Alma Mater Studiorum – Università di Bologna

DOTTORATO DI RICERCA IN

MATEMATICA

Ciclo 35

Settore Concorsuale: 01/A3

Settore Scientifico Disciplinare: MAT/06

A Unified Model for XVA, including Interest Rates and Rating

Presentata da: Kevin Kamm

Coordinatore Dottorato

Prof. Dr. Valeria Simoncini

Supervisore

Prof. Dr. Andrea Pascucci

Co-Supervisore

Dr. Luca Caputo

Esame finale anno 2023

Abstract

We start in Chapter 2 to investigate linear matrix-valued SDEs and the Itô-stochastic Magnus expansion. This is the content of Kamm, Pagliarani and Pascucci (2021, 2022). The Itô-stochastic Magnus expansion provides an efficient numerical scheme to solve matrix-valued SDEs. We show convergence of the expansion up to a stopping time τ and provide an asymptotic estimate of the cumulative distribution function of τ. Moreover, we show how to apply it to solve SPDEs with one and two spatial dimensions by combining it with the method of lines with high accuracy. We will see that the Magnus expansion allows us to use GPU techniques leading to major performance improvements compared to a standard Euler-Maruyama scheme.

In Chapter 3, we study a short-rate model in a Cox-Ingersoll-Ross (CIR) framework for negative interest rates. It is based on Di Francesco and Kamm (2021, 2022). We define the short rate as the difference of two independent CIR processes and add a deterministic shift to guarantee a perfect fit to the market term structure. We show how to use the Gram-Charlier expansion to efficiently calibrate the model to the market swaption surface and price Bermudan swaptions with good accuracy.

Chapter 4 is based on Kamm (2022) and Kamm and Muniz (2022). We are taking two different perspectives for rating transition modelling. In Section 4.4, we study inhomogeneous continuous-time Markov chains (ICTMC) as a candidate for a rating model with deterministic rating transitions. We extend this model by taking a Lie group perspective in Section 4.5, to allow for stochastic rating transitions.

In both cases, we will compare the most popular choices for a change of measure technique and show how to efficiently calibrate both models to the available historical rating data and market default probabilities.

At the very end, we apply the techniques shown in this thesis to minimize the collateral-inclusive Credit/ Debit Valuation Adjustments under the constraint of small collateral postings by using a collateral account dependent on rating triggers.

Funding

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 813261 and is part of the ABC-EU-XVA project.

Acknowledgements

Special thanks goes to both of my supervisors Andrea Pascucci and Luca Caputo. Without them this thesis would not have been possible. They were always available, even in their holidays, for which I am most grateful.

Also, I would like to thank Stefano Pagliarani and Marco Di Francesco, to whom Andrea Pascucci introduced me. I had a lot of fun working together with both of them and thank them for their patience in answering my never ending questions.

Furthermore, I would like to thank Matthias Ehrhardt for introducing me to Michelle Muniz. Michelle and me are still working together on the extension of the rating model presented in Section 4.5 and I am very grateful to her for this wonderful collaboration.

Another big thank you goes to Luca Caputo and the XVA Quantitative team of Banco Santander in Madrid. We had a lot of discussions and struggled together to overcome the problems arising from real world data and applications. In particular, I would like to thank Jerome Vincent Maetz and Michael Belk for their support and valuable insights.

Not only was the research of this thesis very demanding but the times in which it was written also were very challenging. Without my friends and their support, I would have never finished this thesis on time. Therefore, I would like to thank especially Anne-Sophie Sur, Jonas Nispel and Nicolai Pitz for being there for me, whenever I needed them.

Abbreviations

Abbreviation	Meaning
CDF	Cumulative Distribution Function
CDS	Credit Default Swap
CIR	Cox-Ingersoll-Ross
CMS	Constant Maturity Swap
CTMC	Continuous Time Markov Chain
CVA	Credit Valuation Adjustment
DNN	Deep-Neural-Network
DVA	Debit Valuation Adjustment
GAN	Generative Adversarial Network
GRU	Gated Recurrent Unit
HW1	Hull-White one factor
ICTMC	Inhomogeneous Continuous Time Markov Chain
JLT	Jarrow-Lando-Turnbull
LGD	Loss-Given-Default
LSMC	Least-Square-Monte-Carlo/ Longstaff-Schwartz algorithm
LSTM	Long Short Term Memory
PDF	Probability Density Function
PHCTMC	Piecewise Homogeneous Continuous Time Markov Chain
SDE	Stochastic Differential Equation
SPDE	Stochastic Partial Differential Equation
VAE	Variational Autoencoder
XVA	X-Valuation Adjustment
gEM	geometric Euler-Maruyama

Repositories

The code and data for all the experiments conducted in this thesis is available in the following repositories:

Topic	URL
Magnus expansion (Chapter 2)	https://github.com/kevinkamm/StochasticMagnusExpansion
Magnus expansion (Chapter 2)	https://github.com/kevinkamm/MagnusSPDE2D
Magnus expansion (Chapter 2)	https://github.com/kevinkamm/IteratedMagnus
Interest rates (Chapter 3)	https://github.com/kevinkamm/CIR-
Interest rates (Chapter 3)	https://github.com/kevinkamm/CIR--
Rating models (Chapter 4)	https://github.com/kevinkamm/RatingTriggers
Rating models (Chapter 4)	https://github.com/kevinkamm/RatingML
Rating models (Chapter 4)	https://github.com/kevinkamm/LieRatingTriggers

Contents

1 Introduction

To introduce the topic of this thesis, let us consider the following situation:

Suppose that there are three parties, a bank, a company and another entity. The company asks the bank for a loan, which it agrees to, receiving interest on the notional in return. The contract between the company and bank is such that the interest rate changes with the market, resulting in a so-called *floating* position for the company. The company finds another entity, which agrees to swap the floating position with the company. This means that the new entity will pay the floating position to the company and receives a fixed amount from the company. If the payments due to the market interest rates get smaller than the fixed payments, the entity can profit from this interest rate swap. Otherwise, the entity will lose money and therefore takes over the risk inherited from the floating interest rates of the company. The risk for the company coming from the interest rates is therefore eliminated, since it will receive the floating position from the entity and can transfer it to the bank.

In a perfect world, this would be the end of the story. However, in the real world it can happen that the entity due to some reason is not able to pay the floating position to the company, for example if interest rates get very high. This situation is called *default* of an entity and a famous example of such an event was during the financial crisis in 2007, when Lehman Brothers defaulted.

Taking the possibility into account that any counterparty can default, is the subject of *valuation adjustments* for pricing financial derivatives. There are several types of valuation adjustments, called Credit Valuation Adjustment (CVA), Debit Valuation Adjustment (DVA), Funding Valuation Adjustment (FVA), Capital Valuation Adjustment (KVA), . . . The entire family of valuation adjustments is therefore denoted by XVA, where the “X” is a variable for the different types of the valuation adjustments.

In this thesis, we will focus on *collateral-inclusive* CVA and DVA. To reduce the potential loss in the event of a default, an effective method is to post collateral. This is a financial security, which usually is cash or bonds, counter-balancing the exposure to the contracting counterparty. For example, if interest rates get very high and the company and the entity agreed to post collateral, the entity would need to post more collateral into a secondary account to mitigate a potential loss due to a default. If the default would have happened in this situation, the company could take the collateral mitigating its loss and could pay the interest to the bank.

Now, that we understand the dangers of default events and the benefit of posting collateral, let us have a look at the general formula of collateral-inclusive bilateral CVA[1] (cf. Brigo, Morini and Pallavicini (2013): *pp. 312 ff. Chapter 13.2.3 CCVA and CDVA Definitions*)

[1] We disregard the possibility of rehypothecation for simplicity and use the convention $X^+ = \max(X, 0)$, $X^- = \min(X, 0)$.

$$\mathrm{CVA}_t\,(r, \mathrm{V}, \mathrm{C}) := \mathbb{E}^{\mathbb{Q}}\left[\mathrm{LGD}\ \exp\left(-\int_t^{\tau} r_s ds\right) \mathbb{1}_{\tau<T} \left(\mathrm{V}_{\tau}^{+} - \mathrm{C}_{\tau}^{+}\right)^{+} \middle| \mathcal{G}_t \right]. \tag{1.0.1}$$

The evaluation of the CVA takes place under a risk-neutral measure $\mathbb{Q}$ and the filtration $\mathcal{G}_t$ is the enlarged filtration containing information about both the default and the market.

The time of default prior to the end of contracts $T > 0$ of the entity is denoted by τ, the value of the portfolio at time t between the counterparty and the entity (there could be more than one swap contract) is denoted by V_t, the collateral account at time t by C_t and the discount factor seen from time t up to time u by $\exp\left(-\int_t^u r_s ds\right)$. This discount factor serves also as the numeraire under the aforementioned risk-neutral measure $\mathbb{Q}$. We will consider the loss-given-default (LGD) as a deterministic constant, which is usually set to 0.6.

To evaluate the CVA, we will use a Monte-Carlo approach and simulate the short-rate r_t, the value of the portfolio V_t, the collateral C_t, as well as the time of default τ. This divides the thesis into finding a suitable short-rate model and suitable definition of the collateral process.

Interest rate model. For the discount factor appearing in (1.0.1) we want to be able to account for negative interest rates. In a joint work with Marco Di Francesco, we propose in Chapter 3 a short-rate model based on the difference of two Cox-Ingersoll-Ross (CIR) processes, i.e. $r(t) := x(t) - y(t)$

$$dz(t) = k_z(\theta_z - z(t))dt + \sigma_z\sqrt{z(t)}dW_z(t),$$

where $z \in \{x, y\}$ are two CIR processes with standard independent Brownian motions W_z under the risk-neutral measure $\mathbb{Q}$, as well as $k_z, \theta_z, \sigma_z \in \mathbb{R}_{>0}$.

The CIR model and extensions have been studied extensively in the literature due to the fact that a single CIR process is guaranteed to be positive. Historically, this was a desirable feature of an interest rate model. However, recently a paradigm-shift occurred and the possibility of negative interest rates arose. Of course, other factors, such as its analytical tractability and a slightly skewed non-normal distribution made CIR models a popular choice among short-rate models.

We want to preserve as many of the advantageous features of the original CIR model as possible, and decided to use two independent CIR processes and take the difference of them both. Taking the difference of two positive processes makes it possible to reach negative values. In Section 3.3, we show an *endogenous* short-rate model, meaning that the *zero-coupon curve* will be an output of the model. Such models are valuable for fast vanilla option pricing and benchmarking. Afterwards, we show in Section 3.4 how to extend the endogenous model by adding a so-called *deterministic shift*, i.e. $r(t) = x(t) - y(t) + f(t)$, where $f(t)$ is a function containing the market instantaneous forward rate. This allows for a perfect fit to the market zero-coupon curve and gives the possibility to calibrate the short-rate model to market volatilities. In particular, we show how to calibrate this model to the market swaption surface by applying the Gram-Charlier expansion. We will see a close match of Bermudan swaption prices obtained by Bloomberg's Hull-White one factor model and our CIR model. For Constant Maturity Swaps (CMS) we will see similar good results making this simple model a good choice for our application.

Collateral model. Collateral is a good way to minimize CVA and DVA. In fact, setting $\mathrm{C}_t = \mathrm{V}_t$ would ensure that the CVA is always zero. This scenario is called *perfect collateralization* for this reason. However, there are some problems intertwined with this choice. One problem is that collateral cannot be posted in continuous time. Therefore, a rebalancing of the collateral account takes place at pre-set discrete points in time. In between these posting dates the CVA will not be zero anymore and the further the posting dates are apart from each other, the greater the CVA can be. Moreover, suppose that we have counterparties with a lot of contracts, e.g. their portfolio has a value of one million Euros. A perfect collateralization forcing a counterparty to post a million Euros would put a lot of stress on the counterparty maybe also increasing its probability to default.

A second problem, which will be at the heart of this thesis, is the simple fact that posting collateral is expensive. Let us consider a bank with several clients. This bank would like to avoid to post collateral of several million Euros to its clients, since this money could be reinvested and in the case of cash collateral postings, this money usually has to be borrowed.

Conclusively, collateral itself will not decrease the risk of default but decreases the potential loss at default and the size of the valuation adjustments. Hence, we are interested in a minimization of CVA over the collateral under the constraint of posting as little collateral as possible, while simultaneously keeping the potential loss small. One possibility to engage this vaguely formulated optimization problem, is to introduce another measure of a counterparty's risk, namely a concept named *rating*.

A rating is an indicator of the creditworthiness of an entity. A high rating associates less risk to an entity to not fulfill its financial obligations and a low rating a high risk. Ratings are usually denoted by letters **A**, **B**, …, **D**, where **A** denotes the best rating and **D** denotes the worst rating. The rating **D** is special. It means that an entity has defaulted, i.e. it can not fulfill its financial obligation towards a contracting party.

With this new indicator of creditworthiness, we would like to model the collateral as $\mathrm{C}_t := f(\mathrm{V}_t, X_t)$, where X_t is a stochastic process simulating the rating of an entity at time t. Depending on the current rating of the entity, we will set a certain threshold of unsecured money. For example, if the entity is in the best rating, we can set the threshold to infinity, corresponding to an *uncollateralized scenario*, and if the entity is close to default, we can set the threshold to zero and force a perfectly collateralized scenario. By adjusting the thresholds, this serves as a good compromise for posting as little collateral as possible while keeping the potential loss small. But even setting the thresholds to zero for one rating prior to default does not ensure a perfect collateralization, since immediate jumps from high ratings to low ratings are a possibility. We will discuss this phenomenon in Section 4.7.2.

Now, that we have a plan to tackle the constraint optimization problem, we have to understand how we can use the available historical and market data to calibrate a rating process

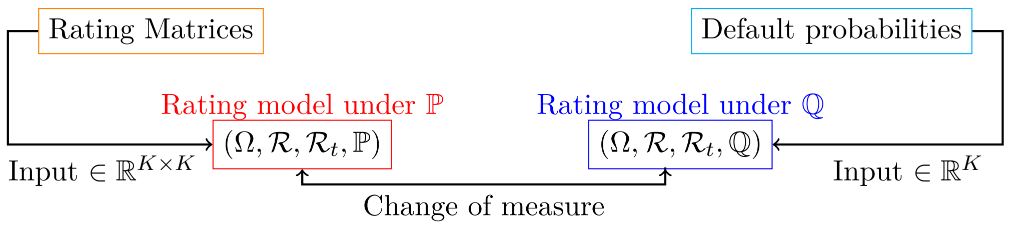

Rating Matrices
Default probabilities
Rating model under $\mathbb{P}$
Rating model under $\mathbb{Q}$
$(\Omega, \mathcal{R}, \mathcal{R}_t, \mathbb{P})$
$(\Omega, \mathcal{R}, \mathcal{R}_t, \mathbb{Q})$
Input $\in \mathbb{R}^{K\times K}$
Input $\in \mathbb{R}^{K}$
Change of measure

Figure 1.1.: Overview of different data for rating modelling.

of an entity and how to model such a process, which is the content of Chapter 4.

There are two major sources of data available. On the one hand, under the risk-neutral measure we have estimates of the probability of default of an entity if Credit Default Swap (CDS) quotes are available in the market. On the other hand, under the historical measure we have access to so-called *rating matrices* published by rating agencies, such as Moody's, S&P and Fitch. Rating matrices have entries containing the probability of transitioning from an initial rating to another rating in a certain amount of time. For example, one gets an estimation of the probability of transitioning from the best rating directly to default in one year.

These matrices are obtained from historical data of a sector of entities. In particular, there are different rating matrices for the financial sector and the corporate sector. In Figure 1.1, we give an overview, how this data can be used to calibrate a rating model. One approach in the literature introduced by Jarrow, Lando and Turnbull (1997) is to model ratings as a Continuous Time Markov Chain (CTMC). In this setting, the rating matrices can be viewed as the transition operators associated to the CTMC and the default data available under the risk-neutral measure corresponds to the default column of this transition operator after applying a change of measure to the CTMC.

In Section 4.4, we will demonstrate how to use a simple inhomogeneous CTMC (ICTMC) to model the ratings of an entity. The proposed model is leveraging on the expertise of Luca Caputo, Jerome Vincent Maetz and Michael Belk, who are part of the XVA Quantitative team of Banco Santander in Madrid. In this approach, the transition operators are assumed to be deterministic and can be calibrated to rating matrices after some work on the data. We will discuss the issues coming from the rating data in more detail in Section 4.3. For the change of measure we will compare two different techniques leading to very different rating models under the risk-neutral measure.

In Section 4.5, we will extend the ICTMC model further, which is joint work with Michelle Muniz. We will take a novel, geometrical point of view of this problem and show how to model stochastic rating transitions leading to matrix-valued SDEs in a certain subset of a Lie group.

Matrix-valued SDEs. In a joint work with Stefano Pagliarani and Andrea Pascucci, we will begin in this thesis by studying matrix-valued linear Itô-SDEs in Chapter 2. To be more precise, we will consider

$$dX_t = B_t X_t dt + \sum_{j=1}^{q} A_t^{(j)} X_t dW_t^j, \quad X_0 = I_d \in \mathbb{R}^{d,d} \tag{1.0.2}$$

where W_t^j are independent standard Brownian motions and the matrix-valued processes B_t, A_t^j, $j = 1, \ldots, q \in \mathbb{N}$ are progressively measurable, bounded processes.

Under these assumptions a unique solution is guaranteed and we want to find a specific expression of the solution. We will, like in the scalar case, show that the solution has an exponential representation $X_t := \exp(Y_t)$ for small times $t > 0$ depending on a stopping time and Y_t can be expressed by a series representation of Itô processes. This method is called *Magnus expansion.* The truncation of the series representation of Y_t will give rise to a numerical scheme for (1.0.2), which has *parallel-in-time* and *parallel-in-simulation* features. Using GPUs and multiple CPUs, we show that this can lead to major performance improvements compared to standard Euler-Maruyama techniques, while having an excellent accuracy, and apply it to solving SPDEs with one and two spatial dimensions. We demonstrate in the case of the stochastic heat equation and the stochastic Langevin equation the merits of this numerical scheme.

The deterministic counterpart of the Magnus expansion is often applied in situations, where a certain geometry needs to be preserved by the numerical scheme. In fact, this gave us the idea to study rating transitions from a Lie group perspective.

Aim and contribution of this thesis. First of all, to the best of our knowledge, this is the first time that the Itô-stochastic Magnus expansion has been derived for the general setting of progressively measurable and bounded coefficients. We show for a wide class of parabolic SPDEs that it leads to significant numerical improvements, making SPDE models viable for many applications.

Second, we show that a simple short-rate model by subtracting two CIR processes leads to very good results in a negative interest rate framework.

Third, we extend the class of models used for rating modelling. In the literature, usually time-homogeneity is assumed and we show that this can be overcome by a simple ICTMC model. Additionally, to the best of our knowledge, this is the first time that rating models were considered from a Lie group perspective, which gives rise to an entirely new methodology to define rating models. Moreover, we show that it is feasible to calibrate stochastic and fully inhomogeneous rating transition models to the available data. We compare the impact of the most popular choices for the change of measure used in the literature and show the benefit of using stochastic models over simpler ICTMC models.

In total, we will see a variety of different concepts and techniques, ranging from matrix-valued SDEs, SPDEs, the Gram-Charlier expansion, the stochastic simulation algorithm to Machine learning for time-series analysis and matrix Lie groups.

The Itô-Stochastic Magnus Expansion

2

The Magnus expansion is a classical tool to solve non-autonomous linear differential equations. Generalizations of the Magnus expansion to *Stratonovich SDEs* are well-known and were proposed by several authors (see for instance Ben Arous (1989), Blanes et al. (2009), Burrage and Burrage (1999), Wang et al. (2020) and the references given in Section 2.1). In this chapter, we derive the Magnus expansion for *Itô SDEs* under general assumptions, which do not guarantee an explicit Itô-Stratonovich conversion, namely progressively measurable stochastic coefficients. Our main results are the convergence of the stochastic Magnus expansion up to a stopping time τ and a novel asymptotic estimate of the cumulative distribution function of τ. The latter improves some previous estimates obtained in purely Markovian settings and is based on an application of Morrey's inequality. We also show, how the Magnus expansion in conjunction with the *method of lines* can be efficiently applied to the numerical solution of stochastic partial differential equations (SPDEs) in Section 2.4.3 and Section 2.4.4.

Let $d, q \in \mathbb{N}$ and consider the linear matrix-valued Itô SDE

$$\begin{cases} dX_t = B_t X_t dt + A_t^{(j)} X_t dW_t^j, \\ X_0 = I_d, \end{cases} \tag{2.0.1}$$

with $A^{(1)}, \dots, A^{(q)}, B$ being real $(d \times d)$-matrix-valued bounded stochastic processes, I_d the identity $(d \times d)$-matrix and $W = (W^1, \dots, W^q)$ a q-dimensional standard Brownian motion. In (2.0.1), as well as anywhere throughout the chapter, we use Einstein summation convention to imply summation of terms containing W^j, over the index j from 1 to q.

In the deterministic case, i.e. $A^{(j)} \equiv 0$, $j = 1, \dots, q$, (2.0.1) reduces to the matrix-valued ODE

$$\begin{cases} \frac{d}{dt} X_t = B_t X_t, \\ X_0 = I_d, \end{cases} \tag{2.0.2}$$

which admits in the time-homogeneous case an explicit solution in terms of the matrix exponential. Namely, if $B_t \equiv B$, the unique solution to (2.0.2) reads as

$$X_t = e^{tB}, \qquad t \geq 0.$$

However, in the non-autonomous case, the ODE (2.0.2) does not admit an explicit solution. In particular, if B_t is not constant, the solution X_t typically differs from $e^{\int_0^t B_s ds}$. This is due to the fact that, in general, B_t and B_s do not commute for $t \neq s$. As it turns out, a representation of the solution in terms of a matrix exponential is still possible, at least for short times, i.e.

$$X_t = e^{Y_t}, \tag{2.0.3}$$

for $t \geq 0$ suitably small and Y_t a real valued $(d \times d)$-matrix. Moreover, Y admits a semi-explicit expansion as a series of iterated integrals involving nested Lie commutators of the function B at different times. Such representation is known as *Magnus expansion* (Magnus (1954)) and its first terms read as

$$\begin{aligned} Y_t = \int_0^t B_s ds + \frac{1}{2}\int_0^t ds \int_0^s [B_s, B_u] du \\ + \frac{1}{6}\int_0^t ds \int_0^s du \int_0^u ([B_s, [B_u, B_r]] + [B_r, [B_u, B_s]]) dr + \cdots , \end{aligned} \tag{2.0.4}$$

where $[A, B] \coloneqq AB - BA$ denotes the Lie commutator. The Magnus expansion has a wide range of physical applications and the related literature has grown increasingly over the last decades (see, for instance, the excellent survey paper Blanes et al. (2009) and the references given therein).

In the stochastic case, when $j = 1$, $B_t \equiv 0$ and A is constant, i.e. $A_t(\omega) \equiv A$, the Itô equation (2.0.1) reduces to

$$\begin{cases} dX_t = AX_t dW_t, \\ X_0 = I_d, \end{cases}$$

whose explicit solution can be easily proven to be of the form (2.0.3), with

$$Y_t = -\frac{1}{2}A^2 t + AW_t, \qquad t \geq 0.$$

In general, when the matrices $A_t^{(j)}, A_s^{(j)}, B_t, B_s$ with $t \neq s$ do not commute, an explicit solution to (2.0.1) is not known. For instance, in the non-commutative case, neither the equation

$$\begin{cases} dX_t = Bdt + AX_t dW_t, \\ X_0 = I_d, \end{cases} \tag{2.0.5}$$

nor the equation

$$\begin{cases} dX_t = A_t X_t dW_t, \\ X_0 = I_d, \end{cases} \tag{2.0.6}$$

admit an explicit solution, save some particular cases (see for instance the example in Section 2.4.2).

Among the approximation tools that were developed in the literature to solve stochastic differential equations, including (2.0.1), some Magnus-type expansions that extend (2.0.3)-

(2.0.4) were derived in different contexts. We now go on to describe our contribution to this stream of literature, and then to firm our results within the existing ones. In particular, a detailed comparison with existing stochastic Magnus expansions previously derived by other authors will be provided below, in Section 2.1.

Description of the main results. In this chapter, we derive a Magnus-type representation formula for the solution to the Itô SDE (2.0.1), which is (2.0.3) together with

$$Y_t = Y_t^{(1)} + Y_t^{(2)} + Y_t^{(3)} + \cdots \qquad t \in [0, \tau], \tag{2.0.7}$$

for τ suitably small, strictly positive stopping time. In analogy with the deterministic Magnus expansion, the general term $Y^{(n)}$ can be expressed recursively, and contains iterated stochastic integrals of nested Lie commutators of the processes $B, A^{(j)}$ at different times.

In the case $j = 1$, the first two terms of the expansion read as

$$\begin{aligned}
Y_t^{(1)} &= \int_0^t B_s ds + \int_0^t A_s dW_s, \\
Y_t^{(2)} &= \frac{1}{2}\int_0^t \left(\left[B_s, \int_0^s B_u du + \int_0^s A_u dW_u \right] - A_s^2 \right) ds \\
&\quad + \frac{1}{2}\int_0^t \left[A_s, \int_0^s B_u du + \int_0^s A_u dW_u \right] dW_s.
\end{aligned}$$

For example, in the case of SDE (2.0.5) the latter can be reduced to

$$\begin{aligned}
Y_t^{(1)} &= Bt + AW_t, \\
Y_t^{(2)} &= [A, B] \left(\frac{1}{2} t W_t - \int_0^t W_s ds \right) - \frac{1}{2} A^2 t.
\end{aligned}$$

Notice that the last expressions do not contain stochastic integrals. In fact, in the general autonomous case, and if $j = 1$, all the iterated stochastic integrals in $Y^{(n)}$ can be solved for any $n \in \mathbb{N}$ (cf. Kloeden and Platen (1992): *p. 171 Corollary 5.2.4*). Therefore, in this case the expansion becomes numerically computable by only approximating Lebesgue integrals, as opposed to approximating stochastic integrals. As we shall see in the numerical tests in Section 2.4.2, this feature allows us to choose a sparser time-grid in order to save computation time. This feature is also preserved in some non-autonomous cases as illustrated in Kamm, Pagliarani and Pascucci (2021).

In Section 2.4.1, we will discuss how the Magnus expansion can be applied *parallel-in-time* and *parallel-in-simulation* for small time intervals. Additionally, we explain how to apply the Magnus expansion iteratively to overcome the restraints due to its convergence radius, which we will see in Theorem 2.0.1, and why this method holds an advantage over the Euler-Maruyama scheme in Section 2.4.2–2.4.4. Furthermore, we will demonstrate how to utilize GPUs to speed up the scheme by taking advantage of its parallel-in-time features. These speed-ups will be most relevant in the case of solving SPDEs numerically, while preserving a high level of accuracy. We will show in Section 2.4.3 and Section 2.4.4 that the Magnus expansion can be applied for a wide range of SPDEs. In all cases, it is faster and more accurate than a corresponding Euler-Maruyama scheme.

As it often happens when deriving convergent (either asymptotically or absolutely) expansions, a formal derivation precedes the rigorous result: that is what we do for Equations (2.0.3)-(2.0.7) in Section 2.2.1. Just like the derivation of the deterministic Magnus expansion relies on the possibility of writing the logarithm Y as the solution to an ODE, in the stochastic case the first step consists in representing Y as the solution to an SDE. Such representation of Y will be more involved compared to the deterministic case because of the presence of the second order derivatives of the exponential map coming from the application of Itô's formula. This is a distinctive feature of our derivation with respect to other analogous results in the Stratonovich setting where the standard chain-rule applies. With the SDE representation for Y at hand, the expansion (2.0.7) stems, like in the deterministic case, from applying a Dyson-type perturbation procedure to the SDE solved by Y.

In the deterministic case, the convergence of the Magnus expansion (2.0.4) to the exact logarithm of the solution to (2.0.2) was studied by several authors, who proved progressively sharper lower bounds on the maximum $\bar{t}$ such that the convergence to the exact solution is assured for any $t \in [0, \bar{t}]$. At the best of our knowledge, the sharpest estimate was given in Moan and Niesen (2008), namely

$$\bar{t} \geq \sup \left\{ t \geq 0 \mid \int_0^t \|B_s\| \, ds < \pi \right\}, \tag{2.0.8}$$

where $\|B_s\|$ denotes the spectral norm. Note that the existence of a real logarithm of X_t is an issue that underlies the study of the convergence of the Magnus expansion. We state here our main result, proven in Section 2.3, which deals with these matters in the stochastic case, when the coefficients $B, A^{(j)}$ in (2.0.1) are progressively measurable processes. We defer a comparison with previous convergence results for stochastic Magnus-type expansions to the next subsection. We denote by $\mathcal{M}^{d\times d}$ the space of the $(d \times d)$-matrices with real entries. Also, for an $\mathcal{M}^{d\times d}$-valued stochastic process $M = (M_t)_{t\in[0,T]}$, we set $\|M\|_T := \| \|M\|_F \|_{L^\infty([0,T]\times\Omega)}$, where $\|\cdot\|_F$ denotes the Frobenius norm.

Theorem 2.0.1. *Let $A^{(1)}, \dots, A^{(q)}$ and B be bounded, progressively measurable, $\mathcal{M}^{d\times d}$-valued processes defined on a filtered probability space $(\Omega, \mathcal{F}, \mathbb{P}, (\mathcal{F}_t)_{t\geq 0})$ equipped with a standard q-dimensional Brownian motion $W = (W^1, \cdots, W^q)$. For $T > 0$ let also $X = (X_t)_{t\in[0,T]}$ be the unique strong solution to (2.0.1) (see Lemma 2.3.3). There exists a strictly positive stopping time $\tau \leq T$ such that:*

1. X_t has a real logarithm $Y_t \in \mathcal{M}^{d\times d}$ up to time τ, i.e.

$$X_t = e^{Y_t}, \qquad 0 \leq t < \tau; \tag{2.0.9}$$

2. the following representation holds $\mathbb{P}$-almost surely:

$$Y_t = \sum_{n=0}^{\infty} Y_t^{(n)}, \qquad 0 \leq t < \tau, \tag{2.0.10}$$

where $Y^{(n)}$ is the n-th term in the stochastic Mangus expansion as defined in (2.2.20) *and* (2.2.23)–(2.2.25)*;*

3. *there exists a positive constant C, only dependent on $\|A^{(1)}\|_T, \dots, \|A^{(q)}\|_T$, $\|B\|_T$, T and d, such that*

$$\mathbb{P}(\tau \le t) \le Ct, \qquad t \in [0,T]. \tag{2.0.11}$$

The first point of Theorem 2.0.1 tells us that the Magnus expansion only converges up to a stopping time: to overcome this restriction, the numerical implementation of the Magnus expansion requires to apply it iteratively in time. Clearly, by (2.0.11) the convergence of the Magnus expansion is problem-dependent, meaning that there is no universal best time step-size for the Magnus expansion. The second point of Theorem 2.0.1 actually yields the numerical scheme by truncating the infinite series (2.0.10). We will see that in practice it is sufficient to consider only two or three terms to obtain a good degree of accuracy.

The proof of the first point in Theorem 2.0.1 relies on the continuity of X together with a standard representation for the matrix logarithm. The key point in the proof of the second point consists in showing that $X_t^{\epsilon,\delta}$ and its logarithm $Y_t^{\epsilon,\delta}$ are holomorphic as functions of (ϵ,δ), where $X_t^{\epsilon,\delta}$ represents the solution of (2.0.1) when $A^{(j)}$ and B are replaced by $\epsilon A^{(j)}$ and δB, respectively. Once this is established, the representation (2.0.10) follows from observing that, by construction, the series in (2.0.10) is exactly the formal power series of $Y_t^{\epsilon,\delta}$ at $(\epsilon,\delta) = (1,1)$. To prove the holomorphicity of $X_t^{\epsilon,\delta}$ we follow the same approach typically adopted to prove regularity properties of stochastic flows. Namely, in Lemma 2.3.3 we state some maximal L^p and Hölder estimates (with respect to the parameters) for solutions to SDEs with random coefficients and combine them with the Kolmogorov continuity theorem. Finally, the proof of the third point owes one more time to the L^p estimates in Lemma 2.3.3 and to a Sobolev embedding theorem to obtain pointwise estimates with respect to the parameters (ϵ,δ) above.

Theorem 2.0.1 has been used in the recent paper Yang et al. (2021) (cf. Lemma 1) where a semi-linear non-commuative Itô-SDEs is studied and Euler, Milstein and derivative-free numerical schemes are developed, with a convergence analysis for those schemes.

In the last part of the chapter, we perform numerical tests with the Magnus expansion. In particular, Section 2.4.3 is devoted to the application of the stochastic Magnus expansion to the numerical solution of parabolic stochastic partial differential equations (SPDEs). The idea is to discretize the SPDE only in space, which is usually called *method-of-lines*, and then approximate the resulting linear matrix-valued SDE by truncating the series in (2.0.9)-(2.0.10). In Section 2.4.4 we add another spatial dimension to the SPDE in Section 2.4.3 and show that the Magnus expansion is an efficient tool for solving even two-dimensional SPDEs. The goal here is to propose the application of stochastic Magnus expansion as novel approximation tools for SPDEs; we study the error of this approximating procedure only numerically. In the case where an explicit benchmark is available, we perform several tests with different parameter choices for the step-size of the iterated Magnus expansion. However, we defer the theoretical error analysis to further studies.

The rest of the chapter is structured as follows. In Section 2.2.1 we prove the key Lemma 2.2.1 with a representations for the first and second order differentials through which the terms $Y^{(n)}$ in (2.0.9)-(2.0.10) will be defined, and some preliminary results that will be used to derive the expansion. Additionally, we will show how to derive the Magnus expansion formulas (2.0.9)-(2.0.10) formally. With the help of further auxiliary results, we prove in Section 2.3 the main result Theorem 2.0.1. Afterwards, in Section 2.4 we first introduce numerical scheme stemming from the truncation of the Magnus expansion series and discuss possible ways to implement this method in Section 2.4.1. This is followed by a numerical test for an SDE with non-commuting coefficients of different generality in Section 2.4.2. Last but not least, we apply the Magnus expansion to the numerical solution of SPDEs with one and two spatial dimension in Section 2.4.3 and Section 2.4.4, respectively.

In the next section, we will review the literature in more detail.

2.1. Literature Review

Stochastic generalizations of the Magnus expansion were proposed by several authors. To the best of our knowledge, we recognize mainly two streams of research.

The beginning of the first one can be traced back to the work Ben Arous (1989), where the author derived *exponential* stochastic Taylor expansions (see also Azencott (1982), Kloeden and Platen (1992) for general stochastic Taylor series) of the solution of a system of Stratonovich SDEs with values on a manifold $\mathcal{M}$, i.e.

$$\begin{cases} dX_t = B(X_t)dt + A^{(j)}(X_t) \circ dW_t^j, \\ X_0 = x_0, \end{cases} \tag{2.1.1}$$

with $B, A^{(j)}$ being smooth, deterministic and autonomous vector fields on $\mathcal{M}$.

The stochastic flow of (2.1.1) is represented in terms of the exponential map of a stochastic vector field Y, i.e.

$$X_t(x_0) = \exp Y_t(x_0),$$

the vector field Y being expressed by an infinite series of iterated stochastic integrals multiplying nested commutators of the vector fields $B, A^{(j)}$. This representation is proved up to a strictly positive stopping time and extends some previous results in Doss (1977), Sussmann (1988) for the commutative case and in Yamato (1979), Kunita (1980), Fliess and Normand-Cyrot (1982) for the nilpotent case. Refinements of Ben Arous (1989) were proven in Castell (1993) making the expansion of Y more explicit. Later, numerical methods based on these representations were proposed in Castell and Gaines (1995) and Castell and Gaines (1996). Such techniques, known as *Castell-Gaines methods*, require the approximation of the solution to a time-dependent ODE besides the approximation of iterated stochastic integrals. Truncating the expansion of Y at a specified order, these schemes turn out to be asymptotically efficient in the sense of Newton (1991).

If $\mathcal{M} = \mathcal{M}^{d\times d}$ and the vector fields are linear, then (2.1.1) reduces to the Stratonovich version of (2.0.1) with $B, A^{(j)}$ constant matrices, and the representation of X given in Ben Arous (1989) can be seen as a stochastic Magnus expansion, in that the exponential map of Y reduces to the multiplication by a matrix exponential. In fact, in this case the expansion in Ben Arous (1989) becomes explicit in terms of iterated stochastic integrals, and can be shown to coincide with the expansion in this paper by applying Itô-Stratonovich conversion formula. In the very interesting paper Lord, Malham and Wiese (2008), the authors study several computational aspects of numerical schemes stemming from the truncated Magnus expansion, in which the iterated stochastic integrals are approximated by their conditional expectation. Besides showing that asymptotic efficiency holds for an arbitrary number of Brownian components, they compare the theoretical accuracy with the one of analogous schemes based on Dyson (or Neumann) series, which are obtained by applying stochastic Taylor expansion directly on the equation. They find that, although the theoretical accuracy of Magnus schemes is not superior, Magnus-based approximations seem more accurate than their Dyson counterparts in practice. They also discuss the computational cost deriving from approximating the iterated stochastic integrals and the matrix exponentiation, in relation to different features of the problem such as the dimension and the number of Brownian motions, as well as to the order of the numerical scheme.

