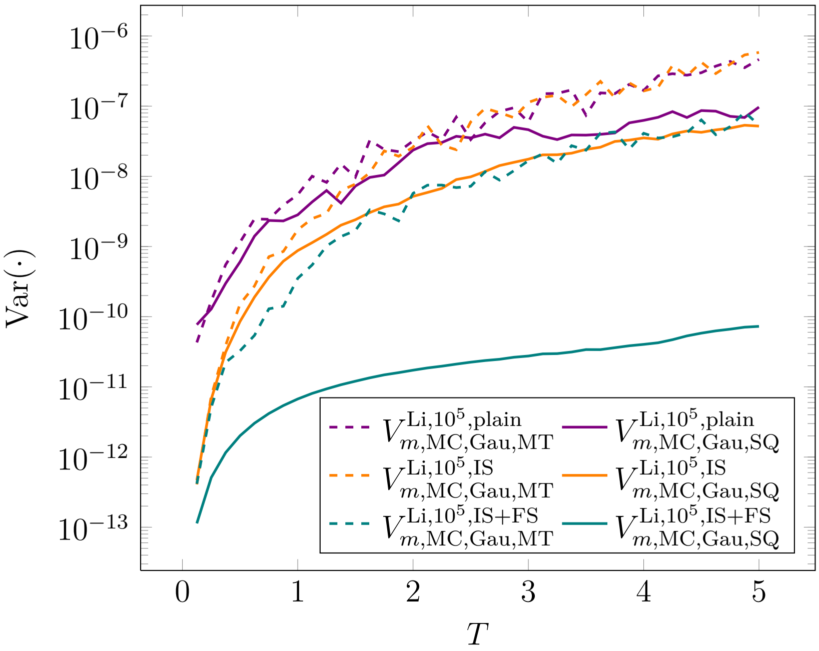


Figure 6.8.: Variances of different Monte–Carlo approximations using a Gaussian copula with correlation matrix Σ (cp. equation (6.55)).

and $r = 0.03$. Additionally, the constant hazard rates are given by vector

$$\boldsymbol{h} = (0.05, 0.01, 0.06, 0.09, 0.03, 0.1, 0.02, 0.07, 0.04, 0.08)^{\mathrm{tr}} \tag{6.54}$$

and the correlation matrix Σ is given by vector

$$\begin{aligned}\boldsymbol{\psi} &= (\psi_1, \dots, \psi_{10})^{\mathrm{tr}} \\ &= (0.75, 0.16, -0.46, -0.93, 0.61, -0.22, 0.86, 0.34, -0.04, 0.57)^{\mathrm{tr}}\end{aligned} \tag{6.55}$$

via $\Sigma = \Sigma^{\mathrm{F}}_1 = \boldsymbol{\psi} \cdot \boldsymbol{\psi}^{\mathrm{tr}} + \operatorname{diag}(1 - \psi_1^2, \dots, 1 - \psi_{10}^2)$.

In Figures 6.8 and 6.9 the variances regarding 30 different loops, each with $M = 10^5$ simulations, are shown. Here, maturity T remains variable for creating different scenarios of application. Furthermore, we use different kinds of random numbers. On the one hand we use “pure” random numbers created with Mersenne Twister (abbr.: MT) and on the other hand we use quasi–random numbers created by means of Sobol’s sequence (abbr.: SQ). Regarding the application of the Clayton copula we have to remark that the usage of quasi–random numbers is not conform to the usage of rejection Algorithm B.2. Here, we have to use the deterministic Algorithm B.3 to conserve the structure of Sobol’s sequence.

Both variance analyses in Figures 6.8 and 6.9 show that these methods, which make use of the factor structure, clearly outperform the other methods. Variance reduction

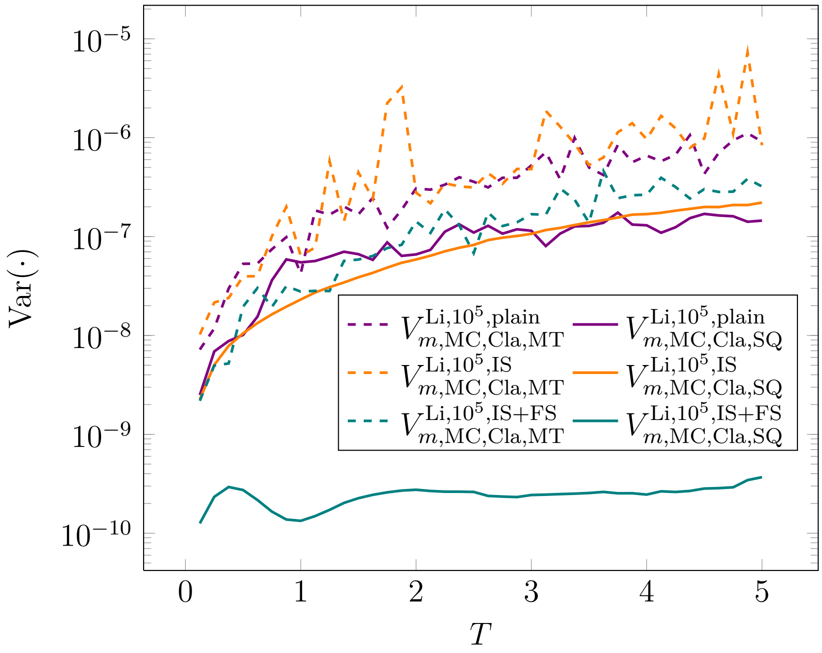


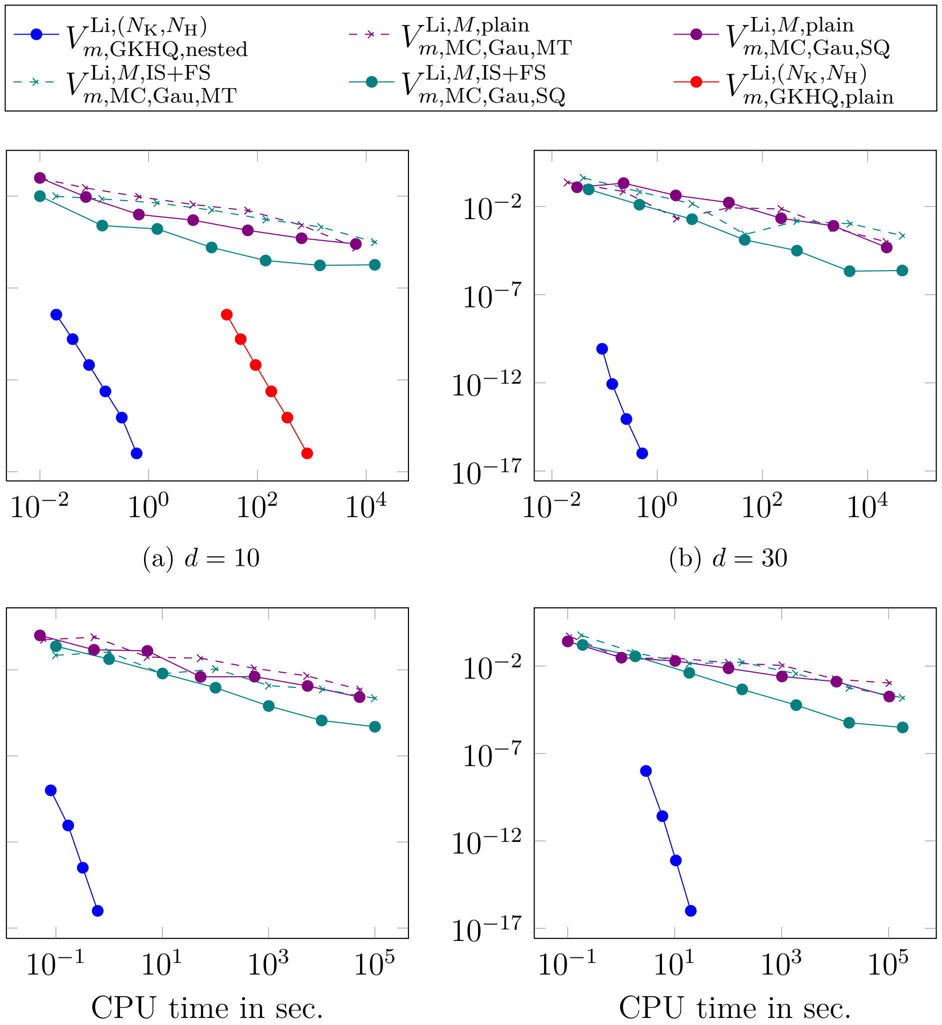
Figure 6.9.: Variances of different Monte–Carlo approximations using a Clayton copula with parameter $\theta = 0.25$.

ratios up to 10^3 are reached. Furthermore, it is obvious that the usage of quasi–random numbers is preferable in this context.

In Figures C.1 and C.2 analogical analyses are shown with respect to the Frank and Gumbel copula. However, the variances of $V^{\mathrm{Li},M,\mathrm{IS}}_{m,\mathrm{MC},\mathrm{Fra}}$ and $V^{\mathrm{Li},M,\mathrm{plain}}_{m,\mathrm{MC},\mathrm{Gum}}$ are neglected in these figures.

In the following we compare the above Monte–Carlo methods to the quadrature methods presented in Section 6.1 as well. For this, we vary dimension d within the current example. The used hazard rates as well as the required factor loads (Gaussian copula) are given in Table C.4. Additionally, we choose a maturity of $T = 1$, a risk free interest rate of $r_t \equiv 0.03$ and a triggering obligor $m = \lfloor d/4 \rfloor$ (Gaussian copula) or $m = \lfloor d/10 \rfloor$ (Clayton copula) respectively.

We only consider the Gaussian and the Clayton copula in the following. For calculating an “exact” value in each case, we apply a quadrature method from Subsections 6.1.3 and 6.1.4 and increase the underlying number of nodes as long as we can observe a change in the resulting value. Thus, we can calculate relative errors ϵ for each method and can plot them against their corresponding CPU times (cp. Figures 6.10 and 6.11). Additional information concerning the choice of N_{LG} within the Clayton copula is given by Remark 6.10.



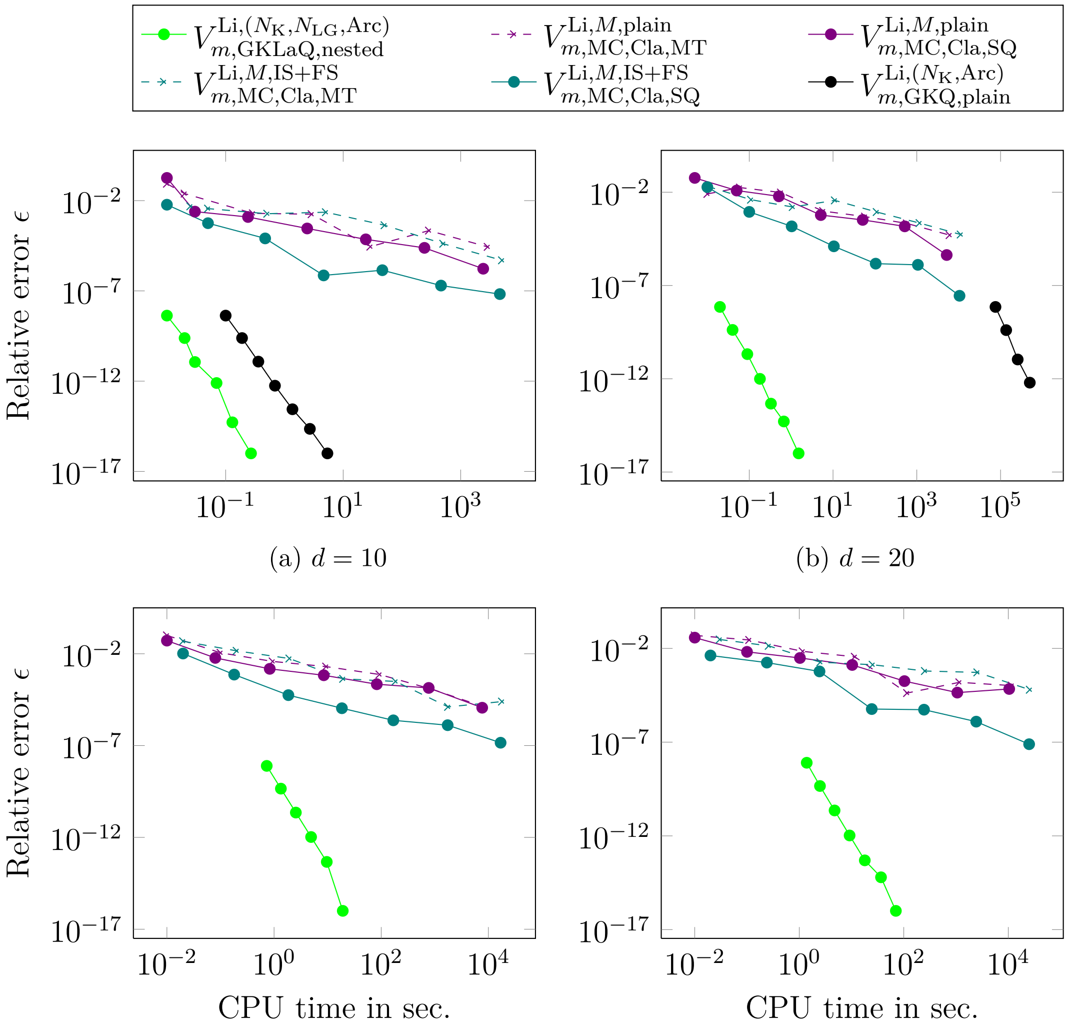
(a) $d = 10$

(b) $d = 30$

(c) $d = 60$

(d) $d = 100$

Figure 6.10.: Comparison of different Monte–Carlo and quadrature methods using the Gaussian copula.



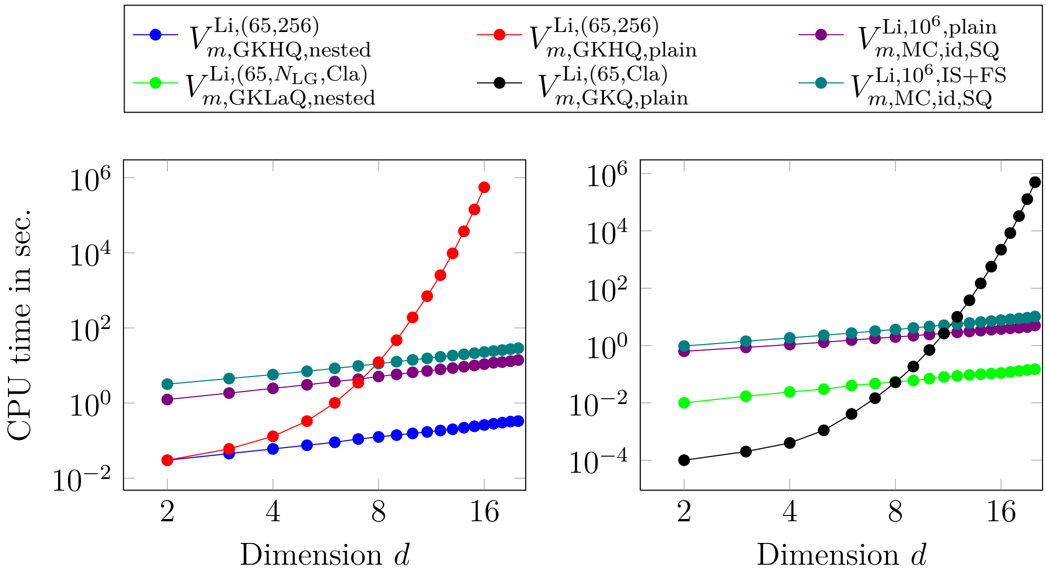
(a) $d = 10$

(b) $d = 20$

(c) $d = 30$

(d) $d = 40$

Figure 6.11.: Comparison of different Monte–Carlo and quadrature methods using the Clayton copula.



(a) Gaussian copula (id=Gau)

(b) Clayton copula (id=Cla)

Figure 6.12.: Comparison of different Monte–Carlo and quadrature methods regarding CPU time and dimension d.

Remark 6.9 (Need of a higher precision). Assuming a high dimension d the calculation of certain binomial coefficients and the evaluation of $\tilde{\eta}_k^{(l)}$ result in very small values as well as in very high values (copula depending). In order to guarantee an accurate pricing the standard precision of the underlying CPU has to be increased. In C++ we can use the package “The GNU Multiple Precision Arithmetic Library” (cp. [GNU12]) for example.

In Figures 6.10 and 6.11 we can finally see the performance of each method. In any dimension we can observe that the quadrature methods significantly outperform the Monte–Carlo methods. Furthermore, we can note that the nested quadrature methods clearly outperform the plain ones. Regarding the Monte–Carlo methods we can observe the dominance of the factor correlation methods using Sobol’s sequence again.

In order to create Figure 6.10 we used *The GNU Multiple Precision Arithmetic Library* for dimensions $d \geq 70$ and in order to create Figure 6.11 we used the same library for dimensions $d \geq 30$ (cp. Remark 6.9 and [GNU12]).