The second stream of literature is explicitly aimed at extending the original Magnus results to stochastic settings and can be traced back to Burrage and Burrage (1999) where the Magnus expansion is derived via formal arguments for a linear system of Stratonovich SDEs with deterministic coefficients. Clearly, in the autonomous case such expansion coincides with the one obtained by Ben Arous (1989), whereas in the non-autonomous case, $B \equiv 0$ and $j = 1$, it is formally equivalent to the deterministic Magnus expansion (2.0.4) with all the Lebesgue integrals replaced by Stratonovich ones. The authors of Burrage and Burrage (1999) do not address the convergence of the Magnus expansion, but rather study computational aspects of the resulting approximation, in particular in comparison with Runge-Kutta stochastic schemes. Marjanovic and Solo (2018) consider the Itô SDE (2.0.1) with constant coefficients, and propose to resolve via Euler method the SDE (2.2.16) for the logarithm of the solution. In Wang et al. (2020) the Magnus expansion for the Stratonovich version of (2.0.1) with deterministic coefficients is applied to solve non-linear SDEs; however, the error analysis of the truncated expansion seems flawed, since the fact that the Magnus series converges only up to a positive stopping time is overlooked. In Muniz et al. (2022a), a general procedure for designing higher strong order methods for Itô SDEs on matrix Lie groups is outlined by applying Runge-Kutta-Munthe-Kaas methods. Most recently an extension to the non-linear case was derived in Muniz et al. (2022b).

We now go on to discuss the contribution of this chapter with respect to the existing literature. In the first place, Theorem 2.0.1 on the convergence of the Magnus expansion requires very weak conditions on the coefficients, which are stochastic processes satisfying the sole assumption of progressive measurability. This is a novel aspect compared to the results in Ben Arous (1989), Castell (1993), which surely cover a wider class of SDEs, but under the assumption of time-independent deterministic coefficients. We point out that this feature is also relevant in light of the fact that our result is stated for Itô SDEs as opposed to Stratonovich ones. Indeed, while this difference might appear as minor in the Markovian case, where a simple conversion formula exists (cf. Correales and Escudero (2018) and Kuo (2006)), it becomes substantial in the case of progressively measurable coefficients. We also point out that, even in the Markovian non-autonomous case, convergence issues were not discussed in Burrage and Burrage (1999) and Lord, Malham and Wiese (2008).

Another novel aspect of our result concerns the estimate (2.0.11) for the cumulative distribution function of the stopping time τ up to which the Magnus series converges to the real logarithm of the solution: this kind of estimate was unknown even in the autonomous case. In Ben Arous (1989): *Theorem 11* (see also Castell (1993)) an asymptotic estimate for the truncation error of the logarithm was provided, which in the linear case studied in this paper would read as

$$Y_t = \sum_{n=1}^{N} Y_t^{(n)} + t^{\frac{N+1}{2}} R_t, \qquad t < T,$$

with R bounded in probability. Although this type of result holds for the general SDE (2.1.1), it is weaker than Theorem 2.0.1 in the linear case. In fact, it can be obtained by (2.0.11) together with the standard estimate $\left\| \sup_{0\le s\le t} \|Y_s^{(n)}\|_F \right\|_{L^2(\Omega)} \le Ct^{\frac{N+1}{2}}$, but not the other way around.

A rigorous error analysis of the Magnus expansion is left for future research, as well as applications to non-linear SDEs (see Wang et al. (2020) for a recent attempt in this direction).

The results of this chapter have recently found applications in the study of so-called signature cumulants Friz, Hager and Tapia (2022), semi-linear Itô SDEs Yang et al. (2021), linear SDEs on matrix Lie groups Muniz et al. (2022a), stochastic modelling of motion in turbulent flows Campana (2022), stability of multi-variate geometric Brownian motion Barrera, Högele and Pardo (2022) and modelling rating transition matrices in quantitative finance Kamm and Muniz (2022).

2.2. Preliminaries and Formal Derivation

In this section, we define the terms in the expansion (2.0.10) and prove Theorem 2.0.1.

Let $\mathcal{M}^{d\times d}$ be the vector space of $(d \times d)$ real-valued matrices. Throughout the paper we denote by $[\cdot,\cdot]$ the standard Lie commutator, i.e.

$$[M, N] := MN - NM, \qquad M, N \in \mathcal{M}^{d\times d},$$

and by $\|\cdot\|$ the spectral norm on $\mathcal{M}^{d\times d}$. Also, we denote by β_k, $k \in \mathbb{N}_0$, the Bernoulli numbers defined as the derivatives of the function $x \mapsto x/(e^x - 1)$ computed at $x = 0$. For sake of convenience we report the first three Bernoulli numbers: $\beta_0 = 1$, $\beta_1 = -\frac{1}{2}$, $\beta_2 = \frac{1}{6}$. Note also that $\beta_{2m+1} = 0$ for any $m \in \mathbb{N}$.

We now define the operators that we will use in the sequel. For a fixed $\Sigma \in \mathcal{M}^{d\times d}$, we let:

- $\mathrm{ad}_\Sigma^j : \mathcal{M}^{d\times d} \to \mathcal{M}^{d\times d}$, for $j \in \mathbb{N}_0$, be the linear operators defined as

$$\begin{aligned}\mathrm{ad}_\Sigma^0(M) &:= M,\\ \mathrm{ad}_\Sigma^j(M) &:= \left[\Sigma, \mathrm{ad}_\Sigma^{j-1}(M)\right], \qquad j \in \mathbb{N}.\end{aligned}$$

  To ease notation we also set $\mathrm{ad}_\Sigma := \mathrm{ad}_\Sigma^1$;

- $e^{\mathrm{ad}_\Sigma} : \mathcal{M}^{d\times d} \to \mathcal{M}^{d\times d}$ be the linear operator defined as

$$e^{\mathrm{ad}_\Sigma}(M) := \sum_{n=0}^{\infty} \frac{1}{n!}\mathrm{ad}_\Sigma^n(M) = e^\Sigma M e^{-\Sigma}, \tag{2.2.1}$$

  where $e^\Sigma := \sum_{j=0}^{\infty} \frac{\Sigma^j}{j!}$ is the standard matrix exponential;

- $\mathcal{L}_\Sigma : \mathcal{M}^{d\times d} \to \mathcal{M}^{d\times d}$ be the linear operator defined as

$$\mathcal{L}_\Sigma(M) := \int_0^1 e^{\mathrm{ad}_{\tau\Sigma}}(M)d\tau = \sum_{n=0}^{\infty} \frac{1}{(n+1)!}\mathrm{ad}_\Sigma^n(M); \tag{2.2.2}$$

- $\mathcal{Q}_\Sigma : \mathcal{M}^{d\times d} \times \mathcal{M}^{d\times d} \to \mathcal{M}^{d\times d}$ be the bi-linear operator defined as

$$\mathcal{Q}_\Sigma(M,N) := \mathcal{L}_\Sigma(M)\,\mathcal{L}_\Sigma(N) + \int_0^1 \tau\left[\mathcal{L}_{\tau\Sigma}(N), e^{\mathrm{ad}_{\tau\Sigma}}(M)\right]d\tau \tag{2.2.3}$$

$$= \sum_{n=0}^{\infty}\sum_{m=0}^{\infty} \frac{\mathrm{ad}_\Sigma^n(M)}{(n+1)!}\frac{\mathrm{ad}_\Sigma^m(N)}{(m+1)!} + \sum_{n=0}^{\infty}\sum_{m=0}^{\infty} \frac{[\mathrm{ad}_\Sigma^n(N), \mathrm{ad}_\Sigma^m(M)]}{(n+m+2)(n+1)!m!}. \tag{2.2.4}$$

In the next lemma, we provide explicit expressions for the first and second order differentials of the exponential map $\mathcal{M}^{d\times d} \ni M \mapsto e^M$. We recall that this map is smooth and in particular, it is continuously twice differentiable.

Lemma 2.2.1. *For any $\Sigma \in \mathcal{M}^{d\times d}$, the first and the second order differentials at Σ of the exponential map $\mathcal{M}^{d\times d} \ni M \mapsto e^M$ are given by*

$$M \mapsto \mathcal{L}_\Sigma(M)\,e^\Sigma = e^\Sigma\,\mathcal{L}_{-\Sigma}(M), \qquad M \in \mathcal{M}^{d\times d}, \tag{2.2.5}$$

$$(M,N) \mapsto \mathcal{Q}_\Sigma(M,N)\,e^\Sigma = e^\Sigma\,\mathcal{Q}_{-\Sigma}(N,M), \qquad M,N \in \mathcal{M}^{d\times d}, \tag{2.2.6}$$

where $\mathcal{L}_\Sigma$ and $\mathcal{Q}_\Sigma$ are the linear and bi-linear operators as defined in (2.2.2)*-*(2.2.3)*.*

We point out that this result, though very basic, is novel and of independent interest (for instance it was recently employed in Friz, Hager and Tapia (2022)).

Proof. The first part of the statement, concerning the first order differential, is a classical result; its proof can be found in Blanes et al. (2009): *Lemma 2* among other references.

We prove the second part.

Fix $M \in \mathcal{M}^{d\times d}$ and denote by $\partial_M e^\Sigma$ the first order directional derivative of e^Σ w.r.t. M, i.e.

$$\partial_M e^{\Sigma} := \frac{d}{dt} e^{\Sigma + tM}\Big|_{t=0}.$$

By the first part, we have

$$\partial_M e^{\Sigma} = \mathcal{L}_{\Sigma}(M)\, e^{\Sigma}, \qquad \Sigma \in \mathcal{M}^{d\times d}. \tag{2.2.7}$$

We now show that, for any $M, N \in \mathcal{M}^{d\times d}$, the second order directional derivative

$$\partial_{N,M} e^{\Sigma} := \frac{d}{dt} \partial_M e^{\Sigma + tN}\Big|_{t=0}$$

is given by

$$\partial_{N,M} e^{\Sigma} = \mathcal{Q}_{\Sigma}(N, M)\, e^{\Sigma}, \qquad \Sigma \in \mathcal{M}^{d\times d}. \tag{2.2.8}$$

We have

$$\begin{aligned} \frac{d}{dt} \partial_M e^{\Sigma+tN} &= \frac{d}{dt}\Big(\mathcal{L}_{\Sigma+tN}(M)\, e^{\Sigma+tN}\Big) \\ &= \mathcal{L}_{\Sigma+tN}(M)\, \mathcal{L}_{\Sigma+tN}(N)\, e^{\Sigma+tN} + \Big(\frac{d}{dt}\mathcal{L}_{\Sigma+tN}(M)\Big)\, e^{\Sigma+tN}. \end{aligned} \tag{2.2.9}$$

We use the definition (2.2.2) and exchange the differentiation and integration signs to obtain

$$\frac{d}{dt}\mathcal{L}_{\Sigma+tN}(M) = \int_0^1 \frac{d}{dt} e^{\mathrm{ad}_{\tau(\Sigma+tN)}}(M) d\tau$$

(by (2.2.1))

$$\begin{aligned} &= \int_0^1 \frac{d}{dt}\Big(e^{\tau(\Sigma+tN)} M e^{-\tau(\Sigma+tN)}\Big) d\tau \\ &= \int_0^1 \Big(\frac{d}{dt} e^{\tau(\Sigma+tN)}\Big)\, M\, e^{-\tau(\Sigma+tN)} d\tau + \int_0^1 e^{\tau(\Sigma+tN)}\, M\, \frac{d}{dt} e^{-\tau(\Sigma+tN)} d\tau \end{aligned}$$

(by employing the two expressions in (2.2.5) for the first-order differential)

$$\begin{aligned} &= \int_0^1 \tau \mathcal{L}_{\tau(\Sigma+tN)}(N)\, e^{\tau(\Sigma+tN)}\, M\, e^{-\tau(\Sigma+tN)} d\tau \\ &\quad - \int_0^1 \tau e^{\tau(\Sigma+tN)}\, M\, e^{-\tau(\Sigma+tN)}\, \mathcal{L}_{\tau(\Sigma+tN)}(N) d\tau \\ &= \int_0^1 \tau \Big[\mathcal{L}_{\tau(\Sigma+tN)}(N), e^{\mathrm{ad}_{\tau(\Sigma+tN)}}(M)\Big]\, d\tau. \end{aligned}$$

This, together with (2.2.9), proves (2.2.8).

To conclude, we prove equality (2.2.4). It is enough to observe that

$$\begin{aligned}\int_0^1 \tau\, \mathcal{L}_{\tau\Sigma}(N)\, e^{\mathrm{ad}_{\tau\Sigma}}(M) d\tau &= \int_0^1 \tau \sum_{n=0}^{\infty}\sum_{m=0}^{\infty} \frac{\mathrm{ad}^n_{\tau\Sigma}(N)}{(n+1)!}\frac{\mathrm{ad}^m_{\tau\Sigma}(M)}{m!} d\tau \\ &= \int_0^1 \sum_{n=0}^{\infty}\sum_{m=0}^{\infty} \frac{\mathrm{ad}^n_{\Sigma}(N)}{(n+1)!}\frac{\mathrm{ad}^m_{\Sigma}(M)}{m!} \tau^{n+m+1} d\tau \\ &= \sum_{n=0}^{\infty}\sum_{m=0}^{\infty} \frac{\mathrm{ad}^n_{\Sigma}(N)\mathrm{ad}^m_{\Sigma}(M)}{(n+m+2)(n+1)!m!}.\end{aligned}$$

The identity $\mathcal{Q}_{\Sigma}(M,N)\, e^{\Sigma} = e^{\Sigma}\, \mathcal{Q}_{-\Sigma}(N,M)$ follows from repeating the same proof but using $e^{\Sigma}\, \mathcal{L}_{-\Sigma}(M)$ instead of $\mathcal{L}_{\Sigma}(M)\, e^{\Sigma}$ in (2.2.7) and the skew-symmetry of the commutator.

□

Proposition 2.2.2 (Itô formula). *Let Y be an $\mathcal{M}^{d\times d}$-valued Itô process of the form*

$$dY_t = \mu_t dt + \sigma_t^j dW_t^j. \tag{2.2.10}$$

Then we have

$$de^{Y_t} = \left(\mathcal{L}_{Y_t}(\mu_t) + \frac{1}{2}\sum_{j=1}^{q} \mathcal{Q}_{Y_t}(\sigma_t^j, \sigma_t^j)\right) e^{Y_t} dt + \sum_{j=1}^{q} \mathcal{L}_{Y_t}(\sigma_t^j)\, e^{Y_t} dW_t^j.$$

or equivalently by using the identities for the derivative operators in Lemma 2.2.1

$$de^{Y_t} = e^{Y_t}\left(\mathcal{L}_{-Y_t}(\mu_t) + \frac{1}{2}\sum_{j=1}^{q} \mathcal{Q}_{-Y_t}(\sigma_t^j, \sigma_t^j)\right) dt + \sum_{j=1}^{q} e^{Y_t}\, \mathcal{L}_{-Y_t}(\sigma_t^j) dW_t^j.$$

Proof. The statement follows from an adaption of the proof of the multi-dimensional Itô formula (see, for instance, Pascucci (2011)) combined with Lemma 2.2.1 and applied to the exponential process e^{Y_t}. □

We also have the following inversion formula for the operator $\mathcal{L}_{\Sigma}$.

Lemma 2.2.3 (Baker, 1905). *Let $\Sigma \in \mathcal{M}^{d\times d}$. The operator $\mathcal{L}_{\Sigma}$ is invertible if and only if the eigenvalues of the linear operator ad_{Σ} are different from $2m\pi$, $m \in \mathbb{Z}\setminus\{0\}$.*

Furthermore, if the spectral norm $\|\Sigma\| < \pi$, then

$$\mathcal{L}_{\Sigma}^{-1}(M) = \sum_{k=0}^{\infty} \frac{\beta_k}{k!}\mathrm{ad}_{\Sigma}^k(M), \qquad M \in \mathcal{M}^{d\times d}. \tag{2.2.11}$$

For a proof to Lemma 2.2.3 we refer the reader to Blanes et al. (2009).

2.2.1. Formal Derivation

In this section, we perform formal computations to derive the terms $Y^{(n)}$ appearing in the ME (2.0.10). Although such computations are heuristic at this stage, they are meant to provide the reader with an intuitive understanding of the principles that underlie the expansion procedure. Their validity will be proven a fortiori, in Section 2.3, in order to prove Theorem 2.0.1.

Let $(\Omega, \mathcal{F}, \mathbb{P}, (\mathcal{F}_t)_{t\geq 0})$ be a filtered probability space. Assume that, for any $\epsilon, \delta \in \mathbb{R}$, the process $X^{\epsilon,\delta} = \left(X_t^{\epsilon,\delta}\right)_{t\geq 0}$ solves the Itô SDE

$$\begin{cases} dX_t^{\epsilon,\delta} = \delta B_t X_t^{\epsilon,\delta} dt + \epsilon A_t^{(j)} X_t^{\epsilon,\delta} dW_t^j, \\ X_0^{\epsilon,\delta} = I_d, \end{cases} \tag{2.2.12}$$

and that it admits the exponential representation

$$X_t^{\epsilon,\delta} = e^{Y_t^{\epsilon,\delta}} \tag{2.2.13}$$

with $Y^{\epsilon,\delta}$ being an $\mathcal{M}^{d\times d}$-valued Itô process. Clearly, if $(\epsilon,\delta) = (1,1)$, then (2.2.12)-(2.2.13) reduce to (2.0.1)-(2.0.3).

Assume now that $Y^{\epsilon,\delta}$ is of the form (2.2.10). Then, our assumption and Proposition 2.2.2 yields

$$\begin{aligned} dX_t &= \delta B_t X_t^{\epsilon,\delta} dt + \epsilon A_t^{(j)} X_t^{\epsilon,\delta} dW_t^j \\ &= \delta B_t \exp\left(Y_t^{\epsilon,\delta}\right) dt + \epsilon A_t^{(j)} \exp\left(Y_t^{\epsilon,\delta}\right) dW_t^j \\ &= d \exp\left(Y_t^{\epsilon,\delta}\right) \\ &= \left(\mathcal{L}_{Y_t^{\epsilon,\delta}}(\mu_t) + \frac{1}{2}\sum_{j=1}^{q} \mathcal{Q}_{Y_t^{\epsilon,\delta}}\left(\sigma_t^j, \sigma_t^j\right)\right) \exp\left(Y_t^{\epsilon,\delta}\right) dt + \mathcal{L}_{Y_t^{\epsilon,\delta}}\left(\sigma_t^j\right) \exp\left(Y_t^{\epsilon,\delta}\right) dW_t^j. \end{aligned}$$

Now, if we compare the coefficients we have

$$\epsilon A_t^{(j)} = \mathcal{L}_{Y_t^{\epsilon,\delta}}\left(\sigma_t^j\right), \qquad j = 1, \dots, q, \tag{2.2.14}$$

$$\delta B_t = \mathcal{L}_{Y_t^{\epsilon,\delta}}(\mu_t) + \frac{1}{2}\sum_{j=1}^{q} \mathcal{Q}_{Y_t^{\epsilon,\delta}}\left(\sigma_t^j, \sigma_t^j\right). \tag{2.2.15}$$

Inverting now (2.2.14)-(2.2.15), in accord with Lemma 2.2.3, one obtains

$$\begin{aligned} \sigma_t^j &= \mathcal{L}^{-1}_{Y_t^{\epsilon,\delta}}\left(\epsilon A_t^{(j)}\right) = \epsilon \sum_{k=0}^{\infty} \frac{\beta_k}{k!} \mathrm{ad}^k_{Y_t^{\epsilon,\delta}}\left(A_t^{(j)}\right), \qquad j = 1, \dots, q, \\ \mu_t &= \mathcal{L}^{-1}_{Y_t^{\epsilon,\delta}}\left(\delta B_t - \frac{1}{2}\sum_{j=1}^{q} \mathcal{Q}_{Y_t^{\epsilon,\delta}}\left(\sigma_t^j, \sigma_t^j\right)\right) \\ &= \sum_{k=0}^{\infty} \frac{\beta_k}{k!} \mathrm{ad}^k_{Y_t^{\epsilon,\delta}}\Bigg(\delta B_t - \frac{1}{2}\sum_{j=1}^{q}\sum_{n=0}^{\infty}\sum_{m=0}^{\infty}\Bigg(\frac{\mathrm{ad}^n_{Y_t^{\epsilon,\delta}}\left(\sigma_t^j\right)}{(n+1)!}\frac{\mathrm{ad}^m_{Y_t^{\epsilon,\delta}}\left(\sigma_t^j\right)}{(m+1)!} \\ &\qquad + \frac{\left[\mathrm{ad}^n_{Y_t^{\epsilon,\delta}}\left(\sigma_t^j\right), \mathrm{ad}^m_{Y_t^{\epsilon,\delta}}\left(\sigma_t^j\right)\right]}{(n+m+2)(n+1)!m!}\Bigg)\Bigg). \end{aligned}$$

Equivalently, $Y^{\epsilon,\delta}$ solves the Itô SDE

$$\begin{cases} dY_t^{\epsilon,\delta} = \mu^{\epsilon,\delta}\left(t, Y_t^{\epsilon,\delta}\right) dt + \sigma_j^{\epsilon}\left(t, Y_t^{\epsilon,\delta}\right) dW_t^j, \\ Y_0^{\epsilon,\delta} = 0, \end{cases} \tag{2.2.16}$$

with

$$\sigma_j^{\epsilon}(t,\cdot) = \epsilon \sum_{n=0}^{\infty} \frac{\beta_n}{n!} \mathrm{ad}_\cdot^n (A_t^{(j)}), \qquad j = 1, \ldots, d, \tag{2.2.17}$$

$$\mu^{\epsilon,\delta}(t,\cdot) = \sum_{n=0}^{\infty} \frac{\beta_n}{n!} \mathrm{ad}_\cdot^n \left(\delta B_t - \frac{1}{2} \sum_{j=1}^{q} \mathcal{Q}_\cdot \left(\sigma_j^{\epsilon}(t,\cdot), \sigma_j^{\epsilon}(t,\cdot) \right) \right). \tag{2.2.18}$$

We now assume that $Y^{\epsilon,\delta}$ admits the representation

$$Y_t^{\epsilon,\delta} = \sum_{n=0}^{\infty} \sum_{r=0}^{n} Y_t^{(r,n-r)} \epsilon^r \delta^{n-r}, \tag{2.2.19}$$

for a certain family $(Y^{(r,n-r)})_{n,r\in\mathbb{N}_0}$ of stochastic processes. In particular, setting $(\epsilon, \delta) = (1, 1)$, (2.2.19) would yield

$$Y_t = \sum_{n=0}^{\infty} Y_t^{(n)} \quad \text{with} \quad Y_t^{(n)} := \sum_{r=0}^{n} Y_t^{(r,n-r)}. \tag{2.2.20}$$

Remark 2.2.4. Note that it is possible to re-order the double series $\sum\limits_{n=0}^{\infty} \sum\limits_{r=0}^{n} Y_t^{(r,n-r)}$ according to any arbitrary choice, for the latter will be proved to be absolutely convergent. The above choice for $Y^{(n)}$ contains all the terms of equal order by weighing ϵ and δ in the same way. A different choice, which respects the probabilistic relation $\sqrt{\Delta t} \approx \Delta W_t$, corresponds to weighing δ as ϵ^2. This would lead to setting

$$Y_t^{(n)} := \sum_{r=0}^{\lfloor \frac{n}{2} \rfloor} Y_t^{(n-2r,r)}$$

in (2.2.20).

Remark 2.2.5. Observe that, if the function $(\epsilon, \delta) \mapsto Y_0^{\epsilon,\delta}$ is assumed to be continuous $\mathbb{P}$-almost surely, then the initial condition in (2.2.16) implies

$$Y_0^{(i,j)} = 0 \quad \mathbb{P}\text{-a.s.}, \qquad i, j \in \mathbb{N}_0,$$

and thus

$$Y_0^{(n)} = 0 \quad \mathbb{P}\text{-a.s.}, \qquad n \in \mathbb{N}_0.$$

We now plug (2.2.19) into (2.2.16) and collect all terms of equal order in ϵ and δ. Let us do this for the first three orders in more detail to get an intuition for the general formula. Also as a side-note, this is in general a good way to compute the expansion formulas for specific cases as done in the appendix of Kamm, Pagliarani and Pascucci (2022) in the case of constant processes A_t and B_t.

Order 1. Let us derive the first-order Magnus expansion, meaning we are interested in all terms with the first power of ϵ and δ. Therefore, with a slight abuse of notation, we will in the following only write down the necessary terms for the function σ_j^ϵ and $\mu_j^{\epsilon,\delta}$. Additionally, we keep Einstein's summation convention over all the terms involving $A_t^{(i)}$ for readability.

We start with inserting $Y_t^{(0)} \equiv 0$ into (2.2.17). Notice, that the zero matrix commutes with all matrices and therefore by definition of $\mathrm{ad}_Y^0(A) = A$

$$\sigma_j^\epsilon\left(t, Y_t^{(0)}\right) = \epsilon A_t^{(j)}.$$

Similarly, we have for (2.2.18)

$$\mu^{\epsilon,\delta}\left(t, Y_t^{(0)}\right) = \delta B_t - \epsilon^2 \left(A_t^{(j)}\right)^2,$$

because $A_t^{(j)}$ commutes with itself. Since,the Itô-correction term is of order ϵ^2 it will not be part of the first-order Magnus expansion and we have

$$\epsilon^1\delta^0: \quad Y_t^{(1,0)} = \int_0^t A_s^{(j)} dW_s^j, \tag{2.2.21}$$

$$\epsilon^0\delta^1: \quad Y_t^{(0,1)} = \int_0^t B_s ds, \tag{2.2.22}$$

after setting our bookkeeping parameters to $\epsilon = \delta = 1$.

Order 2. Let us start again by plugging $Y_t^{(1)}$ into (2.2.17) and only consider the terms up to order 2, i.e. ϵ, δ, δ^2, $\epsilon\delta$ and ϵ^2

$$\begin{aligned}\sigma_j^\epsilon\left(t, Y_t^{(1)}\right) &\approx \frac{1}{0!}\mathrm{ad}^0_{Y_t^{(1)}}\left(\epsilon A_t^{(j)}\right) - \frac{1}{2}\mathrm{ad}^1_{Y_t^{(1)}}\left(\epsilon A_t^{(j)}\right)\\ &= \epsilon A_t^{(j)} - \frac{1}{2}\left[Y_t^{(1)}, \epsilon A_t^{(j)}\right]\\ &= \epsilon A_t^{(j)} - \frac{1}{2}\epsilon^2\left[\int_0^t A_s^{(i)} dW_s^i, A_t^{(j)}\right] - \frac{1}{2}\epsilon\delta\left[\int_0^t B_s ds, A_t^{(j)}\right],\end{aligned}$$

where we already discarded the possible terms of higher order. For (2.2.18) we get

$$\begin{aligned}
\mu^{\epsilon,\delta}\left(t,Y_t^{(1)}\right) \approx{}& \frac{1}{0!}\mathrm{ad}^0_{Y_t^{(1)}}\left(\delta B_t\right) - \frac{1}{2}\mathrm{ad}^1_{Y_t^{(1)}}\left(\delta B_t\right) \\
& - \frac{1}{2}\left(\frac{1}{0!}\left(\frac{\mathrm{ad}^0_{Y_t^{(1)}}\left(\sigma_j^\epsilon\left(t,Y_t^{(1)}\right)\right)}{1!}\frac{\mathrm{ad}^0_{Y_t^{(1)}}\left(\sigma_j^\epsilon\left(t,Y_t^{(1)}\right)\right)}{1!}\right.\right. \\
& \left.\left. + \frac{\left[\mathrm{ad}^0_{Y_t^{(1)}}\left(\sigma_j^\epsilon\left(t,Y_t^{(1)}\right)\right),\mathrm{ad}^0_{Y_t^{(1)}}\left(\sigma_j^\epsilon\left(t,Y_t^{(1)}\right)\right)\right]}{2}\right)\right) \\
={}& \delta B_t - \frac{1}{2}\epsilon\delta\left[\int_0^t A_s^{(i)}dW_s^i, B_t\right] - \frac{1}{2}\delta^2\left[\int_0^t B_s ds, B_t\right] - \frac{1}{2}\left(\left(\sigma_j^\epsilon\left(t,Y_t^{(1)}\right)\right)^2 + 0\right) \\
\approx{}& \delta B_t - \frac{1}{2}\epsilon\delta\left[\int_0^t A_s^{(i)}dW_s^i, B_t\right] - \frac{1}{2}\delta^2\left[\int_0^t B_s ds, B_t\right] - \frac{1}{2}\epsilon^2\left(A_t^{(i)}\right)^2.
\end{aligned}$$

Now, we collect again the terms corresponding to the order two expansion and integrate, i.e.

$$\begin{aligned}
\epsilon^2\delta^0 :\quad & Y_t^{(2,0)} = -\frac{1}{2}\int_0^t\left(A_s^{(j)}\right)^2 ds + \frac{1}{2}\int_0^t\left[A_s^{(j)},\int_0^s A_r^{(i)}dW_r^i\right]dW_s^j, \\
\epsilon^1\delta^1 :\quad & Y_t^{(1,1)} = \frac{1}{2}\int_0^t\left[B_s,\int_0^s A_r^{(j)}dW_r^j\right]ds + \frac{1}{2}\int_0^t\left[A_s^{(j)},\int_0^s B_r dr\right]dW_s^j, \\
\epsilon^0\delta^2 :\quad & Y_t^{(0,2)} = \frac{1}{2}\int_0^t\left[B_s,\int_0^s B_r dr\right]ds,
\end{aligned}$$

for any $t \geq 0$, where we used, one more time, Einstein summation convention to imply summation over the indexes i, j and Remark 2.2.5 to set all the initial conditions equal to zero.

Order 3. Repeating the same argument as for order two leads to

$$
\begin{aligned}
\epsilon^3\delta^0: \quad Y_t^{(3,0)} &= \frac{1}{4}\int_0^t \left[\int_0^t \left(A_s^{(i)}\right)^2 ds, A_t^{(k)}\right] dW_s^k \\
&\quad - \frac{1}{4}\int_0^t \left[\int_0^t \left[A_s^{(i)}, \int_0^s A_r^{(j)} dW_r^j\right] dW_s^i, A_t^{(k)}\right] dW_s^k \\
&\quad + \frac{1}{12}\int_0^t \left[\int_0^s A_r^{(j)} dW_r^j, \left[\int_0^s A_r^{(i)} dW_r^i, A_s^{(k)}\right]\right] dW_s^k \\
&\quad + \frac{1}{12}\int_0^t \left[A_s^{(k)}, \left[A_s^{(k)}, \int_0^s A_r^{(j)} dW_r^j\right]\right] - 3\left[\left(A_s^{(k)}\right)^2, \int_0^s A_r^{(j)} dW_r^j\right] ds, \\
\epsilon^2\delta^1: \quad Y_t^{(2,1)} &= \frac{1}{12}\int_0^t \left[A_s^{(k)}, \left[A_s^{(k)}, \int_0^s B_r dr\right]\right] - 3\left[\left(A_s^{(k)}\right)^2, \int_0^s B_r dr\right] ds \\
&\quad + \frac{1}{4}\int_0^t \left[\int_0^s \left(A_r^{(j)}\right)^2 dr, B_s\right] ds \\
&\quad + \frac{1}{12}\int_0^t \left[\int_0^s B_r dr, \left[\int_0^s A_r^{(i)} dW_r^i, A_s^{(j)}\right]\right] + \left[\int_0^s A_r^{(i)} dW_r^i, \left[\int_0^s B_r dr, A_s^{(j)}\right]\right] dW_s^j \\
&\quad + \frac{1}{12}\int_0^t \left[\int_0^s A_r^{(i)} dW_r^i, \left[\int_0^s A_r^{(j)} dW_r^j, B_s\right]\right] ds \\
&\quad - \frac{1}{4}\int_0^t \left[\int_0^s \left[A_r^{(i)}, \int_0^r A_u^{(j)} dW_u^j\right] dW_r^i, B_s\right] ds \\
&\quad - \frac{1}{4}\int_0^t \left[\int_0^s \left[A_r^{(i)}, \int_0^r B_u du\right] dW_r^i, A_s^{(j)}\right] dW_s^j \\
&\quad - \frac{1}{4}\int_0^t \left[\int_0^s \left[B_r, \int_0^r A_u^{(j)} dW_u^j\right] dr, A_s^{(i)}\right] ds \\
\epsilon^1\delta^2: \quad Y_t^{(1,2)} &= \frac{1}{12}\int_0^t \left[\int_0^s B_r dr, \left[\int_0^s A_r^{(j)} dW_r^j, B_s\right]\right] + \left[\int_0^s A_r^{(j)} dW_r^j, \left[\int_0^s B_r dr, B_s\right]\right] ds \\
&\quad - \frac{1}{4}\int_0^t \left[\int_0^s \left[B_r, \int_0^r A_u^{(j)} dW_u^j\right] dr, B_s\right] ds \\
&\quad - \frac{1}{4}\int_0^t \left[\int_0^s \left[A_r^{(j)}, \int_0^r B_u du\right] dr, B_s\right] ds \\
&\quad - \frac{1}{4}\int_0^t \left[\int_0^s \left[B_r, \int_0^r B_u du\right] dr, A_s^{(j)}\right] dW_s^j \\
&\quad + \frac{1}{12}\int_0^t \left[\int_0^s B_r dr, \left[\int_0^s B_r dr, A_s^{(j)}\right]\right] dW_s^j \\
\epsilon^0\delta^3: \quad Y_t^{(0,3)} &= \frac{1}{12}\int_0^t \left[\int_0^s B_r dr, \left[\int_0^r B_u du, B_s\right]\right] ds \\
&\quad + \frac{1}{4}\int_0^t \left[\int_0^s \left[\int_0^r B_u du, B_r\right] dr, B_s\right] ds
\end{aligned}
$$

Order n. Proceeding by induction, one can obtain a recursive representation for the general term $Y^{(r,n-r)}$ in (2.2.19), namely:

$$
Y_t^{(r,n-r)} = \int_0^t \mu_s^{r,n-r} ds + \int_0^t \sigma_s^{r,n-r,j} dW_s^j, \qquad n \in \mathbb{N}_0,\ r = 0, \dots, n, \tag{2.2.23}
$$

where the terms $\sigma^{r,n-r,j}, \mu^{r,n-r}$ are defined recursively as

$$\sigma_s^{r,n-r,j} := \sum_{i=0}^{n-1} \frac{\beta_i}{i!} S_s^{r-1,n-r,i}(A^{(j)}), \tag{2.2.24}$$

$$\mu_s^{r,n-r} := \sum_{i=0}^{n-1} \frac{\beta_i}{i!} S_s^{r,n-r-1,i}(B) - \frac{1}{2}\sum_{j=1}^{q}\sum_{i=0}^{n-2} \frac{\beta_i}{i!} \sum_{q_1=2}^{r}\sum_{q_2=0}^{n-r} S^{r-q_1,n-r-q_2,i}(Q^{q_1,q_2,j}), \tag{2.2.25}$$

with

$$\begin{aligned} Q_s^{q_1,q_2,j} := \sum_{i_1=2}^{q_1}\sum_{i_2=0}^{q_2}\sum_{h_1=1}^{i_1-1}\sum_{h_2=0}^{i_2}\sum_{p_1=0}^{q_1-i_1}\sum_{p_2=0}^{q_2-i_2}\sum_{m_1=0}^{p_1+p_2}\sum_{m_2=0}^{q_1-i_1-p_1+q_2-i_2-p_2} & \\ \Bigg(\frac{S_s^{p_1,p_2,m_1}\big(\sigma_s^{h_1,h_2,j}\big)}{(m_1+1)!} \frac{S_s^{q_1-i_1-p_1,q_2-i_2-p_2,m_2}\big(\sigma_s^{i_1-h_1,i_2-h_2,j}\big)}{(m_2+1)!} & \\ + \frac{\Big[S_s^{p_1,p_2,m_1}\big(\sigma_s^{i_1-h_1,i_2-h_2,j}\big), S_s^{q_1-i_1-p_1,q_2-i_2-p_2,m_2}\big(\sigma_s^{h_1,h_2,j}\big)\Big]}{(m_1+m_2+2)(m_1+1)!m_2!} \Bigg), & \end{aligned}$$

and with the operators S being defined as

$$\begin{aligned} S_s^{r-1,n-r,0}(A) &:= \begin{cases} A & \text{if } r = n = 1, \\ 0 & \text{otherwise,} \end{cases} \\ S_s^{r-1,n-r,i}(A) &:= \sum_{\substack{(j_1,k_1),\dots,(j_i,k_i)\in\mathbb{N}_0^2 \\ j_1+\cdots+j_i=r-1 \\ k_1+\cdots+k_i=n-r}} \big[Y_s^{(j_1,k_1)}, \big[\dots, \big[Y_s^{(j_i,k_i)}, A_s\big]\dots\big]\big] \\ &= \sum_{\substack{(j_1,k_1),\dots,(j_i,k_i)\in\mathbb{N}_0^2 \\ j_1+\cdots+j_i=r-1 \\ k_1+\cdots k_i=n-r}} \mathrm{ad}_{Y_s^{(j_1,k_1)}} \circ \cdots \circ \mathrm{ad}_{Y_s^{(j_i,k_i)}}(A_s), \qquad i \in \mathbb{N}. \end{aligned}$$

Remark 2.2.6. All the processes $Y^{(r,n-r)}$, with $n \in \mathbb{N}$ and $r = 0, \dots, n$, are well defined according to the recursion (2.2.23)-(2.2.24)-(2.2.25), as long as B and $A^{(1)}, \dots, A^{(q)}$ are bounded and progressively measurable stochastic processes.

Example 2.2.7. As we already pointed out in the introduction, in the case $j = 1$ and $B \equiv 0$, the SDE (2.0.1) admits an explicit solution given by

$$Y_t = -\frac{1}{2}A^2 t + AW_t, \qquad t \geq 0,$$

and the terms in the ME (2.2.20) read as

$$Y_t^{(1)} = AW_t, \qquad Y_t^{(2)} = -\frac{1}{2}A^2 t, \qquad Y_t^{(n)} = 0, \ n \geq 3.$$

In particular, the Mangus expansion coincides with the exact solution with the first two terms.

2.3. Convergence Analysis

In this section, we prove Theorem 2.0.1. To avoid ambiguity, only in this section, we denote by $\mathcal{M}_{\mathbb{R}}^{d\times d}$ and $\mathcal{M}_{\mathbb{C}}^{d\times d}$ the spaces of $(d\times d)$-matrices with real and complex entries, respectively; on these spaces we shall make use of the Frobenius norm denoted by $\|\cdot\|_F$. We say that a matrix-valued function is holomorphic if all its entries are holomorphic functions. We recall that $W=(W^1,\dots,W^q)$ is a q-dimensional standard Brownian motion and $A^{(1)},\dots,A^{(q)},B$ are bounded $\mathcal{M}_{\mathbb{R}}^{d\times d}$-valued progressively measurable stochastic processes defined on a filtered probability space $(\Omega,\mathcal{F},\mathbb{P},(\mathcal{F}_t)_{t\geq 0})$. Also recall that, for any $\mathcal{M}_{\mathbb{R}}^{d\times d}$-valued process $M=(M_t)_{t\in[0,T]}$, we set $\|M\|_T:=\|\|M\|_F\|_{L^\infty([0,T]\times\Omega)}$.

We start with two preliminary lemmas.

Lemma 2.3.1. *Assume that $Y=(Y_t^{\epsilon,\delta})_{\epsilon,\delta\in\mathbb{R},\,t\in\mathbb{R}_{\geq 0}}$ is a $\mathcal{M}_{\mathbb{R}}^{d\times d}$-valued stochastic process that can be represented as a convergent series of the form (2.2.19). If Y solves the SDE (2.2.16) up to a positive stopping time τ, then $Y^{(r,n-r)}$ in (2.2.19) are Itô processes and satisfy (2.2.23)-(2.2.24)-(2.2.25) for any $t<\tau$.*

Proof. We prove (2.2.23)-(2.2.24)-(2.2.25) only for $n=0,1$. Namely, we show that (2.2.21) and (2.2.22) hold up to time τ, $\mathbb{P}$-a.s. The representation for the general term $Y^{(r,n-r)}$ follows by induction.

Since Y is of the form (2.2.19) then $Y_t^{(0,0)}=Y_t^{0,0}$ for any $t<\tau$. Moreover, since Y solves the SDE (2.2.16) then $Y^{0,0}\equiv 0$ on $[0,\tau[$, $\mathbb{P}$-a.s. Thus $Y_t^{(0)}\equiv 0$ holds up to time τ, $\mathbb{P}$-a.s.

Now, (2.2.16) yields

$$Y_t^{\epsilon,0}=\epsilon\int_0^t A_s^{(j)}dW_s^j+\epsilon R_t^\epsilon,\qquad t\in[0,\tau[,\quad \mathbb{P}\text{-a.s.},\tag{2.3.1}$$

where

$$R_t^\epsilon=\int_0^t\Big(\sum_{k=1}^\infty\frac{\beta_k}{k!}\mathrm{ad}^k_{Y_s^{\epsilon,0}}(A_s^{(j)})\Big)dW_s^j \\ -\frac{\epsilon}{2}\int_0^t\mathcal{L}^{-1}_{Y_s^{\epsilon,0}}\Big(\mathcal{Q}_{Y_s^{\epsilon,0}}\Big(\sum_{k=0}^\infty\frac{\beta_k}{k!}\mathrm{ad}^k_{Y_s^{\epsilon,0}}(A_s^{(j)}),\sum_{k=0}^\infty\frac{\beta_k}{k!}\mathrm{ad}^k_{Y_s^{\epsilon,0}}(A_s^{(j)})\Big)\Big)ds.$$

Note that, again by (2.2.16), $R^0\equiv 0$ $\mathbb{P}$-a.s. Moreover, representation (2.2.19) implies continuity of $\epsilon\mapsto Y_t^{\epsilon,0}$ near $\epsilon=0$, which in turn implies the continuity of $\epsilon\mapsto R_t^\epsilon$. Thus we have $\lim_{\epsilon\to 0}R_t^\epsilon=R_t^0$ $\mathbb{P}$-a.s. This, together with (2.3.1) and (2.2.19) implies that (2.2.21) necessarily holds, up to time τ, $\mathbb{P}$-a.s.

Similarly, (2.2.16) yields

$$Y_t^{0,\delta}=\delta\int_0^t B_s ds+\delta Q_t^\delta,\qquad t\in[0,\tau[,\quad P\text{-a.s.},$$

with

$$Q_t^\delta=\int_0^t\Big(\sum_{k=1}^\infty\frac{\beta_k}{k!}\mathrm{ad}^k_{Y_s^{0,\delta}}(B_s)\Big)ds$$

and the same argument employed above yields (2.2.22) up to time τ, $\mathbb{P}$-almost surely. □

Lemma 2.3.2. *Let $M \in \mathcal{M}_{\mathbb{C}}^{d\times d}$ be nonsingular and such that $\|M - I_d\| < 1$ where $\|\cdot\|$ is the spectral norm. Then M has a unique logarithm, which is*

$$\log M = \sum_{n=1}^{\infty} (-1)^{n+1} \frac{(M - I_d)^n}{n} \tag{2.3.2}$$

$$= (M - I_d) \int_0^{\infty} \frac{1}{1+\mu} (\mu I_d + M)^{-1} d\mu. \tag{2.3.3}$$

In particular, we have

$$\|\log M\| \leq -\log\left(1 - \|M - I_d\|\right). \tag{2.3.4}$$

Proof. The first representation is a standard result. The second representation stems from the factorization $M = VJV^{-1}$ with J in Jordan form, under the assumption that M has no non-positive real eigenvalues, i.e. $\lambda \in \mathbb{C}\backslash]-\infty, 0]$ for any λ eigenvalue of M. Alternatively, see Higham (2008): *p. 269 Theorem 11.1* together with the substitution rule using $f(t) := \frac{1}{1+t}$.

Now, $\|M - I_d\| < 1$ implies that there are no non-positive real eigenvalues, because it implies with the definition of the spectral norm that for all $v \in \mathbb{R}^d$ with $|v| = 1$

$$1 > \|M - I_d\| = \max_{|\widetilde{v}|=1} \frac{|(M - I_d)\,\widetilde{v}|}{|\widetilde{v}|} \geq |Mv - v|\,,$$

which in turn implies that, if λ is a real eigenvalue of M and v is one of its normalized eigenvectors, then

$$1 > |Mv - v| = |\lambda v - v| = |\lambda - 1|\,|v| = |\lambda - 1|\,.$$

The inequality (2.3.4) follows from applying the triangular inequality to (2.3.2), the sub-multiplicativity of the spectral norm and using $\log(1 - x) = -\sum_{k=1}^{\infty} \frac{x^k}{k}$, $x \in (0, 1)$, for the scalar logarithm. □

We have one last preliminary lemma, containing some technical results concerning the solutions to (2.2.12). These are semi-standard, in that they can be inferred by combining and adapting existing results in the literature.

Lemma 2.3.3. *For any $T > 0$ and $\epsilon, \delta \in \mathbb{C}$, the SDE (2.2.12) has a unique strong solution $(X_t^{\epsilon,\delta})_{t\in[0,T]}$. For any $p \geq 1$ and $h > 0$ there exists a positive constant κ, only dependent on $\|A^{(1)}\|_T, \dots, \|A^{(q)}\|_T$, $\|B\|_T$, d, T, h and p, such that*

$$\mathbb{E}\Big[\|X_t^{\epsilon,\delta} - X_s^{\epsilon',\delta'}\|_F^{2p}\Big] \leq \kappa\big(\,|t - s|^p + \left(|\epsilon - \epsilon'| + |\delta - \delta'|\right)^{2p}\big), \tag{2.3.5}$$

$$\mathbb{E}\Big[\sup_{0\leq u\leq t} \|X_u^{\epsilon,\delta} - X_0^{\epsilon,\delta}\|_F^{2p}\Big] \leq \kappa t^p (|\epsilon| + |\delta|)^{2p}, \tag{2.3.6}$$

for any $0 \leq t, s \leq T$ and $\epsilon, \delta, \epsilon', \delta' \in \mathbb{C}$ with $|\epsilon|, |\delta|, |\epsilon'|, |\delta'| \leq h$.

Up to modifications, $(X_t^{\epsilon,\delta})_{\epsilon,\delta\in\mathbb{C},\, t\in[0,T]}$ is a continuous process such that:

i) *for any $t \in [0,T]$, the function $(\epsilon,\delta) \mapsto X_t^{\epsilon,\delta}$ is holomorphic;*

ii) *the functions $(t,\epsilon,\delta) \mapsto \partial_\epsilon X_t^{\epsilon,\delta}$ and $(t,\epsilon,\delta) \mapsto \partial_\delta X_t^{\epsilon,\delta}$ are continuous;*

iii) *for any $p \geq 1$ and $h > 0$ there exists a positive constant κ only dependent on $\|A^{(1)}\|_T, \dots, \|A^{(q)}\|_T$, $\|B\|_T$, d, T, h and p, such that*

$$\mathbb{E}\Big[\sup_{0\leq s\leq t}\Big\{\|\partial_\epsilon X_s^{\epsilon,\delta}\|_F^{2p} + \|\partial_\delta X_s^{\epsilon,\delta}\|_F^{2p}\Big\}\Big] \leq \kappa t^p(|\epsilon| + |\delta|)^p, \tag{2.3.7}$$

for any $t \in [0,T]$ and $|\epsilon|, |\delta| \leq h$.

Proof. Existence of the solution and estimates (2.3.5)-(2.3.6) of the moments follow from the results in Section 5, Chapter 2 in Krylov (2008) (in particular, see Corollary 5 on page 80 and Theorem 7 on page 82).

The second part of the statement is a refined version of the Kolmogorov continuity theorem in the form that can be found for instance in Section 2.3 in Kunita (2019). □

Remark 2.3.4. The existence and uniqueness for the solution to (2.0.1) is a particular case of the previous result.

We are now in the position to prove Theorem 2.0.1.

Proof of Theorem 2.0.1. We fix $h > 1$, $T > 0$, and let $(X_t^{\epsilon,\delta})_{\epsilon,\delta\in\mathbb{C},\, t\in[0,T]}$ be the solution of the SDE (2.2.12) as defined in Lemma 2.3.3. Moreover, for $t \in]0,T]$, we set $Q_{t,h} :=]0,t[\times B_h(0)$ where $B_h(0) = \{(\epsilon,\delta) \in \mathbb{C}^2 \mid |(\epsilon,\delta)| < h\}$.

Part (i): as $X_0^{\epsilon,\delta} = I_d$, by continuity the random time defined as

$$\tau := \sup\Big\{t \in [0,T]\,\Big|\,\|X_s^{\epsilon,\delta} - I_d\|_F < 1 - e^{-\pi} \text{ for any } (s,\epsilon,\delta) \in Q_{t,h}\Big\} \tag{2.3.8}$$

is strictly positive. Furthermore, again by continuity,

$$(\tau \leq t) = \bigcup_{(s,\epsilon,\delta)\in\widetilde{Q}_{t,h}} \Big(\|X_s^{\epsilon,\delta} - I_d\|_F \geq 1 - e^{-\pi}\Big), \qquad t \in [0,T],$$

where $\widetilde{Q}_{t,h}$ is a countable, dense subset of $Q_{t,h}$, which implies that τ is a stopping time.

Let $(t,\epsilon,\delta) \in Q_{\tau,h}$: by Lemma 2.3.2 applied to $M = X_t^{\epsilon,\delta}$ we have

$$\begin{aligned} Y_t^{\epsilon,\delta} &:= \log X_t^{\epsilon,\delta} = \sum_{n=1}^{\infty}(-1)^{n+1}\frac{(X_t^{\epsilon,\delta} - I_d)^n}{n} \\ &= (X_t^{\epsilon,\delta} - I_d)\int_0^\infty \frac{1}{1+\mu}(\mu I_d + X_t^{\epsilon,\delta})^{-1}d\mu. \end{aligned} \tag{2.3.9}$$

Notice that $X_t^{\epsilon,\delta}$ (and therefore also $Y_t^{\epsilon,\delta}$) is real for $\epsilon,\delta \in \mathbb{R}$: in particular, $Y_t = Y_t^{1,1}$ is real and this proves Part (i).

Part (ii): since $(\epsilon, \delta) \mapsto X_t^{\epsilon,\delta}$ is holomorphic, we can differentiate (2.3.9) to infer that $(\epsilon, \delta) \mapsto Y_t^{\epsilon,\delta}$ is holomorphic as well: indeed, we have for $(t, \epsilon, \delta) \in Q_{\tau,h}$

$$\begin{aligned}\partial_\epsilon Y_t^{\epsilon,\delta} = \partial_\epsilon X_t^{\epsilon,\delta} \int_0^\infty \frac{1}{1+\mu} (\mu I_d + X_t^{\epsilon,\delta})^{-1} d\mu \\ + (X_t^{\epsilon,\delta} - I_d) \int_0^\infty \frac{1}{1+\mu} (\mu I_d + X_t^{\epsilon,\delta})^{-1} (\partial_\epsilon X_t^{\epsilon,\delta}) (\mu I_d + X_t^{\epsilon,\delta})^{-1} d\mu,\end{aligned}$$

and similarly by differentiating w.r.t. to δ. Then the expansion of $Y_t^{\epsilon,\delta}$ in power series at $(\epsilon, \delta) = (0, 0)$ is absolutely convergent on $B_h(0)$ and the representation (2.2.19) holds on $Q_{\tau,h}$ for some random coefficients $Y_t^{(r,n-r)}$. To conclude we need to show that the latter are as given by (2.2.23)-(2.2.24)-(2.2.25). Then (2.0.10) will stem from (2.2.19) by setting $(\epsilon, \delta) = (1, 1)$.

In light of Lemma 2.3.2, the logarithmic map is continuously twice differentiable on the open subset of $\mathcal{M}_{\mathbb{C}}^{d\times d}$ of the matrices M such that $\|M - I_d\| < 1$: thus $Y_t^{\epsilon,\delta}$ admits an Itô representation (2.2.10) for $(t, \epsilon, \delta) \in Q_{\tau,h}$. Then Proposition 2.2.2 together with (2.2.12) yield (2.2.14)-(2.2.15) $\mathbb{P}$-a.s. up to τ for any $(\epsilon, \delta) \in B_h(0) \cap \mathbb{R}^2$. Furthermore, by estimate (2.3.4) of Lemma 2.3.2 we also have $\left\|Y_t^{\epsilon,\delta}\right\| < \pi$ for $t < \tau$. Therefore, we can apply Baker's Lemma 2.2.3 to invert $\mathcal{L}_{Y_t^{\epsilon,\delta}}$ in (2.2.14)-(2.2.15) and obtain that $Y^{\epsilon,\delta}$ solves (2.2.16) up to τ for any $(\epsilon, \delta) \in B_h(0) \cap \mathbb{R}^2$. Part (ii) then follows from Lemma 2.3.1.

Part (iii): for $t \le T$ let

$$f_t(\epsilon, \delta) := \max_{s\in[0,t]} \|X_s^{\epsilon,\delta} - I_d\|_F, \qquad M_t := \sup_{(\epsilon,\delta)\in B_h(0)} f_t(\epsilon, \delta).$$

By definition (2.3.8), we have with the Markov inequality

$$\mathbb{P}(\tau \le t) \le \mathbb{P}\left(M_t \ge 1 - e^{-\pi}\right) \le \frac{1}{(1 - e^{-\pi})^2} \mathbb{E}\left[M_t^2\right], \tag{2.3.10}$$

and therefore (2.0.11) follows by suitably estimating $\mathbb{E}\left[M_t^2\right]$. To prove such an estimate we will show in the last part of the proof that f_t belongs to the Sobolev space $W^{1,2p}(B_h(0))$ for any $p \ge 1$ and we have

$$\mathbb{E}[\|f_t\|_{W^{1,2p}(B_h(0))}^{2p}] \le Ct^p, \qquad t \in [0, T], \tag{2.3.11}$$

where the positive constant C depends only on $\|A^{(1)}\|_T, \dots, \|A^{(q)}\|_T$, $\|B\|_T$, d, T, h and p. Since $f_t \in W^{1,2p}(B_h(0))$ and $B_h(0) \subseteq \mathbb{R}^4$, by Morrey's inequality (cf., for instance, Corollary 9.14 in Brezis (2011)) for any $p > 2$ we have

$$M_t \le c_0 \|f_t\|_{W^{1,2p}(B_h(0))}, \tag{2.3.12}$$

where c_0 is a a positive constant, dependent only on p and h (in particular, c_0 is independent of $\varnothing$). Combining (2.3.11) with (2.3.12), for a fixed $p>2$ we have

$$\mathbb{E}\left[M_t^2\right] \leq c_0^2 E[\|f_t\|^2_{W^{1,2p}(B_h(0))}] \leq$$

(by Hölder inequality)

$$\leq c_0^2 Ct, \qquad t\in[0,T].$$

This last estimate, combined with (2.3.10), proves (2.0.11).

To conclude, we are left with the proof of (2.3.11). First we have

$$\mathbb{E}\bigg[\int_{B_h(0)} |f_t(\epsilon,\delta)|^{2p} d\epsilon\, d\delta\bigg] = \int_{B_h(0)} \mathbb{E}[|f_t(\epsilon,\delta)|^{2p}]\, d\epsilon\, d\delta \leq Ct^p, \tag{2.3.13}$$

where we used the estimate (2.3.6) of Lemma 2.3.3 in the last inequality. Fix now $t\in]0,T]$, $(\epsilon,\delta),(\epsilon',\delta')\in B_h(0)$ such that $f_t(\epsilon',\delta')\leq f_t(\epsilon,\delta)$ and set

$$\bar{t}\in \operatorname*{arg\,max}_{0\leq s\leq t} \|X_s^{\epsilon,\delta}-I_d\|_F, \qquad \widetilde{t}\in \operatorname*{arg\,max}_{0\leq s\leq t} \|X_s^{\epsilon',\delta'}-I_d\|_F.$$

Note that the arg max above do exist in that the process $g_s(\epsilon,\delta):=X_s^{\epsilon,\delta}-I_d$ is continuous in s and we have

$$\begin{aligned}
|f_t(\epsilon,\delta)-f_t(\epsilon',\delta')| &= \Big|\|g_{\bar{t}}(\epsilon,\delta)\|_F - \|g_{\widetilde{t}}(\epsilon',\delta')\|_F\Big| \leq \big|\|g_{\bar{t}}(\epsilon,\delta)\|_F - \|g_{\bar{t}}(\epsilon',\delta')\|_F\big| \\
&\leq \|g_{\bar{t}}(\epsilon,\delta)-g_{\bar{t}}(\epsilon',\delta')\|_F \leq \sup_{0\leq s\leq t}\|g_s(\epsilon,\delta)-g_s(\epsilon',\delta')\|_F \\
&\leq \big|(\epsilon,\delta)-(\epsilon',\delta')\big| \sup_{0\leq s\leq t} \sup_{\substack{|\bar{\epsilon}-\epsilon|\leq|\epsilon'-\epsilon|\\ |\bar{\delta}-\delta|\leq|\delta'-\delta|}} \|\nabla g_s(\bar{\epsilon},\bar{\delta})\|_F,
\end{aligned}$$

where $\nabla=\nabla_{\epsilon,\delta}$. This, as $(s,\epsilon,\delta)\mapsto\nabla g_s(\epsilon,\delta)$ is continuous on $Q_{t,h}$, implies $f_t\in W^{1,2p}(B_h(0))$ and yields the key inequality

$$|\nabla f_t(\epsilon,\delta)|\leq \sup_{0\leq s\leq t}\|\nabla X_s^{\epsilon,\delta}\|_F, \qquad (\epsilon,\delta)\in B_h(0).$$

Therefore, we have

$$\begin{aligned}
\mathbb{E}\bigg[\int_{B_h(0)} |\nabla f_t(\epsilon,\delta)|^{2p}\, d\epsilon\, d\delta\bigg] &= \int_{B_h(0)} \mathbb{E}[\,|\nabla f_t(\epsilon,\delta)|^{2p}\,]\, d\epsilon\, d\delta \\
&\leq \int_{B_h(0)} \mathbb{E}\bigg[\sup_{0\leq s\leq t}\|\nabla X_s^{\epsilon,\delta}\|_F^{2p}\bigg] d\epsilon\, d\delta \leq Ct^p,
\end{aligned}$$

where we used the estimate (2.3.7) of Lemma 2.3.3 in the last inequality. This, together with (2.3.13), proves (2.3.11) and conclude the proof. □

2.4. Numerical Tests

We present here some numerical tests in order to confirm the accuracy of the approximate solutions to (2.0.1) stemming from the truncation of the series (2.0.10).

In Section 2.4.2, we highlight the features of the Magnus expansion with respect to different types of the coefficient processes A_t and B_t, namely general processes, $C_t := C \cdot c_t$, for $C \in \mathbb{R}^{d\times d}$ and c_t a scalar stochastic process, $C = A, B$, $c = a, b$, which we will call *separable coefficients* and the case of constant coefficients $A_t \equiv A$, $B_t \equiv B$. We will see that the expansion formulas can be refined with more knowledge about the coefficient functions, which leads to increased accuracy and decreased computational times for the Magnus schemes.

We also show, how this approximation can be applied to approximate the solutions to stochastic partial differential equations (SPDEs) of parabolic type with one spatial dimension in Section 2.4.3 and kinetic SPDEs with two spatial dimensions in Section 2.4.4.

For the numerical tests we will implement the exponential of the truncated Magnus expansion up to order $n = 1, 2$ and 3, i.e.

$$X^{(n)} := e^{\sum_{i=1}^{n} Y^{(i)}}, \qquad n = 1, 2, 3, \tag{2.4.1}$$

and compare it with a benchmark solution to (2.0.1).

Error and notations. Throughout this section we will employ the following tags:

1. `exact` to denote the time-discretization of an explicit solution, if available;
2. `euler` for the solution obtained with a corresponding Euler-Maruyama scheme;
3. `m1`, `m2` and `m3` for the time-discretization of the Magnus approximations in (2.4.1), up to order 1,2 and 3, respectively.

For the numerical error analysis we will make use of the following notations. Let us denote by X^{ref} and by X^{app} a benchmark and an approximate solution, respectively.

In Section 2.4.3 and Section 2.4.4 we will show how a space-discretization for SPDEs based on finite-difference techniques will result in matrix-valued SDEs. We will impose zero-boundary conditions for these discretizations leading to errors in the vicinity of the boundary. Therefore, we will only use a central part with varying size of the whole solution matrix at a given time, which is illustrated in Figure 2.1. To vary the size we introduce a new parameter $\kappa = 0, 1, \ldots$ indicating the $2^{-\kappa}$-th central part of the solution matrix $X_t^d \in \mathbb{R}^{d\times d}$, which we will consider for our error analysis and denote the corresponding truncated matrix by $X_t^{\mathrm{ref},d,\kappa}, X_t^{\mathrm{app},d,\kappa} \in \mathbb{R}^{\lfloor \frac{d}{2^\kappa} \rfloor \times \lfloor \frac{d}{2^\kappa} \rfloor}$. Also we set

$$I^\kappa := \left\{ \left\lfloor \frac{d}{2} - \frac{d}{2^{\kappa+1}} \right\rfloor, \ldots, \left\lfloor \frac{d}{2} + \frac{d}{2^{\kappa+1}} \right\rfloor \right\}$$

to collect the corresponding indices.

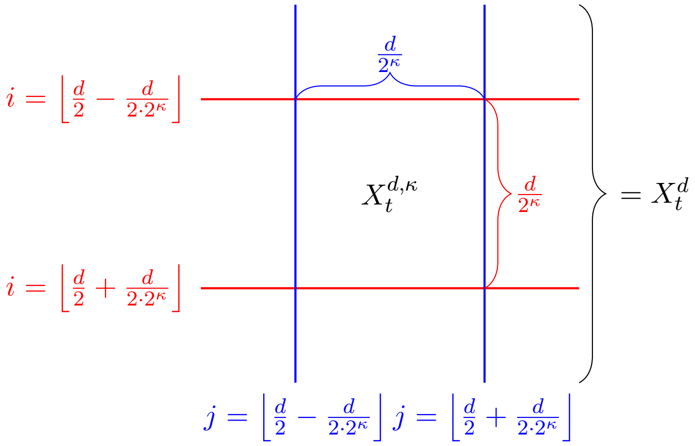

$i = \left\lfloor \frac{d}{2} - \frac{d}{2\cdot 2^\kappa} \right\rfloor$
$i = \left\lfloor \frac{d}{2} + \frac{d}{2\cdot 2^\kappa} \right\rfloor$
$\frac{d}{2^\kappa}$
$X_t^{d,\kappa}$
$\frac{d}{2^\kappa}$
$= X_t^d$
$j = \left\lfloor \frac{d}{2} - \frac{d}{2\cdot 2^\kappa} \right\rfloor$
$j = \left\lfloor \frac{d}{2} + \frac{d}{2\cdot 2^\kappa} \right\rfloor$

Figure 2.1.: Graphical representation of $X_t^{d,\kappa}$ compared to X_t^d for the error analysis to disregard boundary effects.

For our error analysis we consider the following three norms:

$$\begin{aligned}
\mathrm{ME}_t^{d,\kappa} &:= \frac{1}{M} \sum_{m=1}^{M} \left| X_{t,m}^{\mathrm{ref},d,\kappa} - X_{t,m}^{\mathrm{app},d,\kappa} \right| \in \mathbb{R}^{\lfloor \frac{d}{2^\kappa} \rfloor \times \lfloor \frac{d}{2^\kappa} \rfloor} \\
\mathrm{AME}_t^{d,\kappa} &:= \frac{1}{|I^\kappa|^2} \sum_{i,j} \left(\mathrm{ME}_t^{d,\kappa} \right)_{ij} \in \mathbb{R} \\
\mathrm{Err}_t^{d,\kappa} &:= \frac{1}{M} \sum_{m=1}^{M} \frac{\left\| X_{t,m}^{\mathrm{ref},d,\kappa} - X_{t,m}^{\mathrm{app},d,\kappa} \right\|_F}{\left\| X_{t,m}^{\mathrm{ref},d,\kappa} \right\|_F} \in \mathbb{R},
\end{aligned} \tag{2.4.2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The first one is a matrix consisting of a mean absolute error between the reference solution and the approximation for each point in the grid $I^\kappa \times I^\kappa$ by taking the mean over all trajectories. The second error is the average of the first error for the corresponding grid indicated by κ. The larger κ the further away we are from the spatial boundary, as illustrated in Figure 2.1. The third error is the mean of a relative error between the reference solution and the approximation taking all points in the region corresponding to κ into account. It will serve as our main error norm in this section. In particular, we will set $\kappa = 0$ in Section 2.4.2 and $\kappa = 4$ in Section 2.4.3 and Section 2.4.4.

The remainder of this section is structured as follows: In the next Section 2.4.1 we will discuss how to implement the Magnus expansion and its *parallel-in-time* and *parallel-in-simulation* features, as well as, how to deal with the convergence radius of the Magnus expansion. Afterwards, we show in Section 2.4.2 experiments in the case of matrix-valued SDEs with different types of coefficient functions A_t and B_t. This is followed in Section 2.4.3 and Section 2.4.4 by an application to solving SPDEs in one and two spatial dimensions by applying the so-called *method of lines*, leading to matrix-valued SDEs, which can be approximated by the Magnus schemes.

2.4.1. Numerical Scheme and Features

In this section, we will explain how to best implement the numerical schemes based on the the truncation of the Magnus expansion series (2.0.10). In the follwing numerical experiments in Section 2.4.2–2.4.4 we will see good evidence that three terms of the Magnus expansion will lead to satisfactory results. Therefore, we will refer to `m2` ($n = 2$) as the Magnus expansion using two terms and `m3` ($n = 3$) as the one using three terms. An order one expansion is usually not sufficient.

The idea of the Magnus expansion relies on the assumption that we can express the solution as the exponential of some unknown Itô process Y_t, which we will refer to as the Magnus logarithm from now on. This leads to two different steps in the implementation of the scheme: First, we compute the Magnus logarithm for a fixed order $Y_t^{(n)}$ and second we apply the matrix exponential at the desired times t.

Note that contrary to the Euler-Maruyama scheme, we do not need to solve an equation for the Magnus logarithm, we just need to compute iterated Lesbegue and stochastic integrals, as well as the commutators of our coefficients.

Single step Magnus expansion. Let us focus on Figure 2.2 for the moment. Suppose we know that the Magnus expansion will be convergent from $t_0 = 0$ till $t_{11} = T$. Then we use first a cumulative integration scheme, such as `cumsum` for a Riemann-sum approximation of the Lebesgue integrals and the stochastic integrals to obtain values of the Magnus logarithm Y_{t_i}, $i = 1, \dots, 11$ for all points in our time grid.[1] This is usually quite fast and can be evaluated in parallel for all trajectories. Depending on the size of the time grid, the number of simulations and the matrix dimensions, it can be a good idea to use a GPU for this computation.

Now, having the Magnus logarithm evaluated, we can choose for the second step when we want to evaluate the solution of the matrix-valued SDE (2.0.1) by taking the matrix exponential. Therefore, all the selected points in time, say t_3, t_7, t_{11}, can be evaluated in *parallel-in-time* and *parallel-in-simulation*.[2] Using a threaded CPU environment will almost always lead to a speed-up compared to sequential schemes. If a GPU can further decrease the computational time depends again on all the problem parameters and the chosen algorithm for matrix exponentiation.

Usually, the evaluation of a full matrix exponential is more expensive then the evaluation of the Magnus logarithm at one point in time. Therefore, the less matrix-exponentials the better for the overall computational time.

[1] More details about the implementation can be found on https://github.com/kevinkamm/StochasticMagnusExpansion/blob/main/AB_const/AB_const_magnus.m.

[2] More details about the implementation can be found on https://github.com/kevinkamm/StochasticMagnusExpansion/blob/main/AB_const/m_exp.m.

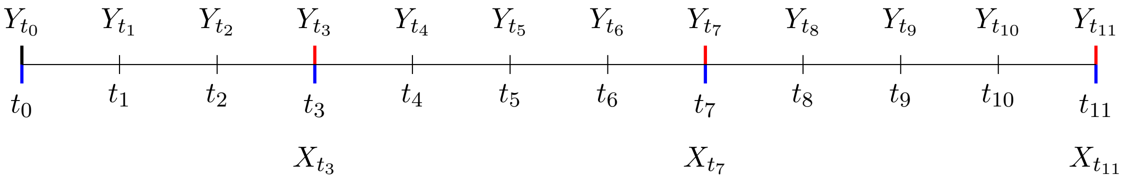

Y_{t_0} Y_{t_1} Y_{t_2} Y_{t_3} Y_{t_4} Y_{t_5} Y_{t_6} Y_{t_7} Y_{t_8} Y_{t_9} $Y_{t_{10}}$ $Y_{t_{11}}$
t_0 t_1 t_2 t_3 t_4 t_5 t_6 t_7 t_8 t_9 t_{10} t_{11}
X_{t_3} X_{t_7} $X_{t_{11}}$

Figure 2.2.: Graphical representation of the evaluation of the Magnus logarithm and exponential.

Iterated Magnus expansion. Since the Magnus expansion is only convergent up to a strictly positive stopping time, the convergence radius can be quite small for some problems. Therefore, let us first of all note that if X_t is the solution to

$$dX_t = B_t X_t dt + A_t X_t dW_t, \quad X_0 = I,$$

then by simple substitution $\widetilde{X}_t := X_t \cdot \widetilde{X}_0$ is the solution to

$$dX_t = B_t X_t dt + A_t X_t dW_t, \quad X_0 = \widetilde{X}_0,$$

where $\widetilde{X}_0$ is a bounded matrix-valued random variable, because the integration acts componentwise.

This allows us to split the evaluation of the SDE on the interval $[0, T]$ into smaller subintervals $(t_0, t_3], (t_3, t_7], (t_7, t_{11}]$. As illustrated in Figure 2.2, we will evaluate the Magnus expansion consecutively on each of them, i.e. use the terminal evaluation highlighted in blue as the initial point of the next sub-interval till the next terminal time highlighted in red.

We will call this method *iterated Magnus expansion.*[3] On each sub-interval, the Magnus expansion still has the usual parallel-in-time features, which we saw in the single step method. Furthermore, due to its relatively large convergence region in time, we can use fewer iterations compared to other iterative methods, e.g. for the Euler-Maruyama scheme, which we will see in Section 2.4.2–2.4.4.

Initial datum. In `Matlab 2022a` the full matrix exponential is computed by a scaling and squaring adjusted Padé-approximation, which uses matrix-matrix multiplications.

If in the application of the Magnus expansion the final step involves a multiplication by a single vector, meaning

$$U_t := X_t \cdot v \approx \exp(Y_t)\, v$$

it is in general a good idea to avoid the calculation of the full matrix exponential. Just by looking at the definition of the matrix exponential, we can see that matrix-matrix multipli-

[3] More details about the implementation can be found on `https://github.com/kevinkamm/IteratedMagnus/blob/main/SDEconst/magnusConstCS.m`.

cations can be replaced by matrix-vector multiplications, i.e.

$$\exp(A)\,v \approx \left(I + A + \frac{1}{2}A^2 + \frac{1}{6}A^3 + \cdots\right) v = v + Av + \frac{1}{2}A\,(Av) + \frac{1}{6}A\,(A\,(Av)) + \cdots$$

This reduces the computational effort tremendously and we recommend the algorithm called `expmvtay2` by Ibáñez et al. (2022) for a fast and GPU-applicable method.[4]

We also tested `expmv` by Al-Mohy and Higham (2011) and a Krylov-subspace implementation called `expv`. However, for our purposes, `expmvtay2` was more accurate and significantly faster than the other methods.

Computational device. We already hinted at the possibilities to either use one or more GPUs or CPUs for the Magnus logarithm or the matrix exponentiation.

Table 2.1.: Heuristic for choosing the computational device for the Magnus expansion with $M = 100$ simulations.

	$X_0 \in \mathbb{R}^{d\times d}$		$X_0 \in \mathbb{R}^{d\times 1}$	
	`expm`	log	`expmvtay2`	log
full	if $d \in [400, 1000]$ GPU, otherwise CPU	if $d \leq 200$ CPU, otherwise GPU	if $d \in [400, 1000]$ GPU, otherwise CPU	if $d \leq 200$ CPU, otherwise GPU
sparse	CPU (GPU not allowed)	if $d \leq 200$ CPU, otherwise GPU	if $d > 10^4$ GPU, otherwise CPU	if $d > 10^4$ GPU, otherwise CPU

Based on our computer architecture with $M = 100$ simulations, we found the following settings in Table 2.1 to work fastest in the case of separable and constant coefficients, which we will define in Section 2.4.2.[5]

We can see, that we distinguish between four different cases for the corresponding device for computing the Magnus logarithm and the matrix (-vector) exponential by considering full or sparse coefficient processes A_t and B_t, as well as the two cases coming from either a vector as initial datum or a matrix.

For each case the computational device is dependent on the matrix dimension of A_t and B_t. For example in the case of full coefficient functions and a matrix as initial datum, we suggest to use a GPU if the matrix dimension is in the interval $[400, 1000]$. The upper limit comes from a restraint of available memory on the GPU. Note that in `Matlab 2022a expm` cannot be computed with sparse matrices as input on a GPU and even though one can use sparse inputs on a CPU, the outputs will in general not be sparse anymore.

In the case of general coefficients, we always recommend to use a GPU for the evaluation of the logarithm, because one has to evaluate many matrix multiplications. For the evaluation of the exponential, we suggest to follow Table 2.1.

[4] More details about the implementation can be found on `https://github.com/kevinkamm/IteratedMagnus/blob/main/SDEconst/expmvtay2.m`.

[5] More details about the implementation can be found on `https://github.com/kevinkamm/IteratedMagnus/blob/main/SDEconst/compMode.m`.

2.4.2. Application to Matrix-Valued SDEs

In this section, we will apply the Magnus expansion to different SDEs of the form (2.0.1) with one Brownian motion. To have an explicit solution as a benchmark for the Magnus expansion, we will first show, that in the case of 2×2 upper triangular matrices one can solve SDE (2.0.1) explicitly by applying an inhomogeneous extension of the Doleans-Dade exponential, which is known as Yoeurp and Yor's formula (cf. Duan and Yan (2008): *p. 2 Theorem 1.1*) in the case of continuoues semimartingales. Let us briefly recall the result and apply it to our example.

Lemma 2.4.1. *Let Z_t, $Z_0 = 0$, and H_t be continuous semimartingales. Then the unique solution of the linear inhomogeneous scalar SDE*

$$X_t = H_t + \int_0^t X_s dZ_s$$

is given by

$$X_t = \mathcal{E}_t(Z)\left(H_0 + \int_0^t \mathcal{E}_s^{-1}(Z)\, dG_s\right)$$

where $\mathcal{E}_t(Z) := \exp\left(Z_t - \frac{1}{2}\langle Z\rangle_t\right)$ denotes the Doleans-Dade exponential *or* stochastic exponential *and $\mathcal{E}_t^{-1}(Z) := \mathcal{E}_t(-Z_\cdot + \langle Z\rangle_\cdot)$ its inverse, as well as*

$$G_t := H_t - \langle H, Z\rangle_t$$

This result can be verified by applying Itô's formula.