Finally, we show the complexity of each method regarding to the corresponding dimension in Figure 6.12. In this figure we fix the number of nodes within the quadrature methods to $N_{\mathrm{K}} = 65$ and $N_{\mathrm{H}} = 256$ (cp. Remark 6.10 for the choice of N_{LG}) and the number of simulations within the Monte–Carlo methods to $M = 10^6$. It falls into place that the nested quadrature methods offer an overall complexity, which is comparable to a Monte–Carlo simulation. Additionally, these quadratures offer a convergence, which is comparable to a plain (or d–dimensional) quadrature (cp. Figures 6.10 and 6.11).

Remark 6.10 (Choice of N_{LG}). Several tests have shown that it is much more efficient to truncate the inner integral of $V_{m,\mathrm{GKLaQ},\mathrm{nested}}^{\mathrm{Li},(N_{\mathrm{K}},N_{\mathrm{LG}},\mathrm{Cla})}$ (cp. equation (6.44)) and solve the corresponding finite integral via a GKQ or GLQ. Thus, in Figures 6.12 and 6.11 the GLaK has been replaced by a GKQ (using $N_{\mathrm{K}} = 513$ nodes in case of Figure 6.12).

6.3. Excursus: European multi–asset options

Besides its application to Basket Default Swaps the theory from Section 6.1 (and Section 6.2) is also applicable to evaluate multi–dimensional European maximum or minimum options. Within this context we just have to replace the m–th smallest default time by the m–th smallest stock price at $t = T$.

For this, let $S := \{S_1, \dots, S_d\}$ denote a basket of stocks (only in this section!) with initial prices $S_{t_0,1}, \dots, S_{t_0,d}$ at $t = t_0$. In the following we assume that $t_0 = 0$ holds, without loss of generality. Furthermore, let $\zeta_1, \dots, \zeta_d$ denote real random variables, which represent the stock prices of the corresponding stocks $S_1, \dots, S_d$ at $t = T$. Additionally, let $\mu_1, \dots, \mu_d$ denote the deterministic growth rates of $S_1, \dots, S_d$ and $\sigma_1, \dots, \sigma_d$ their volatilities.

Let us assume that the stock prices satisfy a geometric Brownian motion. Then, we can state the CDF F_{ζ_j} of $\zeta_j,\ j \in \{1, \dots, d\}$, as (cp. [Sey12, Section 1.8])

$$\forall S_T \in \overline{\mathbb{R}}_{\geq 0}:\ F_{\zeta_j}\left(S_T\right) := \Phi\left(\frac{\ln\left(\frac{S_T}{S_{0,j}}\right) - \left(\mu_j - \frac{\sigma_j^2}{2}\right)T}{\sigma_j\sqrt{T}}\right).$$

In the following these CDFs act as the MDFs of random vector $\boldsymbol{\zeta}$ and replace CDFs F_{τ_j} from Section 6.1.

Then, we can derive the CDF of the m–th smallest stock price at $t = T$ (analogically to Section 6.1) with Lemma 6.1 as

$$\forall S_T \in \overline{\mathbb{R}}_{\geq 0}: F_{\zeta_{\iota_m}}\left(S_T\right) := \sum_{k=m}^{d} c_{k,m} \sum_{\substack{I \subseteq \{1,\dots,d\} \\ |I|=k}} \check{C}_k^I\left(F_{\zeta_{I_1}}\left(S_T\right), \dots, F_{\zeta_{I_k}}\left(S_T\right)\right),$$

in which $\check{C}_d$ denotes the d–copula, which meets (cp. Theorems 1.1 and 1.2)

$$\forall \boldsymbol{S}_T \in \overline{\mathbb{R}}_{\geq 0}^d:\ \check{C}_d\left(F_{\zeta_1}\left(S_{T,1}\right), \dots, F_{\zeta_d}\left(S_{T,d}\right)\right) = F_{\boldsymbol{\zeta}}\left(S_{T,1}, \dots, S_{T,d}\right).$$

Analogically to Theorem 6.1, we can calculate the value $V_{\text{min/max}}^{\text{opt}}$ of a d–dimensional European max/min option with payoff $\Lambda_{\text{min/max}}^{\text{opt}}\left(S_{T,1}, \dots, S_{T,d}\right)$ as (choose $m = 1$ for a minimum option and $m = d$ for a maximum option)

$$V_{\text{min/max}}^{\text{opt}} = \int_0^\infty \cdots \int_0^\infty \Lambda_{\text{min/max}}^{\text{opt}}\left(S_{T,1}, \dots, S_{T,d}\right) \cdot f_{\boldsymbol{\zeta}}(S_{T,1}, \dots, S_{T_d}) \mathrm{d}S_{T,1} \cdots \mathrm{d}S_{T,d} \quad (6.56)$$

$$= \int_0^\infty \widetilde{\Lambda}^{\text{opt}}\left(S_T\right) \cdot f_{\zeta_{\iota_m}}(S_T) \mathrm{d}S_T. \quad (6.57)$$

At this, $f_{\boldsymbol{\zeta}}$ and $f_{\zeta_{\iota_m}}$ denote the PDFs of $F_{\boldsymbol{\zeta}}$ and $F_{\zeta_{\iota_m}}$ and payoff function $\widetilde{\Lambda}^{\text{opt}}_{\text{min/max}}\,(S_T)$, $S_T \in \overline{\mathbb{R}}_{\geq 0}$, depends on the underlying option type. Additionally, we have to replace μ_i in equations (6.56) and (6.57) by the constant risk free interest rate $r_t \equiv r$ to ensure a risk neutral evaluation (cp. [Sey12, Section 1.5])

Example 6.7 (Minimum call option)**.** Assuming the underlying contract to be a d–dimensional European minimum call option with strike price $K \in \mathbb{R}_{\geq 0}$, the payoff function is given by

$$\begin{aligned} \Lambda^{\text{call}}_{\text{min}}\,(S_{T,1},\ldots,S_{T,d}) &= \exp(-rT)\max\{\min\{S_{T,1},\ldots,S_{T,d}\} - K, 0\} \\ \Rightarrow \widetilde{\Lambda}^{\text{call}}(S_T) &= \exp(-rT)\max\{S_T - K, 0\} \end{aligned}$$

for any $\boldsymbol{S}_T \in \overline{\mathbb{R}}^d_{\geq 0}$ and $S_T \in \overline{\mathbb{R}}_{\geq 0}$. The corresponding option price is given by

$$V^{\text{call}}_{\text{min}} = \exp(-rT)\int\limits_K^\infty (S_T - K)\cdot f_{\zeta_{\iota_1}}(S_T)\mathrm{d}S_T.$$

Remark 6.11 (Problem of inhomogeneous baskets)*.* Within this context we do not have to mind inhomogeneous portfolios regarding recovery rate and nominal value anymore (cp. Section 6.1.2). These parameters are not of interest any longer.

Obviously, it is possible to transfer the complete theory from Section 6.1 (and from Section 6.2) to this new context and to evaluate high–dimensional European max/min options efficiently. We only have to choose a copula $\check{C}$ with desired dependence structure.

If we want to price according to the multi–dimensional Black–Scholes model (cp. [BS73]), we will have to choose the Gaussian copula for modeling the joint CDF $F_{\boldsymbol{\zeta}}$. This statement is proven in the following by analyzing the special case of an European minimum call option on two assets ($d = 2$).

Let $V^{\text{call}}_{\text{min}}$ denote the value of an European minimum call option regarding basket $S := \{S_1, S_2\}$. Both stock prices S_1 and S_2 satisfy the Black–Scholes model and their Wiener processes are correlated by $\rho \in [-1, 1]$. The value $V^{\text{call}}_{\text{min}}$ is given by (cp. [Stu82, Hau06])

$$\begin{aligned} V^{\text{call}}_{\text{min}} : = {} & S_{0,1}\Phi^2_{-\rho_1}(y_1, -\delta) + S_{0,2}\Phi^2_{-\rho_2}\left(y_2, \delta - \sigma\sqrt{T}\right) \\ & - K\exp(-rT)\Phi^2_{\rho}\left(y_1 - \sigma_1\sqrt{T}, y_2 - \sigma_2\sqrt{T}\right) \end{aligned} \tag{6.58}$$

with

$$\begin{aligned}
\Phi_\rho^2(u,v) &:= \int_{-\infty}^{u}\int_{-\infty}^{v} \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{x^2+2\rho xy+y^2}{2(1-\rho)}\right) \mathrm{d}x\mathrm{d}y \\
\delta &:= \frac{\ln\left(\frac{S_{0,1}}{S_{0,2}}\right)+\frac{\sigma^2}{2}T}{\sigma\sqrt{T}} & y_{1/2} &:= \frac{\ln\left(\frac{S_{0,1/2}}{K}\right)+\left(r+\frac{\sigma_{1/2}}{2}\right)T}{\sigma_{1/2}\sqrt{T}} \\
\sigma &:= \sqrt{\sigma_1^2+\sigma_2^2-2\rho\sigma_1\sigma_2} & \rho_{1/2} &:= \frac{\sigma_{1/2}-\rho\sigma_{2/1}}{\sigma}.
\end{aligned}$$

The following proposition shows that Theorem 6.1 combined with the usage of the Gaussian copula results in the same option value $V_{\min}^{\text{call}}$ as in (6.58).

Proposition 6.2 (Pricing multi–dimensional European options)**.** Let $V_{\min}^{\text{call}}$ be the value of a two–dimensional European minimum call option on stocks S_1, S_2, whose joint CDF is modeled by using the Gaussian copula with correlation coefficient ρ. Then, $V_{\min}^{\text{call}}$ is given analytically by

$$V_{\min}^{\text{call}} = \exp(-rT)\int_{K}^{\infty}(S_T-K)\cdot f_{\zeta_{\iota_1}}(S_T)\mathrm{d}S_T, \tag{6.59}$$

in which CDF $F_{\zeta_{\iota_1}}$ is given by

$$\begin{aligned}
F_{\zeta_{\iota_1}}(S_T) &= \int_{-\infty}^{S_T} f_{\zeta_{\iota_1}}(u)\mathrm{d}u \\
&= \sum_{i=1}^{2}\Phi\left(\frac{\ln\left(\frac{S_T}{S_{0,i}}\right)-\left(r-\frac{\sigma_i^2}{2}\right)T}{\sigma_i\sqrt{T}}\right) \\
&\qquad - \Phi_\rho^2\left(\frac{\ln\left(\frac{S_T}{S_{0,1}}\right)-\left(r-\frac{\sigma_1^2}{2}\right)T}{\sigma_1\sqrt{T}}, \frac{\ln\left(\frac{S_T}{S_{0,2}}\right)-\left(r-\frac{\sigma_2^2}{2}\right)T}{\sigma_2\sqrt{T}}\right) \\
&=: \Phi\left(z_1(S_T)\right)+\Phi\left(z_2(S_T)\right)-\Phi_\rho^2\left(z_1(S_T),z_2(S_T)\right).
\end{aligned}$$

Proof. To prove (6.59) we show its equivalence to equation (6.58). For this, we transform equation (6.59) as follows:

$$\begin{aligned}
&\exp(-rT)\int_{K}^{\infty}(S_T-K)\cdot f_{\zeta_{\iota_1}}(S_T)\mathrm{d}S_T \\
&\qquad = \exp(-rT)\int_{K}^{\infty} S_T\left(\frac{\partial}{\partial z_1}\Phi\left(z_1\right)\frac{\partial z_1}{\partial S_T}-\frac{\partial}{\partial z_1}\Phi_\rho^2\left(z_1,z_2\right)\frac{\partial z_1}{\partial S_T}\right)\mathrm{d}S_T \\
&+\exp(-rT)\int_{K}^{\infty} S_T\left(\frac{\partial}{\partial z_2}\Phi\left(z_2\right)\frac{\partial z_2}{\partial S_T}-\frac{\partial}{\partial z_2}\Phi_\rho^2\left(z_1,z_2\right)\frac{\partial z_2}{\partial S_T}\right)\mathrm{d}S_T \\
&-K\exp(-rT)\int_{K}^{\infty}\frac{\partial}{\partial S_T}\Phi\left(z_1(S_T)\right)+\frac{\partial}{\partial S_T}\Phi\left(z_2(S_T)\right)-\frac{\partial}{\partial S_T}\Phi_\rho^2\left(z_1(S_T),z_2(S_T)\right)\mathrm{d}S_T
\end{aligned}$$

$$=: I_1 + I_2 - I_3$$

With the notation above we show the equality of I_1 and $S_{0,1}\Phi^2_{-\rho_1}(y_1, -\delta)$ in the following:

$$
\begin{aligned}
I_1 &= \exp(-rT) \int_K^\infty S_T \left(\frac{\partial}{\partial z_1} \Phi(z_1) \frac{\partial z_1}{\partial S_T} - \frac{\partial}{\partial z_1} \Phi^2_\rho (z_1, z_2) \frac{\partial z_1}{\partial S_T} \right) \mathrm{d}S_T \\
&= \exp(-rT) \int_K^\infty \frac{1}{\sqrt{2\pi T}\sigma_1} \exp\left(-\frac{z_1(S_T)^2}{2} \right) \\
&\quad - \int_{-\infty}^{z_2(S_T)} \frac{1}{2\pi\sigma_1\sqrt{(1-\rho^2)T}} \exp\left(-\frac{z_1(S_T)^2 - 2z_1(S_T)\rho t_2 + t_2^2}{2(1-\rho^2)} \right) \mathrm{d}t_2 \mathrm{d}S_T \\
&= \exp(-rT) \int_K^\infty \frac{1}{\sqrt{2\pi T}\sigma_1} \exp\left(-\frac{z_1(S_T)^2}{2} \right) \\
&\quad \cdot \left[1 - \int_{-\infty}^{z_2(S_T)} \frac{1}{\sqrt{(1-\rho^2)2\pi}} \exp\left(-\frac{(t_2 - \rho z_1(S_T))^2}{2(1-\rho^2)} \right) \mathrm{d}t_2 \right] \mathrm{d}S_T \\
&\left\{ \text{Substitution: } u_2(t_2) := \frac{t_2 - \rho z_1(S_T)}{\sqrt{1-\rho^2}} \right\} \\
&= \exp(-rT) \int_K^\infty \frac{1}{\sqrt{2\pi T}\sigma_1} \exp\left(-\frac{z_1(S_T)^2}{2} \right) \\
&\quad \cdot \left[1 - \int_{-\infty}^{\frac{z_2(S_T) - \rho z_1(S_T)}{\sqrt{1-\rho^2}}} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u_2^2}{2} \right) \mathrm{d}u_2 \right] \mathrm{d}S_T \\
&= \exp(-rT) \int_K^\infty \int_{\frac{z_2(S_T) - \rho z_1(S_T)}{\sqrt{1-\rho^2}}}^\infty \frac{1}{2\pi\sqrt{T}\sigma_1} \exp\left(-\frac{1}{2} \left(z_1(S_T)^2 + u_2^2 \right) \right) \mathrm{d}u_2 \mathrm{d}S_T \\
&\left\{ \text{Substitution: } x_1(S_T) := z_1(S_T) - \sigma_1\sqrt{T} \right\} \\
&= S_{0,1} \int_{z_1(K) - \sigma_1\sqrt{T}}^\infty \int_{\frac{\ln\left(\frac{S_{0,1}}{S_{0,2}}\right) + \frac{\sigma}{2}T}{\sigma_2\sqrt{(1-\rho^2)T}} + x_1 \frac{\sigma_1 - \rho\sigma_2}{\sigma_2\sqrt{1-\rho^2}}}^\infty \frac{1}{2\pi} \exp\left(-\frac{1}{2} \left(x_1^2 + u_2^2 \right) \right) \mathrm{d}u_2 \mathrm{d}x_1 \\
&\left\{ \text{Substitution: } v_2(u_2) := u_2 - x_1 \frac{\sigma_1 - \rho\sigma_2}{\sigma_2\sqrt{1-\rho^2}} \right\} \\
&= S_{0,1} \int_{z_1(K) - \sigma_1\sqrt{T}}^\infty \int_{\frac{\ln\left(\frac{S_{0,1}}{S_{0,2}}\right) + \frac{\sigma}{2}T}{\sigma_2\sqrt{(1-\rho^2)T}}}^\infty \frac{1}{2\pi} \exp\left(-\frac{1}{2} \left(x_1^2 + \left(v_2 + x_1 \frac{\sigma_1 - \rho\sigma_2}{\sigma_2\sqrt{1-\rho^2}} \right)^2 \right) \right) \mathrm{d}v_2 \mathrm{d}x_1
\end{aligned}
$$

$$
\begin{aligned}
&\left\{\text{Substitution: } w_2(v_2) := \frac{\sigma_2\sqrt{1-\rho 2}}{\sigma} v_2\right\}\\
&= S_{0,1} \int\limits_{z_1(K)-\sigma_1\sqrt{T}}^{\infty} \int\limits_{\frac{\ln\left(\frac{S_{0,1}}{S_{0,2}}\right)+\frac{\sigma}{2}T}{\sigma\sqrt{T}}}^{\infty} \frac{1}{2\pi}\frac{\sigma}{\sigma_2\sqrt{1-\rho^2}}\\
&\qquad \cdot \exp\left(-\frac{1}{2}\frac{\sigma^2}{\sigma_2^2(1-\rho^2)}\left(x_1^2 - 2x_1 w_2\left(-\frac{\sigma_1-\sigma_2\rho}{\sigma}\right) + w_2^2\right)\right) \mathrm{d}w_2\mathrm{d}x_1\\
&= S_{0,1} \int\limits_{-y_1}^{\infty}\int\limits_{\delta}^{\infty} \frac{1}{2\pi\sqrt{1-(-\rho_1)^2}} \exp\left(-\frac{x_1^2 - 2x_1 w_2(-\rho_1) + w_2^2}{2(1-(-\rho_1)^2)}\right) \mathrm{d}w_2\mathrm{d}x_1\\
&\{\text{Substitution: } x\left(w_2\right) := -w_2,\ y\left(x_1\right) := -x_1\}\\
&= S_{0,1}\Phi^2_{-\rho_1}(y_1, -\delta)
\end{aligned}
$$