Now, let us consider the following example

$$\begin{aligned} dX_t = B_t X_t dt + A_t X_t dt, \quad X_0 &:= \begin{pmatrix} x_0^{11} & x_0^{12} \\ 0 & x_0^{22} \end{pmatrix}, \\ B_t := \begin{pmatrix} b_t^{11} & b_t^{12} \\ 0 & b_t^{22} \end{pmatrix}, \quad A_t &:= \begin{pmatrix} a_t^{11} & a_t^{12} \\ 0 & a_t^{22} \end{pmatrix}, \end{aligned} \tag{2.4.3}$$

where a_t^{ij} and b_t^{ij} are bounded progressively measurable real-valued processes and $x_0^{ij} \in \mathbb{R}$ for $i, j = 1, 2$.

Let us consider each component of the equation, i.e.

$$dX_t^{ij} = \left(b_t^{i1} X_t^{1j} + b_t^{i2} X_t^{2j}\right) dt + \left(a_t^{i1} X_t^{1j} + a_t^{i2} X_t^{2j}\right) dW_t.$$

In the case of the lower left corner of the equation, i.e. $i = 2, j = 1$ we have

$$dX_t^{21} = b_t^{22} X_t^{21} dt + a_t^{22} X_t^{21} dW_t, \quad X_0^{21} = 0.$$

Since $X_0^{21} = 0$, we see that a solution is given by $X_t^{21} \equiv 0$ and therefore by uniqueness the solution. This simplifies the equations for the diagonal entries. Hence, their solutions are

Table 2.2.: Upper triangular case with general coefficients as in (2.4.4). Computational times and errors for each component with $\Delta_t^{\text{Int}} = 10^{-3}$ and $M = 1000$ simulations.

	`euler`	`euler`	`m2`
Step size	$\Delta_t = 10^{-3}$	$\Delta_t = 10^{-4}$	$\Delta_t = 0.1$
$(X_T)_{11}$	$\text{Err}_T = 1 \cdot 10^{-3}$	$\text{Err}_T = 2 \cdot 10^{-4}$	$\text{Err}_T = 3 \cdot 10^{-4}$
$(X_T)_{22}$	$\text{Err}_T = 4 \cdot 10^{-4}$	$\text{Err}_T = 5 \cdot 10^{-5}$	$\text{Err}_T = 1 \cdot 10^{-4}$
$(X_T)_{12}$	$\text{Err}_T = 1.3 \cdot 10^{-2}$	$\text{Err}_T = 3.9 \cdot 10^{-3}$	$\text{Err}_T = 7.3 \cdot 10^{-3}$
Comp. Time	$0.16\, s$	$1.95\, s$	$1.65\, s$

given by the Doleans-Dade exponential

$$X_t^{ii} := x_0^{ii} \mathcal{E}_t \left(\int_0^{\cdot} b_s^{ii} ds + \int_0^{\cdot} a_s^{ii} dW_s \right).$$

The upper right corner remains inhomogeneous with the now known X_t^{22}. Thus, we can apply Lemma 2.4.1 in the case $H_t := x_0^{12} + \int_0^t b_s^{12} X_s^{22} ds + \int_0^t a_s^{12} X_s^{22} dW_s$ $Z_t = \int_0^t b_s^{11} ds + \int_0^t a_s^{11} dW_s$.

Apart from computing stochastic and Lebesgue integrals, we now have an explicit solution to this problem, to which we will compare the Euler-Maruyama scheme and the Magnus expansion. We will compare three different examples depending on the choice of a_t^{ij} and b_t^{ij} to demonstrate the behaviour of the Magnus expansion.

Case 1: general coefficients. In our formal derivation in Section 2.2 of the Magnus expansion, we encountered the general formulas of the expansion. In this paragraph, we will put this general formula of order 2 to the test and compare it to the Euler-Maruyama scheme. We consider

$$\begin{aligned} A_t &:= \begin{pmatrix} 0.0892\, g(W_t^{A,1}) & 0.4015\, g(W_t^{A,2}) \\ 0 & 0.0169\, g(W_t^{A,3}) \end{pmatrix}, \quad g(x) := \sqrt{1 + \frac{1}{1+x^2}} \\ B_t &:= \begin{pmatrix} 0.9234\, f(W_t^{B,1}) & 0.3813\, f(W_t^{B,2}) \\ 0 & 0.5403\, f(W_t^{B,3}) \end{pmatrix}, \quad f(x) := 1 + \frac{1}{1+x^2}, \end{aligned} \tag{2.4.4}$$

where $W_t^{C,i}$, $C = A, B$, $i = 1, 2, 3$, are independent Brownian motions.

The values in the matrices A_t and B_t were chosen at random, with the uniform distribution and will remain the same throughout the following two tests, as well. Since, we want to test the general formula, we decided to apply a bounded function to a different standard Brownian motion in each entry of the upper triangular matrix.

In Table 2.2, we can see the results of our experiment.[6] We can see the mean relative error Err_T for each entry in the matrix, i.e. $(X_T)_{12}$ corresponds to the individual process in the upper right corner of the matrix-valued process X_t. We compare `m2` with step size $\Delta_t = 0.1$ using an integration-discretization of $\Delta_t^{\text{Int}} = 10^{-3}$ to `euler` with $\Delta_t = 10^{-3}$ and $\Delta_t = 10^{-4}$.

In all cases, we can see a slightly better accuracy of `euler` with step size 10^{-4} compared

[6] More details about the implementation can be found on https://github.com/kevinkamm/IteratedMagnus/blob/main/SDEgeneral/main.m.

to `m2` and slightly higher computational time for $M = 1000$ simulations. Additionally, `m2` is almost an order more accurate than `euler` with step size 10^{-3} but significantly slower.

Another test with everything the same but $\Delta_t^{\mathrm{Int}} = 10^{-4}$, showed an increase in accuracy for the Magnus expansion. In the upper right corner the error was $2.5 \cdot 10^{-3}$ and on the diagonal the error was of order $2 \cdot 10^{-7}$, however the computational time increased to roughly 12 seconds.

The reason for the good accuracy in the diagonal entries is due to the fact that A_t and B_t being upper triangular implies that the commutator commutes on the diagonal. With each further commutation another off-diagonal is set to zero, which can lead to exact—up to discretization errors of the integrals—results of the Magnus expansion in the corresponding entries.

Conclusively, the Magnus expansion of order 2 can be useful even in the most general case, especially if there are some structural properties as being upper triangular or nilpotency, and the accuracy of the scheme is very good. But it depends on the specific problem if it is faster than the Euler-Maruyama scheme. An expansion of order 3 is out of question in this general setting, because its computational effort would be too high.

Case 2: separable coefficients. We will see that the more we know about the problem, the more the Magnus expansion can gain from it. We will now consider the case of separable coefficients, i.e.

$$A_t := A\, a_t, \quad B_t := B\, b_t,$$

for $A, B \in \mathbb{R}^{d \times d}$ constant matrices and a_t, b_t are scalar, progressively measurable and bounded processes with values in $\mathbb{R}$. In this case, the general expansion formulas seen in Section 2.2.1 simplify a lot, i.e. for order one we have:

$$Y_t^{(1)} = A \int_0^t a_s dW_s + B \int_0^t b_s ds.$$

This is already a huge benefit compared to the general case, because only one Lebesuge and one stochastic integral has to be computed instead of $d \times d$ each.

For order two, we see an even greater simplification, because the scalar processes a_t and b_t commute and we have

$$Y_t^{(2)} = -\frac{1}{2} A^2 \int_0^t a_s^2 ds + \frac{1}{2} [B, A] \left(\int_0^t b_s \int_0^s a_r dW_r ds - \int_0^t a_s \int_0^s b_r dr dW_s \right).$$

Table 2.3.: Upper triangular case with separable coefficients as in (2.4.5). Computational times and errors for each component with $\Delta_t^{\text{Int}} = 10^{-4}$ and $M = 1000$ simulations.

	`euler`	`euler`	`m2`	`m3`
Step size	$\Delta_t = 10^{-3}$	$\Delta_t = 10^{-4}$	$\Delta_t = 0.1$	$\Delta_t = 0.1$
$(X_T)_{11}$	$\text{Err}_T = 1 \cdot 10^{-3}$	$\text{Err}_T = 1 \cdot 10^{-4}$	$\text{Err}_T = 3 \cdot 10^{-15}$	$\text{Err}_T = 3 \cdot 10^{-15}$
$(X_T)_{22}$	$\text{Err}_T = 4 \cdot 10^{-4}$	$\text{Err}_T = 5 \cdot 10^{-5}$	$\text{Err}_T = 2 \cdot 10^{-15}$	$\text{Err}_T = 2 \cdot 10^{-15}$
$(X_T)_{12}$	$\text{Err}_T = 1.5 \cdot 10^{-2}$	$\text{Err}_T = 5.2 \cdot 10^{-3}$	$\text{Err}_T = 4.1 \cdot 10^{-3}$	$\text{Err}_T = 1.9 \cdot 10^{-3}$
Comp. Time	$0.18\,s$	$1.98\,s$	$0.5\,s$	$1.2\,s$

Now, the order three expansion is numerically feasible and reads as

$$\begin{aligned}
Y_t^{(3)} &= \frac{1}{12}\left[[B,A],A\right]\int_0^t a_s^2 \int_0^s b_r dr ds + \frac{1}{4}\left[B,A^2\right]\int_0^t a_s^2 \int_0^s b_r dr ds \\
&- \frac{1}{4}\left[B,A^2\right]\int_0^t b_s \int_0^s a_r^2 dr ds - \frac{1}{12}\left[[B,A],A\right]\int_0^t \int_0^s a_r dW_r \int_0^s b_r dr a_s dW_s \\
&+ \frac{1}{12}\left[[B,A],A\right]\int_0^t \left(\int_0^s a_r dW_r\right)^2 b_s ds + \frac{1}{4}\left[[B,A],A\right]\int_0^t \int_0^s a_r \int_0^r b_u du dW_r a_s dW_s \\
&- \frac{1}{4}\left[[B,A],A\right]\int_0^t \int_0^s b_r \int_0^r a_u dW_u dr a_s ds + \frac{1}{12}\left[[B,A],B\right]\int_0^t \int_0^s b_r dr \int_0^s a_r dW_r b_s ds \\
&- \frac{1}{4}\left[[B,A],B\right]\int_0^t \int_0^s b_r \int_0^r a_u dW_u dr b_s ds + \frac{1}{4}\left[[B,A],B\right]\int_0^t \int_0^s a_r \int_0^r b_u du dr b_s ds \\
&- \frac{1}{12}\left[[B,A],B\right]\int_0^t \left(\int_0^s b_r dr\right)^2 a_s dW_s.
\end{aligned}$$

In our test, we want to stay as close as possible to the general coefficient case, to have a bit of comparability. In particular, we chose for two independent Brownian motions W_t^1, W_t^2

$$\begin{aligned}
A &:= \begin{pmatrix} 0.0892 & 0.4015 \\ 0 & 0.0169 \end{pmatrix}, \quad a_t := f(W_t^1), \quad g(x) := \sqrt{1 + \frac{1}{1+x^2}} \\
B &:= \begin{pmatrix} 0.9234 & 0.3813 \\ 0 & 0.5403 \end{pmatrix}, \quad b_t := g(W_t^2), \quad f(x) := 1 + \frac{1}{1+x^2}.
\end{aligned} \tag{2.4.5}$$

In Table 2.3 we can see the results of our experiment.[7] We can see the mean relative error Err_T for each entry in the matrix, i.e. $(X_T)_{12}$ corresponds to the individual process in the upper right corner of the matrix-valued process X_t. We compare `m2` and `m3` with step size $\Delta_t = 0.1$ using a integration-discretization of $\Delta_t^{\text{Int}} = 10^{-4}$ to `euler` with $\Delta_t = 10^{-3}$ and $\Delta_t = 10^{-4}$.

In all cases, we can see for the solution in the upper right corner a slightly better accuracy of `m2` and `m3` compared to `euler` with step size 10^{-4}. On the diagonal however, the Magnus expansion of order two is already exact up to discretization errors, which have now a significantly smaller impact than in the general coefficient case.

Comparing the computational times, we can see that `m2`, `m3` is almost four, twice times as

[7] More details about the implementation can be found on https://github.com/kevinkamm/IteratedMagnus/blob/main/SDEseparable/main.m.

Table 2.4.: Upper triangular case with constant coefficients as in (2.4.6). Computational times and errors for each component with $\Delta_t^{\mathrm{Int}} = 10^{-3}$ and $M = 1000$ simulations.

Component Step size	`euler` $\Delta_t = 10^{-3}$	`euler` $\Delta_t = 10^{-4}$	`m2` $\Delta_t = 1$	`m3` $\Delta_t = 1$
$(X_T)_{11}$	$\mathrm{Err}_T = 5 \cdot 10^{-5}$	$\mathrm{Err}_T = 4 \cdot 10^{-4}$	$\mathrm{Err}_T = 3 \cdot 10^{-18}$	$\mathrm{Err}_T = 3 \cdot 10^{-18}$
$(X_T)_{22}$	$\mathrm{Err}_T = 1 \cdot 10^{-5}$	$\mathrm{Err}_T = 1 \cdot 10^{-4}$	$\mathrm{Err}_T = 3 \cdot 10^{-19}$	$\mathrm{Err}_T = 3 \cdot 10^{-19}$
$(X_T)_{12}$	$\mathrm{Err}_T = 1.8 \cdot 10^{-2}$	$\mathrm{Err}_T = 5.2 \cdot 10^{-3}$	$\mathrm{Err}_T = 3.2 \cdot 10^{-2}$	$\mathrm{Err}_T = 2.1 \cdot 10^{-3}$
Comp. Time	$0.055\,s$	$0.56\,s$	$0.082\,s$	$0.086\,s$

fast as `euler` with step size 10^{-4}, respectively, and three, six times slower than `euler` with step size 10^{-3}.

Again, we can see that the Magnus expansion has a good accuracy and an Euler scheme with comparable error is now slower than the Magnus schemes.

If $a_t := g(t)$ and $b_t := f(t)$ are deterministic polynomials, then the stochastic integrals can be expressed by possibly iterated Lebesgue integrals by using Itô's formula, see Lemma A.1.1. A special case with $f \equiv 1 \equiv g$ is part of the next paragraph.

Case 3: constant coefficients. Now, we consider the case, where A_t and B_t are constant matrices, i.e.

$$A := \begin{pmatrix} 0.0892 & 0.4015 \\ 0 & 0.0169 \end{pmatrix}, \quad B := \begin{pmatrix} 0.9234 & 0.3813 \\ 0 & 0.5403 \end{pmatrix}. \tag{2.4.6}$$

The first three terms of the Magnus expansion read as

$$\begin{aligned} Y_t^{(1)} &= Bt + AW_t, \qquad Y_t^{(2)} = [B, A]\left(\int_0^t W_s ds - \frac{1}{2}tW_t\right) - \frac{1}{2}A^2 t, \\ Y_t^{(3)} &= [[B, A], A]\left(\frac{1}{2}\int_0^t W_s^2 ds - \frac{1}{2}W_t \int_0^t W_s ds + \frac{1}{12}tW_t^2\right) \\ &\quad + [[B, A], B]\left(\int_0^t sW_s ds - \frac{1}{2}t\int_0^t W_s ds - \frac{1}{12}t^2 W_t\right). \end{aligned} \tag{2.4.7}$$

and a detailed derivation can be found in the appendix of Kamm, Pagliarani and Pascucci (2022) or apply Lemma A.1.1 to the expansion formulas from the separable case with $a_t \equiv 1 \equiv b_t$ and collect all terms. As we can see, in order to discretize $Y^{(n)}$ it is not necessary to approximate stochastic integrals. This allows us to use a sparser time grid compared to the previous cases, for which the discretization of stochastic integrals were necessary.

In Table 2.4 we can see the results of our experiment.[8] We can see the mean relative error Err_T for each entry in the matrix, i.e. $(X_T)_{12}$ corresponds to the individual process in the upper right corner of the matrix-valued process X_t. We compare `m2` and `m3` with step size $\Delta_t = 1$ using an integration-discretization of $\Delta_t^{\mathrm{Int}} = 10^{-4}$ to `euler` with $\Delta_t = 10^{-3}$ and $\Delta_t = 10^{-4}$.

[8] More details about the implementation can be found on https://github.com/kevinkamm/IteratedMagnus/blob/main/SDEconst/main.m.

In all cases, we can see for the solution in the upper right corner a slightly better accuracy of `m3` compared to `euler` with step size 10^{-4}. On the diagonal however, the Magnus expansion of order two is already exact up to machine precision and this time there are no discretization errors, since the integrals were solved explicitly.

Comparing the computational times, we can see that `m2`, `m3` are both almost six times as fast as `euler` with step size 10^{-4}, and only slightly slower than `euler` with step size 10^{-3}.

Conclusively, in all the previous tests we could see that the Magnus expansion has a good accuracy and an Euler scheme with comparable error is slower than the Magnus schemes. Additionally, the more we now about the problem and the more we can refine the expansion formulas, the greater the computational advantage.

Remark 2.4.2. In Kamm, Pagliarani and Pascucci (2021) we performed additional tests and highlighted the parallel-in-time features of the Magnus expansion. In particular, we showed a test where the upper right corner of the SDE (2.4.3) is deterministically dependent on time and the other entries are constant. In such a case, the stochastic integrals in the Magnus expansion formulas can be rewritten as Lebesgue integrals, which improves the performance similar to the constant case.

2.4.3. Application to One-Dimensional SPDEs

The aim of this subsection is to apply the Magnus expansion for the numerical solution of parabolic stochastic partial differential equations (SPDEs) with one spatial dimension.[9] We derive an approximation scheme for the general case of variable coefficients and perform several tests in the case of the stochastic heat-equation, for which an exact solution is available.

Let $(\Omega, \mathcal{F}, \mathbb{P}, (\mathcal{F}_t)_{t\geq 0})$ be a filtered probability space endowed with a real Brownian motion W. We consider the stochastic Cauchy problem

$$\begin{cases} du_t(x) = \mathbf{B}_t u_t(x)dt + \mathbf{A}_t u_t(x)dW_t, & t > 0,\ x \in \mathbb{R}, \\ u_0(x) = \phi(x), \end{cases} \tag{2.4.8}$$

where $\mathbf{B}_t$ is the elliptic linear operator acting as

$$\mathbf{B}_t u_t(x) = \frac{1}{2} g_t^{xx}(x)\partial_{xx}u_t(x) + f_t^x(x)\partial_x u_t(x) + h_t(x)u_t(x), \tag{2.4.9}$$

and $\mathbf{A}_t$ is the first-order linear operator acting as

$$\mathbf{A}_t u_t(x) = \sigma_t^x(x)\partial_x u_t(x) + \sigma_t(x)u_t(x). \tag{2.4.10}$$

The coefficients $(h, f^x, g^{xx}, \sigma, \sigma^x)$ are random fields indexed by $(t,x) \in [0,\infty[\times\mathbb{R}$ and the initial datum ϕ is a random field on $\mathbb{R}$. A classical solution to (2.4.8) is understood here as a predictable and almost-surely continuous random field $u = u_t(x)$ over $[0,\infty[\times\mathbb{R}$, such that

[9] More details about the implementation can be found on `https://github.com/kevinkamm/IteratedMagnus/tree/main/SPDE1d`.

$u_t \in C^2(\mathbb{R})$ a.s. for any $t > 0$ and

$$u_t(x) = \phi(x) + \int_0^t \mathbf{B}_s u_s(x) ds + \int_0^t \mathbf{A}_s u_s(x) dW_s, \qquad t \geq 0,\ x \in \mathbb{R}. \tag{2.4.11}$$

There is a vast literature on SPDEs and problems of the form (2.4.8), under suitable measurability, regularity and boundedness assumptions on the coefficients and on the initial datum: see, for instance, Krylov and Rozovskii (1977), Mikulevicius (2000), Chow (2015), Pascucci and Pesce (2020) and the references therein.

Note that, in analogy with deterministic PDEs, the solution of the Cauchy problem (2.4.8) can be written, in some cases, as a convolution of the initial datum with a *stochastic fundamental solution* $p(t, x; 0, \xi)$, i.e.

$$u_t(x) = \int_{\mathbb{R}} p(t, x; 0, \xi)\phi(\xi) d\xi, \qquad (t, x) \in]0, \infty[\times \mathbb{R}, \tag{2.4.12}$$

with $p(t, x; 0, \xi)$ being a random field that solves the SPDE in (2.4.8) with respect to the variables (t, x) and which approximates a Dirac delta centered at ξ as t approaches 0.

In the following subsections, we will demonstrate how to derive an SDE of the form (2.0.1) to approximate the SPDE (2.4.8) and apply the Magnus expansion to it. In the end, we will recall the Euler-Maruyama scheme for the approximating SDE.

Space discretization and Magnus expansion. We proceed in this paragraph by the so-called *method of lines*. In this context, it means that we discretize the spatial derivatives but do not discretize time.

Therefore, we introduce the following homogeneous grid $\mathbb{X}^{n_x}_{a_x,b_x}$ with $n_x + 2$ points on the subset $[a_x, b_x] \subset \mathbb{R}$ for the space variable

$$\mathbb{X}^{n_x}_{a_x,b_x} := \left\{ x_i^{n_x} \in [a_x, b_x] : x_i^{n_x} = a_x + i\Delta x,\ i = 0, \dots, n_x + 1 \right\}, \qquad \Delta x := \frac{b_x - a_x}{n_x + 1}, \tag{2.4.13}$$

Moreover, let us define

$$u_t^{n_x,1} := (u_t(x_i))_{i=1,\dots,n_x}, \qquad \phi^{n_x,1} := (\phi(x_i))_{i=1,\dots,n_x}.$$

We consider the $n_x \times 1$-dimensional SDE

$$dU_t^{n_x,1} = B_t U_t^{n_x,1} dt + A_t U_t^{n_x,1} dW_t, \quad U_0^{n_x,1} = \phi^{n_x,1}, \tag{2.4.14}$$

where A_t and B_t are matrix-valued processes, which will be defined via space-finite differences in the next paragraph in a way that $U_t^{n_x,1} \approx u_t^{n_x,1}$ with respect to a suitable norm.

Since $\phi^{n_x,1}$ is now a vector in $\mathbb{R}^{n_x,1}$ we can use the standard basis $e_j \in \mathbb{R}^{n_x}$ to express it as a linear combination, i.e. there exists $\lambda_j \in \mathbb{R}$, $j = 1, \dots, n_x$, such that

$$\phi^{n_x,1} = \sum_{j=1}^{n_x} \lambda_j e_j.$$

Since the SDE (2.4.14) is linear, we can solve (2.4.14) taking the standard basis vectors as initial datum first, multiply the solution by its corresponding coordinate λ_j and add up the n_x-solutions to obtain the solution for the Cauchy-problem with initial datum $\phi^{n_x,1}$.

In fact, this is equivalent to solving a matrix-valued SDE first and multiply it by the initial datum to obtain the solution to the Cauchy problem, i.e.

$$\begin{aligned} dX_t &= B_t X_t dt + A_t X_t dW_t, \quad X_0 = I \in \mathbb{R}^{n_x \times n_x}, \\ U_t^{n_x,1} &= X_t \phi^{n_x,1}. \end{aligned} \tag{2.4.15}$$

Now, the SDE (2.4.15) corresponds to the approximation of the fundamental solution $p\,(t, x; 0, \xi)$ and can be approximated with the Magnus expansion, i.e. $X_t \approx \exp\,(Y_t)$.

We will derive A_t and B_t by means of finite-difference techniques, leading to sparse matrices, which will be advantageous for the implementation and is subject of the next paragraph.

Derivation of A_t and B_t. The idea is to discretize the first and second-order derivatives, for which we will use central differences with zero-boundary conditions. Therefore, let us introduce the following matrices corresponding to the finite differences

$$D^x := \frac{1}{2\Delta x} \text{tridiag}^{n_x,n_x}\,(-1, 0, 1) \in \mathbb{R}^{n_x \times n_x}, \quad D^{xx} := \frac{1}{(\Delta x)^2} \text{tridiag}^{n_x,n_x}\,(1, -2, 1) \in \mathbb{R}^{n_x \times n_x}$$

Additionally, we will introduce the following matrix-valued processes corresponding to the coefficient functions on the discretized spatial grid for $t \geq 0$

$$Z_t^w := \text{diag}^{n_x,n_x}\left((z_t^w(x_i))_{i=1,\dots,n_x} \right)$$

for $Z = F, G, H, \Sigma$, $z = f, g, h, \sigma$, respectively, and $w \in \{x, xx\}$.

Let us start with discretizing the first-order derivative with respect to x. First, we replace the partial derivative by the first-order central differences and assume zero-boundary conditions, leading to

$$f_t^x(x_i) \partial_x u_t(x_i) \approx f_t^x(x_i) \frac{u_t(x_{i+1}) - u_t(x_{i-1})}{2\Delta x}$$

for all $i = 1, \dots, n_x$. As aforementioned, we need to extract the correct matrix-valued process for the equation (2.4.15).

In our notations, a derivative in x is a multiplication of the corresponding finite-difference matrix from the left to $u_t^{n_x,1}$, i.e.

$$\left(\frac{u_t(x_{i+1}) - u_t(x_{i-1})}{2\Delta x} \right)_{i=1,\dots,n_x} = D^x U_t^{n_x,1}.$$

The coefficients are now a multiplication from the left-hand side with the diagonal matrix F_t^x corresponding to the values $f_t(x_i)$, i.e. we have

$$(f_t^x(x_i)\partial_x u_t(x_i))_{i=1,\dots,n_x} \approx F_t^x D^x U_t^{n_x,1}.$$

This reasoning holds true for all other derivatives as well and conclusively, we have

$$B_t := H_t + F_t^x D^x + \frac{1}{2} G_t^{xx} D^{xx}, \qquad\qquad A_t := \Sigma_t + \Sigma_t^x D^x.$$

Euler-Maruyama To also have an approximation for the fundamental solution, we derive the Euler-Maruyama scheme for (2.4.15). In this case, we have to discretize the time-derivative as well and with the same notation and reasoning from above we obtain

$$X_{t_{k+1}} \approx X_{t_k} + \left(H_{t_k} + F_{t_k}^x D^x + \frac{1}{2} G_{t_k}^{xx} D^{xx}\right) X_{t_k} \Delta t + \left(\Sigma_{t_k} + \Sigma_{t_k}^x D^x\right) X_{t_k} \Delta W_{t_k}, \tag{2.4.16}$$

where $\Delta t := t_{k+1} - t_k > 0$ for any k and $\Delta W_{t_k} := W_{t_{k+1}} - W_{t_k}$.

Preliminaries for the numerical tests. In the following numerical experiments we will use the notations in Table 2.5 in the plots and descriptions.

For the space grid $\mathbb{X}_{a_x,b_x}^{n_x}$ we will use a symmetric grid around zero with a cut-off region of $[-4, 4]$ and d grid points.

Regarding the integration discretization Δ_t^{Int}, we have observed that the computational times in general increase with smaller discretization but the final errors change insignificantly for $\Delta_t^{\text{Int}} < 10^{-5}$.

We suggest to use an integration-discretization equal to 10^{-4} or 10^{-5} in the separable coefficient case and to 10^{-3} or 10^{-4} in the constant coefficient case. Henceforth, we will set it to $\Delta_t^{\text{Int}} = 10^{-4}$ for all tests.

We have verified a linear behaviour (with a slope less than one until the GPU is fully saturated) in the number of simulations M. In our experiments, we decided to use $M = 100$ simulations and display always the average computational times for one simulation.

Also the computational effort with respect to the finite time horizon T scales linearly for all methods. Thus, we use $T = 1$ as our terminal time.

We used for the calculations `Matlab 2022a` with the Parallel Computing Toolbox running on Debian GNU/Linux 10 (buster), on a machine with the following specifications: processor 2x AMD EPYC 7301 CPU @ 2.20 GHz, 256 GB RAM and a NVIDIA Tesla V100 PCIe (32 GB HBM2 RAM). We limit ourselves to 12 CPU cores to highlight that the Magnus expansion not only works very well on clusters but also on an average desktop computer.

The next subsection is structured as follows: First, we derive the explicit solution of the fundamental solution of the heat equation. Then, we discuss the impact of the step-size Δ_t for the iterated Magnus scheme regarding computational times and errors. Next, we look at the boundary effects over time. This is followed by a comparison of the Magnus scheme with the Euler-Maruyama scheme with different sizes of the space grid. Last but not least, we will show a similar behaviour in the separable coefficient case compared to the constant case.

Table 2.5.: Notations for the numerical experiments.

`euler`	Euler-Maruyama scheme (2.4.16)
`m1`	Iterated Magnus scheme of order 1
`m2`	Iterated Magnus scheme of order 2
`m3`	Iterated Magnus scheme of order 3
M	Number of simulations
d	Number of grid points in $\mathbb{X}$
Δ_t	step-size of `euler` or Magnus
Δ_t^{Int}	discretization of the Lebesgue and stochastic integrals for Magnus
M2, x	Magnus order 2 with step-size $\Delta_t = x$
M3, x	Magnus order 3 with step-size $\Delta_t = x$
E, x	`euler` with step-size $\Delta_t = x$

2.4.3.1. The Magnus Expansion for the Stochastic Heat Equation with Constant Coefficients

In this subsection, we apply the Magnus expansion to the stochastic heat equation.[10]

We consider a special case of (2.4.8) with

$$h \equiv f^x \equiv \sigma \equiv 0, \quad g_t^{xx}(x) := a \in \mathbb{R}_{>0}, \quad \sigma_t^x(x) := \sigma \in \mathbb{R}_{>0}, \tag{2.4.17}$$

leading to

$$du_t = \frac{a}{2}\partial_{xx}u_t(x)dt + \sigma\partial_x u_t(x)dW_t, \qquad t > 0,\ x \in \mathbb{R},$$

with $a > \sigma^2$, whose stochastic fundamental solution is given explicitly by

$$p(t, x; 0, \xi) := \frac{1}{\sqrt{2\pi(a-\sigma^2)t}} \exp\left(-\frac{(x + \sigma W_t - \xi)^2}{2\,(a-\sigma^2)\,t}\right), \qquad t > 0,\ x, \xi \in \mathbb{R}. \tag{2.4.18}$$

The components of X_t in (2.4.15) can be regarded as approximations of the integrals of the fundamental solution of the SPDE in (2.4.8), when it exists, on each sub-interval $[\frac{1}{2}(x_{j-1} + x_j), \frac{1}{2}(x_j + x_{j+1})]$, namely

$$(X_t)_{i,j} \approx \int_{\frac{1}{2}(x_{j-1}^d + x_j^d)}^{\frac{1}{2}(x_j^d + x_{j+1}^d)} p\,(t, x_i; 0, \xi)\, d\xi, \qquad i, j = 1, \ldots, d. \tag{2.4.19}$$

The processes A_t and B_t in (2.4.15) now read as

$$A_t \equiv \frac{a}{2\,(\Delta x)^2}\begin{pmatrix} -2 & 1 & \cdots & 0 \\ 1 & -2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 0 & \cdots & 1 & -2 \end{pmatrix}, \qquad B_t \equiv \frac{\sigma}{2\Delta x}\begin{pmatrix} 0 & 1 & \cdots & 0 \\ -1 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 0 & \cdots & -1 & 0 \end{pmatrix}.$$

[10] More details about the implementation can be found on https://github.com/kevinkamm/IteratedMagnus/blob/main/SPDE1d/Const/Code/main.m.

In particular, they do not commute and are constant for fixed d.

Having an exact benchmark solution we will now perform some numerical tests to judge the performance of the iterated Magnus scheme.[11] Henceforth, the parameters for the stochastic heat equation will be $a = 1.1$ and $\sigma = \frac{1}{\sqrt{10}}$, so that $a - \sigma^2 = 1 > 0$.

Computational effort and errors with respect to the number of iterations. For this experiment we fix the number of grid points to $d = 200$ but vary the step-size of the Magnus scheme Δ_t. In Figure 2.3 we can see the corresponding results. The left y-axis shows the average computational times for one simulation in a log scale and the right y-axis the mean relative errors $\mathrm{Err}_T^{d,4}$ also in a log scale. The computational times (in seconds) of `m2` are depicted in light blue and of `m3` in dark blue. Moreover, the mean relative errors for `m2` are orange and for `m3` red.

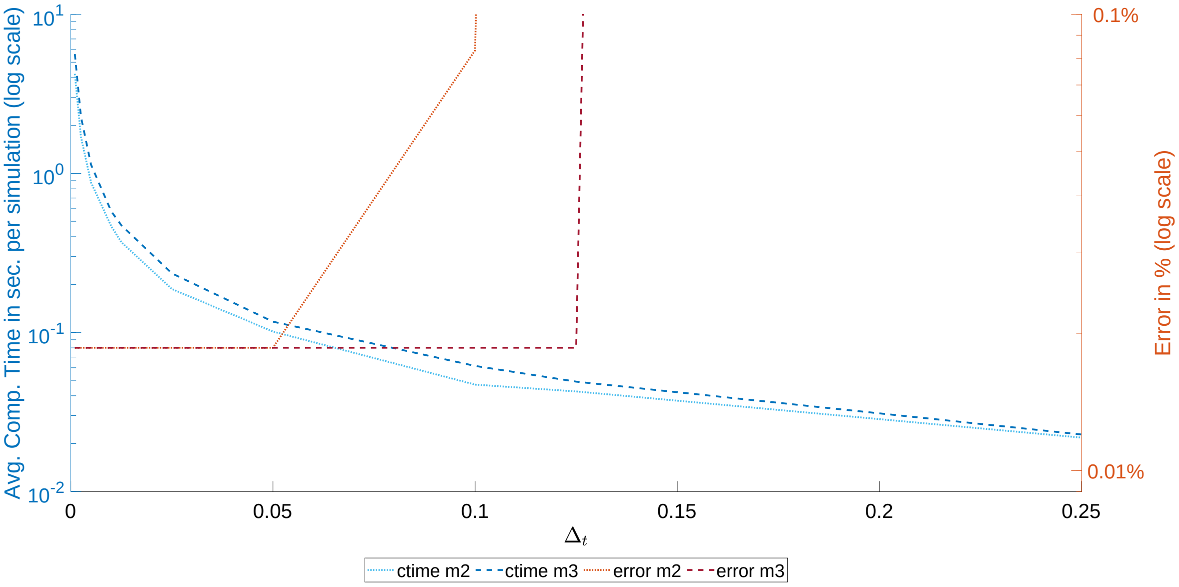


Figure 2.3.: Constant coefficients as in (2.4.17): Computational times and errors of the Magnus expansion for varying step-size Δ_t with fixed spatial dimension $d = 200$.

We can see an almost constant error up until step size 0.05. At 0.05 the error for `m2` starts to increase and explodes around 0.1. Similarly, for `m3` we see an explosion around 0.0125. The explosions for large step-sizes are not surprising, since the step-size is determined by the underlying stopping times for the convergence of the Magnus scheme. Therefore, this experiment indicates that any step-size less than 0.05, 0.1 for $d = 200$ is well within the convergence radius of `m2`, `m3`, respectively, and yields stable results.

For other spatial dimensions d this breaking point might be different. Moreover, we can see that the computational time increases more and more for smaller step-sizes, while the error for both methods stays almost constant and close to each other.

This suggests that one should choose the step-size as large as possible for the iterated

[11] More details about the implementation can be found on `https://github.com/kevinkamm/IteratedMagnus/blob/main/SPDE1d/Const/Code/coefficients.m`, function *exactHeatEquation1*.

Magnus scheme to gain the maximal performance. However, being too greedy will lead to blow-ups of some trajectories.

Also as a side note, usually if one increases the spatial dimension d, then one has to choose a smaller time step-size for the Magnus methods as well: this will be shown in Figure 2.6–2.8.

Mean errors and boundary effects over time. For this experiment we fix the grid points in each space grid to $d = 300$.

In Figure 2.5, we can see the mean absolute errors of the entire spatial grid as a two-dimensional plot. A deep blue color indicates a small error and a bright yellow color an error up to 10^{-1}. The black rectangle is the corresponding region for $\kappa = 1$. The black number within the rectangle is the average mean absolute error of the corresponding region. The picture on the left-hand side is the area of errors at $t = 0.25$ and on the right-hand side at $t = 1$.

We can see that the errors on the diagonal from the upper right to the lower left corners are decreasing over time. To explain this, one should note that the fundamental solution of the heat equation starts as a dirac-delta and spreads out over time (see Figure 2.4). We can see that the Magnus expansion is able to recover from its initial larger errors on the diagonal over time and the error is smoothing out.

Comparison to the Euler-Maruyama scheme For this experiment, we will compare different choices of parameters for both the Magnus scheme and Euler-Maruyama scheme. There are essentially two major parameters contributing to the possible accuracy. One is the time step-size of the individual schemes and the other one the space discretization. Hence, we compare Euler and Magnus methods with different time step-sizes for different space discretizations $d = 100, 200, 300$ to increase the level of accuracy. In the Figures 2.6, 2.7 and 2.8 the left y-axis shows the average computational times in a log scale and the right y-axis the mean relative errors $\mathrm{Err}_T^{d,4}$ also in a linear scale. The computational times (in seconds) are depicted in the left blue columns and the mean relative errors in the red right columns for each method.

As mentioned in Table 2.5, “E, x” denotes Euler with step-size $\Delta_t = x$ and “M2, x”, “M3, x” denotes Magnus with step-size $\Delta_t = x$ for order 2 and 3, respectively. In Figure 2.6 we compare the errors and computational times of the methods with spatial dimension $d = 100$, in Figure 2.7 with $d = 200$ and in Figure 2.8 with $d = 300$.