Analogically to the calculation above, we can show the equality of $S_{0,2}\Phi^2_{-\rho_2}(y_2, \delta - \sigma\sqrt{T})$ and I_2. Next, we show the equality of I_3 and $K\exp(-rT)\Phi^2_\rho(y_1 - \sigma_1\sqrt{T}, y_2 - \sigma_2\sqrt{T})$:

$$
\begin{aligned}
I_3 = K\exp(-rT)&\left(\int\limits_K^\infty \frac{\partial}{\partial S_T}\Phi\left(z_1(S_T)\right)\mathrm{d}S_T + \int\limits_K^\infty \frac{\partial}{\partial S_T}\Phi\left(z_2(S_T)\right)\mathrm{d}S_T\right.\\
&\quad \left. - \int\limits_K^\infty \frac{\partial}{\partial S_T}\Phi^2_\rho\left(z_1(S_T), z_2(S_T)\right)\mathrm{d}S_T\right)\\
=:&\ K\exp(-rT)\left(I_{3,1} + I_{3,2} - I_{3,3}\right)
\end{aligned}
$$

Considering $I_{3,1}$ in detail we get:

$$
\begin{aligned}
I_{3,1} &= \frac{\partial}{\partial S_T}\int\limits_K^\infty \Phi\left(z_1(S_T)\right)\mathrm{d}S_T = \Phi\left(z_1(\infty)\right) - \Phi\left(z_1(K)\right) = \int\limits_{z_1(K)}^\infty \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{u_1^2}{2}\right)\mathrm{d}u_1\\
&\{\text{Substitution: } t_1\left(u_1\right) := -u_1\}\\
&= \int\limits_{-\infty}^{-z_1(K)} \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{t_1^2}{2}\right)\mathrm{d}t_1\\
&= \int\limits_{-\infty}^{-z_1(K)} \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{t_1^2}{2}\right)\int\limits_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi(1-\rho^2)}}\exp\left(-\frac{\left(t_2-\rho t_1\right)^2}{2(1-\rho^2)}\right)\mathrm{d}t_2\mathrm{d}t_1\\
&= \int\limits_{-\infty}^{-z_1(K)}\int\limits_{-\infty}^{\infty}\frac{1}{2\pi\sqrt{(1-\rho^2)}}\exp\left(-\frac{t_1^2 - 2\rho t_1 t_2 + t_2^2}{2(1-\rho^2)}\right)\mathrm{d}t_2\mathrm{d}t_1\\
&= \Phi^2_\rho\left(-z_1(K), \infty\right)
\end{aligned}
$$

Analogically we can show equality $I_{3,2} = \Phi_\rho^2\left(\infty, -z_2(K)\right)$. A detailed consideration of $I_{3,3}$ leads to:

$$
\begin{aligned}
I_{3,3} &= \frac{\partial}{\partial S_T} \int\limits_K^\infty \Phi_\rho^2\left(z_1(S_T), z_2(S_T)\right) \mathrm{d}S_T = 1 - \Phi_\rho^2\left(z_1(K), z_2(K)\right) \\
&= 1 - \int\limits_{-\infty}^{z_1(K)} \int\limits_{-\infty}^{z_2(K)} \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{u_1^2 - 2\rho u_1 u_2 + u_2^2}{2(1-\rho^2)}\right) \mathrm{d}u_2 \mathrm{d}u_1 \\
&= \int\limits_{-\infty}^{\infty} \int\limits_{z_2(K)}^{\infty} \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{u_1^2 - 2\rho u_1 u_2 + u_2^2}{2(1-\rho^2)}\right) \mathrm{d}u_2 \mathrm{d}u_1 \\
&\quad + \int\limits_{z_1(K)}^{\infty} \int\limits_{-\infty}^{\infty} \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{u_1^2 - 2\rho u_1 u_2 + u_2^2}{2(1-\rho^2)}\right) \mathrm{d}u_2 \mathrm{d}u_1 \\
&\quad - \int\limits_{z_1(K)}^{\infty} \int\limits_{z_2(K)}^{\infty} \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{u_1^2 - 2\rho u_1 u_2 + u_2^2}{2(1-\rho^2)}\right) \mathrm{d}u_2 \mathrm{d}u_1 \\
&\quad \left\{\text{Substitution: } t_1\left(u_1\right) := -u_1,\ t_2\left(u_2\right) := -u_2\right\} \\
&= \int\limits_{-\infty}^{\infty} \int\limits_{-\infty}^{-z_2(K)} \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{t_1^2 - 2\rho t_1 t_2 + t_2^2}{2(1-\rho^2)}\right) \mathrm{d}t_2 \mathrm{d}t_1 \\
&\quad + \int\limits_{-\infty}^{-z_1(K)} \int\limits_{-\infty}^{\infty} \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{t_1^2 - 2\rho t_1 t_2 + t_2^2}{2(1-\rho^2)}\right) \mathrm{d}t_2 \mathrm{d}t_1 \\
&\quad - \int\limits_{-\infty}^{-z_1(K)} \int\limits_{-\infty}^{-z_2(K)} \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{t_1^2 - 2\rho t_1 t_2 + t_2^2}{2(1-\rho^2)}\right) \mathrm{d}t_2 \mathrm{d}t_1 \\
&= \Phi_\rho^2\left(\infty, -z_2(K)\right) + \Phi_\rho^2\left(-z_1(K), \infty\right) - \Phi_\rho^2\left(-z_1(K), -z_2(K)\right)
\end{aligned}
$$

If we sum up the terms $I_{3,i}$, we will get

$$
I_3 = I_{3,1} + I_{3,2} - I_{3,3} = \Phi_\rho^2\left(-z_1(K), -z_2(K)\right) = \Phi_\rho^2\left(y_1 - \sigma_1\sqrt{T}, y_2 - \sigma_2\sqrt{T}\right)
$$

and consequently equality

$$
\begin{aligned}
I_1 + I_2 - I_3 &= S_{0,1}\Phi_{-\rho_1}^2\left(y_1, -\delta\right) + S_{0,2}\Phi_{-\rho_2}^2\left(y_2, \delta - \sigma\sqrt{T}\right) \\
&\quad - K\exp(-rT)\Phi_\rho^2\left(y_1 - \sigma_1\sqrt{T}, y_2 - \sigma_2\sqrt{T}\right) \overset{\text{eq. (6.58)}}{=} V_{\text{min}}^{\text{call}}
\end{aligned}
$$

is proven. □

As we have shown in the proposition above, it is possible to evaluate d–dimensional max/min options according to the multi–dimensional Black–Scholes model. Additionally, we can model other dependence structures by choosing any different copula than the Gaussian one. Alternatively, we can evaluate m–th min/max options, which depend on the m–th smallest/highest stock price at $t = T$.

We finish this section by calculating a practical example.

Example 6.8 (Minimum put option on 13 different DAX® corporations)**.** Let $V_{\min}^{\text{put}}$ be the value of a 13–dimensional minimum put option on 13 different DAX® corporations $S_1, \dots, S_{13}$ and let $\zeta_1, \dots, \zeta_{13}$ denote the random variables, which represent the stock prices of these corporations at $t = T = 1$. The initial stock prices and the corresponding volatilities are shown in Table 6.7 and were imported from [com11]. The deterministic growth rates are only of theoretical interest and are denoted by $\mu_1, \dots, \mu_{13}$.

i	Symbol	$S_{0,i}$	β_i	σ_i		i	Symbol	$S_{0,i}$	β_i	σ_i
1	ADS[13]	53.50	0.9162	0.2399		8	HEI[20]	49.07	1.4585	0.3493
2	BAYN[14]	57.98	0.8898	0.2141		9	HEN3[21]	47.90	0.6902	0.2275
3	BEI[15]	43.90	0.5332	0.1951		10	SDF[22]	53.04	0.8689	0.2571
4	DAI[16]	50.65	1.3138	0.3103		11	MEO[23]	47.87	0.9405	0.2465
5	DBK[17]	42.57	1.1991	0.2688		12	RWE[24]	43.52	0.7432	0.2008
6	DB1[18]	55.55	1.0002	0.2479		13	SAP[25]	42.97	0.5879	0.1763
7	FME[19]	50.18	0.3417	0.1698		D	DAXK[26]	—	1.0000	0.1679

Table 6.7.: Daily closing prices at 05/05/2011, beta factor loadings and volatilities of different DAX® corporations and of the DAX® stock index respectively.

The option value $V_{\min}^{\text{put}}$ with respect to strike K is given as

$$V_{\min}^{\text{put}} = \int_0^K \widetilde{\Lambda}^{\text{put}}\left(S_T\right) \cdot f_{\zeta_{\iota_1}}\left(S_T\right) \mathrm{d}S_T = \exp(-rT) \int_0^K F_{\zeta_{\iota_1}}\left(S_T\right) \mathrm{d}S_T,$$

in which $\widetilde{\Lambda}^{\text{put}}\left(S_T\right) = \exp(-rT)\left(K - S_T\right)$ holds for any $S_T \in \overline{\mathbb{R}}_{\geq 0}$.

Because each of the 13 corporations is listed in the DAX® stock index, it is reasonable to model their correlation by using a one–factor model, in which the DAX®

[13] Adidas AG, Adi-Dassler-Straße 1, 91074 Herzogenaurach, Germany

[14] Bayer AG, Gebäude W 11, 51368 Leverkusen, Germany

[15] Beiersdorf AG, Unnastraße 48, 20245 Hamburg, Germany

[16] Daimler AG, Mercedesstraße 137, 70327 Stuttgart, Germany

[17] Deutsche Bank AG, Taunusanlage 12, 60325 Frankfurt am Main, Germany

[18] Deutsche Börse AG, 60485 Frankfurt am Main, Germany

[19] Fresenius Medical Care AG & Co. KGaA, 61352 Bad Homburg, Germany

[20] HeidelbergCement AG, Berliner Straße 6, 69120 Heidelberg, Germany

[21] Henkel AG & Co. KGaA, Henkelstraße 67, 40589 Düsseldorf, Germany

[22] K+S Aktiengesellschaft, Bertha-von-Suttner-Straße 7, 34131 Kassel, Germany

[23] Metro AG, Schlüterstrasse 1, 40235 Düsseldorf, Germany

[24] RWE AG, Opernplatz 1, 45128 Essen, Germany

[25] SAP AG & Co. KG, Hasso-Plattner-Ring 7, 69190 Walldorf, Germany

[26] DAX® – German stock index (price index)

stock index (with deterministic growth rate μ_{D} and volatility σ_{D}) represents the actual factor.

For this, let $\widetilde{\zeta}_1 := \ln\left(\zeta_1/S_{0,1}\right), \ldots, \widetilde{\zeta}_{13} := \ln\left(\zeta_{13}/S_{0,13}\right)$ denote the random variables, which represent the logarithmic stock returns of the corresponding stocks. Following [Sey12, Section 1.8], random variables $\widetilde{\zeta}_i,\ i \in \{1, \ldots, 13\}$, are distributed according to the Gaussian distribution with mean $\widetilde{\mu}_i$ and variance σ_i^2

$$\widetilde{\zeta}_i \sim \mathcal{N}\Big(\underbrace{\mu_i - \sigma_i^2/2}_{=:\widetilde{\mu}_i}, \sigma_i^2\Big) \tag{6.60}$$

and are modeled by using the one–factor model

$$\widetilde{\zeta}_i = \beta_i \cdot \widetilde{\zeta}_{\mathrm{D}} + \vartheta_i \cdot \epsilon_i. \tag{6.61}$$

At this, $\widetilde{\zeta}_{\mathrm{D}}$ denotes the logarithmic index return of the DAX® stock index, which is also distributed according to the Gaussian distribution

$$\widetilde{\zeta}_{\mathrm{D}} \sim \mathcal{N}\Big(\underbrace{\mu_{\mathrm{D}} - \sigma_{\mathrm{D}}^2/2}_{=:\widetilde{\mu}_{\mathrm{D}}}, \sigma_{\mathrm{D}}^2\Big).$$

Furthermore, $\epsilon_i \sim \Phi$ denotes a random variable, which meets $\forall j \in \{1, \ldots, 13\}\ j \neq i : \mathrm{Cov}(\epsilon_i, \epsilon_j) = 0$ as well as $\mathrm{Cov}(\epsilon_i, \widetilde{\zeta}_{\mathrm{D}}) = 0$.

To fit the distribution from equation (6.60) to the factor model of equation (6.61) with respect to the first two moments, we have to show

$$\beta_i = \frac{\widetilde{\mu}_i}{\widetilde{\mu}_{\mathrm{D}}} \tag{6.62}$$

and

$$\vartheta_i^2 = \sigma_i^2 - \beta_i^2 \cdot \sigma_{\mathrm{D}}^2 \tag{6.63}$$

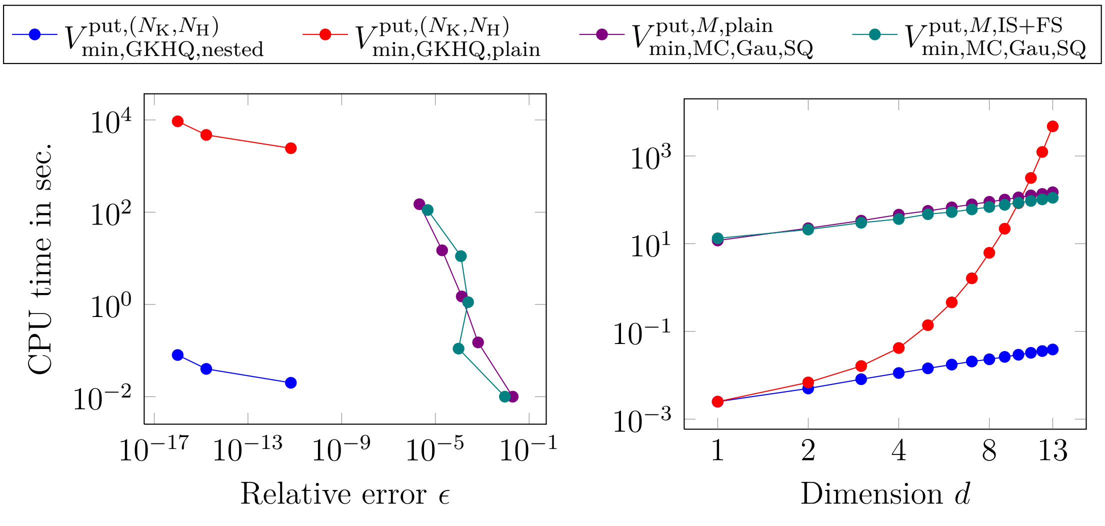
However, we avoid the restriction from equation (6.62) by estimating β_i directly by market data (cp. Table 6.7). Then, we can define ϑ_i according to equation (6.63) and can consequently satisfy factor model (6.61) with $i \in \{1, \ldots, 13\}$.

This fact also offers significant advantages for estimating the correlation (cp. Subsection 3.1.1), because the correlation coefficient $\rho_{i,j},\ i, j \in \{1, \ldots, d\},\ i \neq j$, is given by

$$\rho_{i,j} = \frac{\beta_i \beta_j \sigma_{\mathrm{D}}^2}{\sigma_i \sigma_j} = \psi_i \cdot \psi_j$$

and thus we can write the resulting correlation matrix in factor structure as

$$\Sigma = \Sigma_1^{\mathrm{F}} = \boldsymbol{\psi} \cdot \boldsymbol{\psi}^{\mathrm{tr}} + \mathrm{diag}_{13}\left(1 - \psi_1^2, \ldots, 1 - \psi_{13}^2\right).$$



(a) $d = 13$

(b) $N_\text{K} = 33$, $N_\text{H} = 256$, $M = 10^7$

Figure 6.13.: CPU times and relative errors ϵ of different quadrature and Monte–Carlo methods for pricing a d–dimensional minimum put option, $d \in \{1, \dots, 13\}$.

In the current example we furthermore assume that no dividends are paid and that a constant risk free interest rate of $r_t \equiv 0.03$ is observed within $[0, T]$. Additionally, strike K is set to $K = 40$ and the desired copula is the Gaussian one.

In Figure 6.13a we apply quadrature methods $V^{(N_\text{K},N_\text{H})}_{m,\text{GKHQ,nested}}$ and $V^{(N_\text{K},N_\text{H})}_{m,\text{GKHQ,plain}}$ as well as Monte–Carlo methods $V^{M,\text{plain}}_{m,\text{MC,Gau,SQ}}$ and $V^{M,\text{IS+FS}}_{m,\text{MC,Gau,SQ}}$ to the context of option pricing. The resulting convergence (cp. Figure 6.13a) is similar to the results we obtained in previous examples.

In Figure 6.13b we additionally plot the required CPU time against the variable dimension $d \in \{1, \dots 13\}$. At this, we use parameters $N_\text{K} = 33$, $N_\text{H} = 256$ and $M = 10^7$, which cause relative errors of $\mathcal{O}(10^{-15})$ within the quadrature methods and $\mathcal{O}(10^{-6})$ within the Monte–Carlo methods.

6.4. Calculating sensitivities (“Greeks”)

Besides the evaluation of an mBDS, the calculation of its sensitivities (also known as “Greeks”) is of major importance in practice. Especially in risk management the sensitivities of the value to small changes in certain parameters are essential. Questions like “How will the value of an mBDS change if the underlying hazard rate h_i changes?” or “How much equity has to exist in order to compensate a change in the underlying risk free interest rate of 0.5%?” are daily occurrences within this context.

Another aspect illustrating the importance of an accurate and efficient calculation of sensitivities is its “non–tradable” character. While the values of certain credit derivatives are often calibrated by market data or are even read off the market, this is not possible within the context of sensitivities.

A widely used example for a sensitivity calculation is the famous *delta–hedge*, which is known from option price theory, cp. [Sey12, Appendix A4].

In the following we develop different quadrature and Monte–Carlo methods for calculating certain sensitivities. These methods are based on the results of Sections 6.1 and 6.2.

Let V_m denote the value of an mBDS, which is given by (cp. Theorem 6.1)

$$V_m = \int\limits_{\mathbb{R}^d} \Lambda_m(\boldsymbol{t})\,\mathrm{d}F_{\boldsymbol{\tau}}(\boldsymbol{t}) = \widetilde{\Lambda}_\infty - \int\limits_0^T \widetilde{\Lambda}'(t) \cdot F_{\tau_{\iota_m}}(t)\mathrm{d}t. \tag{6.64}$$

Here, the first equality is used to establish Monte–Carlo methods and the second one for quadrature methods. Additionally, the following theorem is of high importance for this section.

Theorem 6.2 (Leibniz–rule)**.** *Let $J_1 := [l_1, u_1] \subseteq \mathbb{R}$ and $J_2 := [l_2, u_2] \subseteq \mathbb{R}$ be compact intervals and let $g : J_1 \times J_2 \mapsto \mathbb{R}$ be any mapping. If $g(x, y)$ is continuously differentiable with respect to x for any fixed $y \in J_2$,*

$$\frac{\mathrm{d}}{\mathrm{d}x}\int\limits_{l_2}^{u_2} g(x,y)\mathrm{d}y = \int\limits_{l_2}^{u_2} \frac{\partial}{\partial x} g(x,y)\mathrm{d}y$$

will hold.

Proof. We refer to [Heu83]. □

In [Gla04, Section 7.2] the Leibniz–rule was first applied to the calculation of sensitivities within a financial background.

6.4.1. Sensitivity with respect to the interest rate

In this subsection we establish a technique for calculating the sensitivity of V_m with respect to the underlying interest rate r_t, i.e. we have to calculate $\partial V_m / \partial r_t$. By using equation (6.64) and Theorem 6.2 we can easily simplify this quotient to

$$\frac{\partial V_m}{\partial r_t} = \int_{\mathbb{R}^d} \frac{\partial \Lambda_m}{\partial r_t}(\boldsymbol{t})\, \mathrm{d}F_{\boldsymbol{\tau}}(\boldsymbol{t}) = \frac{\partial \widetilde{\Lambda}_\infty}{\partial r_t} - \int_0^T \frac{\widetilde{\Lambda}'}{\partial r_t}(t) \cdot F_{\tau_{\iota_m}}(t)\mathrm{d}t, \tag{6.65}$$

because neither the joint CDF $F_{\boldsymbol{\tau}}$ nor the CDF of the m–th smallest default time $F_{\tau_{\iota_m}}$ depends on the interest rate r_t.

In order to prove transformation (6.65) we just have to show that the payoff functions Λ_m, $\widetilde{\Lambda}$ and $\widetilde{\Lambda}'$ are continuously differentiable with respect to r_t. This proof has to be done context–dependent, but in most cases this restriction is satisfied.

Example 6.9 (Interest–Greek of *SWN Synthia 2009*)**.** Let $V_{\bar{t}}^{\mathrm{Syn}}$ be the value of the Credit Linked Note *SWN Synthia 2009* (cp. Subsection 6.1.5). Then, payoff $\Lambda_{\bar{t}}^{\mathrm{Syn}}$ is given by

$$\begin{aligned}
&\Lambda_{\bar{t}}^{\mathrm{Syn}}\left(\Delta t_{\mathrm{B}}, \Delta t_{\mathrm{D}}, \Delta t_{\mathrm{T}}\right) := \widetilde{\Lambda}_{\bar{t}}^{\mathrm{Syn}}\left(\Delta t_{\min}\right) \\
&:= \begin{cases}
\exp\left(-r_{\Delta t_{\min}} \Delta t_{\min}\right) B_{\min} \cdot N, & \text{if } t_{\mathrm{iss}} < t_{\min} \le t_{z_1} \\
\sum\limits_{i=1}^{j} \exp\left(-r_{\Delta t_{z_i}} \Delta t_{z_i}\right) r_f N \cdot \mathbf{1}_{\{t_{z_i} > \bar{t}\}} & \\
\qquad\qquad + \exp\left(-r_{\Delta t_{\min}} \Delta t_{\min}\right) B_{\min} \cdot N, & \text{if } t_{z_j} < t_{\min} \le t_{z_{j+1}},\ j \in \{1, 2\}\ , \\
\sum\limits_{i=1}^{3} \exp\left(-r_{\Delta t_{z_i}} \Delta t_{z_i}\right) r_f N \cdot \mathbf{1}_{\{t_{z_i} > \bar{t}\}} & \\
\qquad\qquad + \exp\left(-r_{\Delta T} \Delta T\right) N, & \text{else}
\end{cases}
\end{aligned}$$

in which the notation from Subsection 6.1.5 remains valid. In the following we assume a constant interest rate $r_t \equiv r$. Then, derivatives $\frac{\partial \Lambda_{\bar{t}}^{\mathrm{Syn}}}{\partial r}$, $\frac{\partial \widetilde{\Lambda}_{\bar{t}}^{\mathrm{Syn}}}{\partial r}$ and $\frac{\partial^2 \widetilde{\Lambda}_{\bar{t}}^{\mathrm{Syn}}}{\partial t_{\min} \partial r}$ are calculated easily and sensitivity $\frac{\partial V_{\bar{t}}^{\mathrm{Syn}}}{\partial r}$ is calculated with the same effort as for $V_{\bar{t}}^{\mathrm{Syn}}$.

In Table 6.8 we show a comparison of three different methods for calculating this sensitivity (regarding the Gaussian copula). On the one hand we calculate $\frac{\partial V_{\bar{t}}^{\mathrm{Syn}}}{\partial r}$ directly by applying equation (6.65) and a suitable quadrature method, i.e. we have to calculate (cp. equation (6.47))

$$\begin{aligned}\frac{\partial V_{\bar{t}}^{\text{Syn}}}{\partial r} &= \left(\lim_{s\nearrow\Delta t_{z_2}} \frac{\partial \widetilde{\Lambda}_{\bar{t}}^{\text{Syn}}}{\partial r}(s) - \lim_{q\searrow\Delta t_{z_2}} \frac{\partial \widetilde{\Lambda}_{\bar{t}}^{\text{Syn}}}{\partial r}(q)\right) F_{\tau_{\iota_1}}(\Delta t_{z_2}) - \int\limits_0^{\Delta t_{z_2}} \frac{\partial \widetilde{\Lambda}_{\bar{t}}^{\text{Syn}\prime}}{\partial r}(t) F_{\tau_{\iota_1}}(t)\mathrm{d}t \\ &+ \left(\lim_{s\nearrow\Delta T} \frac{\partial \widetilde{\Lambda}_{\bar{t}}^{\text{Syn}}}{\partial r}(s) - \lim_{q\searrow\Delta T} \frac{\partial \widetilde{\Lambda}_{\bar{t}}^{\text{Syn}}}{\partial r}(q)\right) F_{\tau_{\iota_1}}(\Delta T) - \int\limits_{\Delta t_{z_2}}^{\Delta T} \frac{\partial \widetilde{\Lambda}_{\bar{t}}^{\text{Syn}\prime}}{\partial r}(t) F_{\tau_{\iota_1}}(t)\mathrm{d}t \\ &+ \underbrace{\lim_{s\nearrow\infty} \frac{\partial \widetilde{\Lambda}_{\bar{t}}^{\text{Syn}}}{\partial r}(s)}_{=\frac{\partial \widetilde{\Lambda}_{\infty}^{\text{Syn}}}{\partial r}}.\end{aligned}$$

The resulting integrals are approximated analogically to $V_{\bar{t},\text{GKHQ,nested}}^{\text{Syn},(N_\text{K},N_\text{H})}$ (cp. Section 6.1.5). The corresponding approximation of $\frac{\partial V_{\bar{t}}^{\text{Syn}}}{\partial r}$ is denoted by "∂r–quad" and can be found in Table 6.8.

On the other hand we approximate equation (6.65) by means of a Monte–Carlo simulation. This approximation is based on the method $V_{m,\text{MC,Gau,SQ}}^{M,\text{plain}}$ and is denoted by "∂r–MC".

Finally, we approximate sensitivity $\frac{\partial V_{\bar{t}}^{\text{Syn}}}{\partial r}$ by means of a finite differences approach, which uses the estimator

$$\frac{\partial V_{\bar{t}}^{\text{Syn}}}{\partial r} = \lim_{h\searrow 0} \frac{V_{\bar{t},\text{GKHQ,nested}}^{\text{Syn},(N_\text{K},N_\text{H})}(r+h) - V_{\bar{t},\text{GKHQ,nested}}^{\text{Syn},(N_\text{K},N_\text{H})}(r-h)}{2h}.$$

The resulting approximation is denoted by "∂r–FD". In the applied quadrature methods we fix the number of nodes $N_\text{H} = 256$ within the GHQ and vary the number of nodes $N_\text{K} = 2^j,\ j \in \{7,8,9\}$, within the GKQ. Furthermore, we also vary the number of simulations $M = 10^j,\ j \in \{7,8,9\}$, within the Monte–Carlo simulation.

In Table 6.8 it is easy to see that methods ∂r–quad and ∂r–FD offer a similar convergence. However, we have to note that method ∂r–FD needs a suitable value for h in order to deliver good results. Furthermore, method ∂r–FD needs twice of the CPU time ∂r–quad requires. Although the ∂r–MC only offers a slow convergence, it has a more robust character than the ∂r–FD method, because it does not depend on h.

Remark 6.12 (Non–constant interest rate). If the interest rate r_t is not assumed to be constant, but is calibrated by market data (analogically to Subsection 6.1.5), we will be able to calculate sensitivities with respect to single nodes of the interest rate analogically (cp. for example [Sch10, Section 9]).

j	h	-(∂r–quad) · e-2	Sec.	-(∂r–MC) · e-2	Sec.	-(∂r–FD) · e-2	Sec.
7	1e-4	**1.7248379**084	0.16	**1.7248**540773	35.42	**1.7248379**178	0.32
	1e-5					**1.7248379**085	0.32
	1e-6					**1.7248379**084	0.33
	1e-7					**1.7248379**888	0.32
8	1e-4	**1.72483784**22	0.32	**1.72483**89411	356.3	**1.7248378**516	0.62
	1e-5					**1.72483784**23	0.62
	1e-6					**1.72483784**22	0.63
	1e-7					**1.72483784**27	0.62
9	1e-4	**1.72483784**84	0.61	**1.72483**86520	3555	**1.7248378**578	1.26
	1e-5					**1.72483784**85	1.27
	1e-6					**1.72483784**84	1.27
	1e-7					**1.72483784**91	1.26

Table 6.8.: Comparison of different methods for calculating $-\frac{\partial V_{\bar{t}}^{\mathrm{Syn}}}{\partial r} \cdot 10^{-2}$ regarding convergence and CPU time (in sec.).

6.4.2. Sensitivity with respect to the maturity

The calculation of the sensitivity with respect to maturity T represents a special case of the previous Subsection 6.4.1. If we use the joint CDF $F_{\boldsymbol{\tau}}$ and payoff Λ_m for a Monte–Carlo simulation, we will be able to proceed completely analogically to the subsection above. However, if we use the distribution of the m–th smallest default time, we will additionally have to calculate a derivative with respect to a limit of integration.

Thus, the sensitivity with respect to maturity T is given by

$$\frac{\partial V_m}{\partial T} = \int_{\mathbb{R}^d} \frac{\partial \Lambda_m}{\partial T}(\boldsymbol{t})\,\mathrm{d}F_{\boldsymbol{\tau}}(\boldsymbol{t}) = -\widetilde{\Lambda}'(T) F_{\tau_{\iota m}}(T). \tag{6.66}$$

Example 6.10 (Maturity–Greek of Li's example)**.** Let V_m^{Li} be the value of the mBDS from Subsection 6.2.3. Then, payoff functions Λ_m^{Li} and $\widetilde{\Lambda}^{\mathrm{Li}}$ are given by

$$\Lambda_m^{\mathrm{Li}}(\boldsymbol{t}) = \widetilde{\Lambda}^{\mathrm{Li}}\left(t_{\tau_{\iota m}}\right) := \exp(-rt_{\iota m}) \cdot \mathbf{1}_{\{t_{\iota m} \leq T\}}$$

for any $\boldsymbol{t} \in \overline{\mathbb{R}}^d_{\geq 0}$. Obviously, the left hand side of equation (6.66) is not applicable here, because derivative $\frac{\partial \Lambda_m^{\mathrm{Li}}}{\partial T}$ does not exist. This means that it is not possible to calculate the desired sensitivity directly by a Monte–Carlo simulation.

Nevertheless, this sensitivity is easily calculated by using a quadrature method.

N_{H}	h	$\widetilde{\Lambda}^{\mathrm{Li}}(T)\frac{F_{\tau_{\iota_m}}(T+h)-F_{\tau_{\iota_m}}(T-h)}{2h}$	Sec.	$\frac{V_m^{(\mathrm{Li})}(T+h)-V_m^{(\mathrm{Li})}(T-h)}{2h}$	Sec.
64	1e-4	**0.0015664872**518878	1e-3	**0.0015664872**516409	0.61
	1e-5	**0.001566487248**8970	1e-3	**0.001566487248**8952	0.62
	1e-6	**0.001566487248**6130	1e-3	**0.001566487248**6187	0.61
	1e-7	**0.0015664872**542947	1e-3	**0.0015664872**540397	0.61
128	1e-4	**0.0015664872**518878	2e-3	**0.0015664872**516404	1.20
	1e-5	**0.001566487248**8707	2e-3	**0.001566487248**8681	1.20
	1e-6	**0.001566487248**5604	2e-3	**0.001566487248**5645	1.20
	1e-7	**0.0015664872**506121	3e-3	**0.0015664872**502450	1.21
256	1e-4	**0.0015664872**518862	5e-3	**0.0015664872**516388	2.33
	1e-5	**0.001566487248**9023	4e-3	**0.001566487248**9006	2.34
	1e-6	**0.001566487248**5078	4e-3	**0.001566487248**5103	2.32
	1e-7	**0.0015664872**511381	4e-3	**0.0015664872**513292	2.35

Table 6.9.: Comparison of different methods for calculating $\frac{\partial V_m^{\mathrm{Li}}}{\partial T}$ regarding convergence and CPU time (in sec.).

Considering equation (6.53), sensitivity $\frac{\partial V_m^{\mathrm{Li}}}{\partial T}$ is given by

$$\frac{\partial V_m^{\mathrm{Li}}}{\partial T} = \widetilde{\Lambda}^{\mathrm{Li}}(T) f_{\tau_{\iota_m}}(T). \tag{6.67}$$

Hence, we have to evaluate the PDF $f_{\tau_{\iota_m}}$, which is very expensive. Due to the fact that we need this PDF only at the argument $t = T$, though, it makes sense to approximate it by means of a difference quotient.

Let us assume $d = 10$, $m = 5$, $r = 0.03$ and $T = 1$. Furthermore, let $\boldsymbol{h}$ and $\boldsymbol{\psi}$ be defined according to equations (6.54) and (6.55). For the approximation in (6.67) we first have to approximate $f_{\tau_{\iota_m}}(T)$ by

$$f_{\tau_{\iota_m}}(T) \approx \frac{F_{\tau_{\iota_m}}(T+h) - F_{\tau_{\iota_m}}(T-h)}{2h}, \tag{6.68}$$

in which $F_{\tau_{\iota_m}}$ is approximated by means of a GHQ using N_{H} nodes (Gaussian copula). Alternatively to this method, we approximate the desired sensitivity by

$$\frac{\partial V_m^{\mathrm{Li}}}{\partial T} = \frac{V_m^{\mathrm{Li}}(T+h) - V_m^{\mathrm{Li}}(T-h)}{2h}, \tag{6.69}$$

in which V_m^{Li} is approximated by $V_{m,\mathrm{GKHQ,nested}}^{\mathrm{Li},(N_{\mathrm{K}},N_{\mathrm{H}})}$. Here, the number of nodes within the applied GKQ is fixed at $N_{\mathrm{K}} = 513$ and the number of nodes N_{H} within the GHQ remains variable for both estimators.