Let us focus on Figure 2.6 with $d = 100$. We can see that four different methods are compared: the Euler method with step-size $\Delta_t = 10^{-3}$ and $\Delta_t = 10^{-4}$, as well as the Magnus method with step-size $\Delta_t = 0.1$ of order 2 and order 3. It is notable that the Euler method with step-size $\Delta_t = 10^{-4}$ and the Magnus methods perform almost the same with respect to the error. The Euler method with step-size $\Delta_t = 10^{-3}$ is exploding already. Overall, the Magnus methods were the fastest methods. The Magnus methods are roughly 70 times faster than the Euler method with step-size $\Delta_t = 10^{-4}$ and has a slightly better accuracy. Surprisingly, the Magnus method of order 3 is faster than order 2 in this picture, but the computational times are small, so this can be attributed to some fluctuations of the computational device or internal caching of Matlab.

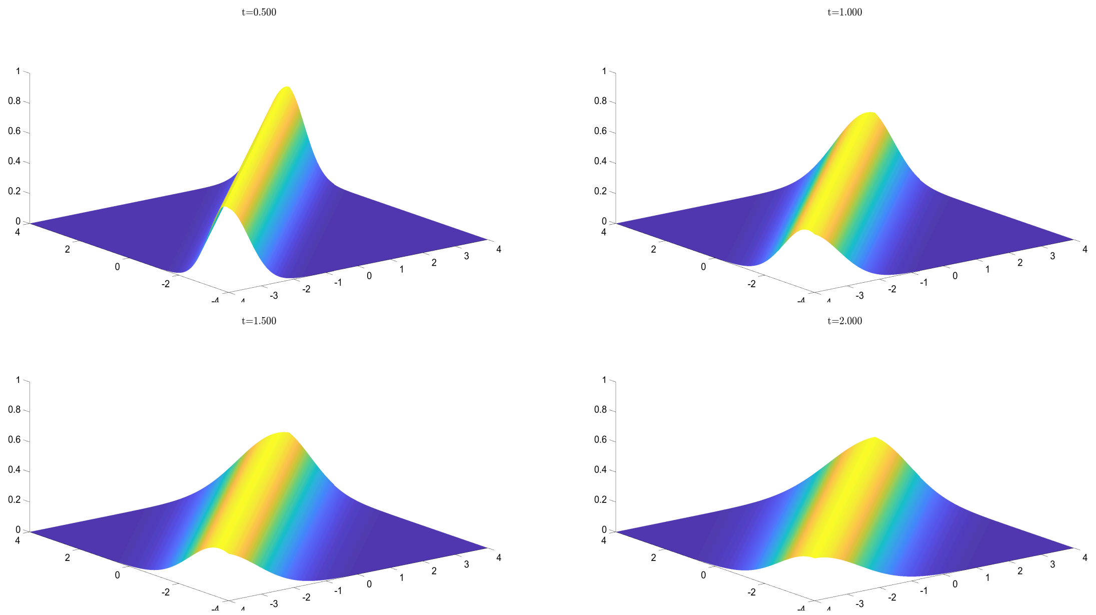


Figure 2.4.: One trajectory of the fundamental solution of the constant heat equation (2.4.18) at $t = 0.5$ (upper left), $t = 1$ (upper right), $t = 1.5$ (bottom left) and $t = 2$ (bottom right).

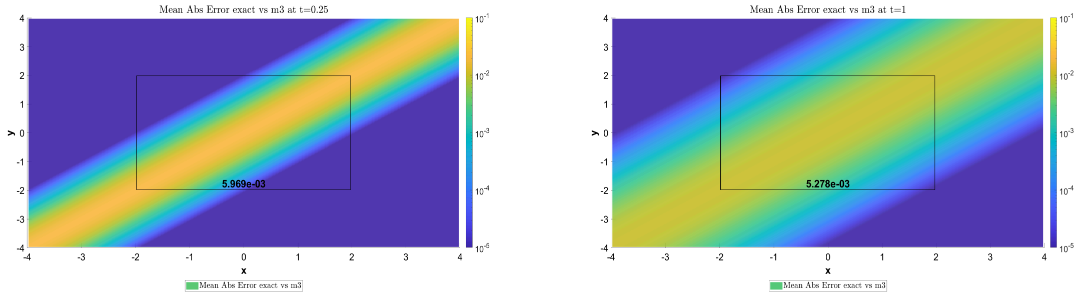


Figure 2.5.: Constant coefficients as in (2.4.17): Absolute Errors of `m3` compared to `exact` using $d = 300$ grid points at $t = 0.25$ (left) and $t = 1$ (right).

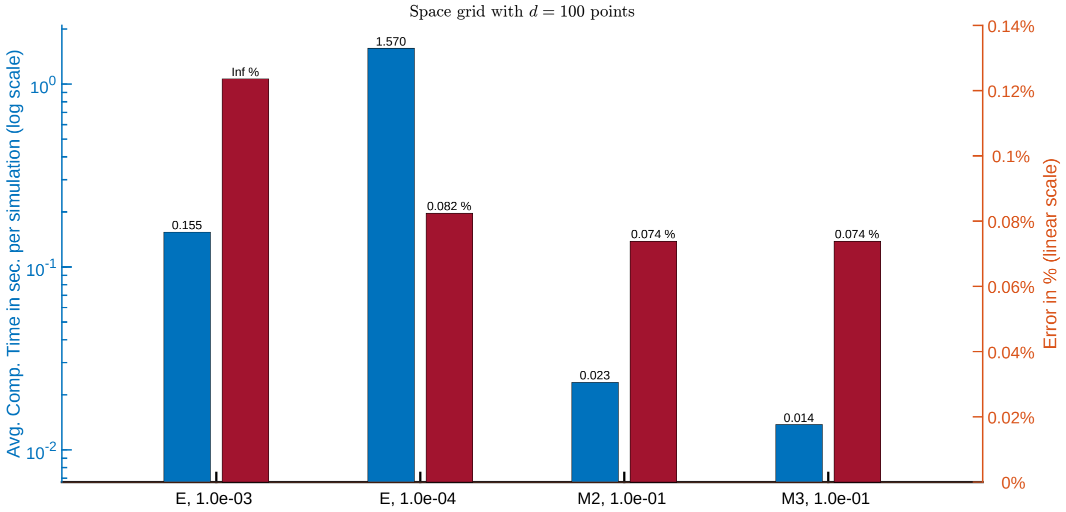


Figure 2.6.: Constant coefficients as in (2.4.17): Computational times and errors of the Magnus expansion and Euler scheme for $d = 100$.

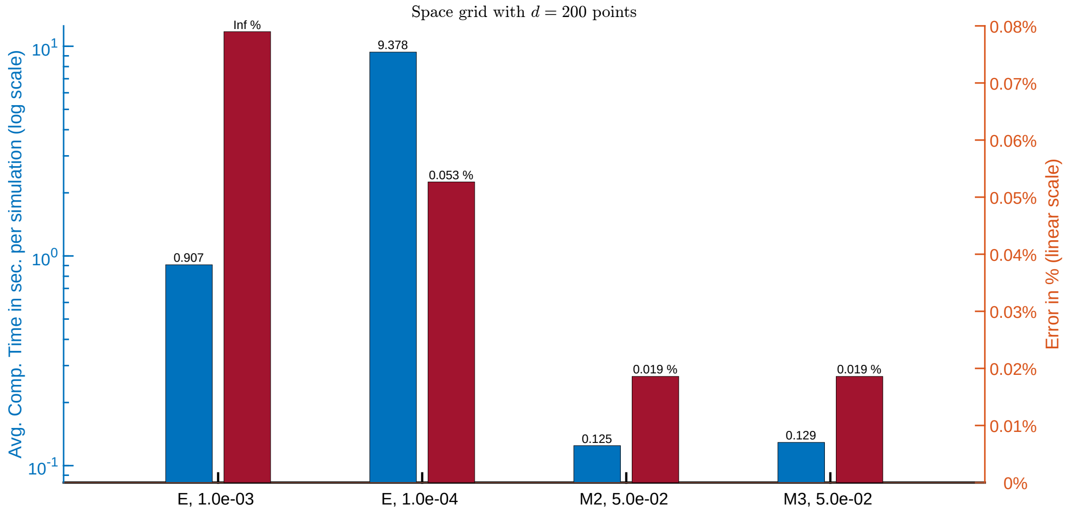


Figure 2.7.: Constant coefficients as in (2.4.17): Computational times and errors of the Magnus expansion and Euler scheme for $d = 200$.

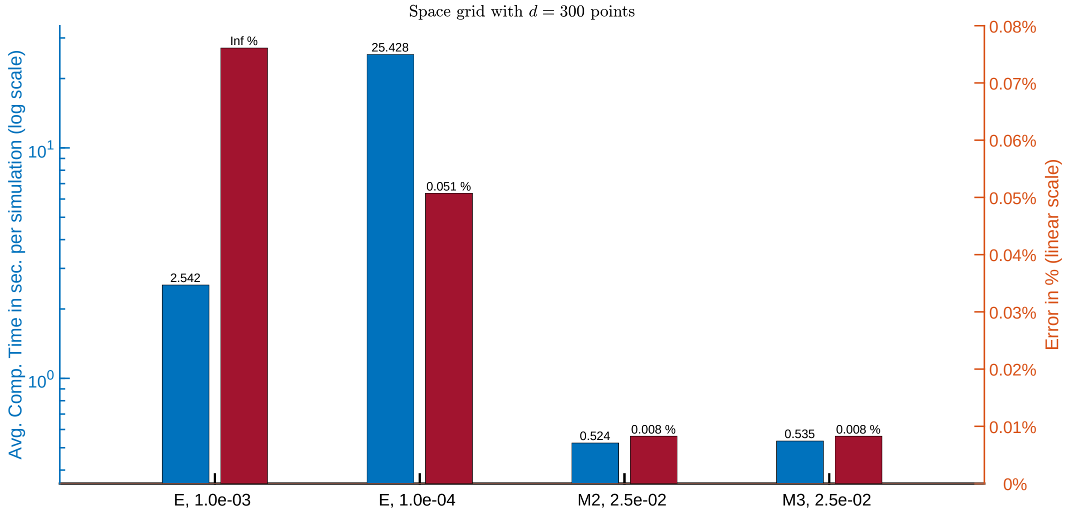


Figure 2.8.: Constant coefficients as in (2.4.17): Computational times and errors of the Magnus expansion and Euler scheme for $d = 300$.

Now, let us consider Figure 2.7 with $d = 200$. Again, we can see that two Euler methods and two Magnus methods are compared to each other but this time we have a step-size $\Delta_t = 0.05$ for the Magnus methods. Similar to Figure 2.6, we can see that the Euler method with step-size $\Delta_t = 10^{-3}$ is exploding and the Magnus methods are best in terms of accuracy. However, this time the Euler method with step-size $\Delta_t = 10^{-4}$ has roughly twice the error compared to the Magnus methods and is still 70 times slower than the Magnus methods.

In Figure 2.8 with $d = 300$ the Euler method with step-size $\Delta_t = 10^{-3}$ is also exploding. Therefore, we compare the Euler method with step-size $\Delta_t = 10^{-4}$ to the Magnus method with with step-size $\Delta_t = 0.025$ with order two and three. This time the Euler method is six times worse in terms of accuracy and 50 times slower than Magnus.

Overall, from these observation it is clear that an Euler scheme with a fine time-discretization is essential to make it comparable to the iterated Magnus scheme in terms of accuracy. More-

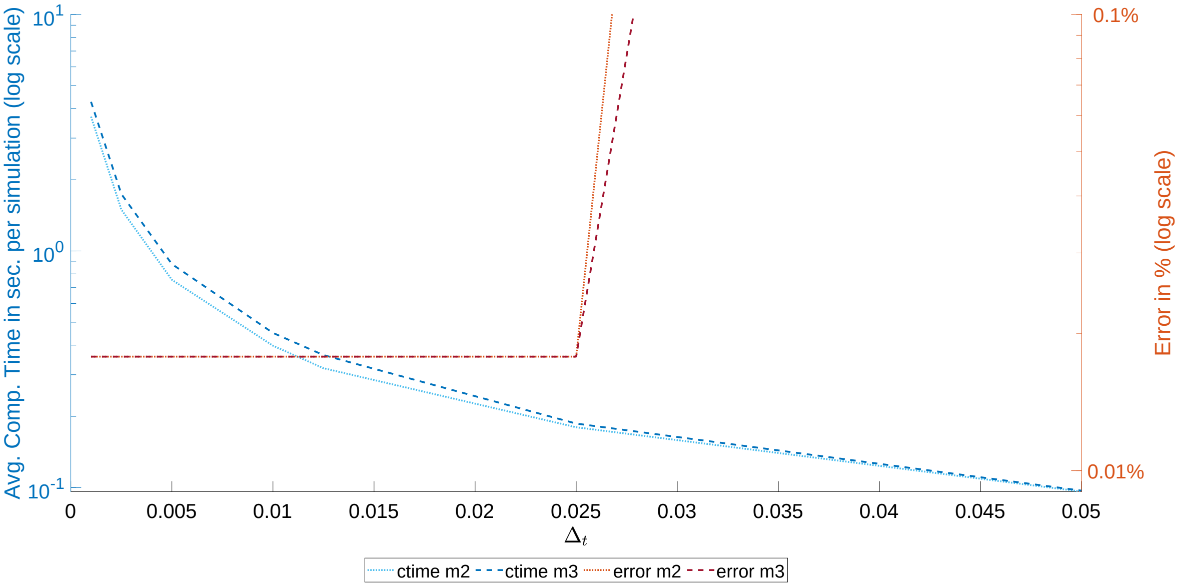


Figure 2.9.: Variable coefficients as in (2.4.20): Computational times and errors compared to Euler with $\Delta_t = 10^{-5}$ of the Magnus expansion for varying step-size Δ_t with fixed spatial dimension $d = 200$.

over, increasing the number of grid points is leading to less accurate errors using the Euler method with step-size $\Delta_t = 10^{-4}$ compared to the Magnus schemes with corresponding step-sizes, while the Magnus methods remain roughly 50 times faster in all tests.

2.4.3.2. The Magnus Expansion for the Stochastic Heat Equation with Separable Coefficients

In this brief subsection, we will perform some tests in the case of separable coefficients.[12] In particular, we choose bounded, smooth coefficients of the form

$$\begin{aligned} h \equiv f^x \equiv \sigma \equiv 0, \\ g_t^{xx}(x) := a \cdot f\left(B_t\right), \quad a \in \mathbb{R}_{\geq 0}, \quad f(x) := \left(1 + \frac{1}{x^2+1}\right), \\ \sigma_t^x(x) := \sigma \cdot g\left(B_t\right), \quad a \in \mathbb{R}_{\geq 0}, \quad g(x) := \sqrt{\left(1 + \frac{1}{x^2+1}\right)}, \end{aligned} \tag{2.4.20}$$

with B_t a standard Brownian motion and $a = 1.1$, $\sigma = \frac{1}{\sqrt{10}}$ satisfying $g^{xx}(x) - (\sigma^x(x))^2 > 0$, as in the constant coefficient case.

Analog to Figure 2.3, we show in Figure 2.9 the case of varying step-sizes for the Magnus method with fixed spatial discretization $d = 200$. The average computational times of `m2` and `m3` in seconds, per simulation, are again depicted in light blue and dark blue, respectively. The errors are this time with respect to the Euler method with $\Delta_t = 10^{-5}$, since an exact solution is not available, and again illustrated as orange for `m2` and red for `m3`.

[12] More details about the implementation can be found on https://github.com/kevinkamm/IteratedMagnus/blob/main/SPDE1d/Separable/Code/main.m.

Table 2.6.: Separable coefficients as in (2.4.20): Computational times and errors compared to Euler with $\Delta_t = 10^{-5}$ of the Magnus expansion for different matrix dimension $d = 100, 200, 300$.

Method	Mean Rel. Error (in %)	Comp. Time (in sec./simulation)
	$d = 100$	
`euler`, $\Delta_t = 10^{-3}$	0.193 %	0.152
`euler`, $\Delta_t = 10^{-4}$	0.049 %	1.611
`euler`, $\Delta_t = 10^{-5}$	–	16.85
`m2`, $\Delta_t = 0.1$	0.018 %	0.13
`m3`, $\Delta_t = 0.1$	0.018 %	0.24
	$d = 200$	
`euler`, $\Delta_t = 10^{-3}$	∞ %	0.81
`euler`, $\Delta_t = 10^{-4}$	0.049 %	8.11
`euler`, $\Delta_t = 10^{-5}$	–	82.7
`m2`, $\Delta_t = 0.025$	0.019 %	0.241
`m3`, $\Delta_t = 0.025$	0.019 %	0.242
	$d = 300$	
`euler`, $\Delta_t = 10^{-3}$	∞ %	2.8
`euler`, $\Delta_t = 10^{-4}$	0.05 %	28.54
`euler`, $\Delta_t = 10^{-5}$	–	286.28
`m2`, $\Delta_t = 0.01$	0.019 %	4.28
`m3`, $\Delta_t = 0.01$	0.019 %	5.0

We can see that Figure 2.9 looks almost identical to Figure 2.3. The only difference is, that the explosion of the Magnus methods happens earlier than in the constant coefficient case due to the different convergence radius.

Any method with step size less or equal than 0.025 yield roughly the same error but the computational effort increases with decreasing step size.

In our next experiment, similarly to Figure 2.14–2.16, we show in Table 2.6 the mean relative errors and computational times for $d = 100, 200, 300$. This time we use an Euler method with $\Delta_t = 10^{-5}$ as our reference solution and compare an Euler method with $\Delta_t = 10^{-4}$, as well as `m2` and `m3` to it. The results are given in Table 2.6.

We can see that computational times are higher for the Magnus scheme than in the constant coefficient case. This increase of computational effort can be attributed to the more complex expansion formulas and the smaller step size for the corresponding space grid. The Euler scheme did not suffer from a performance decrease.

All in all, we come to the same conclusion as in the constant coefficient case. The accuracy of the Magnus schemes is very high and it is faster than an Euler method with similar accuracy.

2.4.4. Application to Two-Dimensional SPDEs

In this section, we investigate the application of the Mangus expansion to the numerical solution of a class of hypoelliptic stochastic partial differential equations (SPDEs) that naturally arise in physics and mathematical finance.[13] The deterministic prototype of such SPDEs is the classical Langevin equation

$$\frac{1}{2}\partial_{vv}u_t + v\partial_x u - \partial_t u_t = 0, \tag{2.4.21}$$

where the variables $t \geq 0$, $x \in \mathbb{R}$ and $v \in \mathbb{R}$ respectively stand for time, position and velocity, and the unknown $u_t = u_t(x, v)$ stands for the density of a particle in the phase space. Notice that (2.4.21) is a degenerate, non-uniformly parabolic PDE. Perturbations of (2.4.21) with variable coefficients appear in linear and non-linear form in several applications in kinetic theory (cf. Lions (1994), Desvillettes and Villani (2001) and Cercignani (1988)); also, (2.4.21) describes path-dependent financial derivatives such as Asian options and volatility contracts (cf. Pascucci (2011) and Di Francesco and Pascucci (2004)).

We consider here the stochastic version of (2.4.21), which is the kinetic SPDE

$$du_t = \left(\frac{a_t}{2}\partial_{vv} + v\partial_x + b_t\partial_v + c_t\right)u_t dt + \left(\sigma_t\partial_v + \beta_t\right)u_t dW_t, \tag{2.4.22}$$

where a_t, b_t, c_t, σ_t and β_t are non-constant (i.e., for example, $a_t = a_t(x, v)$) and possibly random coefficients. Here W denotes a Wiener process defined on a complete probability space $(\Omega, \mathcal{F}, \mathbb{P})$ endowed with a filtration $(\mathcal{F}_t)_{t\geq 0}$ satisfying the usual conditions. SPDE (2.4.22) naturally appears in stochastic filtering theory: as shown in Pascucci and Pesce (2022a); Pascucci and Pesce (2022b), the fundamental solution of (2.4.22) is the conditional transition density of a two-dimensional stochastic process representing the position and the velocity of a particle under partial observation.

The numerical solution of (2.4.22) is a challenging issue, as standard techniques, such as Euler methods, can be excessively time-consuming due to the two spatial variables.

We proceed similar to the previous section. First, we show how to discretize a more general parabolic-type SPDE by an SDE of the type (2.0.1). After that, we show the numerical experiments in the special case of the stochastic Langevin equation with constant coefficients where the exact solution is available in closed form. In this case we will perform several tests concerning the parameters of the Magnus expansion and discuss their impact. Then we compare the performance of the iterated Magnus expansion with that of standard Euler-Maruyama schemes. In Section 2.4.4.2 we consider a more general case with separable coefficients.

Now, let us show how to derive a numerical scheme for a general parabolic SPDE by combining space-discretization and Magnus expansion. We will perform some formal computations, which hold even for a fairly general class of SPDEs. Since the computations are understood in a formal manner, we do not impose any further conditions on the coefficients

[13] More details about the implementation can be found on https://github.com/kevinkamm/MagnusSPDE2D/blob/main/IteratedMagnus/main.m.

of the following type of SPDE

$$\begin{cases} du_t(x,v) = \Big(h_t(x,v)u_t(x,v) + f_t^x(x,v)\partial_x u_t(x,v) + f_t^v(x,v)\partial_v u_t(x,v) \\ \qquad + \frac{1}{2}g_t^{xx}(x,v)\partial_{xx}u_t(x,v) + g_t^{xv}(x,v)\partial_{xv}u_t(x,v) + \frac{1}{2}g_t^{vv}(x,v)\partial_{vv}u_t(x,v)\Big)dt \\ \qquad + \left(\sigma_t(x,v)u_t(x,v) + \sigma_t^x(x,v)\partial_x u_t(x,v) + \sigma_t^v(x,v)\partial_v u_t(x,v)\right)dW_t \\ u_0(x,v) = \phi(x,v). \end{cases} \tag{2.4.23}$$

In the following subsections, we will demonstrate how to derive an SDE of the form (2.0.1) to approximate the SPDE (2.4.23) and apply the Magnus expansion to it. In the end, we will recall the Euler-Maruyama scheme for the approximating SDE.

Space discretization and Magnus expansion. As in Section 2.4.3, we will use the method of lines, meaning that we discretize the space variables but not time. After that we will vectorize the equation to derive a matrix-valued equation. We introduce the following two homogeneous grids for position and velocity respectively: $\mathbb{X}^{n_x}_{a_x,b_x}$ with n_x+2 points on the subset $[a_x,b_x]\subset\mathbb{R}$, $\mathbb{V}^{n_v}_{a_v,b_v}$ with n_v+2 points on the subset $[a_v,b_v]\subset\mathbb{R}$:

$$\mathbb{X}^{n_x}_{a_x,b_x} := \{x_i^{n_x}\in[a_x,b_x] : x_i^{n_x} = a_x + i\Delta x,\ i=0,\dots,n_x+1\}, \qquad \Delta x := \frac{b_x-a_x}{n_x+1}, \tag{2.4.24}$$

$$\mathbb{V}^{n_v}_{a_v,b_v} := \left\{v_j^{n_v}\in[a_v,b_v] : v_j^{n_v} = a_v + j\Delta v,\ j=0,\dots,n_v+1\right\}, \qquad \Delta v := \frac{b_v-a_v}{n_v+1}. \tag{2.4.25}$$

Let us denote by vec the isomorphism of transforming a matrix to a larger column-vector by stacking each column in the matrix below each other, i.e.

$$\begin{aligned} &\mathrm{vec} : \mathbb{R}^{n_x\times n_v} \to \mathbb{R}^{n_x\cdot n_v\times 1}, \\ &\qquad A=[a_{ij}] \mapsto \mathrm{vec}(A) := [a_{1,1},\dots,a_{n_x,1},a_{1,2},\dots,a_{n_x,2},\dots,a_{1,n_v},\dots,a_{n_x,n_v}]^\top. \end{aligned}$$

Moreover, let us define

$$u_t^{n_x,n_v} := (u_t(x_i,v_j))_{\substack{i=1,\dots,n_x\\ j=1,\dots,n_v}}, \qquad \phi^{n_x,n_v} := (\phi(x_i,v_j))_{\substack{i=1,\dots,n_x\\ j=1,\dots,n_v}}, \qquad \Phi^{n_x n_v} := \mathrm{vec}\left(\phi^{n_x,n_v}\right).$$

We consider the $n_x n_v$-dimensional SDE

$$dU_t^{n_x n_v} = B_t U_t^{n_x n_v}dt + A_t U_t^{n_x n_v}dW_t, \quad U_0^{n_x n_v} = \Phi^{n_x n_v}, \tag{2.4.26}$$

where A_t and B_t are $(n_x n_v \times n_x n_v)$-matrix-valued processes, which will be defined via space-finite differences in the next paragraph in a way that $U_t^{n_x n_v} \approx \mathrm{vec}\left(u_t^{n_x,n_v}\right)$ with respect to a suitable norm. This equation can be solved by first computing its fundamental solution, then by multiplying it with the initial datum, i.e.

$$\begin{aligned} dX_t &= B_t X_t dt + A_t X_t dW_t, \quad X_0 = I \in \mathbb{R}^{n_x n_v \times n_x n_v}, \\ U_t^{n_x n_v} &= X_t \Phi^{n_x n_v}. \end{aligned}$$

The fundamental solution now can be approximated with the Magnus expansion, i.e. $X_t \approx \exp(Y_t)$. Since A_t and B_t will be very large in this case, we will utilize the sparsity of A_t and B_t, as well as using a special algorithm specifically designed to compute the matrix-exponential times a vector denoted by `expmvtay2`, which does not need to compute the whole matrix-exponential first. This is crucial for the implementation and explained in further detail in Ibáñez et al. (2022).

Derivation of A_t and B_t. The idea is to discretize the first and second-order space derivatives, for which we will use central differences with zero-boundary conditions. Therefore, let us introduce the following matrices corresponding to the finite differences

$$\begin{aligned} D^x &:= \frac{1}{2\Delta x} \text{tridiag}^{n_x,n_x}(-1,0,1), & D^v &:= \frac{1}{2\Delta v} \text{tridiag}^{n_v,n_v}(-1,0,1), \\ D^{xx} &:= \frac{1}{(\Delta x)^2} \text{tridiag}^{n_x,n_x}(1,-2,1), & D^{vv} &:= \frac{1}{(\Delta v)^2} \text{tridiag}^{n_v,n_v}(1,-2,1). \end{aligned}$$

Additionally, we will introduce the following matrices corresponding to the coefficient functions on the discretized space grid

$$Z_t^w := \left(z_t^w(x_i, v_j)\right)_{\substack{i=1,\dots,n_x \\ j=1,\dots,n_v}}, \qquad \Sigma_t^w := \left(\sigma_t^w(x_i, v_j)\right)_{\substack{i=1,\dots,n_x \\ j=1,\dots,n_v}}$$

for $Z = F, G, H$, $z = f, g, h$, respectively, and $w \in \{x, v, xx, xv, vv\}$.

Let us start with discretizing the first-order derivative with respect to x. First, we replace the partial derivative by the first-order central differences and assume zero-boundary conditions, leading to

$$f_t^x(x_i, v_j) \partial_x u_t(x_i, v_j) \approx f_t^x(x_i, v_j) \frac{u_t(x_{i+1}, v_j) - u_t(x_{i-1}, v_j)}{2\Delta x}$$

for all $i = 1, \dots, n_x$ and $j = 1, \dots, n_v$. As aforementioned, we need to extract the correct coefficient matrix for the vectorized equation (2.4.26).

In our notations, a derivative in x is a multiplication of the corresponding finite-difference matrix from the left to $u_t^{n_x,n_v}$, i.e.

$$\left(\frac{u_t(x_{i+1}, v_j) - u_t(x_{i-1}, v_j)}{2\Delta x} \right)_{\substack{i=1,\dots,n_x \\ j=1,\dots,n_v}} = D^x u_t^{n_x,n_v}.$$

Using the Kronecker product, it is well-known for compatible matrices $D_1 U D_2 = C$ that

$$\operatorname{vec}(C) = \operatorname{vec}(D_1 U D_2) = \left(D_2^\top \otimes D_1 \right) \operatorname{vec}(U) .$$

In our case, this leads to

$$\operatorname{vec}\left(D^x u_t^{n_x, n_v}\right) = \left(I_{n_v} \otimes D^x\right) U_t^{n_x n_v} .$$

Now, we need to deal with the coefficients as well. Denoting by $\odot$ the Hadamard, or elementwise, product, it is easy to see that

$$\operatorname{vec}\left(F_t^x \odot \left(D^x u_t^{n_x, n_v}\right)\right) = \operatorname{vec}\left(F_t^x\right) \odot \operatorname{vec}\left(D^x u_t^{n_x, n_v}\right) = \operatorname{diag}\left(\operatorname{vec}\left(F_t^x\right)\right) \operatorname{vec}\left(D^x u_t^{n_x, n_v}\right) .$$

Using these two observations together yields

$$\left(f_t^x(x_i, v_j) \partial_x u_t(x_i, v_j)\right)_{\substack{i=1,\dots,n_x \\ j=1,\dots,n_v}} \approx \operatorname{diag}\left(\operatorname{vec}\left(F_t^x\right)\right) \left(I_{n_v} \otimes D^x\right) U_t^{n_x n_v} .$$

This reasoning holds true for all other derivatives as well, i.e. an operation in x is a matrix multiplication from the left and in v it is the matrix multiplication from the right with the transposed matrix.

Conclusively, we have

$$\begin{aligned} B_t &:= \operatorname{diag}\left(\operatorname{vec}\left(H_t\right)\right) + \operatorname{diag}\left(\operatorname{vec}\left(F_t^x\right)\right) \left(I_{n_v} \otimes D^x\right) + \operatorname{diag}\left(\operatorname{vec}\left(F_t^v\right)\right) \left(D^v \otimes I_{n_x}\right) \\ &\quad + \frac{1}{2} \operatorname{diag}\left(\operatorname{vec}\left(G_t^{xx}\right)\right) \left(I_{n_v} \otimes D^{xx}\right) + \operatorname{diag}\left(\operatorname{vec}\left(G_t^{xv}\right)\right) \left(D^v \otimes D^x\right) \\ &\quad + \frac{1}{2} \operatorname{diag}\left(\operatorname{vec}\left(G_t^{vv}\right)\right) \left(D^{vv} \otimes I_{n_x}\right) , \\ A_t &:= \operatorname{diag}\left(\operatorname{vec}\left(\Sigma_t\right)\right) + \operatorname{diag}\left(\operatorname{vec}\left(\Sigma_t^x\right)\right) \left(I_{n_v} \otimes D^x\right) + \operatorname{diag}\left(\operatorname{vec}\left(\Sigma_t^v\right)\right) \left(D^v \otimes I_{n_x}\right) . \end{aligned}$$

Remark 2.4.3. We can see that the sparsity of B_t and A_t is mostly determined by the finite-difference matrices due to the elementwise product of the coefficient functions. The coefficient functions can further reduce or increase the sparsity by zero entries or cancellations. Additionally, this implies that the number of non-zero diagonals is independent of the size of B_t and A_t respectively even though the density of non-zero elements will decline for increasing dimensions.

An illustration of the sparsity pattern can be seen in Figure 2.10: In the upper left corner is the pattern for A, followed by B, $[B, A]$, $[[B, A], A]$ and $[[B, A], B]$. The blue lines represent non-zero entries. The pictures are ordered by the number of non-zero diagonals, i.e. a diagonal with at least one non-zero entry, which are 2, 5, 5, 8 and 10 respectively. We can see that the sparsity decreases with the order of the commutator.

Euler-Maruyama In this case, we do not need to vectorize the equation but we have to discretize the time-derivative as well. With the same notation and reasoning from above we

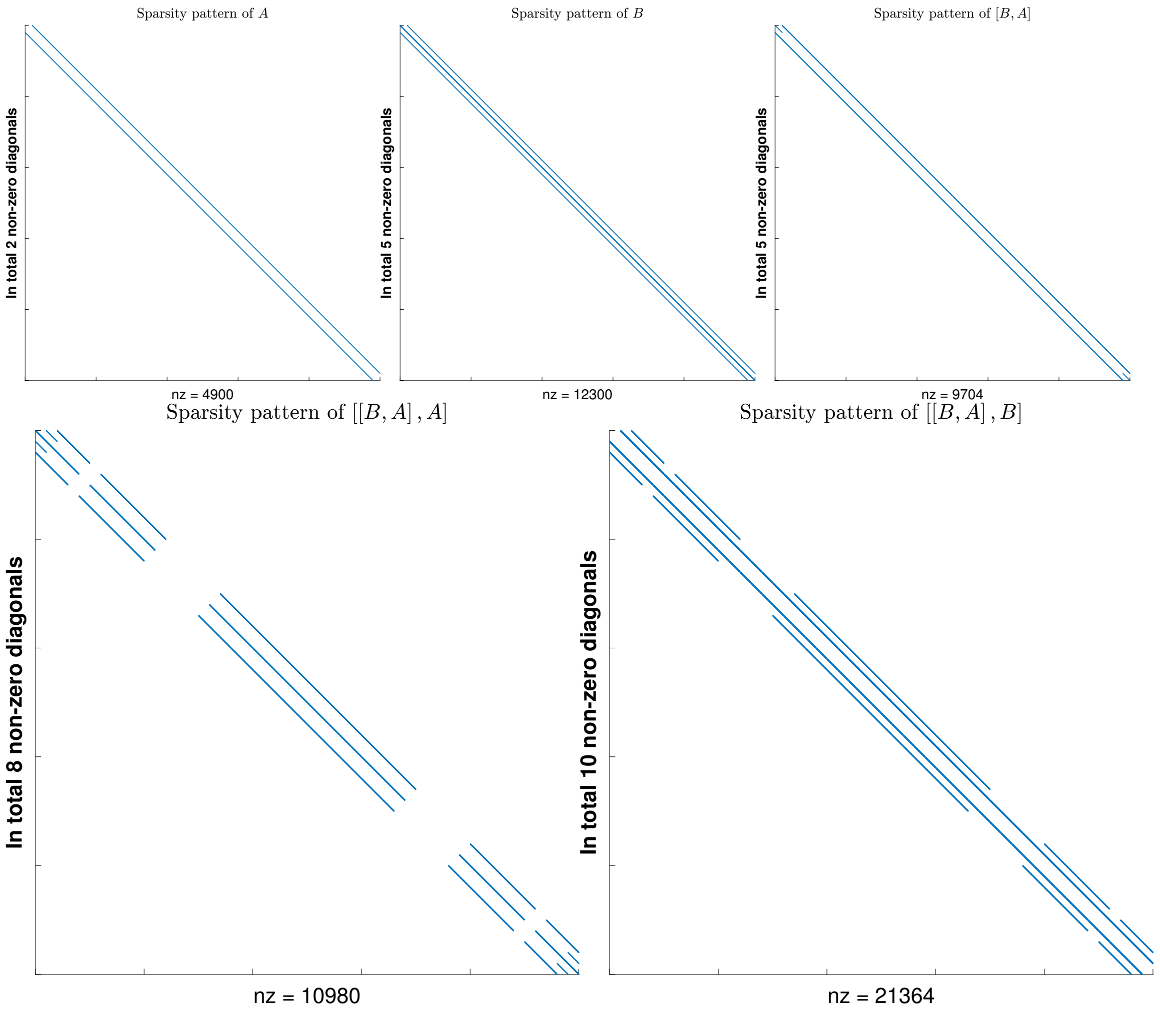


Figure 2.10.: Sparsity patterns of A, B and commutators of the SPDE from Section 2.4.4.1 with coefficients (2.4.28) and $d = 50$. nz stands for the number of non-zero entries.

Table 2.7.: Notations for the numerical experiments.

`euler`	Euler-Maruyama scheme (2.4.27)
`m1`	Iterated Magnus scheme of order 1
`m2`	Iterated Magnus scheme of order 2
`m3`	Iterated Magnus scheme of order 3
M	Number of simulations
d	Number of grid points in $\mathbb{X}$ and $\mathbb{V}$
Δ_t	step-size of `euler` or Magnus
Δ_t^{Int}	discretization of the integrals for Magnus
M2, x	Magnus order 2 with step-size $\Delta_t = x$
M3, x	Magnus order 3 with step-size $\Delta_t = x$
E, x	`euler` with step-size $\Delta_t = x$

obtain

$$\begin{aligned} u_{t_{k+1}}^{n_x,n_v} \approx u_{t_k}^{n_x,n_v} + \Bigg(& H_{t_k} \odot u_{t_k}^{n_x,n_v} + F_{t_k}^x \odot \left(D^x u_{t_k}^{n_x,n_v} \right) + F_{t_k}^v \odot \left(u_{t_k}^{n_x,n_v} \left(D^v \right)^\top \right) \\ & + \frac{1}{2} G^{xx} \odot \left(D^{xx} u_{t_k}^{n_x,n_v} \right) + G_{t_k}^{xv} \odot \left(D^x u_{t_k}^{n_x,n_v} \left(D^v \right)^\top \right) + \frac{1}{2} G_{t_k}^{vv} \odot \left(u_{t_k}^{n_x,n_v} \left(D^{vv} \right)^\top \right) \Bigg) \Delta t \\ + \Bigg(& \Sigma_{t_k} \odot u_{t_k}^{n_x,n_v} + \Sigma_{t_k}^x \odot \left(D^x u_{t_k}^{n_x,n_v} \right) + \Sigma_{t_k}^v \odot \left(u_{t_k}^{n_x,n_v} \left(D^v \right)^\top \right) \Bigg) \Delta W_{t_k}, \end{aligned} \tag{2.4.27}$$

where $\Delta t := t_{k+1} - t_k > 0$ for any k and $\Delta W_{t_k} := W_{t_{k+1}} - W_{t_k}$.