Table 6.9 shows that the convergence of both estimators is similar. However, a comparison of the underlying CPU time offers an effort reduction factor of about 600 by using equation (6.67) and estimator (6.68).

6.4.3. Sensitivity with respect to the hazard rate

In this subsection we calculate the sensitivity $\frac{\partial V_m}{\partial h_\delta}$ of an mBDS value V_m with respect to a single hazard rate h_δ, $\delta \in \{1, \ldots, d\}$. This calculation clearly differs from both techniques, which were introduced before. Hazard rate h_δ is not a component of payoffs Λ_m and $\widetilde{\Lambda}$, but represents an important parameter of the joint CDF $F_{\boldsymbol{\tau}}$ and the CDF of the m-smallest default time $F_{\tau_{\iota_m}}$. In the following we assume that $F_{\boldsymbol{\tau}}$ and $F_{\tau_{\iota_m}}$ are continuously differentiable with respect to h_δ.

Then, the desired sensitivity $\frac{\partial V_m}{\partial h_\delta}$ is calculated according to

$$\frac{\partial V_m}{\partial h_\delta} = \int\limits_{\mathbb{R}^d} \Lambda_m(\boldsymbol{t})\, \mathrm{d}\frac{\partial F_{\boldsymbol{\tau}}}{\partial h_\delta}(\boldsymbol{t}) = \widetilde{\Lambda}_\infty - \int\limits_0^T \widetilde{\Lambda}'(t) \cdot \frac{\partial F_{\tau_{\iota_m}}}{\partial h_\delta}(t)\mathrm{d}t. \tag{6.70}$$

For establishing a Monte–Carlo method, which approximates the integral above on the left hand side of (6.70), we have to calculate the joint PDF $f_{\boldsymbol{\tau}}$ first and have to differentiate it with respect to h_δ afterwards. Here, this expensive technique is neglected, we refer to [SH12b]. In the following we consider the right hand side of equation (6.70), i.e. we establish an efficient quadrature method for calculating $\frac{\partial F_{\tau_{\iota_m}}}{\partial h_\delta}$.

For this, we assume the independence of $\tau_1, \ldots, \tau_d$ again (cp. especially Subsection 6.1.2). In this case $F_{\tau_{\iota_m}}(t)$ is given recursively by

$$F_{\tau_{\iota_m}}(t) = \sum_{k=m}^{d} \underbrace{(-1)^{k-m}\binom{k-1}{m-1}}_{=:c_{k,m}} \sum_{\substack{I \subseteq \{1,\ldots,d\} \\ |I|=k}} \prod_{l=1}^{k} F_{\tau_{I_l}}(t) = \sum_{k=m}^{d} c_k \cdot \widetilde{\eta}(k, d-k+1)(t)$$

for any $t \in \overline{\mathbb{R}}_{\geq 0}$, in which $\widetilde{\eta}(k,l)(t)$ is defined recursively as

$$\widetilde{\eta}(k,l)(t) := \begin{cases} F_{\tau_1}(t), & \text{if } k = l = 1 \\ \widetilde{\eta}(k, l-1)(t) + F_{\tau_l}(t), & \text{if } k = 1 \text{ and } l > 1 \\ \widetilde{\eta}(k-1, l)(t) \cdot F_{\tau_k}(t), & \text{if } k > 1 \text{ and } l = 1 \\ \widetilde{\eta}(k-1, l)(t) \cdot F_{\tau_{k+l-1}}(t) + \widetilde{\eta}(k, l-1)(t), & \text{if } k > 1 \text{ and } l > 1 \end{cases} \tag{6.71}$$

for each $k \in \{1, \ldots, d\}$ and $l \in \{1, \ldots, \min\{d-k+1, d-m+1\}\}$ (cp. Lemma 6.4).

Remark 6.13 (Special case $m = 1$). The efficient calculation of $F_{\tau_{\iota_m}}$ for the special case $m = 1$ (cp. Lemma 6.3) is neglected here, because calculating sensitivities in this special case is almost trivial.

For the calculation of $\frac{\partial F_{\tau_{\iota_m}}}{\partial h_\delta}$ we first remark that parameter h_δ only occurs within the CDF F_{τ_δ}. Furthermore we can observe that CDF F_{τ_δ} only occurs for the choice

$k+l-1 \geq \delta$ within $\widetilde{\eta}(k,l)$. Thus, the former recursion (6.71) can be used easily for this new context, we just have to change it for special cases $k+l-1=\delta$ and $k+l-1>\delta$ to the new function

$$\widetilde{\eta}_\delta(k,l)(t) := \begin{cases} \frac{\partial F_{\tau_1}}{\partial h_\delta}(t), & \text{if } k=l=1 \text{ and } k+l-1=\delta \\ \frac{\partial F_{\tau_l}}{\partial h_\delta}(t), & \text{if } k=1\text{ , } l>1 \text{ and } k+l-1\geq\delta \\ \widetilde{\eta}_\delta(k-1,l)(t)\cdot\frac{\partial F_{\tau_k}}{\partial h_\delta}(t), & \text{if } k>1\text{ , } l=1 \text{ and } k+l-1=\delta \\ \widetilde{\eta}_\delta(k-1,l)(t)\cdot\frac{\partial F_{\tau_{k+l-1}}}{\partial h_\delta}(t), & \text{if } k>1\text{ , } l>1 \text{ and } k+l-1=\delta \end{cases} \tag{6.72}$$

Analogically to Subsections 6.1.3 and 6.1.4, we can apply the recursion above to any Archimedean copula or to a Gaussian copula with a factor structure.

Example 6.11 (Hazard–Greek of Li’s example)**.** For a better illustration of this recursion we calculate the sensitivity $\frac{\partial V_m^{\mathrm{Li}}}{\partial h_\delta}$ of the mBDS from Subsection 6.2.3. For this, we use the Gaussian copula and the input parameter $d=10$, $m=3$, $\delta=5$, $r=0.03$ and $T=1$. Additionally, ψ_i and h_i are defined for each $i\in\{1,\ldots,d\}$ according to equations (6.54) and (6.55).

In order to transfer recursion (6.72) to the application of the Gaussian copula we have to adjust the definition of input matrix $\mathbb{X}\in\mathbb{R}^{d\times N_{\mathrm{K}}\times N_{\mathrm{H}}}$ from equation (6.35) to

$$\mathbb{X}' = \left(\mathbb{X}'\right)_{i,j,k} := \begin{cases} \Phi_i\left(t_j^{(\mathrm{K})},x_k^{(\mathrm{H})}\right), & \text{if } i\neq\delta \\ \frac{\partial\Phi_i\left(t_j^{(\mathrm{K})},x_k^{(\mathrm{H})}\right)}{\partial h_\delta}, & \text{if } i=\delta \end{cases}. \tag{6.73}$$

Furthermore, we have to adjust Algorithm 6.3 (`CalculateSum`), in which the underlying recursion is implemented, to Algorithm 6.13 (`CalculateSum2`)

By using Algorithm 6.1, input matrix $\mathbb{X}'$ and Algorithm `CalculateSum2` we can finally calculate the desired sensitivity $\frac{\partial V_m^{\mathrm{Li}}}{\partial h_\delta}$.

In Table 6.10 the results of the sensitivity calculation above are shown and are denoted by “∂h_δ–quad”. Here, we proceed analogically to approximation $V_{m,\mathrm{GKHQ,nested}}^{\mathrm{Li},(N_{\mathrm{K}},N_{\mathrm{H}})}$ and approximate

$$\frac{\partial V_m^{\mathrm{Li}}}{\partial h_\delta} = \widetilde{\Lambda}^{\mathrm{Li}}(T)\cdot\frac{\partial F_{\tau_{\iota_m}}}{\partial h_\delta}(T) - \int\limits_0^T \widetilde{\Lambda}^{\mathrm{Li}'}(t)\cdot\frac{\partial F_{\tau_{\iota_m}}}{\partial h_\delta}(t)\mathrm{d}t \tag{6.74}$$

directly. Additionally, we calculate the finite differences approximation “∂h_δ–FD” as

$$\frac{\partial V_m^{\mathrm{Li}}}{\partial h_\delta} = \lim_{\xi\searrow 0}\frac{V_{m,\mathrm{GKHQ,nested}}^{\mathrm{Li},(N_{\mathrm{K}},N_{\mathrm{H}})}(h_\delta+\xi) - V_{m,\mathrm{GKHQ,nested}}^{\mathrm{Li},(N_{\mathrm{K}},N_{\mathrm{H}})}(h_\delta-\xi)}{2\xi}. \tag{6.75}$$

At this, we vary the number of nodes N_{K} within the GKQ and fix the number of nodes $N_{\mathrm{H}}=256$ within the GHQ.

Algorithm 6.13: `CalculateSum2`(m, $\mathbb{X}'$, i, j)

```
Input  : $m$, $\mathbb{X}'$, $i$, $j$
Output: $\mathcal{X} \in \mathbb{R}$
Global : $d$, $N_{\mathrm{K}}$, $N_{\mathrm{H}}$
Initialize $\mathcal{X} \leftarrow 0$ and $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{d-m+1}$
for $k = 1$ to $k = d$ do
    for $l = 1$ to $l = \min\{d-m+1, d-k+1\}$ do
        if $(k + l - 1 \neq \delta)$ then
            if $(k > 1$ and $l > 1)$ then
                $\mathcal{Y}_l \leftarrow \mathcal{Y}_l \cdot \mathbb{X}'_{k+l-1,i,j} + \mathcal{Y}_{l-1}$
            else
                if $(k > 1$ and $l = 1)$ then
                    $\mathcal{Y}_l \leftarrow \mathcal{Y}_l \cdot \mathbb{X}'_{k,i,j}$
                else
                    if $(k = 1$ and $l > 1)$ then
                        if $(l > \delta)$ then
                            $\mathcal{Y}_l \leftarrow \mathbb{X}'_{\delta,i,j}$
                        else
                            $\mathcal{Y}_l \leftarrow \mathcal{Y}_{l-1} + \mathbb{X}'_{l,i,j}$
                    else
                        $\mathcal{Y}_l \leftarrow \mathbb{X}'_{1,i,j}$
        else
            if $(k > 1$ and $l > 1)$ then
                $\mathcal{Y}_l \leftarrow \mathcal{Y}_l \cdot \mathbb{X}'_{k+l-1,i,j}$
            else
                if $(k > 1$ and $l = 1)$ then
                    $\mathcal{Y}_l \leftarrow \mathcal{Y}_l \cdot \mathbb{X}'_{k,i,j}$
                else
                    if $(k = 1$ and $l > 1)$ then
                        $\mathcal{Y}_l \leftarrow \mathbb{X}'_{l,i,j}$
                    else
                        $\mathcal{Y}_l \leftarrow \mathbb{X}'_{1,i,j}$
    if $k >= m$ then
        $\mathcal{X} \leftarrow \mathcal{X} + c_{k,m}\mathcal{Y}_{d-k+1}$
```

N_{K}	ξ	∂h_δ–quad	Sec.	∂h_δ–FD	Sec.
128	1e-4	**0.12197916529687**54	0.29	**0.1219792**249129194	0.6
	1e-5			**0.121979165**8934807	0.61
	1e-6			**0.12197916529**96849	0.6
	1e-7			**0.121979165**3447877	0.6
256	1e-4	**0.12197916529687**59	0.58	**0.1219792**249129194	1.22
	1e-5			**0.121979165**8933072	1.19
	1e-6			**0.12197916529**96849	1.21
	1e-7			**0.121979165**3447877	1.19
512	1e-4	**0.121979165296876**0	1.17	**0.1219792**249129367	2.37
	1e-5			**0.121979165**8934807	2.38
	1e-6			**0.12197916529**96849	2.37
	1e-7			**0.121979165**3447877	2.37

Table 6.10.: Comparison of different methods for calculating $\frac{\partial V_m}{\partial h_\delta}$ regarding convergence and CPU time (in sec.).

Obviously, method ∂h_δ–quad clearly outperforms the other method concerning convergence and CPU time. Additionally, method ∂h_δ–quad does not need any input parameter ξ and is therefore more robust than method ∂h_δ–FD.

6.4.4. Sensitivity with respect to the correlation

If we want to calculate a sensitivity with respect to a correlation parameter, we will first have to distinguish between the usage of a Gaussian copula or an Archimedean copula.

If we choose the Gaussian copula for modeling the underlying correlation, it will be possible to calculate the sensitivity $\frac{\partial V_m}{\partial \psi_\delta}$ with respect to any factor loading ψ_δ. This is done completely analogically to the previous Subsection 6.4.3, because factor loading ψ_δ only appears within F_{τ_δ}, too. Consequently, we just have to replace the term $\frac{\partial \Phi_i\left(t_j^{(\mathrm{K})}, x_k^{(\mathrm{H})}\right)}{\partial h_\delta}$ in equation (6.73) by $\frac{\partial \Phi_i\left(t_j^{(\mathrm{K})}, x_k^{(\mathrm{H})}\right)}{\partial \psi_\delta}$. The remain of the calculation proceeds analogically.

If we choose any Archimedean copula for modeling the underlying correlation structure, we will only be able to calculate a sensitivity $\frac{\partial V_m}{\partial \theta}$ with respect to parameter θ, which controls the overall correlation within the underlying portfolio. The appearance of parameter θ within $F_{\boldsymbol{\tau}}$ and $F_{\boldsymbol{\tau}_{\iota m}}$ depends crucially on the applied generator

ϕ_θ. Usually, θ appears in nested ways frequently within $F_{\boldsymbol{\tau}}$ and $F_{\tau_{\iota_m}}$, which makes an analytical calculation of $\frac{\partial F_{\boldsymbol{\tau}}}{\partial \theta}$ and $\frac{\partial F_{\tau_{\iota_m}}}{\partial \theta}$ very difficult or even impossible. Due to this, we suggest a calculation according to the finite differences approaches

$$\frac{\partial V_m}{\partial \theta} \approx \frac{V_{m,\text{GKLaQ,nested}}^{(N_\text{K},N_\text{LG},\text{Arc})}(\theta + h) - V_{m,\text{GKLaQ,nested}}^{(N_\text{K},N_\text{LG},\text{Arc})}(\theta - h)}{2h} \tag{6.76}$$

or

$$\frac{\partial V_m}{\partial \theta} \approx \frac{V_{m,\text{GKQ,plain}}^{(N_\text{K},N_\text{LG},\text{Arc})}(\theta + h) - V_{m,\text{GKQ,plain}}^{(N_\text{K},N_\text{LG},\text{Arc})}(\theta - h)}{2h} \tag{6.77}$$

with small h.

Remark 6.14 (Correlation–Greek (Archimedean copula)). If we use an Archimedean copula for modeling the correlation, sensitivity $\frac{\partial V_m}{\partial \theta}$ will not be very meaningful or rather it will not be scaled. Thus, it makes more sense to calculate

$$\frac{\partial V_m}{\partial \tau^\text{K}} = \frac{\partial V_m}{\partial \theta} \cdot \frac{\partial \theta}{\partial \tau^\text{K}},$$

by using equations (6.76) or (6.77) in connection with Table 4.2.

Part III.

Conclusion

7. Summary

In the following we summarize the achievements, which were presented in the previous part. Here, we limit ourselves to the semi–dynamic pricing model from Chapter 6. In Section 6.1 we introduced the pricing of a d–dimensional mBDS as an integral problem of the same dimension. However, considering tradable products in detail we were able to observe that only the m–th smallest default time influences the payments of an mBDS directly. The consideration of each single default time causes superfluous effort.

Therefore, we developed a formula for the distribution function $F_{\tau_{\iota_m}}$ of the m–th smallest default time (cp. Lemma 6.1). With the help of this formula we were able to transform the pricing of a d–dimensional mBDS to an one–dimensional integral problem (cp. Theorem 6.1 and Proposition 6.1). At first sight we were able to break the so–called *curse of dimensionality* here, but the appearances were deceiving. The evaluation of $F_{\tau_{\iota_m}}$ brought a complexity of $\mathcal{O}\left(2^d\right)$ about, which implicates a new exponential dependence on dimension d. Nevertheless, we already reduced the complexity of pricing an mBDS from $\mathcal{O}\left(N^d\right)$ (using a d–dimensional quadrature with N nodes) to $\mathcal{O}\left(N^2 2^d\right)$ in the worst case.

In order to overcome this exponential behavior we first developed two nested recursion schemes (cp. Lemma 6.3 and 6.4) in case of independent default times. Hereby, we were able to reduce the complexity of $F_{\tau_{\iota_m}}$ to at worst $\mathcal{O}\left(d^2\right)$. Later on we were additionally able to apply these recursion schemes to dependent default times, whose dependence is modeled by means of factor models. In Subsections 6.1.3 and 6.1.4 we proved the validity of this result for the application of the Gaussian copula as well as for any Archimedean copula. Finally, the pricing of a d–dimensional mBDS equaled a two–dimensional integral problem, whose integrand shows a complexity of $\mathcal{O}\left(d^2\right)$. Hence, the resulting quadrature method shows a complexity of $\mathcal{O}\left(N^2 d^2\right)$. In Subsection 6.1.5 we applied this method to the pricing of the product *SWN Synthia 2009*, which is traded at the Stuttgart stock exchange. The resulting performance offered a very quick convergence and a massive effort reduction regarding a plain evaluation of $F_{\tau_{\iota_m}}$ or the application of a d–dimensional quadrature. In summary, we reduced the effort of an mBDS evaluation according to

$$\mathcal{O}\left(N^d\right) \Downarrow \mathcal{O}\left(N^2 \cdot 2^d\right) \Downarrow \mathcal{O}\left(N^2 \cdot d^2\right)$$

for any dimension $d \in \mathbb{N}_{\geq 2}$ and any number of nodes $N \in \mathbb{N}$.

Further advantages of the approach above consist in its deterministic nature and its multifunctional applicability. In Section 6.3 we showed that it is easily possible to transfer this new technique to the context of pricing multi–dimensional European maximum or minimum options. In addition we proved that the application of the Gaussian copula within this context implicates a pricing regarding the famous multi–dimensional Black–Scholes pricing model (cp. [BS73]). By using this technique we can consequently expand the analytical formulas for 2–dimensional European maximum or minimum options given in [Stu82, Hau06] to any dimension. We priced a 13–dimensional European minimum put option at the end of Section 6.3 for a better understanding.

Besides the context of option pricing we were furthermore able to transform the pricing via $F_{\tau_{\iota m}}$ to the context of calculating sensitivities in Section 6.4. For each sensitivity we showed a very quick convergence and a huge effort reduction compared to existing methods. Considering the transformation to option pricing again, we can easily calculate corresponding sensitivities like *Delta*, *Vega* or *Theta*.

In Section 6.2 we introduced existing Monte–Carlo methods and developed two new methods, which accelerate the underlying simulation by using importance sampling. For this, we transferred the knowledge of existing methods regarding the Gaussian copula to the context of applying Archimedean copulae. In the derivation of both methods we pointed out that they are applicable to any arbitrary Archimedean copula. Further numerical tests showed that the resulting methods offer a significantly faster convergence and produced a clearly lower variance than a plain simulation. Variance reduction ratios of 10^4 were observed.

On the whole, we developed an innovative quadrature method, which clearly outperforms the existing methods in terms of convergence and complexity. This method is applicable to a Gaussian copula showing factor correlation as well as to any Archimedean copula. Furthermore, by transferring algorithms referring to the Gaussian copula, we established two new Monte–Carlo approaches regarding Archimedean copulae. These methods dominate the existing methods conspicuously in terms of convergence and variance. For closing this summary we briefly suggest which method fits best to certain preconditions.

If the underlying copula is the Gaussian copula, we will distinguish:

1. If the correlation matrix Σ shows a factor correlation and
   a) the underlying basket S is homogeneous regarding recovery rate and nominal value, the best method will be the quadrature method, which applies a nested evaluation of $F_{\tau_{\iota_m}}$, choose: $V_{m,\text{GKHQ,nested}}^{(N_{\text{K}},N_{\text{H}})}$. This method was presented in this thesis for the first time.

   b) the underlying basket S is inhomogeneous regarding recovery rate and nominal value, the best method will be the Monte–Carlo method, which applies importance sampling and exploits the factor correlation, choose: $V_{m,\text{MC,Gau}}^{M,\text{IS+FS}}$. This method was presented in [JK04, CG08].

2. If the correlation matrix Σ does not show a factor correlation, the best method will be the Monte–Carlo method, which only applies importance sampling, choose: $V_{m,\text{MC,Gau}}^{M,\text{IS}}$. This method was presented in [JK04, CG08].

If we choose an Archimedean copula for modeling the joint CDF of default times we will distinguish:

1. If the underlying basket S is homogeneous in terms of recovery rate and nominal value and
   a) the dimension d is moderate (i.e. $d \leq 8$), the best method will be the quadrature method, which applies a plain evaluation of $F_{\tau_{\iota_m}}$, choose: $V_{m,\text{GKQ,plain}}^{(N_{\text{K}},\text{Arc})}$. This method was presented in this thesis for the first time.

   b) the dimension d is huge (i.e. $d > 8$), the best method will be the quadrature method, which applies a nested evaluation of $F_{\tau_{\iota_m}}$, choose: $V_{m,\text{GKLaQ,nested}}^{(N_{\text{K}},N_{\text{LG}},\text{Arc})}$. This method was presented in this thesis for the first time.

2. If the underlying basket S is inhomogeneous regarding recovery rate and nominal value, the best method is the Monte–Carlo method, which exploits the factor correlation of the Archimedean copula, choose: $V_{m,\text{MC,Arc}}^{M,\text{IS+FS}}$. This method was presented in this thesis for the first time.

8. Outlook

In this thesis we mainly considered a semi–dynamic pricing model (cp. [Li00] or [Sch03]), which represents an extension of the static pricing model (cp. Chapter 5). Of course each method presented in Chapter 6 can easily by applied to the static pricing model, which is not presented within this thesis due to its simplicity. The only modification, which has to be applied is a transformation of the underlying MDFs $F_{\tau_1}, \dots, F_{\tau_d}$.

A more complex model is presented in [SS01, Sch03] and is named the dynamic pricing model. In this model the default of obligor S_i implicates a jump in the hazard rate h_j of obligor S_j for each $i, j \in \{1, \dots, d\},\ j \neq i$. Consequently, it is not straightforward to apply a quadrature regarding distribution $F_{\tau_{\iota_m}}$ here. Of course, the application of plain Monte–Carlo methods is possible, but they offer a poor ratio of convergence. Hence, a remaining challenge is the transformation of the presented importance sampling techniques to the context of a dynamic pricing model.

Finally, the efficient pricing of multi–dimensional European options calls for a corresponding modification to American options, which so far is not satisfying and deserves further study.

Part IV.

Appendix

A. Different distributions

Definition A.1 (Gamma distribution)**.** The CDF of the gamma distribution $\mathcal{G}\left(\alpha,\nu\right)$ with $\alpha > 0$ and $\nu > 0$ is defined by its continuous PDF

$$\forall x \in \mathbb{R}_{>0} : f^{\mathcal{G}}_{\alpha,\nu}\left(x\right) := \frac{\alpha^{\nu} x^{\nu-1}}{\Gamma\left(\nu\right)} \exp\left(-\alpha \cdot x\right).$$

At this, $\Gamma(t)$ denotes the gamma function, which is given by (cp. [Fel71, Section II.2])

$$\forall t \in \mathbb{R}_{>0} : \Gamma(t) := \int\limits_{0}^{\infty} x^{t-1} \exp\left(-x\right) \mathrm{d}x.$$

Definition A.2 (Logarithmic distribution)**.** The CDF of the logarithmic distribution $\mathcal{L}_{\xi}$ with $0 < \xi < 1$ is defined by its discrete PDF

$$\forall x \in \mathbb{N} : f^{\mathcal{L}}_{\xi}\left(x\right) := -\frac{\xi^{x}}{x \ln\left(1-\xi\right)}.$$

Alternatively, it is defined recursively by (cp. [JKK05, Def. 7.1.1])

$$\forall x \in \mathbb{N}_{>1} : f^{\mathcal{L}}_{\xi}\left(x\right) := \frac{(x-1)\xi}{x} f^{\mathcal{L}}_{\xi}\left(x-1\right).$$

Definition A.3 (Stable distribution)**.** A random variable X is distributed according to the CDF of the Stable distribution $\mathcal{S}\left(\alpha,\beta,\gamma,\delta,k\right)$ if and only if X is distributed according to

$$X \sim \begin{cases} \begin{cases} \gamma \cdot Z + \delta, & \text{if } \alpha \neq 1 \\ \gamma \cdot Z + \left(\delta + \frac{2\beta}{\pi}\gamma \ln(\gamma)\right), & \text{if } \alpha = 1 \end{cases}, & \text{if } k = 1 \\ \begin{cases} \gamma \cdot \left(Z - \beta \tan\left(\frac{\pi\alpha}{2}\right)\right) + \delta, & \text{if } \alpha \neq 1 \\ \gamma \cdot Z + \delta, & \text{if } \alpha = 1 \end{cases}, & \text{if } k = 0 \end{cases},$$

in which Z has the characteristic function (cp. [Nol12, Definition 1.7, Definition 1.8])

$$\mathbb{E}\left(\exp\left(\mathrm{i}uZ\right)\right) \begin{cases} \exp\left(-|u|^{\alpha}\left[1 - \mathrm{i}\beta \tan\left(\frac{\pi\alpha}{2}\right)\operatorname{sgn}(u)\right]\right), & \text{if } \alpha \neq 1 \\ \exp\left(-|u|\left[1 + \mathrm{i}\frac{\beta 2}{\pi}\operatorname{sgn}(u) \ln|u|\right]\right), & \text{if } \alpha = 1 \end{cases}.$$

B. Sampling random numbers

Algorithm B.1: Sample $V \sim \mathcal{L}_\xi,\ 0 < \xi < 1$

```
Input  : $0 < \xi < 1$
Output: $V$
Note   : cp. [Kem81, Algorithm LK] and Definition A.2

Initialize $h \leftarrow \ln(1-\xi)$, $x \leftarrow 1$ and draw $u_1 \sim \mathcal{U}[0,1]$
if $u_1 > \xi$ then
    $V \leftarrow x$
else
    Draw $u_2 \sim \mathcal{U}[0,1]$ and set $q \leftarrow 1 - \exp(u_2 \cdot h)$
    if $u_1 < q^2$ then
        $V \leftarrow \lfloor 1 + \ln(u_1) / \ln(q) \rfloor$
    else
        if $u_1 > q$ then
            $V \leftarrow x$
        else
            $V \leftarrow 2$
```

Algorithm B.2: Sample $V \sim \mathcal{G}_{1,\nu},\ \nu > 0$

```
Input  : $\nu > 0$
Output: $V$
Note   : cp. [Knu00, 3.4.1.E] and Definition A.1
Initialize $j \leftarrow 0$
if $\nu > 1$ then
    while $j = 0$ do
        Draw independently distributed $u_1, u_2, u_3 \sim \mathcal{U}[0,1]$
        $Y \leftarrow \tan(u_1 \cdot \pi)$
        $X \leftarrow \sqrt{2\nu - 1} \cdot Y + \nu - 1$
        if $(X > 0$ and $(1 + Y^2) \cdot \exp\left((\nu - 1) \cdot \ln\left(\frac{X}{\nu-1}\right) - \sqrt{2\nu - 1} \cdot Y\right) \geq u_2)$
        then
            $V \leftarrow X,\ j++$
else
    if $\nu = 1$ then
        Draw $u_1 \sim \mathcal{U}[0,1],\ V \leftarrow -\ln(u_1)$
    else
        if $\nu = 0.5$ then
            Draw $u_1 \sim \mathcal{U}[0,1],\ u_1 \leftarrow \Phi^{-1}(u_1),\ V \leftarrow 0.5 \cdot u_1^2$
        else
            $Y \leftarrow \frac{\exp(1)}{\nu + \exp(1)}$
            while $j = 0$ do
                Draw independently distributed $u_1, u_2, u_3 \sim \mathcal{U}[0,1]$
                if $u_1 < Y$ then
                    $V \leftarrow u_2^{1/\nu},\ X \leftarrow \exp(-V)$
                else
                    $V \leftarrow 1 - \ln(u_2),\ X \leftarrow V^{\nu-1}$
                if $X > u_3$ then
                    $j++$
```

Algorithm B.3: Sample $V \sim \mathcal{G}_{1,\nu},\ \nu \in \mathbb{N}$

```
Input  : $\nu \in \mathbb{N}$
Output : $V$
Note   : cp. [PFTV07, 7.3.10] and Definition A.1
Initialize $V \leftarrow 1$ and draw independently distributed $u_1, \dots, u_\nu \sim \mathcal{U}[0,1]$
for $i \leftarrow 1$ to $\nu$ do
    $V \leftarrow V \cdot u_i$
$V \leftarrow -\ln(V)$
```

Algorithm B.4: Sample $V \sim \mathcal{S}(\alpha, \beta, \gamma, \delta, 1)$

```
Input  : $\alpha, \beta, \gamma, \delta$
Output : $V$
Note   : cp. [Nol12, Theorem 1.19] and Definition A.3
Draw independently distributed $u_1, u_2 \sim \mathcal{U}[0,1]$
Initialize $\Theta \leftarrow \pi\left(u_1 - \frac{1}{2}\right)$ and $W \leftarrow -\ln(u_2)$
if $\alpha = 1$ then
    $V \leftarrow \frac{2}{\pi}\left[\left(\frac{\pi}{2} + \beta\Theta\right)\tan(\Theta) - \beta\ln\left(\frac{\frac{W\pi}{2}\cos(\Theta)}{\frac{\pi}{2}+\beta\Theta}\right)\right]$
    $V \leftarrow \gamma \cdot V + \left(\delta + \frac{2\beta}{\pi}\gamma\ln(\gamma)\right)$
else
    $\theta_0 \leftarrow \arctan\left(\beta\tan\left(\frac{\alpha\pi}{2}\right)\right)$
    $V \leftarrow \frac{\sin(\alpha(\theta_0+\Theta))}{(\cos(\alpha\theta_0)\cos(\Theta))^{1/\alpha}}\left[\frac{\cos(\alpha\theta_0+(\alpha-1)\Theta)}{W}\right]^{(1-\alpha)/\alpha}$
    $V \leftarrow \gamma \cdot V + \delta$
```

C. Various tables and figures

C.1. Gauss–Hermite quadrature for approximating the multivariate Gaussian distribution

$\lvert\psi_j\rvert \downarrow$	$N_{\mathrm{H}} \rightarrow$	4	8	16	32	64	128
≤ 1	$\overline{\epsilon}_{N_{\mathrm{H}},k}$	2.03e-01	8.69e-02	4.10e-02	1.96e+00	1.73e-01	7.37e-02
	$\overline{\sigma}_{N_{\mathrm{H}},k}$	3.74e-01	2.29e-01	1.66e+00	6.17e+02	3.90e+01	1.60e+01
	$\overline{\mathcal{M}}_{N_{\mathrm{H}},k}$	7.21e-02	7.86e-03	2.34e-04	5.19e-07	9.39e-12	1.64e-15
≤ 0.95	$\overline{\epsilon}_{N_{\mathrm{H}},k}$	8.44e-02	1.40e-02	1.01e-03	1.80e-05	3.07e-08	1.43e-12
	$\overline{\sigma}_{N_{\mathrm{H}},k}$	1.26e-01	3.43e-02	4.69e-03	1.97e-04	8.40e-07	1.11e-10
	$\overline{\mathcal{M}}_{N_{\mathrm{H}},k}$	3.24e-02	1.52e-03	8.68e-06	1.18e-09	3.03e-15	7.01e-16
≤ 0.9	$\overline{\epsilon}_{N_{\mathrm{H}},k}$	4.38e-02	3.34e-03	7.54e-05	1.97e-07	1.91e-11	1.10e-15
	$\overline{\sigma}_{N_{\mathrm{H}},k}$	7.69e-02	9.79e-03	4.81e-04	4.11e-06	1.48e-09	1.36e-15
	$\overline{\mathcal{M}}_{N_{\mathrm{H}},k}$	1.54e-02	3.16e-04	3.77e-07	3.05e-12	1.86e-15	6.55e-16
≤ 0.85	$\overline{\epsilon}_{N_{\mathrm{H}},k}$	2.56e-02	9.07e-04	5.83e-06	2.55e-09	3.49e-14	1.06e-15
	$\overline{\sigma}_{N_{\mathrm{H}},k}$	5.93e-02	3.40e-03	5.48e-05	7.09e-08	5.25e-12	1.16e-15
	$\overline{\mathcal{M}}_{N_{\mathrm{H}},k}$	7.20e-03	6.73e-05	1.82e-08	9.24e-15	1.64e-15	6.45e-16
≤ 0.8	$\overline{\epsilon}_{N_{\mathrm{H}},k}$	1.85e-02	3.84e-04	4.78e-07	2.69e-11	1.78e-15	1.05e-15
	$\overline{\sigma}_{N_{\mathrm{H}},k}$	5.62e-02	3.18e-03	4.99e-06	1.01e-09	1.05e-15	1.14e-15
	$\overline{\mathcal{M}}_{N_{\mathrm{H}},k}$	3.31e-03	1.39e-05	8.49e-10	1.99e-15	1.63e-15	6.51e-16
≤ 0.75	$\overline{\epsilon}_{N_{\mathrm{H}},k}$	1.56e-02	2.73e-04	8.93e-08	3.30e-13	1.77e-15	1.04e-15
	$\overline{\sigma}_{N_{\mathrm{H}},k}$	5.37e-02	2.74e-03	2.34e-06	2.04e-11	9.56e-16	1.11e-15
	$\overline{\mathcal{M}}_{N_{\mathrm{H}},k}$	1.59e-03	3.04e-06	4.13e-11	1.38e-15	1.63e-15	6.57e-16
≤ 0.7	$\overline{\epsilon}_{N_{\mathrm{H}},k}$	1.50e-02	2.77e-04	8.86e-08	1.08e-14	1.75e-15	1.02e-15
	$\overline{\sigma}_{N_{\mathrm{H}},k}$	5.27e-02	2.84e-03	3.28e-06	9.75e-13	9.13e-16	1.08e-15
	$\overline{\mathcal{M}}_{N_{\mathrm{H}},k}$	8.57e-04	7.70e-07	2.31e-12	1.29e-15	1.65e-15	6.54e-16

Table C.1.: Different statistics of error estimator ϵ using $k = 10$ and $N_{\boldsymbol{u},\boldsymbol{\psi}} = 10^5$.