Preliminaries for the numerical tests. In the following numerical experiments, we will use the notations in Table 2.7 in the plots and descriptions.

For the space grids $\mathbb{X}_{a_x,b_x}^{n_x}$ and $\mathbb{V}_{a_v,b_v}^{n_v}$ we will use symmetric grids centered around zero with a cut-off region of $[-4, 4]$ and d grid points each.

Regarding the integration discretization Δ_t^{Int} we have observed that the computational times in general increase with smaller discretization but the final errors change insignificantly for $\Delta_t^{\text{Int}} < 10^{-5}$.

We suggest to use an integration-discretization equal to 10^{-4} or 10^{-5} in the separable coefficient case and to 10^{-3} or 10^{-4} in the constant coefficient case. Henceforth, we will set it to $\Delta_t^{\text{Int}} = 10^{-4}$ for all tests.

We have verified a linear behaviour (with a slope less than one until the GPU is fully saturated) in the number of simulations M. In our experiments, we decided to use $M = 100$ simulations and display always the average computational times for one simulation.

Also the computational effort with respect to the finite time horizon T scales linearly for all methods. Thus, we use $T = 1$ as our terminal time.

We used for the calculations `Matlab 2022a` with the Parallel Computing Toolbox running on Debian GNU/Linux 10 (buster), on a machine with the following specifications: processor 2x AMD EPYC 7301 CPU @ 2.20 GHz, 256 GB RAM and a NVIDIA Tesla V100 PCIe (32 GB HBM2 RAM). We limit ourselves to 12 CPU cores to highlight that the Magnus expansion not only works very well on clusters but also on an average desktop computer.

The next subsection is structured as follows: First, we derive the explicit solution of the Langevin equation. Then, we discuss the impact of the number of intervals Δ_t for the iterated Magnus scheme regarding computational times and errors. Next, we look at the boundary effects over time. This is followed by a comparison of the Magnus scheme with the Euler-Maruyama scheme with different sizes of the space grid.

2.4.4.1. The Magnus Expansion for the Stochastic Langevin Equation with Constant Coefficients

In this subsection, we apply the Magnus expansion to the stochastic Langevin equation. For further details and a solution theory in Hölder spaces under the weak Hörmander condition we refer the reader to Pascucci and Pesce (2022b).

In the constant coefficient case, the Langevin SPDE can be recovered from (2.4.23) by setting

$$h \equiv f^v \equiv g^{xx} \equiv g^{xv} \equiv \sigma \equiv \sigma^x \equiv 0, \quad f_x(x,v) := -v, \quad g^{vv} \equiv a \in \mathbb{R}_{>0}, \quad \sigma^v(x,v) \equiv \sigma \in \mathbb{R}_{>0}. \tag{2.4.28}$$

In this special case, there exists an explicit fundamental solution Γ for $0 < \sigma < \sqrt{a}$ (cf. Pascucci and Pesce (2022b): *p. 4 Proposition 1.1.*), which is given by

$$\begin{aligned}\Gamma\left(t,z;0,\zeta\right) &:= \Gamma_0\left(t, z - m_t(\zeta)\right), \\ \Gamma_0(t,(x,v)) &:= \frac{\sqrt{3}}{\pi t^2(a-\sigma^2)} \exp\left(-\frac{2}{a-\sigma^2}\left(\frac{v^2}{t} - \frac{3vx}{t^2} + \frac{3x^2}{t^3}\right)\right)\end{aligned}$$

where $\zeta := (\xi,\eta)$ is the initial point and

$$m_t(\zeta) := \begin{pmatrix} \xi + t\eta - \sigma\int_0^t W_s ds \\ \eta - \sigma W_t \end{pmatrix}.$$

Having the fundamental solution, we can solve the Cauchy-problem by integrating against the initial datum, i.e.

$$u_t(z) = \int_{\mathbb{R}^2} \Gamma(t,z;0,\zeta)\phi(\zeta)d\zeta, \qquad z = (x,v).$$

To get an explicit solution (up to the stochastic integral $\int_0^t W_s ds$) for the double integral we will choose ϕ to be Gaussian, i.e.

$$\phi\left(\xi,\eta\right) := \exp\left(-\frac{(\xi^2+\eta^2)}{2}\right). \tag{2.4.29}$$

The formula for the exact solution is lengthy and its specific form is not instructive for the following experiments, therefore we decided to exclude it from this presentation. The interested reader can find it in the corresponding `Matlab 2022a` code, which is publicly available.[14]

Having an exact benchmark solution we will now perform some numerical tests to judge the performance of the iterated Magnus scheme. Henceforth, the parameters for the stochastic Langevin equation will be $a = 1.1$ and $\sigma = \frac{1}{\sqrt{10}}$, so that $a - \sigma^2 = 1 > 0$.

Computational effort and errors with respect to the number of iterations. For this experiment, we fix the number of grid points in each space grid to $d = 200$ but vary step-size of the Magnus scheme Δ_t. In Figure 2.11 we can see the corresponding results. The left y-axis shows the average computational times for one simulation in a log scale and the right y-axis the mean relative errors $\mathrm{Err}_T^{d,4}$ also in a log scale. The computational times (in seconds) of `m2` are depicted in light blue and of `m3` in dark blue. Moreover, the mean relative errors for `m2` are orange and for `m3` red.

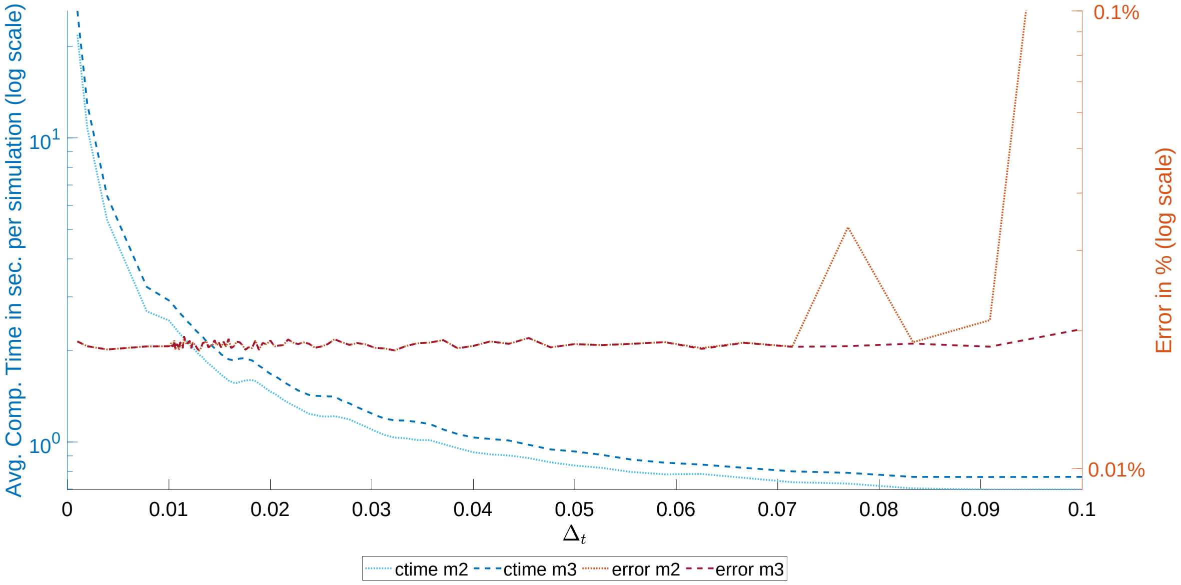


Figure 2.11.: Constant coefficients as in (2.4.28): Computational times and errors of the Magnus expansion for varying step-size Δ_t with fixed spatial dimension $d = 200$.

We can see that the mean relative errors start to fluctuate for a step-size larger than 0.065 for `m2` and see an explosion of `m2` around 0.085. For `m3` this phenomenon starts outside of the picture. The fluctuations begin right after $\Delta_t = 0.1$ and an explosion can be seen after $\Delta_t = 0.33$. The explosions for large step-sizes are not surprising, since the step-size is determined by the underlying stopping times for the convergence of the Magnus scheme. The fluctuations beforehand are most likely due to an interplay between error propagations due to larger step-sizes and the necessary Taylor-terms in `expmv`, and it indicates that `m3` is more stable than `m2`. Therefore, this experiment indicates that any step-size less than 0.05, 0.1 for $d = 200$ is well within the convergence radius of `m2`, `m3`, respectively, and yields stable results.

[14] More details about the implementation can be found on https://github.com/kevinkamm/MagnusSPDE2D/blob/main/IteratedMagnus/exact.m.

For other spatial dimensions d this breaking point might be different. Moreover, we can see that the computational time increases more and more for smaller step-size, while the error for both methods stays almost constant and close to each other.

This suggests that one should choose the step-size as large as possible for the iterated Magnus scheme to gain the maximal performance. However, being too greedy will lead to blow-ups of some trajectories.

Also as a side note, usually with increasing spatial dimension d, one has to choose a smaller time step-size for the Magnus methods as well: this will be shown in Figure 2.14–2.16.

Mean errors and boundary effects over time. For this experiment we fix the grid points in each space grid to $d = 300$.

In Figure 2.13 we can see the mean absolute errors of the entire spatial grid as a two-dimensional plot. A deep blue color indicates a small error and a bright yellow color an error up to 10^{-2}. The black rectangle is the corresponding region for $\kappa = 1$. The black number within the rectangle is the average mean absolute error of the corresponding region. The picture on the left-hand side is the area of errors at $t = 0.25$ and on the right-hand side at $t = 1$.

We can see that the errors in the upper right and lower left corners are significantly increasing over time. To explain this, one should note that the Langevin equation with this specific initial datum looks like a two-dimensional Gaussian (see Figure 2.12) at first and its shape changes on the diagonal from the lower left corner to the upper right corner over time more than on the other diagonal. Therefore, the cut-off region is getting too small for larger times leading to boundary effects in the error plots. This also explains why the upper left and lower right corner remain a stable small error. In the center of the error plots we can see an increasing error over time. If this is due to the boundary effects, error propagation due to the iterated scheme, the error due to the order 3 truncation or the algorithm used for the matrix-vector exponential is not apparent in this illustration, we suspect a mixture of all of them.

Comparison to the Euler-Maruyama scheme For this experiment, we will compare different choices of parameters for both the Magnus scheme and Euler-Maruyama scheme. There are essentially two major parameters contributing to the possible accuracy. One is the time step-size of the individual schemes and the other one the space discretization. Hence, we compare Euler and Magnus methods with different time step-sizes for different space discretizations $d = 100, 200, 300$ to increase the level of accuracy. In the Figures 2.14, 2.15 and 2.16 the left y-axis shows the average computational times in a log scale and the right y-axis the mean relative errors $\mathrm{Err}_T^{d,4}$ also in a linear scale. The computational times (in seconds) are depicted in the left blue columns and the mean relative errors in the red right columns for each method.

As mentioned in Table 2.7, “E, x” denotes Euler with step-size $\Delta_t = x$ and “M2, x”, “M3, x” denotes Magnus with step-size $\Delta_t = x$ for order 2 and 3, respectively. In Figure 2.14 we compare the errors and computational times of the methods with spatial dimension $d = 100$,

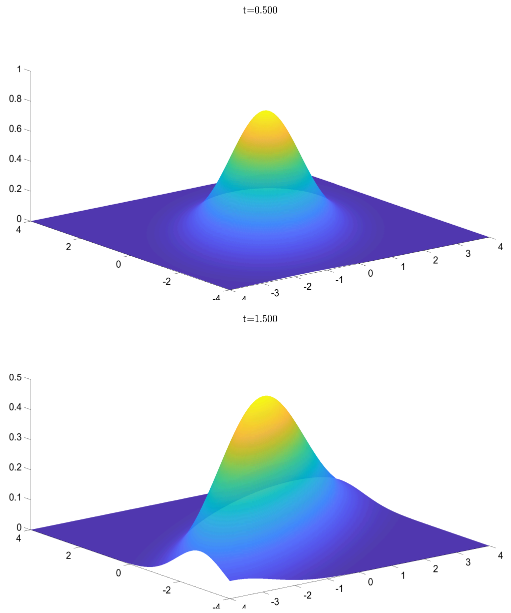


Figure 2.12.: One trajectory of the constant Langevin equation (2.4.28) at $t = 0.5$ (upper left), $t = 1$ (upper right), $t = 1.5$ (bottom left) and $t = 2$ (bottom right).

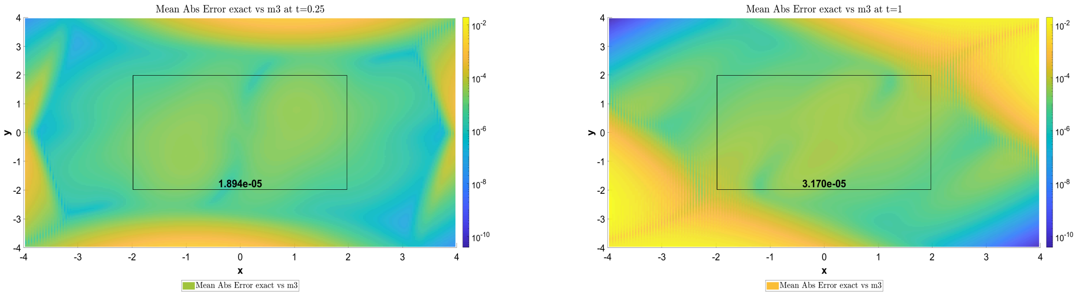


Figure 2.13.: Constant coefficients as in (2.4.28): Absolute Errors of `m3` compared to `exact` using $d = 300$ grid points at $t = 0.25$ (left) and $t = 1$ (right).

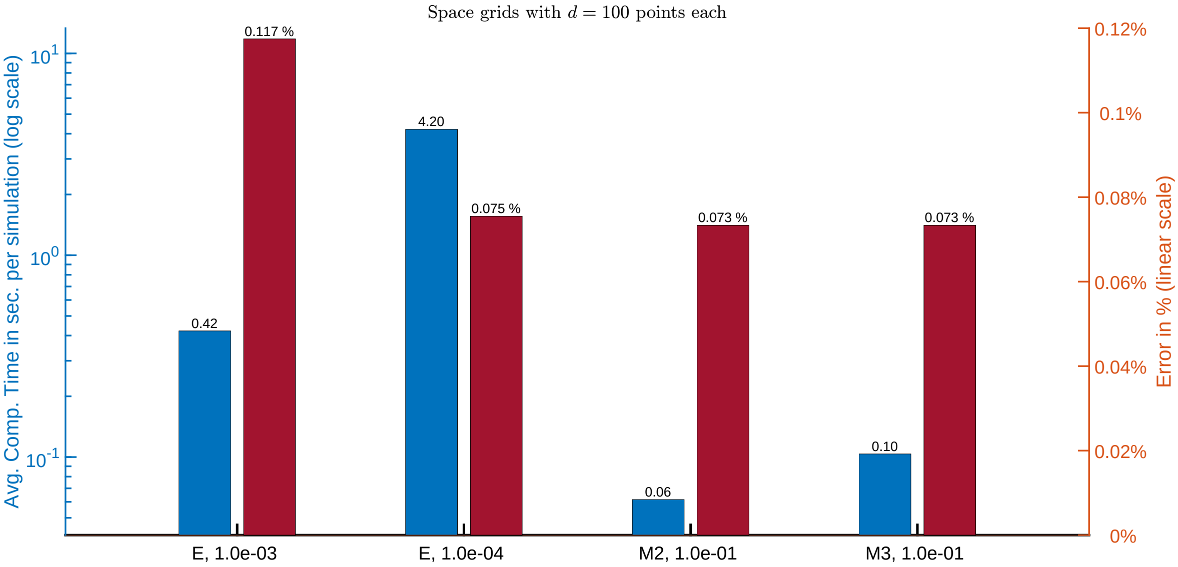


Figure 2.14.: Constant coefficients as in (2.4.28): Computational times and errors of the Magnus expansion and Euler scheme for $d = 100$.

in Figure 2.15 with $d = 200$ and in Figure 2.16 with $d = 300$.

Let us focus on Figure 2.14 with $d = 100$. We can see that four different methods are compared: the Euler method with step-size $\Delta_t = 10^{-3}$ and $\Delta_t = 10^{-4}$, as well as the Magnus method with step-size $\Delta_t = 0.1$ of order 2 and order 3. It is notable that the Euler method with step-size $\Delta_t = 10^{-4}$ and the Magnus methods perform almost the same with respect to the error. The Euler method with step-size $\Delta_t = 10^{-3}$ has roughly double the error of the method with step-size $\Delta_t = 10^{-4}$ but is ten times faster. Overall, the Magnus methods were the fastest methods. The Magnus method of order two, three is 70, 42 times, respectively, faster than Euler method with step-size $\Delta_t = 10^{-4}$ and has a slightly better accuracy.

Now, let us consider Figure 2.15 with $d = 200$. Again, we can see that two Euler methods and two Magnus methods are compared but this time we have a step-size $\Delta_t = 0.05$ for the Magnus methods. Similarly, to Figure 2.14, we can see that the Euler method with step-size $\Delta_t = 10^{-3}$ performed worst and the Magnus methods best in terms of accuracy. However, this time the Euler method with step-size $\Delta_t = 10^{-4}$ has twice the error compared to the Magnus methods and is 25, 23 times slower than the Magnus method with order 2, 3, respectively.

In Figure 2.16 with $d = 300$ the Euler method with step-size $\Delta_t = 10^{-3}$ is exploding, since its stability criterion is violated and its errors is ∞. Therefore, we compare the Euler method with step-size $\Delta_t = 10^{-4}$ to the Magnus method with with step-size $\Delta_t = 0.025$ with order two and three. This time the Euler method is four times worse in terms of accuracy and 30, 27 times slower than Magnus with order two, three, respectively. We also performed tests with an Euler method using step-size $\Delta_t = 10^{-5}$. Its accuracy was still slightly worse compared to the Magnus methods and its computational time ten times slower than the Euler scheme in the figure. This results in a speed-up of order 250 of the Magnus method compared to an Euler scheme with similar accuracy.

Overall, from these observations it is clear that an Euler scheme with a fine time-discretization

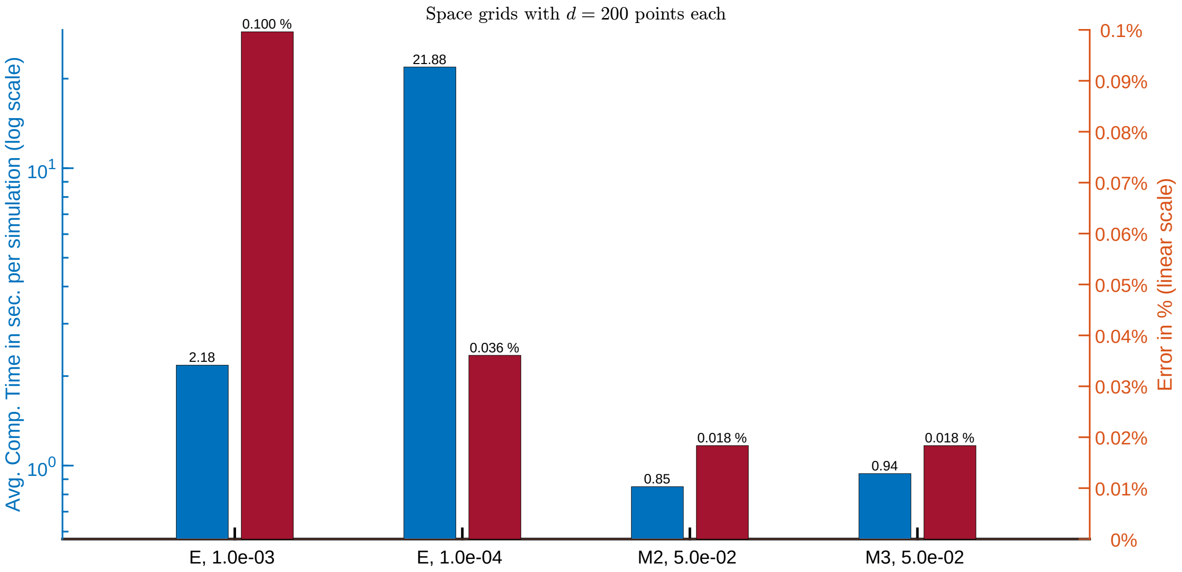


Figure 2.15.: Constant coefficients as in (2.4.28): Computational times and errors of the Magnus expansion and Euler scheme for $d = 200$.

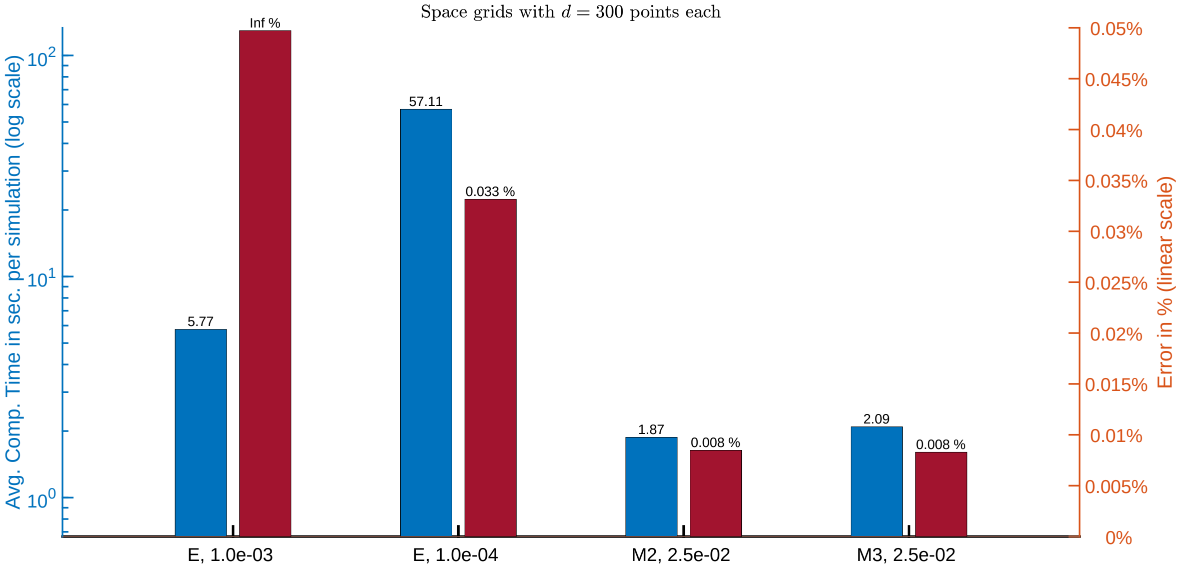


Figure 2.16.: Constant coefficients as in (2.4.28): Computational times and errors of the Magnus expansion and Euler scheme for $d = 300$.

is essential to make it comparable to the iterated Magnus scheme in terms of accuracy. Moreover, increasing the number of grid points is leading to less accurate errors using the Euler method with step-size $\Delta_t = 10^{-4}$ compared to the Magnus schemes with corresponding step-sizes, while the Magnus methods remain roughly 30 times faster in all tests.

Remark 2.4.4. If we have a close look at all tests from above we can see that all of them share a common feature, namely for reasonable parameters `m2` and `m3` were always close. Therefore, this leads to a natural step-size control in time by comparing the results of Magnus order 2 to order 3. If they are closer than a given tolerance then the step-size is small enough, otherwise make it smaller by a given factor.

With this method, the computation of the Magnus logarithms up to order 3 can be reused for Magnus order 2. However, two matrix-vector exponentials for each trajectory are necessary to determine if the time-step is rejected. For implementations with a lot of trajectories, one can think about using less randomly chosen trajectories to determine the correct step-size to increase the overall performance.

Remark 2.4.5. The Magnus expansion holds an advantage over all other finite-difference method in the deterministic case. Inspecting the approximation formulas in the case $A \equiv 0$ reveals immediately that the Magnus expansion is exact, at order 1, up to the initial space discretization for x and v, meaning that its accuracy is far more superior than e.g. explicit and implicit Euler-schemes.

2.4.4.2. The Magnus Expansion for the Stochastic Langevin Equation with Separable Coefficients

In this brief subsection, we will perform some tests in the case of variable coefficients.[15] In particular, we choose bounded, smooth coefficients of the form

$$\begin{aligned}
&h \equiv f^v \equiv g^{xx} \equiv g^{xv} \equiv \sigma \equiv \sigma^x \equiv 0,\\
&f_x(x,v) := -v, \quad g^{vv}(x,v) = a \cdot f(B_t), \quad \sigma^v(x,v) \equiv \sigma \cdot g(B_t),\\
&f(x) := 1 + \frac{1}{1+x^2}, \quad g(x) := \sqrt{1 + \frac{1}{1+x^2}}
\end{aligned} \tag{2.4.30}$$

with B_t a standard Brownian motion, $a = 1.1, \sigma = \frac{1}{\sqrt{10}}$ satisfying $g^{vv}(x,v) - (\sigma^v(x,v))^2 > 0$, as in the constant coefficient case. We will also use the same initial condition as in (2.4.29).

In this case, the coefficient processes A_t and B_t look like

$$\begin{aligned}
&B_t := B_1 + B_2 b_t,\\
&B_t^1 := \operatorname{diag}\left(\operatorname{vec}\left(F^x\right)\right)\left(I_{n_v} \otimes D^x\right), \quad B_t^2 := \frac{1}{2}\operatorname{diag}\left(\operatorname{vec}\left(G^{vv}\right)\right)\left(D^{vv} \otimes I_{n_x}\right),\\
&A_t := A a_t, \quad A := \operatorname{diag}\left(\operatorname{vec}\left(\Sigma^v\right)\right)\left(D^v \otimes I_{n_x}\right)
\end{aligned}$$

where $a_t := f\left(B_t\right)$ and $b_t := g\left(B_t\right)$.

[15] More details about the implementation can be found on https://github.com/kevinkamm/IteratedMagnus/blob/main/SPDE2d/Separable/Code/main.m.

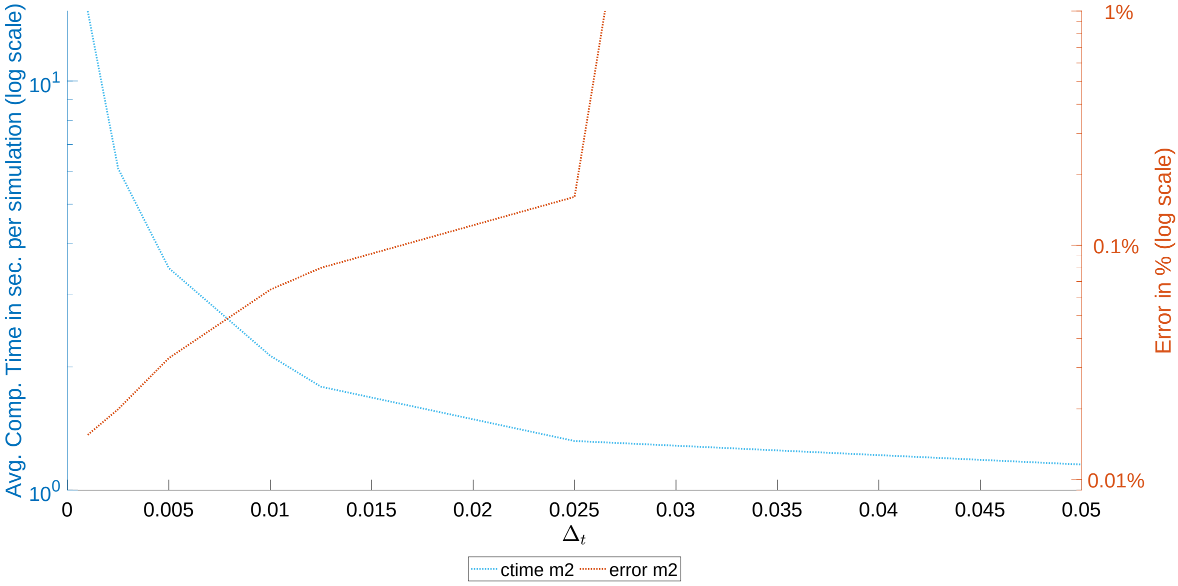


Figure 2.17.: Separable coefficients as in (2.4.30): Computational times and errors compared to Euler with $\Delta_t = 10^{-5}$ of the Magnus expansion for varying step-size Δ_t with fixed spatial dimension $d = 200$.

For this experiment, we decided to only show the first two orders of the Magnus expansion, which read as follows: For the order one expansion we have

$$Y_t^{(1)} = A \int_0^t a_s dW_s + B_1 t + B_2 \int_0^t b_s ds.$$

The second order expansion is given by

$$\begin{aligned} Y_t^{(2)} = -\frac{1}{2} A^2 \int_0^t a_s^2 ds &+ \frac{1}{2} [B_1, A] \left(\int_0^t \int_0^s a_r dW_r ds - \int_0^t s a_s dW_s \right) \\ &+ \frac{1}{2} [B_2, A] \left(\int_0^t b_s \int_0^s a_r dW_r ds - \int_0^t a_s \int_0^s b_r dr dW_s \right) \\ &+ \frac{1}{2} [B_1, B_2] \left(\int_0^t \int_0^s b_r dr ds - \int_0^t s b_s ds \right). \end{aligned}$$

Analog to Figure 2.11, we show in Figure 2.17 the case of varying step-sizes for the Magnus method with fixed spatial discretization $d = 200$. The average computational times of `m2` and `m3` in seconds, per simulation, are again depicted in light blue and dark blue, respectively. The errors are this time with respect to the Euler method with $\Delta_t = 10^{-5}$, since an exact solution is not available, and again illustrated as orange for `m2` and red for `m3`.

The computational times look similar to the constant coefficient case and we can notice a continuous increase for smaller step sizes.

The errors look different than in the constant coefficient case, where it was almost constant for small step sizes. Here, we see that the errors decrease with smaller step sizes, which might be due to the increased complexity of the formulas.

In our next experiment, similarly to Figure 2.14–2.16, we show in Table 2.8 the mean

Table 2.8.: Separable coefficients as in (2.4.30): Computational times and errors compared to Euler with $\Delta_t = 10^{-5}$ of the Magnus expansion for different spatial dimension $d = 100, 200, 300$.

Method	Mean Rel. Error (in %)	Comp. Time (in sec./simulation)
	$d = 100$	
`euler`, $\Delta_t = 10^{-3}$	0.158 %	0.47
`euler`, $\Delta_t = 10^{-4}$	0.041 %	4.62
`euler`, $\Delta_t = 10^{-5}$	–	49.6
`m2`, $\Delta_t = 0.1$	0.663 %	0.258
`m2`, $\Delta_t = 0.05$	0.320 %	0.179
	$d = 200$	
`euler`, $\Delta_t = 10^{-3}$	∞ %	2.46
`euler`, $\Delta_t = 10^{-4}$	0.042 %	25.11
`euler`, $\Delta_t = 10^{-5}$	–	25.17
`m2`, $\Delta_t = 0.025$	0.162 %	1.32
`m2`, $\Delta_t = 0.01$	0.066 %	2.13
	$d = 300$	
`euler`, $\Delta_t = 10^{-3}$	∞ %	7.13
`euler`, $\Delta_t = 10^{-4}$	0.041 %	69.8
`euler`, $\Delta_t = 10^{-5}$	–	725.74
`m2`, $\Delta_t = 0.01$	0.064 %	3.272
`m2`, $\Delta_t = 0.005$	0.032 %	4.906

relative errors and computational times for $d = 100, 200, 300$. This time we use an Euler method with $\Delta_t = 10^{-5}$ as our reference solution and compare an Euler method with $\Delta_t = 10^{-4}$, as well as `m2` and `m3` to it. The results are given in Table 2.8. As we already noticed in Figure 2.17, we have contrary to Figure 2.14–2.16 the possibility to reduce the error of the Magnus scheme by reducing the step size Δ_t. Therefore, we show two different step sizes for `m2` for each grid size d. We can see for $d = 100$ that a smaller step size does not necessarily imply a larger computational time, if the step sizes are rather large. This can be explained by the fact that the computation of the matrix-vector exponential takes longer, since more Taylor terms are required to achieve the target accuracy. Another important difference is the error of `m2`. It is larger than the error of the `euler` schemes and a lot worse than in our previous tests. For larger dimensions, we can see a similar behaviour of `m2` to our previous tests and come to the same conclusions as in the constant coefficient case. The Magnus expansion is faster than the Euler scheme, and performs well in terms of accuracy.

In Kamm, Pagliarani and Pascucci (2022) we performed another test in the case of deterministic autonomous coefficients, but with space dependency, giving similar results as in the constant case as well.

2.5. Conclusion

In this chapter, we have seen how the Itô-stochastic Magnus expansion can be derived and have shown that it is convergent up to a strictly positive stopping time. In our numerical experiments, we demonstrated that order two or three is usually sufficient to yield accurate results. Additionally, we have shown, how the restrictions coming from the stopping time can be overcome in practice by applying the Magnus expansion iteratively. Furthermore, we found that the Magnus expansion can be applied for a wide class of problems very efficiently by considering different generalities of the coefficient processes A_t and B_t. Especially in the case of one and two dimensional SPDEs, we saw significant improvements in terms of both accuracy and computational effort compared to the Euler-Maruyama scheme using a single GPU.

3 The Cox-Ingersoll-Ross Model in a Negative Interest Rate Framework

The Cox-Ingersoll-Ross model (hereafter referred to as CIR model) has been regarded as the reference model in interest rate modelling by both practitioners and academics for several decades, not only because of its analytical tractability as an affine model, but also because of its derivation from a general equilibrium framework (see for example Cox, Ingersoll and Ross (1981)), among other reasons. The well-known feature of the CIR model that ultimately led to this chapter is that interest rates never become negative. This long-standing paradigm of non-negative interest rates made the CIR model and its extensions one of the most appropriate models for interest rate modelling.

Today, however, negative interest rates are very common and thus the need for models that can handle this paradigm shift is highly desirable, provided that they have as few shortcomings as possible compared to the original CIR models.

We can categorize short-rate models into two different classes: *endogenous* and *exogenous*. In endogenous models the observed term-structure is an output depending on the model parameters and in exogenous models the observed term-structure is an input.

In this chapter, we propose a simple endogenous and exogenous model to address the problem of negative interest rates. The principle idea is to subtract two independent CIR processes to account for negative interest rates, which leads to an endogenous model and is described in Section 3.3. Applying the deterministic-shift extension technique (cf. Brigo and Mercurio (2006)) to this model and its calibration to swaption prices will be subject of Section 3.4.

To fix notations let us recall basic interest rate related market instruments. For this we will follow Brigo and Mercurio (2006): *pp. 2ff.* closely.

3.1. Market Instruments

In this section, we will briefly recall some interest rate instruments, which we will need for our further discussions. So let us start by introducing the *bank account* as in Brigo and Mercurio (2006): *pp. 2ff.* which can be regarded as the natural choice of numeraire (cf. Björk (2004): *pp. 154ff.*).

Definition 3.1.1. We define $\mathrm{M}(t)$ to be the *value of a bank account* at time $t \geq 0$. We assume $\mathrm{M}(0) = 1$ and

$$\mathrm{M}(t) = \exp\left(\int_0^t r_s ds\right).$$

where r_t is a possible negative stochastic process called *instantaneous (spot) rate* or *short* *rate*. Moreover, the *(stochastic) discount factor* $D(t,T)$ is given by

$$D(t,T) = \frac{\mathrm{M}(t)}{\mathrm{M}(T)} = \exp\left(-\int_t^T r_t ds\right).$$

In the next subections, we will first introduce zero-coupon bonds in Section 3.1.1, followed by interest rate swaps in Section 3.1.2 and a variant called constant maturity swaps in Section 3.1.3. Afterwards, we will elaborate more on swaptions in Section 3.1.4 and Bermudan swaptions in Section 3.1.5.

3.1.1. Zero-Coupon Bonds

The most essential instrument for us will be the so-called *zero-coupon bonds* (cf. Brigo and Mercurio (2006): *p. 4 Definition 1.2.1. Zero-coupon bond.*).

Definition 3.1.2. A T-maturity *zero-coupon bond* is a contract that guarantees its holder the payment of one unit of currency at time T, with no intermediate payments. The contract value at time $t < T$ is denoted by $P(t,T)$ and we have $P(T,T) = 1$ for all T.

Let us make the following quick observation: Both the discount factor, as well as a zero-coupon bond will yield one unit of currency at the time T, i.e. $D(T,T) = P(T,T)$. However, this is in general not true for $t < T$. The zero-coupon bond is the value or price of a contract, while the discount factor is the time t amount of one unit of currency at time T. Therefore, if we view this from today, i.e. $t = 0$, the zero-coupon price will be a real number, but the discount factor will still be stochastic. Their relation is given by (cf. Björk (2004): *p. 370 Proposition 23.3 (Risk-neutral valuation)*)

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

under a risk-neutral measure $\mathbb{Q}$. Therefore, they coincide if the short-rate is deterministic.

Definition 3.1.3. The *instantaneous forward interest rate* at time t for the maturity $T > t$ is denoted by $f(t,T)$ and is defined as

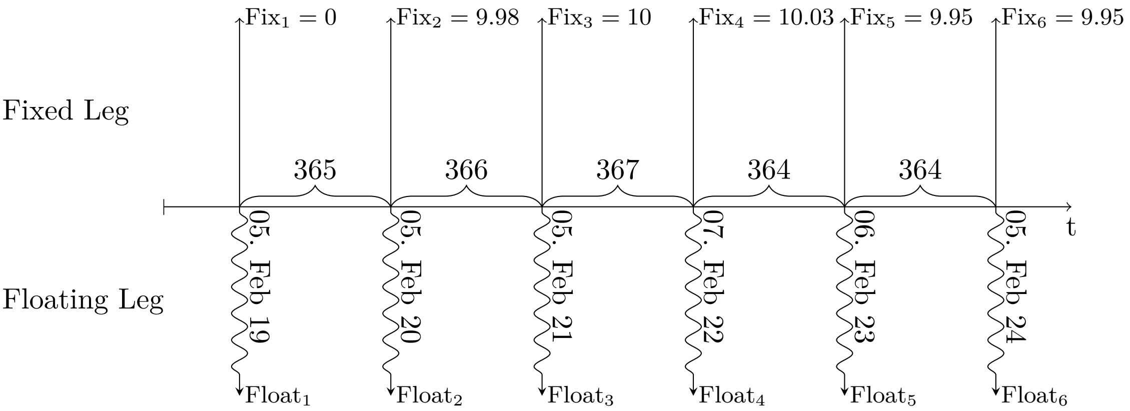
$$f\left(t,T\right) := -\left(\partial_T \ln P\right)\left(t,T\right),$$

so that we also have

$$P\left(t,T\right) = \exp\left(-\int_t^T f\left(t,u\right) du\right).$$

3.1.2. Interest Rate Swaps

We are following Miron and Swannell (1991): *pp. 9ff.* Björk (2004): *pp. 428ff. Chapter 27.7 Swaps*, Brigo and Mercurio (2006): *pp. 19ff.* and Schrager and Pelsser (2006): *pp. 3ff.* in this section. Let us start by considering an example of an interest rate swap and discover all necessary features of this financial instrument meanwhile.



1. Trade Date: 01. Feb 19
2. Termination Date: 05. Feb 24
3. Effective Date: 05. Feb 19
4. Fixed Rate: 9.84 %
5. Fixed Rate day-count fraction: Actual/360
6. Principal: 100 $
7. Payment Days: anually or modified business day including last excluding first

Figure 3.1.: Example of an interest rate swap.

Example 3.1.4. This example (cf. Miron and Swannell (1991): *p. 28 Example 1*) with all relevant parameters will be illustrated in Figure 3.1. Lets call our two participants in this swap contract Fix (Fixed Rate Payer) and Float (Floating Rate Payer) and take the view of Fix.

Moreover, assume that Fix borrowed money for 5 years from another counterparty and pays interest every year. Now, this interest rate can change as time progresses and therefore introduces a certain amount of risk.

In order to minimize the risk arising from this contract Fix wishes to rather pay a fixed amount of interest on his borrowings every year. Thus, he wants to *swap* his floating, risky position to a fixed position.

In order to do this he agrees with Float upon a 5-year interest rate swap at 01. Feb 19, which will be effective on the 05. Feb 19. The fixed and floating payments will be made each year on 05. Feb except if its a weekend or holiday, then on the next following business day, which is responsible for e.g. the payment date 07. Feb 22.

There are several different conventions regarding the day-count (cf. Brigo and Mercurio (2006): *pp. 5–6*), in this example we have the actual day difference (including leap years) divided by 360 (which comes from the convention that one month has 30 days).

The day-count convention comes into play when we calculate the fixed payments via

$$\text{Principle} \cdot \text{Fixed Rate} \cdot \text{Day-count convention}(\text{date}_1, \text{date}_2).$$

The first payments will be made at 05. Feb 20, therefore

$$\mathrm{Fix}_2 = 100 \cdot \frac{9.84}{100} \cdot \frac{365}{360} \approx 9.98.$$

With this formula one can calculate the other fixed rates as well and are displayed in Figure 3.1 as the values of Fix_i, $i = 1, \ldots, 6$.

Therefore, the risk of changes in the LIBOR will now be shifted to Float.

We have seen that the only parameter—which was previously given—to decide whether this contract is fair, is the fixed rate.

Now, let us put our observation in Example 3.1.4 into more technical terms.

A *swap* is a financial contract between two counterparties with fixed resettlement dates $T_0, T_1, \ldots, T_N$, $N \in \mathbb{N}$. The contract itself contains two cashflows, one—called the floating leg—are payments of future interest rates and the other—called the fixed leg—is a fixed amount of payments. The receiver of a swap will receive at the fixed dates the amount of the fixed leg and pays the amounts of the floating leg to the other counterparty, giving it its name: the floating leg is swapped for the fixed leg. Additionally, a payer swap refers to the case, when the floating leg is received and the fixed rate payed. We will distinguish the different kinds by introducing the factor ζ, which will be equal to $+1$ in case of a payer swap and -1 in case of a receiver swap.

Such a contract with maturity T_0 and tenor $T_N - T_0$ and resettlements $T_0, \ldots, T_N$ is commonly called a $T_0 \times (T_N - T_0)$ *swap.*

The net value of a $T_0 \times (T_N - T_0)$ payer and receiver swap at time $t < T_0$ is given by

$$\mathrm{Swap}_{T_0}^{T_N}(t; K, \zeta) := \zeta \left(P(t, T_0) - P(t, T_N) - K \sum_{i=1}^{N} \alpha_i P(t, T_i) \right) \tag{3.1.1}$$

where $\alpha_i = T_i - T_{i-1}$ is the day-count convention and K the fixed rate, see for instance Björk (2004): *pp. 429ff.* To ease notation, we will suppress the explicit dependency on the T_0 and T_N whenever there is no confusion.

A particular fixed rate K, called *par or forward swap rate*, is of special interest and is usually quoted in the market. It is the one, such that $\mathrm{Swap}_{T_0}^{T_N}(t; K, \zeta) = 0$, i.e. the contract is fair and is independent of ζ. We will denote it a bit more generally by

$$\mathrm{R}_n^N(t) := \frac{P(t, T_n) - P(t, T_N)}{\sum_{i=n+1}^{N} \alpha_i P(t, T_i)}, \quad n = 0, \ldots, N-1.$$

Moreover, we will denote the so-called *accrual factor* or *present value of a basis point* by

$$\mathrm{S}_n^N(t) := \sum_{i=n+1}^{N} \alpha_i P(t, T_i), \quad n = 0, \ldots, N-1.$$

3.1.3. Constant Maturity Swaps

In this section, we recall the definition of a constant maturity swap. For more details, we refer the reader to Brigo and Mercurio (2006): *pp. 557ff. Section 13.7 Constant-Maturity-*

Swaps and Tanaka, Yamada and Watanabe (2010): *pp. 7ff.*

Definition 3.1.5. A *constant maturity swap (CMS)* is a variant of an interest rate swap between two parties, such that at each payment date starting at T_0 and ending at T_N a fixed rate K is swapped with a c-year swap rate.[1]

Analogously, we distinguish between payer and receiver CMS. In receiver CMS the fixed rate is received and the floating rate paid, vice versa for payer CMS.

Also, we will assume annual settlements between the effective date T_0 and maturity T_N and denote the payment dates by $\mathcal{T} := \{T_0, T_1, \ldots, T_N\}$. The net value of a $T_0 \times T_N + c$ *CMS* with fixed rate K and index c at time 0 under the risk-neutral measure is

$$\mathrm{CMS}_{T_0}^{T_N}(0; K, c, \zeta) := \mathbb{E}^{\mathbb{Q}}\left[\sum_{i=1}^{N} \exp\left(-\int_0^{T_{i-1}} r(s)ds\right) \zeta\alpha_i \left(R_{i-1}^{i-1+c}(T_{i-1}) - K\right)\right]. \tag{3.1.2}$$

By rearranging (3.1.2), we can compute the par CMS rates by setting it to zero and solve for K, i.e.

$$\begin{aligned} K &= \frac{\mathbb{E}^{\mathbb{Q}}\left[\sum_{i=1}^{N} \alpha_i \exp\left(-\int_0^{T_{i-1}} r(s)ds\right) R_{i-1}^{i-1+c}(T_{i-1})\right]}{\mathbb{E}^{\mathbb{Q}}\left[\sum_{i=1}^{N} \alpha_i \exp\left(-\int_0^{T_{i-1}} r(s)ds\right)\right]} \\ &= \frac{\mathbb{E}^{\mathbb{Q}}\left[\sum_{i=1}^{N} \alpha_i \exp\left(-\int_0^{T_{i-1}} r(s)ds\right) R_{i-1}^{i-1+c}(T_{i-1})\right]}{\sum_{i=1}^{N} \alpha_i P(0, T_{i-1})}. \end{aligned}$$

3.1.4. Swaption

Now, we are able to discuss swaptions (cf. Björk (2004): *pp. 430ff.*).

Definition 3.1.6. A $T_0 \times (T_N - T_0)$ payer, receiver swaption with swaption strike K is a contract, which at maturity T_0 gives the holder the right to enter into a $T_0 \times (T_N - T_0)$ payer, receiver swap with fixed rate K.[2]

Its arbitrage free price at time $t < T_0$ is given by

$$\mathrm{Swaption}_{T_0}^{T_N}(t; K, \zeta) = \mathbb{E}^{\mathbb{Q}}\left[\exp\left(-\int_t^{T_0} r(s)ds\right)\left(\zeta\left(\mathrm{R}_0^N(T_0) - K\right)\right)^+ \mathrm{S}_0^N(T_0) \middle| \mathcal{F}_t\right] \tag{3.1.3}$$

While this formulation is often beneficial for Monte-Carlo simulations, it can be useful to remove the stochastic discount factor in (3.1.3) by a clever change of measure. For fixed T_0, the T_0-forward measure $\mathbb{Q}^{T_0}$ is defined as the martingale measure for the numeraire process $P(t, T_0)$ (cf. Björk (2004): *pp. 403ff. Chapter 26.4 Forward measure*) and we have the following:

[1] More details about the implementation can be found on `https://github.com/kevinkamm/CIR--/blob/main/constantMaturitySwap.m`.

[2] More details about the implementation can be found on `https://github.com/kevinkamm/CIR--/blob/main/swaption_matrix.m`.

Proposition 3.1.7. *For any T_0-claim X we have*

$$\mathbb{E}^{\mathbb{Q}}\left[e^{-\int_t^{T_0} r(s)ds} X \middle| \mathcal{F}_t\right] = P(t,T_0)\mathbb{E}^{\mathbb{Q}^{T_0}}\left[X \middle| \mathcal{F}_t\right].$$

Thus, the price at time $t < T_0$ of a payer ($\zeta = 1$), receiver ($\zeta = -1$) swaption under the T_0-forward measure is given by

$$\text{Swaption}\,(t;K,\zeta) = P(t,T_0)\mathbb{E}^{\mathbb{Q}^{T_0}}\left[\left(\zeta\left(\mathrm{R}_0^N(T_0) - K\right)\right)^+ \mathrm{S}_0^N(T_0)\middle| \mathcal{F}_t\right] \tag{3.1.4}$$

3.1.5. Bermudan Swaption

In this section, we want to discuss how to compute the prices of Bermudan swaptions by using the Least-Square-Monte-Carlo (LSMC) technique. A popular choice of literature on this subject is e.g Brigo and Mercurio (2006): *pp. 588 ff. Section 13.15 LFM: Pricing Bermudan Swaptions*, Glasserman (2004): *pp. 421 ff. Chapter 8 Pricing American Options* or more recently Gatarek and Jabłecki (2021) and Oosterlee and Grzelak (2019): *pp. 422 ff. Section 13.3.2 European and Bermudan option example.*

Now, let us define which type of Bermudan swaptions we are interested in.

Definition 3.1.8. A T_N *no-call* T_0 or $T_N\,\mathrm{nc}\,T_0$ *Bermudan swaption* with annual exercise dates gives its holder the right but not the obligation to enter at any time $\mathcal{T}_E^N := \{T_0, T_1, \ldots, T_{N-1}\}$ into an interest rate swap with first reset $T \in \mathcal{T}_E$, last payment T_N and fixed rate K.[3]

Let us give a quick example of a 10 nc 2 Bermudan swaption with annually spaced exercise dates. The holder can exercise this option starting from year 2 and afterwards at the beginning of each consecutive year but not later than year 9. After exercising the option, the holder enters into a swap contract—for simplicity with annual settlements—ending at year 10.

Accordingly, the price at time t of a $T_N\,\mathrm{nc}\,T_0$ Bermudan swaption is the solution to the following optimal stopping problem

$$\text{BSwaption}_{T_0}^{T_N}(t;K,\zeta) := \sup_{\substack{\tau \in \mathcal{T}_E^N \\ \tau \text{ stopping time}}} \mathbb{E}_t^{\mathbb{Q}}\left[e^{-\int_t^{\tau} r(s)ds} S_\tau^N(\tau)\left(\zeta\left(K - R_\tau^N(\tau)\right)\right)^+\right],$$

where the filtration is generated by the forward swap rate, i.e. $\mathcal{F}_t := \sigma\left(R_s^N(s) : s \leq t\right)$ augmented such that it satisfies the usual hypothesis.

For the implementation we are interested in the special case of today's price, i.e. $t = 0$. We will use backward induction to compare the exercise value to the continuation value and compute the conditional expectations by the least square Monte Carlo (LSMC) method (cf. Longstaff and Schwartz (2001)). Let us be more precise:

[3] More details about the implementation can be found on https://github.com/kevinkamm/CIR--/blob/main/bermudanSwaption.m.

We know that the price at time T_{N-1} is given by

$$\begin{aligned}\text{BSwaption}_{T_0}^{T_N}\left(T_{N-1};K,\zeta\right)&=\mathbb{E}_{T_{N-1}}^{\mathbb{Q}}\left[e^{-\int_{T_{N-1}}^{T_{N-1}}r(s)ds}S_{N-1}^N(T_{N-1})\left(\zeta\left(K-R_{N-1}^N\left(T_{N-1}\right)\right)\right)^+\right]\\&=P\left(T_{N-1},T_N\right)\left(\zeta\left(K-R_{N-1}^N\left(T_{N-1}\right)\right)\right)^+\end{aligned}$$

by definition and measurablility as well as the fact that the stopping time can only be equal to T_{N-1} in this case. This gives us the opportunity to inductively calculate the Bermudan swaption price backwards. Thus, let us now assume that $\text{BSwaption}_{T_0}^{T_N}\left(T_{i+1};K,\zeta\right)$ for $i=N-2,\ldots,0$ is known.

We would like to compare the so-called *continuation value*, which is the expected future payoff if the option is not exercised to the exercise value at all times $\mathcal{T}_E^N$, and is defined as

$$c(T_i):=\mathbb{E}_{T_i}^{\mathbb{Q}}\left[e^{-\int_{T_i}^{T_{i+1}}r(s)ds}\text{BSwaption}_{T_0}^{T_N}\left(T_{i+1};K,\zeta\right)\right].$$

Since the optimal stopping time will pathwise choose the maximum of continuing the option or exercising it, we have a dynamic programming principle

$$\text{BSwaption}_{T_0}^{T_N}\left(T_i;K,\zeta\right)=\begin{cases}P\left(T_{N-1},T_N\right)\left(\zeta\left(K-R_{N-1}^N\left(T_{N-1}\right)\right)\right)^+, & i=N-1\\ \max\left(c(T_i),S_i^N(T_i)\left(\zeta\left(K-R_i^N\left(T_i\right)\right)\right)^+\right), & i=0,\ldots,N-2.\end{cases}$$

The price at time $t=0$ is then given by

$$\text{BSwaption}_{T_0}^{T_N}\left(0;K,\zeta\right)=\mathbb{E}^{\mathbb{Q}}\left[e^{-\int_0^{T_0}r(s)ds}\text{BSwaption}_{T_0}^{T_N}\left(T_0;K,\zeta\right)\right].$$

For completeness we explain how to approximate the conditional expectation with the LSMC method in Appendix B.5. For the numerical implementation we choose the polynomial basis.

Now, let us give a short literature review concerning short-rate modelling.

3.2. Literature Review

Historically, the theory of interest-rate modelling started on the assumption of specific one-dimensional dynamics for the instantaneous spot rate process r. These models are convenient for defining all fundamental quantities (rates and bonds) by no-arbitrage arguments as the expectation of a functional of the process r. Indeed, the price at time $t>0$ of a contingent claim with payoff H_T, $T>t$, under the risk-neutral measure Q is given by (cf. Pascucci (2011): *p. 356 Definition 10.46*)

$$H_t=\mathbb{E}^{\mathbb{Q}}\left[e^{-\int_t^T r(s)ds}H_T\middle|\mathcal{F}_t\right], \tag{3.2.1}$$

where $\mathbb{E}^{\mathbb{Q}}[\cdot|\mathcal{F}_t]$ denotes the conditional expectation with respect to some filtration $\mathcal{F}_t$ under measure $\mathbb{Q}$. In particular, choosing $H_T := P(T,T) = 1$, where $P(t,T)$ denotes a zero-coupon bond.

The literature on interest rate modelling is very vast and our short literature review is by no means exhaustive. We refer to Björk (2004), Brigo and Mercurio (2006) and Hull (2006) for a comprehensive review and description of these models.

Among all possible classifications, we can divide these models into two major categories: the endogenous and exogenous models. In chronological order, the first short-rate models belong to the first group: the Vasicek model Vasicek (1977), the Dothan model and the Cox, Ingersoll & Ross (CIR) Cox, Ingersoll and Ross (1985). In particular, the CIR model has been regarded as the reference model in interest rate modelling by both practitioners and academics for several decades for several reasons. First of all, it was derived from a general equilibrium framework. Secondly, it generates more realistic interest rate distributions with skewness and fatter tail with respect to normal distribution. Thirdly, it avoids negative interest rates. There is a rich literature on extensions to the classical CIR model in order to obtain more sophisticated models, which could fit the market data better, allowing to price interest rate derivatives more accurately.

There is a rich literature on extensions to the classical CIR model in order to obtain more sophisticated models, which could fit the market data better, allowing to price interest rate derivatives more accurately. For example, Baker et al. (1996) proposed a three-factor model; Brigo and Mercurio (2006) proposed a jump diffusion model (JCIR). In order to include time dependent coefficients. Brigo and Mercurio (2001a) proposed to add a deterministic function. This model, called CIR++, is able to fit the observed term structure of interest rates exactly, while preserving the positivity of the short-rate process. Brigo and El-Bachir (2006) generalized the CIR++ model by adding a jump term described by a time-homogeneous Poisson process and Brigo and Mercurio (2006) studied the CIR2++ model by adding another risk-factor. Another way to generalize the CIR model by including time dependent coefficients in equation (3.4.1) was introduced by Jamshidian (1990) and Maghsoodi (1996), which are known as extended CIR models.

But in the last decade the financial industry encountered a paradigm shift by allowing the possibility of negative interest rates, making the classical CIR model unsuitable.

Recently, Orlando et al. suggest in several papers (cf. Orlando, Mininni and Bufalo (2019a), Orlando, Mininni and Bufalo (2019b) and Orlando, Mininni and Bufalo (2020)) a new framework, which they call CIR# model, that fits the market term-structure of interest rates. Additionally, it preserves the market volatility, as well as the analytical tractability of the original CIR model. Their new methodology consists in partitioning the entire available market data sample, which usually consists of a mixture of probability distributions of the same type. They use a technique to detect suitable sub-samples with normal or gamma distributions. In a next step, they calibrate the CIR parameters to shifted market interest rates, such that the interest rates are positive, and use a Monte Carlo scheme to simulate the expected value of interest rates.

Beside historical reasons, endogenous models are important for their simplicity and analytical tractability, in particular for the possibility of pricing bonds and bond options analytically. But there are some drawbacks. Since these models use only a few constant parameters, they are not able to reproduce simultaneously a given term-structure and volatility curve satisfactorily. Moreover, some shapes of the zero-coupon curve can never be reproduced (for example an inverted shape curve with the Vasicek model). The need for an exact fit to the currently observed yield curve led some authors to introduce exogenous term-structure models. The first model was proposed by Ho & Lee (see Ho and Lee (1986)), but we believe the most popular among practitioners is the Hull & White extended Vasicek model (see Hull and White (1990)). A generalization of this model with a good calibration to swaption market prices was found in Di Francesco (2012), while Mercurio and Pallavicini in Mercurio and Pallavicini (2005) proposed a mixing Gaussian model coupled with parameter uncertainty. Moreover, in Russo and Torri (2019) the authors calibrate a one- and two-factor Hull-White model using swaptions under a market-consistent framework compatible with negative interest rates.

On the one hand, these models can handle negative interest rates with a very good analytical tractability. On the other hand, the distribution of continuously compounded interest rates show all the undesirable features of the Gaussian distribution.

3.3. Endogenous Model

In this section, we tackle negative interest rates by using the difference of two independent CIR processes.

We will propose a term structure in the risk-neutral world suitable for the difference of two independent affine processes and obtain a pricing formula for default-free zero-coupon bonds by deriving the associated Riccati equations arising from this no-arbitrage framework. In the special case of two CIR processes we will then solve the Riccati equations explicitly, which preserves the analytical tractability of its non-negative interest rate counterpart.

Afterwards, we will show some numerical experiments to demonstrate the merits of this approach in practice.

Let us consider the following affine dynamics

$$\begin{cases} dx(t) = \left(\lambda_x(t)x(t) + \eta_x(t)\right)dt + \sqrt{\gamma_x(t)x(t) + \delta_x(t)}dW_x(t) \\ x(0) = x_0, \end{cases} \tag{3.3.1}$$

$$\begin{cases} dy(t) = \left(\lambda_y(t)y(t) + \eta_y(t)\right)dt + \sqrt{\gamma_y(t)y(t) + \delta_y(t)}dW_y(t) \\ y(0) = y_0, \end{cases} \tag{3.3.2}$$

where henceforth throughout the whole section W_y and W_x are two independent standard Brownian motions on a stochastic basis $\left(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\in[0,T]}, \mathbb{Q}\right)$, $\mathbb{Q}$ is a martingale measure for the zero-coupon market (see for instance the *martingale approach* for short-rate modelling described in Björk (2004): *p. 374 Chapter 23 Result 24.1.1.*) and $T > 0$ is a finite time horizon. The initial values x_0, $y_0 \in \mathbb{R}$ are real-valued constants and the coefficients

$\lambda_z, \eta_z, \gamma_z, \delta_z,\ z \in \{x, y\}$, are all real-valued deterministic functions, such that (3.3.1) and (3.3.2) are well-defined.

Furthermore, let the instantaneous short-rate process be given by

$$r(t) := x(t) - y(t). \tag{3.3.3}$$

In the case where $y \equiv 0$, this reduces to the standard affine one-factor short rate model class. If additionally $\delta_x(t) \equiv 0$, $\lambda_x(t) \equiv -k_x$ $\eta_x(t) \equiv k_x\theta_x$ and $\gamma_x(t) \equiv \sigma_x^2$, where $k_x, \sigma_x, \theta_x \in \mathbb{R}_{\geq 0}$, it reduces to the standard CIR model

$$dx(t) = k_x(\theta_x - x(t))dt + \sigma_x\sqrt{x(t)}dW_x(t), \tag{3.3.4}$$

which lets (3.3.3) preserve all the features of a standard CIR model in a non-negative interest rate setting.

3.3.1. Description of the Main Results

The main result consists of two main parts. First of all, we derive the zero-coupon bond price for (3.3.3) in the case of the difference of (3.3.1) and (3.3.2) being two independent CIR processes as in (3.3.4). Secondly, we provide numerical experiments to demonstrate the features of this model in Section 3.3.3.

Theorem 3.3.1. *Let $\left(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\in[0,T]}, \mathbb{Q}\right)$ be a stochastic basis, where $\mathbb{Q}$ is a martingale measure as above, making the discounted zero-coupon price processes martingales, $T > 0$ a finite time horizon and let the σ-algebra $(\mathcal{F}_t)_{t\in[0,T]}$ fulfill the usual conditions and support two independent standard Brownian motions W_x and W_y.*

The price of a zero-coupon bond in the model $r(t) = x(t) - y(t)$ with x and y being two independent CIR processes as in (3.3.4) is given by

$$P(t,T) = A_x(t,T)e^{-B_x(t,T)x(t)}A_y(t,T)e^{B_y(t,T)y(t)}, \tag{3.3.5}$$

where $t \leq T$ and for $z \in \{x, y\}$

$$\begin{aligned} A_z(t,T) &= \left(\frac{\phi_1^z e^{\phi_2^z(T-t)}}{\phi_2^z\left(e^{\phi_1^z(T-t)} - 1\right) + \phi_1^z}\right)^{\phi_3^z} \\ B_z(t,T) &= \frac{e^{\phi_1^z(T-t)} - 1}{\phi_2^z\left(e^{\phi_1^z(T-t)} - 1\right) + \phi_1^z} \end{aligned} \tag{3.3.6}$$

with $\phi_i^z \geq 0$, $i = 1, 2, 3$, $z \in \{x, y\}$, defined as

$$\begin{aligned} \phi_1^x &:= \sqrt{k_x^2 + 2\sigma_x^2}, & \phi_2^x &:= \frac{k_x + \phi_1^x}{2}, & \phi_3^x &:= \frac{2k_x\theta_x}{\sigma_x^2} \\ \phi_1^y &:= \sqrt{k_y^2 - 2\sigma_y^2}, & \phi_2^y &:= \frac{k_y + \phi_1^y}{2}, & \phi_3^y &:= \frac{2k_y\theta_y}{\sigma_y^2}. \end{aligned} \tag{3.3.7}$$

Remark 3.3.2. As stated in Theorem 3.3.1, we only need for pricing the zero-coupon bonds that the coefficients ϕ_i^z, $z \in \{x, y\}$, $i = 1, 2, 3$, are real, positive and defined as in (3.3.7).

However, for the numerical implementation, we will assume henceforth the so-called Feller-condition as well, i.e. $2k_z\theta_z \geq \sigma_z^2$ for both CIR processes.

The Feller-condition has no impact on the existence and uniqueness of the solutions to (3.3.4) or on the validity of (3.3.5) but guarantees that the solution remains strictly positive instead of just non-negative, whose violation can cause problems in some numerical schemes.

We tested for the presented data the case, when we do not assume the Feller-condition as well and could not see an improvement in terms of accuracy compared to the case where we assumed it. However, this might be due to our chosen data, because the Feller-condition was only slightly violated. Therefore, we decided to impose the Feller-condition for our numerical tests and leave a detailed investigation of the violation of this condition with different numerical schemes for future research.

For a thorough discussion on the Feller-condition and existence and uniqueness of the solution to (3.3.4), we refer to Andersen, Jäckel and Kahl (2010), Gikhman (2011) and Liao (2018).

The technical part of the proof is quite standard and is reported in Section B.1 with a description of how to derive this result in Section 3.3.2. Formula (3.3.5) provides the necessary ingredient for the numerical experiments in Section 3.3.3 to calibrate the model to the market term structure.

This remainder of the section is organized as follows. In Section 3.3.2 we introduce the model in a general affine model setup and describe our main result Theorem 3.3.1. We will derive the Riccati equations associated with the proposed term structure suitable for the difference of two independent affine processes and solve those explicitly in a CIR framework.

After that, in Section 3.3.3, we will conduct some numerical experiments. First, we calibrate our model via (3.3.5) to the market data at 30/12/2019 and 30/11/2020 in Section 3.3.3.2. Subsequently, we simulate the model by using the Euler-Maruyama scheme in Section 3.3.3.3 and study the mean, variance and distribution of the model in Section 3.3.3.4. Then we price swaptions in Section 3.3.3.5 and summarize the results of the sections in Section 3.3.4.

3.3.2. A Model for Negative Interest Rates

We will now describe how Theorem 3.3.1 can be derived. As aforementioned, we consider all dynamics under the risk-neutral measure $\mathbb{Q}$ and give now a heuristic argument, why it makes sense to choose the term structure in Theorem 3.3.1 as in (3.3.5).

Suppose, that $x(t)$ and $y(t)$ are both independent affine processes. Then the price of a zero-coupon bond for each of them separately (cf. Brigo and Mercurio (2006) p. 69) is given by

$$P(t,T) = \mathbb{E}^{\mathbb{Q}}\left[e^{-\int_t^T z(s)ds} \middle| \mathcal{F}_t \right] = A_z(t,T)e^{-B_z(t,T)z(t)}, \tag{3.3.8}$$

where $z \in \{x, y\}$ and $E_t^{\mathbb{Q}}$ denotes the conditional expectation with respect to $\mathcal{F}_t$ under the measure $\mathbb{Q}$. Now, consider $r(t) = x(t) - y(t)$, then we have by linearity and independence

$$\begin{aligned} P(t,T) &= \mathbb{E}^{\mathbb{Q}}\left[e^{-\int_t^T r(s)ds} \middle| \mathcal{F}_t \right] = \mathbb{E}^{\mathbb{Q}}\left[e^{-\int_t^T (x(s)-y(s))ds} \middle| \mathcal{F}_t \right] \\ &= \mathbb{E}^{\mathbb{Q}}\left[e^{-\int_t^T x(s)ds} \middle| \mathcal{F}_t \right] \mathbb{E}^{\mathbb{Q}}\left[e^{\int_t^T y(s)ds} \middle| \mathcal{F}_t \right]. \end{aligned}$$

If we concentrate in (3.3.8) only on the right-hand side, it would make sense for two independent processes x and y that we can apply these formulas with a change of sign in front of B_y, leading to

$$P(t,T) \overset{!}{=} A_x(t,T)e^{-B_x(t,T)x(t)}A_y(t,T)e^{B_y(t,T)y(t)}.$$

In the following Lemma we will make this argument rigorous.

Lemma 3.3.3. *Let everything be as in Theorem 3.3.1 but let $x(t)$ and $y(t)$ follow the general affine dynamics described in (3.3.1) and (3.3.2).*

Then, the price of a Zero-coupon bond is given by

$$P(t,T) = \mathbb{E}^{\mathbb{Q}}\left[e^{-\int_t^T r(s)ds} \middle| \mathcal{F}_t \right] = A_x(t,T)e^{-B_x(t,T)x(t)}A_y(t,T)e^{B_y(t,T)y(t)}, \tag{3.3.9}$$

where A_z and B_z, $z \in \{x, y\}$, are deterministic functions and are a classical solution to the following system of Riccati equations

$$\begin{cases} -1 - B_x(t,T)\lambda_x(t) - (\partial_t B_x)(t,T) + \frac{1}{2}B_x^2(t,T)\gamma_x(t) = 0, & B_x(T,T) = 0 \\ -B_x(t,T)\eta_x(t) + \frac{1}{2}B_x^2(t,T)\delta_x(t) + \partial_t(\log A_x)(t,T) = 0, & A_x(T,T) = 1 \\ 1 + B_y(t,T)\lambda_y(t) + (\partial_t B_y)(t,T) + \frac{1}{2}B_y^2(t,T)\gamma_y(t) = 0, & B_y(T,T) = 0 \\ B_y(t,T)\eta_y(t) + \frac{1}{2}B_y^2(t,T)\delta_y(t) + \partial_t(\log A_y)(t,T) = 0, & A_y(T,T) = 1. \end{cases} \tag{3.3.10}$$

The proof of this Lemma is referred to Section B.1. The independence of x and y ensures that the Riccati equations for A_x and B_x are decoupled from the ones for A_y and B_y, making it possible to use the existing literature on explicit solutions in the context of short rate models to construct easily a solution for our difference process (3.3.3) in the case where x (3.3.1) and y (3.3.2) are CIR processes.

Remark 3.3.4. One can immediately use Lemma 3.3.3 to construct solutions to other popular one-factor affine short rate models, where an explicit solution is available, e.g. the Vasicek model, provided that x and y are independent.

Introducing dependence between x and y suggests a coupling of A_x and B_x to A_y and B_y and might have an impact on the analytical tractability, but is left for future research.

It is well-known that the processes $x(t)$ and $y(t)$ are non-negative for every $t \geq 0$ (see for instance Cox, Ingersoll and Ross (1985) or Jeanblanc, Yor and Chesney (2009)). We underline that even if the processes $x(t)$ and $y(t)$ are positive, the instantaneous spot rate

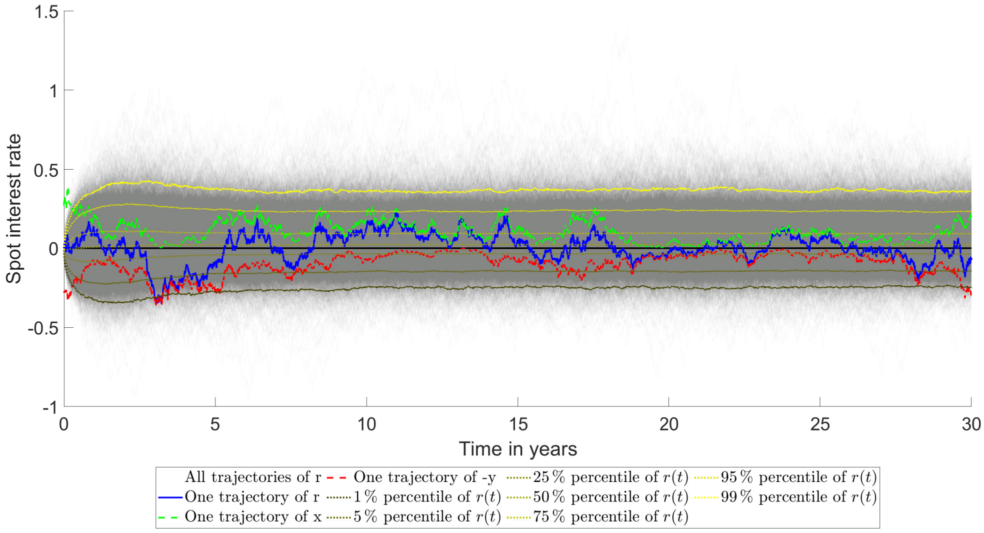


Figure 3.2.: An example of a trajectory with negative interest rates $r = x - y$ and its decomposition in x and $-y$, obtained with the market data on 30/12/2019 and parameters given in Table 3.2.

$r(t)$ could be negative since it is defined as the difference of $x(t)$ and $y(t)$ for every $t > 0$, which is illustrated in Figure 3.2 together with several percentiles of $r(t)$.

3.3.3. Numerical Tests

We will now perform some numerical experiments in our model.[4] In Section 3.3.3.1 we will briefly discuss the market data, which we will use to perform all numerical tests in the subsequent sections. Afterwards, we will describe the calibration procedure of our model to the zero-coupon curves at 30/12/2019 and 30/11/2020 in Section 3.3.3.2. This is followed by a short subsection on simulating the model with the Euler-Maruyama scheme in Section 3.3.3.3 and we investigate the mean, variance and distribution of the short rate model in Section 3.3.3.4. Last but not least, Section 3.3.3.5 will show results on pricing swaptions in our model.

We used for the calculations `Matlab 2022a` with the (Global) Optimization Toolbox running on Windows 10 Pro, on a machine with the following specifications: processor Intel(R) Core(TM) i7-8750H CPU @ 2.20 GHz and 2x32 GB (Dual Channel) Samsung SODIMM DDR4 RAM @ 2667 MHz.

3.3.3.1. Market Data

To obtain the market zero-coupon bond term structure, we first build the EUR Euribor-swap curve which is created from the most liquid interest rate instruments available in the market and constructed as follows: We consider deposit rates and Euribor rates with maturity from

[4] More details about the implementation can be found on https://github.com/kevinkamm/CIR-/blob/main/main.m.

one day to one year and par-swap rates versus six-month Euribor rates with maturity from two years to thirty years. Then the zero interest curve and the zero-coupon bond curve are calculated using a standard "bootstrapping" technique in conjunction with cubic spline interpolation of the continuously compounded rate (cf. Miron and Swannell (1991) for more details).

We choose two different dates and we take the data at the end of each business day. In particular, we test our model at 30/12/2019 and at 30/11/2020. At the first date, the zero interest rates were negative up to year six, while at the second date the entire zero interest rate structure was negative. In Table B.11 and in Table B.12 we report the zero interest rate curve and the zero-coupon bond curve at the two different dates.

Furthermore, in Section 3.3.3.5 we need the market volatility surface to compute the market swaption prices with Bachelier's formula and the strikes to compute the model swaption prices. The volatility surface, strikes and market swaption prices, are for both dates in the Appendix in Table B.2, Table B.3, Table B.4, Table B.5, Table B.6 and Table B.7, respectively.

All data has been downloaded from Bloomberg and is used in the following subsections for our numerical experiments. We start in the next subsection with calibrating our model to the zero-coupon curve.

3.3.3.2. Calibration

In this subsection, we will discuss how we calibrate our model to the market zero-coupon curve given in Table B.11 and Table B.12 by using the formula derived in (3.3.5).

Let us denote $\Pi := [\phi_1^x, \phi_2^x, \phi_3^x, \phi_1^y, \phi_2^y, \phi_3^y, x_0, y_0]^T \in \mathbb{R}^8$. We will formulate the calibration procedure as a constraint minimization problem in $\mathbb{R}^8$ for the parameters Π with objective function

$$f(\Pi) := \sum_{i=1}^{n} \left(\frac{P^M(0, T_i)}{P(\Pi; 0, T_i)} - 1 \right)^2, \tag{3.3.11}$$

where $n \in \mathbb{N}$ is the number of time points, where market data is available, and T_i, $i = 1, \ldots, n$ are these maturities. The market zero-coupon curve is denoted by $P^M(0, T_i)$ and $P(\Pi; 0, T_i)$ is the price of a zero-coupon bond in our model given by (3.3.5) with parameters Π.

The objective function describes the relative square difference between the market zero-coupon bond prices and the theoretical prices from the model given by (3.3.5).