$\lvert\psi_j\rvert \downarrow$	$N_{\mathrm{H}} \rightarrow$	4	8	16	32	64	128
≤ 1	$\overline{\epsilon}_{N_{\mathrm{H}},k}$	8.16e+04	3.79e-01	1.20e-01	9.99e-01	5.32e+03	1.42e+03
	$\overline{\sigma}_{N_{\mathrm{H}},k}$	2.58e+07	4.14e+01	2.58e+00	2.15e+02	1.66e+06	4.49e+05
	$\overline{\mathcal{M}}_{N_{\mathrm{H}},k}$	3.40e-01	1.07e-01	1.48e-02	5.01e-04	1.40e-06	4.18e-11
≤ 0.95	$\overline{\epsilon}_{N_{\mathrm{H}},k}$	2.38e-01	6.82e-02	8.72e-03	3.17e-04	1.83e-06	5.54e-10
	$\overline{\sigma}_{N_{\mathrm{H}},k}$	2.03e-01	8.73e-02	1.92e-02	1.47e-03	2.96e-05	2.71e-08
	$\overline{\mathcal{M}}_{N_{\mathrm{H}},k}$	1.84e-01	3.36e-02	1.41e-03	4.81e-06	2.14e-10	1.18e-15
≤ 0.9	$\overline{\epsilon}_{N_{\mathrm{H}},k}$	1.41e-01	2.53e-02	1.43e-03	1.29e-05	7.24e-09	6.00e-14
	$\overline{\sigma}_{N_{\mathrm{H}},k}$	1.27e-01	3.58e-02	3.85e-03	9.51e-05	2.54e-07	6.14e-12
	$\overline{\mathcal{M}}_{N_{\mathrm{H}},k}$	1.06e-01	1.13e-02	1.67e-04	8.17e-08	1.22e-13	6.73e-16
≤ 0.85	$\overline{\epsilon}_{N_{\mathrm{H}},k}$	8.33e-02	9.37e-003	2.33e-04	5.03e-07	2.98e-11	9.80e-16
	$\overline{\sigma}_{N_{\mathrm{H}},k}$	8.11e-02	1.48e-02	7.58e-04	4.26e-06	1.70e-09	1.33e-15
	$\overline{\mathcal{M}}_{N_{\mathrm{H}},k}$	5.92e-02	3.61e-03	1.87e-05	1.35e-09	2.50e-15	6.21e-16
≤ 0.8	$\overline{\epsilon}_{N_{\mathrm{H}},k}$	4.86e-02	3.31e-03	3.66e-05	2.09e-08	1.41e-13	9.65e-16
	$\overline{\sigma}_{N_{\mathrm{H}},k}$	5.57e-02	6.02e-03	1.54e-04	3.71e-07	1.04e-11	1.08e-15
	$\overline{\mathcal{M}}_{N_{\mathrm{H}},k}$	3.16e-02	1.05e-03	1.75e-06	1.90e-11	1.68e-15	6.15e-16
≤ 0.75	$\overline{\epsilon}_{N_{\mathrm{H}},k}$	2.80e-02	1.08e-03	4.79e-06	6.45e-10	1.99e-15	9.60e-16
	$\overline{\sigma}_{N_{\mathrm{H}},k}$	4.28e-02	2.49e-03	2.41e-05	1.49e-08	2.44e-14	1.06e-15
	$\overline{\mathcal{M}}_{N_{\mathrm{H}},k}$	1.58e-02	2.65e-04	1.24e-07	1.54e-13	1.56e-15	6.15e-16
≤ 0.7	$\overline{\epsilon}_{N_{\mathrm{H}},k}$	1.68e-02	3.37e-04	5.09e-07	1.46e-11	1.74e-15	9.74e-16
	$\overline{\sigma}_{N_{\mathrm{H}},k}$	4.06e-02	1.82e-03	4.13e-06	6.99e-10	1.04e-15	1.05e-15
	$\overline{\mathcal{M}}_{N_{\mathrm{H}},k}$	7.18e-03	5.48e-05	6.23e-09	3.00e-15	1.58e-15	6.31e-16

Table C.2.: Different statistics of error estimator ϵ using $k = 20$ and $N_{\boldsymbol{u},\boldsymbol{\psi}} = 10^5$.

$\lvert\psi_j\rvert \downarrow$	$N_{\mathrm{H}} \rightarrow$	4	8	16	32	64	128
≤ 1	$\overline{\epsilon}_{N_{\mathrm{H}},k}$	6.78e-01	8.15e+02	1.03e+02	1.94e+04	1.26e+08	2.53e+00
	$\overline{\sigma}_{N_{\mathrm{H}},k}$	1.59e+00	2.57e+05	3.19e+04	6.13e+06	3.72e+10	5.66e+02
	$\overline{\mathcal{M}}_{N_{\mathrm{H}},k}$	6.04e-01	2.82e-01	7.49e-02	7.15e-03	1.18e-04	8.30e-08
≤ 0.95	$\overline{\epsilon}_{N_{\mathrm{H}},k}$	3.96e-01	1.53e-01	2.91e-02	1.78e-03	1.89e-05	1.45e-08
	$\overline{\sigma}_{N_{\mathrm{H}},k}$	2.57e-01	1.37e-01	4.14e-02	4.81e-03	1.40e-04	4.74e-07
	$\overline{\mathcal{M}}_{N_{\mathrm{H}},k}$	3.68e-01	1.15e-01	1.29e-02	2.00e-04	9.30e-08	1.10e-13
≤ 0.9	$\overline{\epsilon}_{N_{\mathrm{H}},k}$	2.65e-01	7.23e-02	7.19e-03	1.38e-04	1.94e-07	4.94e-12
	$\overline{\sigma}_{N_{\mathrm{H}},k}$	1.77e-01	6.92e-02	1.17e-02	4.86e-04	2.18e-06	2.58e-10
	$\overline{\mathcal{M}}_{N_{\mathrm{H}},k}$	2.40e-01	5.15e-02	2.70e-03	9.39e-06	2.68e-10	9.85e-16
≤ 0.85	$\overline{\epsilon}_{N_{\mathrm{H}},k}$	1.77e-01	3.41e-02	1.80e-03	1.12e-05	2.45e-09	3.08e-15
	$\overline{\sigma}_{N_{\mathrm{H}},k}$	1.24e-01	3.55e-02	3.36e-03	5.25e-05	4.87e-08	1.49e-13
	$\overline{\mathcal{M}}_{N_{\mathrm{H}},k}$	1.57e-01	2.26e-02	5.39e-04	4.20e-07	8.33e-13	6.82e-16
≤ 0.8	$\overline{\epsilon}_{N_{\mathrm{H}},k}$	1.16e-01	1.53e-02	4.13e-04	8.10e-07	2.15e-11	9.42e-16
	$\overline{\sigma}_{N_{\mathrm{H}},k}$	8.69e-02	1.76e-02	9.18e-04	4.81e-06	4.86e-10	1.03e-15
	$\overline{\mathcal{M}}_{N_{\mathrm{H}},k}$	9.89e-02	9.22e-03	9.57e-05	1.52e-08	3.70e-15	6.41e-16
≤ 0.75	$\overline{\epsilon}_{N_{\mathrm{H}},k}$	7.20e-02	6.21e-03	8.04e-05	4.57e-08	1.89e-13	9.22e-16
	$\overline{\sigma}_{N_{\mathrm{H}},k}$	5.95e-02	7.99e-03	2.13e-04	4.53e-07	1.65e-11	1.01e-15
	$\overline{\mathcal{M}}_{N_{\mathrm{H}},k}$	5.84e-02	3.34e-03	1.32e-05	3.43e-10	1.79e-15	6.20e-16
≤ 0.7	$\overline{\epsilon}_{N_{\mathrm{H}},k}$	4.28e-02	2.28e-03	1.29e-05	1.86e-09	2.31e-15	9.31e-16
	$\overline{\sigma}_{N_{\mathrm{H}},k}$	4.12e-02	3.34e-03	4.29e-05	2.21e-08	3.65e-14	1.01e-15
	$\overline{\mathcal{M}}_{N_{\mathrm{H}},k}$	3.22e-02	1.06e-03	1.42e-06	4.90e-12	1.54e-15	6.18e-16

Table C.3.: Different statistics of error estimator ϵ using $k = 30$ and $N_{\boldsymbol{u},\boldsymbol{\psi}} = 10^5$.

C.2. Input Parameters for Li's example

i	h_i	ψ_i		i	h_i	ψ_i		i	h_i	ψ_i
1	0.0787	0.516		35	0.095	0.811		68	0.0222	-0.5
2	0.025	-0.449		36	0.0438	-0.111		69	0.0225	-0.494
3	0.0711	0.379		37	0.0322	-0.32		70	0.0434	-0.119
4	0.0947	0.804		38	0.0532	0.0583		71	0.0987	0.877
5	0.00193	-0.865		39	0.0256	-0.439		72	0.0211	-0.52
6	0.0405	-0.171		40	0.00455	-0.818		73	0.0853	0.636
7	0.0251	-0.448		41	0.0505	0.00896		74	0.0284	-0.389
8	0.00227	-0.859		42	0.0696	0.353		75	0.0836	0.605
9	0.0521	0.0372		43	0.00912	-0.736		76	0.0367	-0.24
10	0.0345	-0.28		44	0.0907	0.733		77	0.0935	0.783
11	0.0274	-0.406		45	0.00309	-0.844		78	0.086	0.648
12	0.0561	0.11		46	0.0152	-0.626		79	0.0491	-0.0164
13	0.014	-0.648		47	0.0982	0.867		80	0.00221	-0.86
14	0.0544	0.0789		48	0.062	0.217		81	0.0362	-0.248
15	0.0522	0.0394		49	0.0299	-0.362		82	0.0736	0.426
16	0.0857	0.643		50	0.0361	-0.25		83	0.0519	0.0349
17	0.05	-0.000406		51	0.0481	-0.0343		84	0.0423	-0.139
18	0.0419	-0.145		52	0.0298	-0.364		85	0.0029	-0.848
19	0.0744	0.44		53	0.0285	-0.387		86	0.0909	0.737
20	0.0249	-0.451		54	0.0924	0.764		87	0.0112	-0.698
21	0.0239	-0.469		55	0.0629	0.232		88	0.098	0.864
22	0.032	-0.324		56	0.0755	0.459		89	0.0688	0.338
23	0.0911	0.739		57	0.0714	0.385		90	0.0487	-0.0228
24	0.0165	-0.603		58	0.0723	0.401		91	0.0736	0.425
25	0.0246	-0.458		59	0.00698	-0.774		92	0.0589	0.16
26	0.0198	-0.543		60	0.0487	-0.0237		93	0.0528	0.0512
27	0.0716	0.389		61	0.00889	-0.74		94	0.0456	-0.0798
28	0.0968	0.842		62	0.076	0.467		95	0.0828	0.591
29	0.0769	0.485		63	0.0424	-0.137		96	0.087	0.667
30	0.00807	-0.755		64	0.0597	0.175		97	0.0491	-0.0153
31	0.046	-0.0721		65	0.00864	-0.744		98	0.097	0.845
32	0.0257	-0.437		66	0.0273	-0.409		99	0.0889	0.7
33	0.0777	0.499		67	0.0624	0.223		100	0.0138	-0.651
34	0.0584	0.151								

Table C.4.: Hazard rates and factor loadings for Li's example from Subsection 6.2.3.

C.3. Credit portfolio

i	N_i	p_i	$1 - R_i$
1	4.6164	0.0017	0.5233
2	3.0402	0.0444	0.4008
3	2.3508	0.0198	0.5543
4	2.3427	0.0039	0.2953
5	2.2087	0.0296	0.7318
6	2.0984	0.0009	0.0448
7	1.7929	0.0198	0.4312
8	1.6011	0.0039	0.6334
9	1.4852	0.0132	0.4487
10	1.2556	0.0007	0.2763
11	1.2530	0.0198	0.4025
12	1.2305	0.0007	0.5714
13	1.2015	0.0132	0.0490
14	1.1917	0.0039	0.5543
15	1.1532	0.0039	0.2962
16	1.1092	0.0003	0.5543
17	0.9810	0.0296	0.6769
18	0.8839	0.0004	0.5497
19	0.6993	0.0667	0.6431
20	0.6083	0.0088	0.6769
21	0.5468	0.0132	0.7318
22	0.5231	0.0039	0.6334
23	0.5167	0.0059	0.6334
24	0.4742	0.0296	0.2928
25	0.4430	0.0039	0.6334
26	0.4282	0.0296	0.6769
27	0.4263	0.0039	0.6334
28	0.4181	0.0059	0.6334
29	0.3710	0.0026	0.5497
30	0.2647	0.0132	0.6334
31	0.2536	0.0296	0.5543
32	0.2440	0.0088	0.5478
33	0.2412	0.0004	0.5119
34	0.2238	0.0132	0.6303
35	0.2199	0.0667	0.4824
36	0.2180	0.0017	0.5119
37	0.2033	0.0009	0.6030
38	0.1910	0.0004	0.5119
39	0.1890	0.0132	0.6334
40	0.1844	0.0296	0.6769
41	0.1683	0.0005	0.5720
42	0.1496	0.0132	0.5056
43	0.1458	0.0088	0.4824
44	0.1388	0.0296	0.2795
45	0.1247	0.1000	0.5119
46	0.1228	0.0026	0.6334
47	0.1197	0.0088	0.6334
48	0.1190	0.0088	0.7473
49	0.1141	0.0444	0.5119
50	0.1127	0.0296	0.6334
51	0.0975	0.0132	0.6334
52	0.0892	0.0296	0.2796
53	0.0832	0.0004	0.7318
54	0.0815	0.0088	0.5211
55	0.0727	0.0088	0.5119
56	0.0671	0.0004	0.5497
57	0.0658	0.0009	0.5497
58	0.0642	0.0132	0.6334
59	0.0628	0.0296	0.2672
60	0.0620	0.0026	0.6334
61	0.0568	0.0088	0.6334
62	0.0565	0.0059	0.7318
63	0.0542	0.0005	0.6334
64	0.0518	0.0012	0.5497
65	0.0505	0.0088	0.6334
66	0.0474	0.0198	0.6334
67	0.0470	0.0132	0.5119
68	0.0458	0.0012	0.5119
69	0.0416	0.0026	0.6334
70	0.0383	0.0667	0.5543
71	0.0370	0.0026	0.4824
72	0.0348	0.0009	0.7003
73	0.0282	0.0059	0.6334
74	0.0107	0.0444	0.6334
75	0.0095	0.0039	0.6334

Table C.5.: Exemplary credit portfolio of the health care business.

C.4. Variance analyses of different Monte–Carlo methods

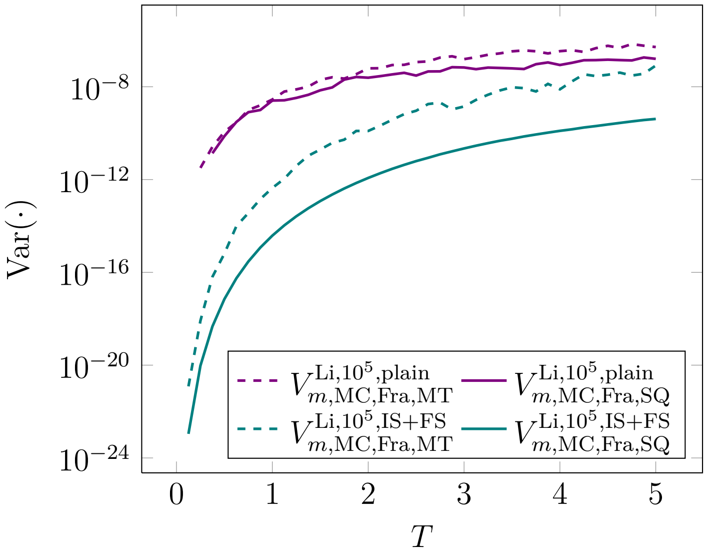


Figure C.1.: Variances of different Monte–Carlo approximations using a Frank copula with parameter $\theta = 1.0$.

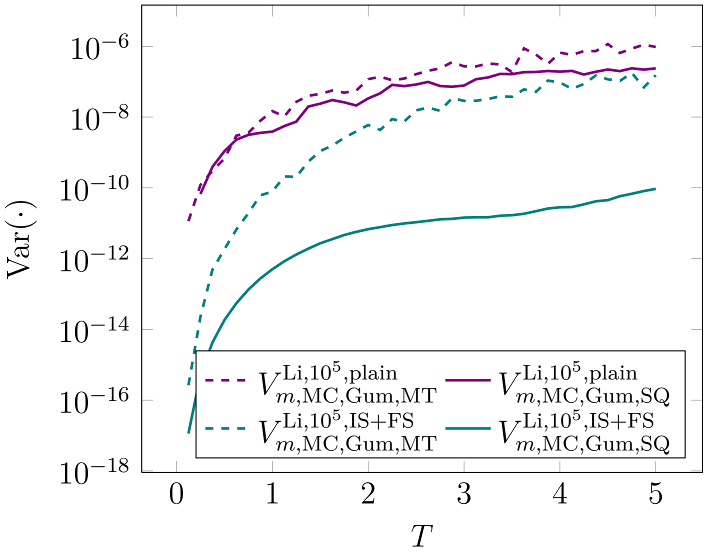


Figure C.2.: Variances of different Monte–Carlo approximations using a Gumbel copula with parameter $\theta = 1.2$.

Bibliography

[AS70] Abramowitz, M. and Stegun, I.A.: *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables.* Dover publications, 1970.

[BH10] Borsdorf, R. and Higham, N.J.: *A Preconditioned Newton Algorithm for the Nearest Correlation Matrix.* IMA Journal of Numerical Analysis, 30(1):94–107, 2010.

[BHR10] Borsdorf, R., Higham, N.J., and Raydan, M.: *Computing a Nearest Correlation Matrix with Factor Structure.* SIAM Journal on Matrix Analysis and Applications, 31:2603–2622, 2010.

[BK06] Bacigál, T. and Komorniková, M.: *Fitting Archimedean Copulas to Bivariate Geodetic Data.* In *Proc. COMPSTAT*, pages 649–656, 2006.

[BS73] Black, F. and Scholes, M.: *The Pricing of Options and Corporate Liabilities.* The journal of political economy, pages 637–654, 1973.

[CD62] Curnow, R.N. and Dunnett, C.W.: *The Numerical Evaluation of Certain Multivariate Normal Integrals.* The Annals of Mathematical Statistics, pages 571–579, 1962.

[CG08] Chen, Z. and Glasserman, P.: *Fast Pricing of Basket Default Swaps.* Operations Research-Baltimore, 56(2):286–303, 2008.

[CK07] Cheney, E.W. and Kincaid, D.R.: *Numerical Mathematics and Computing.* Brooks/Cole Publishing Co., 2007.

[com11] comdirect bank AG: *Market Data.* Website, 2011. Online available at: `http://www.comdirect.de/inf/index.html`, access on 29/09/2011.

[Del12] DELL COMPUTER CORPORATION: *Dell PowerEdge R710 Rack-Server.* Website, 2012. Online available at: `http://www.dell.com/downloads/global/products/pedge/en/server-poweredge-r710-specs-en.pdf`, access on 19/04/2012.

[DG96] DHAENE, J. and GOOVAERTS, M.J.: *Dependency of Risks and Stopp-Loss Order.* Astin Bulletin, 26(2):201–212, 1996.

[DNR00] DURRLEMAN, V., NIKEGHBALI, A., and RONCALLI, T.: *Which Copula is the Right One?* Credit Lyonnais, 2000.

[DS03] DUFFIE, D. and SINGLETON, K. J.: *Credit Risk: Pricing, Measurement and Management.* Princeton University Press, 2003.

[ELM03] EMBRECHTS, P., LINDSKOG, F., and MCNEIL, A.: *Modelling Dependence with Copulas and Applications to Risk Management.* Handbook of heavy tailed distributions in finance, 8(1):329–384, 2003.

[EMS02] EMBRECHTS, P., MCNEIL, A., and STRAUMANN, D.: *Correlation and Dependence in Risk Management: Properties and Pitfalls.* Risk management: value at risk and beyond, pages 176–223, 2002.

[Fel71] FELLER, W.: *An Introduction to Probability Theory and Its Applications*, volume II. Wiley Series in Probability and Mathematical Statistics, 2nd edition, 1971.

[FF92] FAMA, E.F. and FRENCH, K.R.: *The Cross-Section of Expected Stock Returns.* The Journal of Finance, 47(2):427–465, 1992.

[FF93] FAMA, E.F. and FRENCH, K.R.: *Common Risk Factors in the Returns on Stocks and Bonds.* Journal of financial economics, 33(1):3–56, 1993.

[FH07] FREUND, R.W. and HOPPE, R.H.W.: *Stoer/Bulirsch: Numerische Mathematik 1.* Springer Verlag, 2007.

[Gal11] GALLIER, J.: *Discrete Mathematics.* Springer Verlag, 2011.

[Gau04] GAUTSCHI, W.: *Orthogonal Polynomials: Computation and Approximation.* Oxford University Press, 2004.

[GB09] GENZ, A. and BRETZ, F.: *Computation of Multivariate Normal and T Probabilities.* Springer Verlag, 2009.

[Gla04] Glasserman, P.: *Monte Carlo Methods in Financial Engineering.* Springer Verlag, 2004.

[GLN$^+$01] Georges, P., Lamy, A.G., Nicolas, E., Quibel, G., and Roncalli, T.: *Multivariate Survival Modelling: A Unified Approach with Copulas.* Groupe de Recherche Opérationnelle Crédit Lyonnais, 2001.

[GNU12] GNU–Project: *The GNU Multiple Precision Arithmetic Library.* Website, 2012. Online available at: `hhttp://gmplib.org/`, access on 29/02/2012.

[GS07] Glasserman, P. and Suchintabandid, S.: *Correlation Expansions for CDO Pricing.* Journal of Banking & Finance, 31(5):1375–1398, 2007.

[Hau06] Haug, E.G.: *The Complete Guide to Option Pricing Formulas.* McGraw-Hill Companies, 2006.

[Heu83] Heuser, H.: *Lehrbuch der Analysis*, volume 2. B. G. Teubner, 2nd edition, 1983.

[Hig02] Higham, N.J.: *Computing the Nearest Correlation Matrix – A Problem from Finance.* IMA Journal of Numerical Analysis, 22(3):329–343, 2002.

[Hoe40] Hoeffding, W.: *Massstabinvariante Korrelationstheorie.* Schriften des Mathematischen Seminars und des Instituts für Angewandte Mathematik der Universität Berlin, 5:181–233, 1940. Dissertation.

[Hof08] Hofert, M.: *Sampling Archimedean Copulas.* Computational Statistics & Data Analysis, 52(12):5163–5174, 2008.

[Hul09] Hull, J.: *Options, Futures and Other Derivatives.* Pearson Prentice Hall, 7th edition, 2009.

[HWW55] Hastings, Jr., C., Wayward, J.T., and Wong, Jr., J.P.: *Approximations for Digital Computers.* Princeton University Press, 1955.

[JK04] Joshi, M.S. and Kainth, D.: *Rapid and Accurate Development of Prices and Greeks for n–th to Default Credit Swaps in the Li Model.* Quantitative Finance, 4(3):266–275, 2004.

[JKK05] Johnson, N.L., Kemp, A.W., and Kotz, S.: *Univariate Discrete Distributions*, volume 444. Wiley-Interscience, 2005.

[Joe97] Joe, H.: *Multivariate Models and Dependence Concepts.* Chapman & Hall, 1997.

[JRR04] Jouanin, J.F., Riboulet, G., and Roncalli, T.: *Financial Applications of Copula Functions.* Risk Measures for the 21st Century, pages 273–301, 2004.

[Kem81] Kemp, A.W.: *Efficient Generation of Logarithmically Distributed Pseudo-Random Variables.* Applied Statistics, pages 249–253, 1981.

[Kim74] Kimberling, C.H.: *A Probabilistic Interpretation Of Complete Monotonicity.* Aequationes Mathematicae, 10(2):152–164, 1974.

[KMN89] Kahaner, D., Moler, C.B., and Nash, S.: *Numerical Methods and Software.* Prentice Hall, Englewood Cliffs, 1989.

[Knu00] Knuth, D.E.: *Seminumerical Algorithms*, volume 2 of *The Art of Computer Programming.* Addison-Wesley Publishing Company, 3rd edition, 2000.

[Kro65] Kronrod, A.S.: *Nodes and Weights of Quadrature Formulas: Sixteen-Place Tables.* New York: Consultants Bureau, 1965.

[Lau97] Laurie, D.P.: *Calculation of Gauss-Kronrod Quadrature Rules.* Mathematics of computation, 66(219):1133–1146, 1997.

[LBW09a] Landesbank Baden-Württemberg, Zertifikate: *SWN Synthia 2009, Endgültige Bedingungen.* Website, 2009. Online available at: `https://www.lbbw-markets.de/cmp-portalWAR/ShowPropertyServlet?nodePath=%2FMarkets+Repository%2Fcmp%2Fde%2FZertifikate%2FUpload-Dokumente%2FBasisprospekte%2FVDE000LBW47W9.pdf`, access on 05/12/2011.

[LBW09b] Landesbank Baden-Württemberg, Zertifikate: *SWN Synthia 2009, Produktdetails.* Website, 2009. Online available at: `https://www.lbbw-markets.de/cmp-portalWAR/appmanager/LBBW/Markets?_nfpb=true&_nfls=false&_pageLabel=p_lbbw_zertifikate_hidden_superseite&lang=de&SUPERSEITE_ISIN=DE000LBW47W9&_sourcePageLabel=LEER&holderId=root&portletId=&query=DE000LBW47W9&startDatum=null&endDatum=null&suche=1`, access on 05/12/2011.

[LBW09c] Landesbank Baden-Württemberg, Zertifikate: *SWN Synthia 2009, Produktinformationsblatt.* Website, 2009. Online available at: `https://www.lbbw-markets.de/cmp-portalWAR/ShowPropertyServlet?nodePath=%2FMarkets+Repository%2Fcmp%2Fde%2FZertifikate%2FUpload-Dokumente%2FTermsheets%2FHDE000LBW47W9.pdf`, access on 05/12/2011.

[LG05] Laurent, J.P. and Gregory, J.: *Basket Default Swaps, CDOs and Factor Copulas.* Journal of Risk, 7(4):103–122, 2005.

[Li00] Li, D.X.: *On Default Correlation: A Copula Function Approach.* Journal of Fixed Income, 9(4):43–54, 2000.

[LMS03] Lindskog, F., McNeil, A., and Schmock, U.: *Kendall's Tau for Elliptical Distributions.* Credit risk: Measurement, evaluation and management, pages 149–156, 2003.

[Mar63] Marsaglia, G.: *Expressing the Normal Distribution with Covariance Matrix A+B in Terms of One with Covariance Matrix A.* Biometrika, pages 535–538, 1963.

[Maz63] Mazkewitsch, D.: *The n-th Derivative of a Product.* The American Mathematical Monthly, 70(7):739–742, 1963.

[McN08] McNeil, A.: *Sampling Nested Archimedean Copulas.* Journal of Statistical Computation and Simulation, 78(6):567–581, 2008.

[Mel03] Melchiori, M.R.: *Which Archimedean Copula Is the Right One?* Yield Curve, 37:1–20, 2003.

[MFE05] McNeil, A.J., Frey, R., and Embrechts, P.: *Quantitative Risk Management: Concepts, Techniques and Tools.* Princeton University Press, 2005.

[MN98] Matsumoto, M. and Nishimura, T.: *Mersenne Twister: A 623-dimensionally Equidistributed Uniform Pseudo-Random Number Generator.* ACM Transactions on Modeling and Computer Simulation (TOMACS), 8(1):3–30, 1998.

[MO88] Marshall, A.W. and Olkin, I.: *Families of Multivariate Distributions.* Journal of the American Statistical Association, pages 834–841, 1988.

[Mor95] Moro, B.: *The Full Monte.* Risk, 8(2):57–58, 1995.

[MRW06] Martin, M.R.W., Reitz, S., and Wehn, C.: *Kreditderivate und Kreditrisikomodelle.* Vieweg & Teubner Verlag, 2006.

[MV04] Meneguzzo, D. and Vecchiato, W.: *Copula Sensitivity in Collateralized Debt Obligations and Basket Default Swaps.* Journal of Futures Markets, 24(1):37–70, 2004.

[MZM94] Marsaglia, G., Zaman, A., and Marsaglia, J.C.W.: *Rapid Evaluation of the Inverse of the Normal Distribution Function.* Statistics & Probability Letters, 19(4):259–266, 1994.

[Nel06] Nelsen, R.B.: *An Introduction to Copulas.* Springer Verlag, 2006.

[Nol12] Nolan, J. P.: *Stable Distributions - Models for Heavy Tailed Data.* Birkhauser, Boston, 2012. In progress, Chapter 1 online at academic2.american.edu/∼jpnolan.

[OB73] Oberhettinger, F. and Badii, L.: *Tables of Laplace Transforms*, volume 207. Springer Verlag, 1973.

[PdDKÜK83] Piessens, R., Doncker-Kapenga, E. de, Überhuber, C.W., and Kahaner, D.K.: *Quadpack – A Subroutine Package for Automatic Integration*, volume 1 of *Springer Series in Computational Mathematics.* Springer Verlag, 1983.

[PFTV07] Press, W.H., Flannery, B.P., Teukolsky, S.A., and Vetterling, W.T.: *Numerical Recipes – The Art of Scientific Computing.* Cambridge University Press, 3rd edition, 2007.

[QS07] Qi, H. and Sun, D.: *A Quadratically Convergent Newton Method for Computing the Nearest Correlation Matrix.* SIAM Journal on Matrix Analysis and Applications, 28(2):360–385, 2007.

[Ros76] Ross, S.A.: *The Arbitrage Theory of Capital Asset Pricing.* Journal of Economic Theory, 13(3):341–360, 1976.

[Sch03] Schönbucher, P.J.: *Credit Derivatives Pricing Models: Models, Pricing and Implementation.* John Wiley & Sons Inc., 2003.

[Sch10] Schröter, A.: *Copula Modelle und Modellrisiken.* Diploma thesis, University of Cologne, 2010.

[Sey12] Seydel, R.U.: *Tools for Computational Finance.* Springer Verlag, 5th edition, 2012.

[SH12a] Schröter, A. and Heider, P.: *An Analytical Formula for Pricing m–th to Default Swaps.* Journal of Applied Mathematics and Computing, 2012. `http://dx.doi.org/10.1007/s12190-012-0589-1`, accepted for publication.

[SH12b] Schröter, A. and Heider, P.: *Numerical Methods to Quantify the Model Risk of Basket Default Swaps.* Preprint, 2012.

[Sha64] Sharpe, W.F.: *Capital Asset Prices: A Theory of Market Equilibrium under Conditions of Risk.* The Journal of Finance, 19(3):425–442, 1964.

[Sha08] Shampine, L.F.: *Vectorized Adaptive Quadrature in Matlab.* Journal of Computational and Applied Mathematics, 211(2):131–140, 2008.

[SK09] Schwarz, H.R. and Köckler, N.: *Numerische Mathematik.* Vieweg & Teubner / GWV Fachverlage GmbH, 7th edition, 2009.

[Skl73] Sklar, A.: *Random Variables, Joint Distribution Functions, and Copulas.* Kybernetika, 9(6):449–460, 1973.

[Sob76] Sobol, I.M.: *Uniformly Distributed Sequences with an Additional Uniform Property.* USSR Computational Mathematics and Mathematical Physics, 16(5):236–242, 1976.

[SS01] Schönbucher, P. and Schubert, D.: *Copula-dependent Defaults in Intensity Models.* Technical report, Department of Statistics, Faculty of Economics, Bonn University, 2001.

[Stu82] Stulz, R.M.: *Options on the Minimum or the Maximum of Two Risky Assets: Analysis and Applications.* Journal of Financial Economics, 10(2):161–185, 1982.

[Tak67] Takacs, L.: *On the Method of Inclusion and Exclusion.* Journal of the American Statistical Association, 62(317):102–113, 1967, ISSN 0162-1459.

[TM11] The MathWorks®, Inc.: *Product Documentation MATLAB®*. Website, 2011. Online available at: `http://www.mathworks.de/help/techdoc/`, access on 18/11/2011.

Erklärung

Ich versichere, dass ich die von mir vorgelegte Dissertation selbständig angefertigt, die benutzten Quellen und Hilfsmittel vollständig angegeben und die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken im Wortlaut oder dem Sinn nach entnommen sind, in jedem Einzelfall als Entlehnung kenntlich gemacht habe; dass diese Dissertation noch keiner anderen Fakultät oder Universität zur Prüfung vorgelegen hat; dass sie - abgesehen von unten angegebenen Teilpublikationen - noch nicht veröffentlicht worden ist sowie, dass ich eine solche Veröffentlichung vor Abschluss des Promotionsverfahrens nicht vornehmen werde. Die Bestimmungen der Promotionsordnung sind mir bekannt. Die von mir vorgelegte Dissertation ist von Herrn Priv.-Doz. Dr. Pascal Heider betreut worden.

Nachfolgend genannte Teilpublikationen liegen vor:

Schröter, A. and Heider, P.: *Numerical Methods to Quantify the Model Risk of Basket Default Swaps*. Preprint, 2012, under review.

Schröter, A. and Heider, P.: *An Analytical Formula for Pricing m–th to Default Swaps*. Journal of Applied Mathematics and Computing, 2012, `http://dx.doi.org/10.1007/s12190-012-0589-1`, accepted for publication.

Ich versichere, dass ich alle Angaben wahrheitsgemäß nach bestem Wissen und Gewissen gemacht habe und verpflichte mich, jedmögliche, die obigen Angaben betreffenden Veränderungen, dem Dekanat unverzüglich mitzuteilen.

....

Ort, Datum Unterschrift