The set of admissible parameters $\mathcal{A}$ will consist of the following constraints arising from the well-definedness of the formulas (3.3.7):

1. First of all, let us note that there is a one-to-one correspondence between the parameters

Π and k_z, σ_z and θ_z if one is looking for positive real solutions only. We have

$$\begin{aligned} k_x &= 2\phi_2^x - \phi_1^x, & k_y &= 2\phi_2^y - \phi_1^y, \\ \sigma_x &= \sqrt{2\left(\phi_2^x\phi_1^x - (\phi_2^x)^2\right)}, & \sigma_y &= \sqrt{-2\left(\phi_2^y\phi_1^y - (\phi_2^y)^2\right)}, \\ \theta_x &= -\frac{\phi_x^2\phi_x^3(\phi_x^1 - \phi_x^2)}{\phi_x^1 - 2\phi_x^2}, & \theta_y &= \frac{\phi_y^2\phi_y^3(\phi_y^1 - \phi_y^2)}{\phi_y^1 - 2\phi_y^2}. \end{aligned} \tag{3.3.12}$$

2. We require $\sigma_z \in \mathbb{R}_{\geq 0}$, $z \in \{x, y\}$. By rearranging (3.3.12), these conditions are equivalent to $\phi_1^x \geq \phi_2^x$ and $\phi_2^y \geq \phi_1^y$;
3. A positive mean-reversion speed, i.e. $k_z \geq 0$, is equivalent to $2\phi_2^z \geq \phi_1^z$, $z \in \{x, y\}$;
4. The Feller condition $2k_z\theta_z \geq \sigma_z^2$ is equivalent to $\phi_3^z \geq 1$, $z \in \{x, y\}$;
5. A positive mean for each CIR process, i.e. $\theta_z \geq 0$, is by positivity of σ_z^2 and k_z equivalent to $\phi_3^z \geq 0$, which is already satisfied by the Feller condition;
6. The parameter ϕ_1^z, assuming that it is real-valued, is positive by definition, meaning that by the positivity of the mean reversion speed, ϕ_2^z will be as well. Therefore, all ϕ are positive;
7. As both CIR processes x_t and y_t, individually, are positive processes, we additionally require $x_0 \geq 0$ and $y_0 \geq 0$.

The advantage of using the parameters Π instead of k_z, σ_z and θ_z is that we can rewrite these conditions as a system of linear inequality constraints in matrix notation $A \cdot \Pi \leq 0$ (where less-or-equal sign is to be understood in a element-wise sense), where

$$A := \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 1 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -2 & 0 & 0 & 0 \end{bmatrix}$$

with lower bounds $\Pi_i \geq 0$, $i = 1, \ldots, 8$, and $\Pi_3 = \phi_3^x \geq 1$, as well as $\Pi_6 = \phi_3^y \geq 1$.

In total, the set of admissible parameters is given by

$$\mathcal{A} := \left\{\Pi = [\Pi_1, \ldots, \Pi_8]^T \in \mathbb{R}_{\geq 0}^8 : \Pi_3 \geq 1, \Pi_6 \geq 1 \text{ and } A \cdot \Pi \leq 0\right\}. \tag{3.3.13}$$

Finally, a solution Π^* to the calibration problem is a minimizer of

$$\min_{\Pi \in \mathcal{A}} f(\Pi). \tag{3.3.14}$$

To solve (3.3.14) numerically, we want to use `Matlab`'s function `fmincon` in the (Global) Optimization Toolbox. In order to use this function, we need an initial guess of the parameter Π and the computational time will depend on that choice. In Table 3.1 we present a few choices for initial guesses of Π. The first row for each date 30/12/2019 or 30/11/2020 refers to `Matlab`'s function `ga` in the (Global) Optimization Toolbox, which uses a generic global optimization algorithm to find a solution of (3.3.14) without starting from an initial guess, which takes a long time to compute, roughly 35 to 43 seconds. In the following three rows

Table 3.1.: Calibration times and corresponding mean relative errors (MRE) for different initial parameters at 30/12/2019 and 30/11/2020.

Inital Parameter	Times (in s)	MRE (in %)
Calibration at 30/12/2019		
`ga`	42.136	0.142014 %
0.50001 0.50001 1.5 0.50001 0.50001 1.5 0.50001 0.50001	0.287	0.143798 %
1 1 2 1 1 2 1 1	0.229	0.146769 %
1e-05 1e-05 1 1e-05 1e-05 1 1e-05 1e-05	0.276	0.145207 %
0.048808 0.72079 1.4275 0.64469 0.32152 1.4794 0.2556 0.25427	0.334	0.146509 %
0.66073 0.78158 1.5547 0.3779 0.24209 1.249 0.74017 0.63334	103.949	0.14374 %
Calibration at 30/11/2020		
`ga`	35.842	0.135885 %
0.50001 0.50001 1.5 0.50001 0.50001 1.5 0.50001 0.50001	0.280	0.137577 %
1 1 2 1 1 2 1 1	0.188	0.138228 %
1e-05 1e-05 1 1e-05 1e-05 1 1e-05 1e-05	40.864	0.13642 %
0.87647 0.89591 1.4508 0.10496 0.63629 1.3009 0.3297 0.74067	0.185	0.140284 %
0.51541 0.83366 1.9624 0.79757 0.13068 1.6517 0.54064 0.43674	0.342	0.14292 %

are three manual initial guesses. We can see that the first two choices work for both dates exceptionally fast (0.3 seconds) and the accuracy is almost identical to all other choices, making this model a good choice if live calibration to the data is needed, which we also use in the following numerical experiments. In the last two rows we used random starting parameters to demonstrate that the error remains stable but the computational time varies.

For the algorithms used by `Matlab` we refer the reader to Gilli, Maringer and Schumann (2011), in the context of financial mathematics.

The results of the aforementioned calibration procedure are displayed in Table 3.2 for both dates 30/12/2019 and 30/11/2020. On the left-hand side, one can see the parameters Π^* and on the right-hand side the corresponding model parameters derived from Π^*. At both dates we obtain good results in fitting the market term structure. The mean relative error (MRE), i.e. $\frac{1}{n}\sum_{i=1}^{n}\left|\frac{P^M(0,T_i)}{P(\Pi^*;0,T_i)}-1\right|$, over the entire term structure is 0.144 % at the first date and 0.138 % at the second date.

3.3.3.3. Euler-Monte-Carlo Simulation

In order to forecast the future expected interest rate, we use the Euler-Maruyama scheme to simulate the instantaneous spot rate r (3.3.3). We refer to Dereich, Neuenkirch and Szpruch (2012) and the references therein for a list of different Euler-type methods to simulate a CIR process. In our experiments, we simulate the processes $x(t)$ and $y(t)$ by the truncated Euler scheme defined as follows:

First of all, we fix a homogeneous time grid $0 = t_0 \leq t_1 \leq \cdots \leq t_N = T$ for the interval $[0,T]$ with $N+1$ time points and mesh $\Delta t_i := t_{i+1} - t_i \equiv \Delta := \frac{T}{N}$ for all $i = 0, \ldots, N-1$. Secondly, we simulate the two independent Brownian motions W_z, $z \in \{x,y\}$, and define their time increment as $\Delta W_z(t_i) := W_z(t_{i+1}) - W_z(t_i)$. In total, we compute $r(t_{i+1}) := x(t_{i+1}) - y(t_{i+1})$

Table 3.2.: Calibration parameters Π^*, model parameters and mean relative errors (MRE) at 30/12/2019 and 30/11/2020, obtained with the market data given in Table B.11 and Table B.12.

Parameter	30/12/2019	30/11/2020	Parameter	30/12/2019	30/11/2020
ϕ_1^x	0.710501	0.767497	k_x	0.578626	0.631802
ϕ_2^x	0.644564	0.699649	σ_x	0.291551	0.308122
ϕ_3^x	1.60862	1.6014	θ_x	0.118155	0.120319
x_0	0.268914	0.257145			
ϕ_1^y	0.468673	0.523363	k_y	0.59774	0.665895
ϕ_2^y	0.533206	0.594629	σ_y	0.262334	0.291125
ϕ_3^y	1.50249	1.49966	θ_y	0.0864925	0.0954364
y_0	0.280095	0.270007			
$f\left(\Pi^*\right)$	$3.247465e-04$	$3.548162e-04$			
MRE	0.144 %	0.138 %			

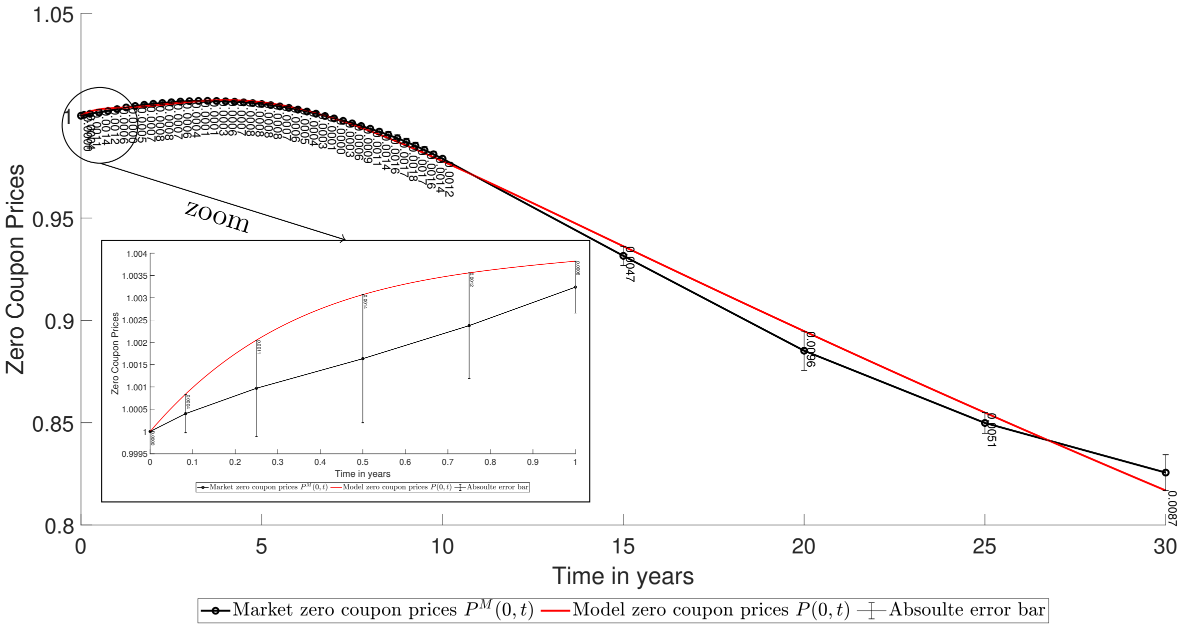


Figure 3.3.: A comparison of the market zero-coupon prices (Table B.11) to the model zero-coupon prices with absolute errors at 30/12/2019 with parameters given in Table Table 3.2.

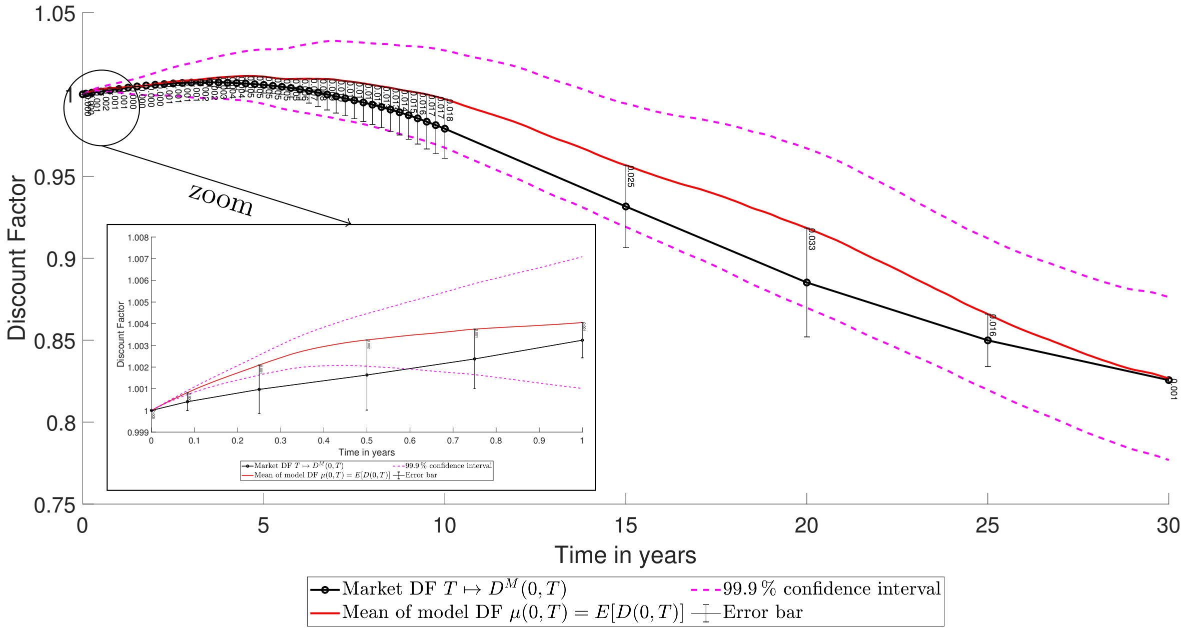


Figure 3.4.: A comparison of the market discount factors (Table B.11) to the mean of the model discount factors with absolute errors at 30/12/2019 with parameters given in Table 3.2 by using (3.3.15) with $\Delta = \frac{1}{256}$ and $M = 10000$.

for $i = 0, \dots, N-1$, where

$$\begin{aligned} x(t_{i+1}) &= x(t_i) + k_x(\theta_x - x(t_i))\Delta t_i + \sigma_x\sqrt{\max(x(t_i),0)}\Delta W_x(t_i) \\ y(t_{i+1}) &= y(t_i) + k_y(\theta_y - y(t_i))\Delta t_i + \sigma_y\sqrt{\max(y(t_i),0)}\Delta W_y(t_i). \end{aligned} \tag{3.3.15}$$

We choose the max inside the square-root to ensure that the square-root remains real, because due to discretization effects the positivity of $x(t_i)$ and $y(t_i)$ might be violated.

In the following experiments we choose $\Delta = \frac{1}{256}$ and use $M = 10000$ samples for each of the Brownian motions. In Figure 3.4 we show the mean and 99.9 % confidence interval (under the assumption of the central limit theorem) of the model discount factors, i.e. $D(0,T) := \exp\left(-\int_0^T r(s)ds\right)$ with simulated short rate $r(s)$, compared to the market discount factors, i.e. $D^M(0,T) = P^M(0,T)$ from Table B.11, at 30/12/2019. One can see that the mean does not differ from the market discount factors very much till 5 years with an error of magnitude 0.005 and increases slightly to a magnitude of 0.05 afterwards till 30 years.

A more detailed comparison of the mean absolute errors, i.e. the absolute value of the difference of the mean over all simulations of our model to the market data, at each maturity can be found in the appendix in Table B.1.

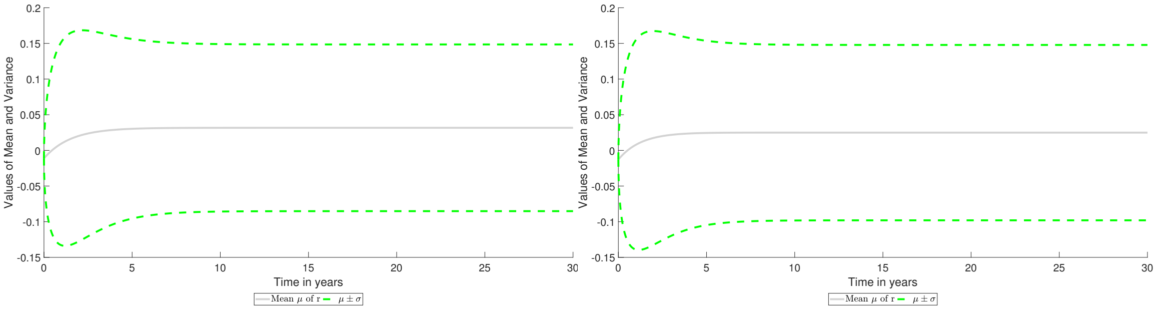


Figure 3.5.: Mean and standard deviation of $r(t)$ using the calibrated parameters in Table 3.2, obtained with (3.3.16) and (3.3.17). The left picture shows the results at 30/12/2019 and the right at 30/11/2020.

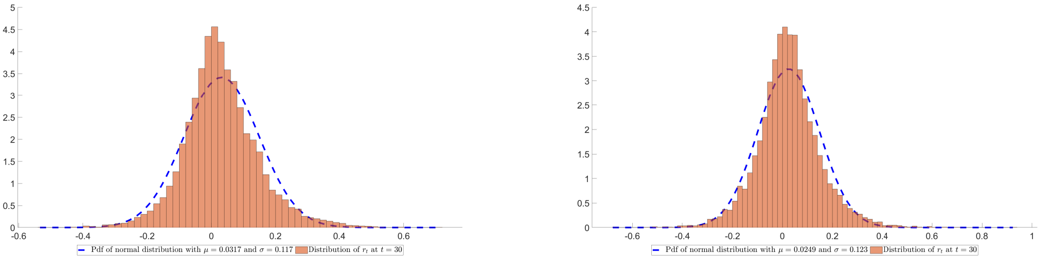


Figure 3.6.: Distribution of the simulated short rate $r(t)$ compared to the normal distribution at $t = 30$ using the calibrated parameters in Table 3.2, $\Delta = \frac{1}{256}$ and $M = 10000$. The left picture shows the results at 30/12/2019 and the right at 30/11/2020.

3.3.3.4. Mean and Variance

The $\mathcal{F}_s$-conditional mean and variance of the CIR process are well-known (cf. Brigo and Mercurio (2006): *p. 66 Equation 3.23*) and are given by

$$\begin{aligned}\mathbb{E}^{\mathbb{Q}}\left[z(t)|\,\mathcal{F}_s\right] &= z(s)e^{-k_z(t-s)} + \theta_z\left(1-e^{-k_z(t-s)}\right)\\ \mathrm{Var}^{\mathbb{Q}}\left[z(t)|\,\mathcal{F}_s\right] &= z(s)\frac{\sigma_z^2}{k_z}\left(e^{-k_z(t-s)} - e^{-2k_z(t-s)}\right) + \theta_z\frac{\sigma_z^2}{2k_z}\left(1-e^{-k_z(t-s)}\right)^2,\end{aligned}$$

where $z \in \{x, y\}$. In the case of the difference of two CIR processes we have

$$\mathbb{E}^{\mathbb{Q}}\left[r(t)|\,\mathcal{F}_s\right] = \mathbb{E}^{\mathbb{Q}}\left[x(t)-y(t)|\,\mathcal{F}_s\right] = \mathbb{E}^{\mathbb{Q}}\left[x(t)|\,\mathcal{F}_s\right] - \mathbb{E}^{\mathbb{Q}}\left[y(t)|\,\mathcal{F}_s\right] \tag{3.3.16}$$

and by independence

$$\mathrm{Var}^{\mathbb{Q}}\left[r(t)|\,\mathcal{F}_s\right] = \mathrm{Var}^{\mathbb{Q}}\left[x(t)|\,\mathcal{F}_s\right] + \mathrm{Var}^{\mathbb{Q}}\left[y(t)|\,\mathcal{F}_s\right]. \tag{3.3.17}$$

In Figure 3.6 we show for each date the histogram of the short rate distribution after 30 years. To describe the distribution of $r(t)$ after 30 years better, we also compare it to the density of a normal random variable with the same mean and variance. As one expects, the distribution of r shows a slight skewness and fatter tail with respect to the normal distribution.

Table 3.3.: 30/12/2019: difference between swaption model price and swaption market price

Tenor Maturity	1	2	5	7	10
1	4.82602 %	3.72741 %	1.56009 %	0.601718 %	−0.554628 %
2	4.70047 %	3.56775 %	1.18057 %	0.0467543 %	−1.37405 %
5	3.60511 %	2.4125 %	−0.249531 %	−1.64559 %	−3.50471 %
7	3.23362 %	1.97696 %	−0.945175 %	−2.48803 %	−4.54551 %
10	2.85895 %	1.49467 %	−1.69265 %	−3.37829 %	−5.63478 %
15	2.63259 %	1.17927 %	−2.21714 %	−4.05043 %	−6.53224 %
20	2.48029 %	1.01554 %	−2.47166 %	−4.39298 %	−6.98019 %

Table 3.4.: 30/11/2020: difference between swaption model price and swaption market price

Tenor Maturity	1	2	5	7	10
1	4.87829 %	3.71571 %	1.62322 %	0.745822 %	−0.307305 %
2	4.62191 %	3.4627 %	1.22337 %	0.167669 %	−1.18931 %
5	3.82762 %	2.596 %	−0.0624237 %	−1.4526 %	−3.34358 %
7	3.45837 %	2.13705 %	−0.80375 %	−2.35496 %	−4.46368 %
10	3.24577 %	1.77121 %	−1.56541 %	−3.3177 %	−5.74088 %
15	3.02788 %	1.4328 %	−2.26341 %	−4.2787 %	−7.03714 %
20	3.16548 %	1.42331 %	−2.6344 %	−4.8421 %	−7.85129 %

3.3.3.5. Pricing Swaptions

In this subsection, we test if our model is market consistent, in the sense whether the model is able to reproduce market swaption prices or not.

We compare market swaption prices to model swaption prices with different tenors $(1, 2, 5, 7, 10$ years) and maturities $(1, 2, 5, 7, 10, 15, 20$ years). The market swaption prices (Table B.6 and Table B.7) are computed by Bachelier's formula from normal volatilities quoted in the market (Table B.2 and Table B.3) whereas the model swaption prices are from the simulated future zero-coupon prices in (3.3.5). The difference between market price to model prices for 30/12/2019 and 30/11/2020 are reported in Table 3.3 and Table 3.4, respectively. We notice that, similar to one-factor short interest rate model, our model fails to capture the full swaption volatility surface. This result is not surprising, since the model uses essentially a single volatility factor due to the fact that the model parameters are constant and the Brownian motion are independent.

3.3.4. Conclusion

In this section, we propose a new model to handle the challenges arising from negative interest rates, while preserving the analytical tractability of the original CIR model without introducing any shift to the market interest rates. The strength of our model is that it is very simple, fast to calibrate and fits the present market term structure very well for an essentially one-factor short rate model.

Let us briefly summarize our discoveries of the numerical section. We show that the distribution of the short rate after 30 years has similar features compared to the original CIR model in terms of skewness and fat tail. Moreover, we notice that we require an extension of the model to price swaptions more accurately, which is part of the next section.

3.4. Exogenous Model

In this section, we transform the endogenous model from Section 3.3 into an exogenous one, in which the observed term-structure is an input, to fit the short-rate model to the swaption surface.

A basic strategy for this, is the inclusion of time-dependent parameters to exactly reproduce the observed term-structure. In fact, matching the term-structure exactly is equivalent to solving a system with an infinite number of equations. However, this is only possible by introducing an infinite number of parameters or, equivalently, a deterministic function of time. We follow the method illustrated in Brigo and Mercurio (2006): *pp. 95ff. Section 3.8 A General Deterministic-Shift Extension* to extend any time-homogeneous short-rate model, so as to exactly reproduce any observed term-structure of interest rates while preserving the possible analytical tractability of the original model.

To be more precise, we consider the CIR dynamics for $z \in \{x, y\}$

$$dz(t) = k_z(\theta_z - z(t))dt + \sigma_z\sqrt{z(t)}dW_z(t), \quad z(0) = z_0 \tag{3.4.1}$$

under a martingale measure $\mathbb{Q}$ with $k_z, \theta_z, \sigma_z \in \mathbb{R}_{>0}$ and define the short-rate as

$$r(t) := x(t) - y(t) + \psi\left(t\right), \tag{3.4.2}$$

where W_y and W_x are two independent standard Brownian motions on a stochastic basis $\left(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\in[0,T]}, \mathbb{Q}\right)$ and $\psi(t) := f^M(0,t) - f(0,t)$ is a deterministic function defined as the difference of the market and model instantaneous forward rate.

Since the market term-structure is now an input, we can calibrate the model parameters to the swaption surface. However, simple Monte-Carlo techniques are in general very slow and memory demanding. Therefore, we resort to an approximation formula known as Gram-Charlier expansion (cf. Tanaka, Yamada and Watanabe (2010)) in our model. This allows for a fast and accurate calibration procedure.

3.4.1. Description of the Main Results

In this section, we will first of all extend the results of Section 3.3 by using a deterministic shift extension. The zero-coupon price in the extended model (3.4.2) is given in the next Lemma.

Lemma 3.4.1. *Let $\left(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\in[0,T]}, \mathbb{Q}\right)$ be a stochastic basis, where $\mathbb{Q}$ is a martingale measure, $T > 0$ a finite time horizon and let the σ-algebra $(\mathcal{F}_t)_{t\in[0,T]}$ fulfill the usual conditions and support two independent standard Brownian motions W_x and W_y. The price of a zero-*

coupon bond in the model $r(t) := x(t) - y(t) + \psi(t)$ is given by

$$P(t,T) = \frac{P^M(0,T)}{P^M(0,t)} \frac{P^{CIR\text{-}}(0,t)}{P^{CIR\text{-}}(0,T)} P^{CIR\text{-}}(t,T),$$

where $P^{CIR\text{-}}(t,T)$ is the zero-coupon price from Theorem 3.3.1 and $P^M(0,T)$ the market zero-curve.

The derivation of this result is straightforward and referred to Section 3.4.2.

We will see that it is necessary to study the so-called *swap moments* to derive the Gram-Charlier expansion. In our model, we will find explicit formulas allowing for fast swaption pricing and it is part of the next technical Lemma.

Lemma 3.4.2. *Let everything be as in Lemma 3.4.1. The so-called* swap moments *at time $t < T_0$ of order $m \in \mathbb{N}$ are given by*

$$M^m(t) := \mathbb{E}^{\mathbb{Q}^{T_0}}\left[\left(\mathrm{Swap}_{T_0}^{T_N}(T_0;K,\zeta)\right)^m \middle| \mathcal{F}_t\right] = \left(\frac{P^{CIR\text{-}}(0,T_0)}{P^M(0,T_0)}\right)^m \frac{1}{P^{CIR\text{-}}(t,T_0)}$$
$$\sum_{\substack{0 \le k_0,\dots,k_N \le N \\ k_0+\dots+k_N = m}} \frac{m!}{k_0! \cdots k_N!} \widetilde{a}_0^{k_0} \cdots \widetilde{a}_N^{k_N} \left(M_x(t,T_0) e^{-N_x(t,T_0)x(t)} M_y(t,T_0) e^{N_y(t,T_0)y(t)}\right)$$

where we suppress the dependency of N_z, M_z on k_i for readability. The coefficients $\widetilde{a}_i$ are given by

$$\widetilde{a}_0 := \zeta \frac{P^M(0,T_0)}{P^{CIR\text{-}}(0,T_0)}, \quad \widetilde{a}_N := -\zeta\left(1 + K\alpha_N\right) \frac{P^M(0,T_N)}{P^{CIR\text{-}}(0,T_N)}, \quad \widetilde{a}_i := -\zeta K \alpha_i \frac{P^M(0,T_i)}{P^{CIR\text{-}}(0,T_i)},$$

for $i = 1, \dots, N-1$, year fractions α_i, fixed swap rate K and swap type $\zeta = 1$ for a payer swap and $\zeta = -1$ for a receiver swap.

Moreover, the functions M_z, N_z, $z \in \{x,y\}$ are defined as

$$M_z(t,T_0) = a_z \left(\frac{\phi_1^z \exp\left(\phi_2^z(T_0 - t)\right)}{\phi_1^z + \phi_2^z\left(\exp\left(\phi_1^z(T_0-t)\right) - 1\right)\left(1 + b_z\left(\phi_1^z - \phi_2^z\right)\right)}\right)^{\phi_3^z}, \quad a_z = \prod_{j=0}^{N} A_z(T_0,T_j)^{k_j}$$

$$N_z(t,T_0) = \frac{b_z\phi_1^z + \left(\exp\left(\phi_1^z(T_0-t)\right) - 1\right)\left(1 + b_z\left(\phi_1^z - \phi_2^z\right)\right)}{\phi_1^z + \phi_2^z\left(\exp\left(\phi_1^z(T_0-t)\right) - 1\right)\left(1 + b_z\left(\phi_1^z - \phi_2^z\right)\right)}, \quad b_z = \sum_{j=0}^{N} k_j B_z(T_0,T_j),$$

where A_z, B_z are the functions defined in Theorem 3.3.1. The swap cumulants $c_l(t)$ at time t are now given by the formulas in Section B.4 by setting $\mu_i := M^l(t)$, $l = 1, \dots, m$.

For the proof of this Lemma we follow Tanaka, Yamada and Watanabe (2010) closely, which is referred to Section 3.4.3.1.

The main result of this section is the approximation of swaption prices by the Gram-Charlier expansion with short-rate (3.4.2), which follows immediately from Lemma 3.4.2 by using Proposition 3.4.4 and is referred to Section 3.4.3.2.

Theorem 3.4.3. *Let everything be as in Lemma 3.4.2.*

The time t price of a $T_0 \times (T_N - T_0)$ payer ($\zeta = 1$) and receiver ($\zeta = -1$) swaption is given by

$$\mathrm{Swaption}_{T_0}^{T_N}(t; K, \zeta) = P(t, T_0)\left(C_1 \mathcal{N}\left(\frac{C_1}{\sqrt{C_2}}\right) + \sqrt{C_2}\varphi\left(\frac{C_1}{\sqrt{C_2}}\right)\left(1 + \sum_{l=3}^{\infty}(-1)^l q_l H_{l-2}\right)\right),$$

where $\mathcal{N}$ denotes the cdf of the normal distribution, φ is the pdf or the normal distribution and H_l are the probabilist's Hermite polynomials (see Section B.3). The coefficients $q_0 = 1$, $q_1 = q_2 = 0$, and for $n \geq 3$

$$q_n = \sum_{m=1}^{\lfloor \frac{n}{3} \rfloor} \sum_{\substack{k_1+\cdots+k_m=n \\ k_i \geq 3}} \frac{C_{k_1} \cdots C_{k_m}}{m! k_1! \cdots k_m!} \left(\frac{1}{\sqrt{C_2}}\right)^n$$

for $C_l := c_l(t)P(t, T_n)^l$ with $c_l(t)$ being the swap cumulants from Lemma 3.4.2 for fixed $t \geq 0$.

This formula will provide the necessary ingredient for the numerical experiments in Section 3.4.4 making it possible to calibrate the model to the swaption surface very efficiently. After successfully calibrating the model, we apply it to find constant maturity swap rates in Section 3.4.4.4 and Bermudan swaption pricing in Section 3.4.4.5 using the Least-Square Monte Carlo technique. We will see a good performance of this model compared to the reference data downloaded from Bloomberg.

We performed tests on two different dates 30/12/2019 and 30/11/2020. At the first date, the market zero rates were partially negative and at the second date they were completely negative. We saw similar numerical results at both dates and decided for the sake of brevity to only present the results at 30/12/2019. For the interested reader we will make the data at 30/11/2020 as well as the code of the numerical implementation available online.[5]

The section is organized as follows. In Section 3.4.2, we first introduce the deterministic shift extension and the corresponding zero-coupon price.

In Section 3.4.3, we will derive the Gram-Charlier expansion. This is done by first recalling how a probability density of a random variable can be approximated by Hermite polynomials. We will see that it is necessary to study the cumulants or equivalently the moments of this random variable. In our case, this will be the *swap moments* and we will show, how to derive them from the so-called *bond moments* by solving some Riccati equations, which will have explicit solutions in our model, making it possible to compute swaption prices very fast.

After that, in Section 3.4.4, we will conduct some numerical experiments. First, we calibrate our model to the market swaption surface at 30/12/2019 in Section 3.4.4.2. Subsequently, we simulate the model by using the Euler-Maruyama scheme in Section 3.4.4.3 and compute CMS rates in Section 3.4.4.4. We conclude our numerical tests by pricing Bermudan swaptions in Section 3.4.4.5. Finally, we summarize the results of the chapter in Section 3.5 and discuss possible extensions for future research.

[5] More details about the implementation can be found on https://github.com/kevinkamm/CIR--.

3.4.2. A Model for Negative Interest Rates with Perfect Fit to the Term-Structure

Let us define $\alpha := (\alpha_x, \alpha_y)$, $\alpha_z := (k_z, \theta_z, \sigma_z)$, $z \in \{x, y\}$. We want to use the general deterministic shift extension by Brigo and Mercurio (2006): *pp. 95ff. Chapter 3.8 A General Deterministic-Shift Extension* or Brigo and Mercurio (2001b) in the case of multifactor models. We note that contrary to the presented ideas in the aforementioned papers, we do not need to introduce another probability space for our purposes and will use the same risk-neutral measure for all dynamics.

Thus, we are interested in the following short rate model on $(\Omega, \mathcal{F}, \mathbb{Q})$

$$r(t;\alpha) := r^{\text{CIR-}}(t;\alpha) + \psi(t;\alpha) = x(t;\alpha_x) - y(t;\alpha_y) + \psi(t;\alpha_x), \quad \psi(0;\alpha) = 0, \tag{3.4.3}$$

where $r^{\text{CIR-}}$ denotes the short-rate model without the deterministic shift extension. We will suppress the dependency on the parameters α for readability whenever there is no confusion.

Likewise, we recall from Theorem 3.3.1 that the price of the zero-coupon bond for the non-extended model is given by

$$P^{\text{CIR-}}(t,T) = A_x(t,T)e^{-B_x(t,T)x(t)}A_y(t,T)e^{B_y(t,T)y(t)}.$$

Analogue to Brigo and Mercurio (2001b): *p. 5 Theorem 3.1* we derive easily the price of a zero-coupon bond in the short-rate model (3.4.3) for given parameters α

$$\begin{aligned} P(t,T) = \mathbb{E}^{\mathbb{Q}}\left[\exp\left(-\int_t^T r(s)ds\right)\middle|\mathcal{F}_t\right] &= \mathbb{E}^{\mathbb{Q}}\left[\exp\left(-\int_t^T x(s) - y(s) + \psi(s)ds\right)\middle|\mathcal{F}_t\right] \\ &= \exp\left(-\int_t^T \psi(s)ds\right)P^{\text{CIR-}}(t,T) \end{aligned}$$

because ψ is deterministic.

To ensure a perfect fit to the initial term-structure, we set as in Brigo and Mercurio (2001b): *pp. 5–6 Corollary 3.2*

$$\psi(t;\alpha) = f^M(0,t) - f^\alpha(0,t),$$

where $f^M(0,t)$ is the instantaneous market forward rate and

$$f^\alpha(0,t) = -\frac{\partial_T(A_x(0,t))}{A_x(0,t)} + \partial_T(B_x(0,t))\,x(0) - \frac{\partial_T(A_y(0,t))}{A_y(0,t)} - \partial_T(B_y(0,t))\,y(0)$$

is instantaneous market rate for $r^{\text{CIR-}}$ (see Section B.2).

More conveniently, we observe that this is equivalent to asking that the following equation holds

$$\exp\left(-\int_t^T \psi^\alpha(s)ds\right) = \frac{P^M(0,T)}{P^M(0,t)}\frac{P^{\text{CIR-}}(0,t)}{P^{\text{CIR-}}(0,T)},$$

where we used $P^z(t,T) = \exp\left(-\int_t^T f^z(t,s)ds\right)$, $z \in \{M, \text{CIR-}\}$.

In total, this leads to the following formula for the zero-coupon price of the deterministic shift extended model

$$P(t,T) = \frac{P^M(0,T)}{P^M(0,t)} \frac{P^{\text{CIR-}}(0,t)}{P^{\text{CIR-}}(0,T)} P^{\text{CIR-}}(t,T) \tag{3.4.4}$$

and $P(0,T) = P^M(0,T)$ is guaranteed.

3.4.3. Gram-Charlier Expansion

We will use all the results available in Tanaka, Yamada and Watanabe (2010): *pp. 3ff. Section 2.1 Gram-Charlier expansion* and apply them to our case.

Let us first of all make the following observation: The payer ($\zeta = 1$) and receiver ($\zeta = -1$) swap value (3.1.1) can both be rewritten as

$$\text{Swap}_{T_0}^{T_N}(t; K, \zeta) := \sum_{i=0}^{N} a_i^\zeta P(t, T_i),$$

where a_i^ζ are equal to

$$a_0^\zeta := \zeta, \qquad a_N^\zeta := -\zeta\left(1 + K\alpha_N\right), \qquad a_i^\zeta := -\zeta K \alpha_i, \quad i = 1, \dots, N-1.$$

Now, with this notation, we can rewrite the swaption prices (3.1.4) to get

$$\begin{aligned} \text{Swaption}(t; K, \zeta) &= P(t, T_0)\mathbb{E}^{\mathbb{Q}^{T_0}}\left[\left(\text{Swap}_{T_0}^{T_N}(T_0; K, \zeta)\right)^+ \middle| \mathcal{F}_t\right] \\ &\overset{!}{=} P(t, T_0)\int_0^\infty x f(x) dx, \end{aligned}$$

for an unknown density function f. For the remainder of this section we will drop the dependency on ζ, K, T_0 and T_N to ease the notation. The idea of the Gram-Charlier expansion is to approximate this density function f by using the orthonormal basis of Hermite polynomials (see Appendix B.3), which is the content of the next Proposition (cf. Tanaka, Yamada and Watanabe (2010): *p. 3 Proposition 2.1* and Cheng (2013): *p. 5 Proposition 2.1.2*).

Proposition 3.4.4. *Assume that a random variable Y has the continuous density function f and has finite cumulants c_k, $k \geq 1$. Then the following holds:*

1. *f can be expanded as*

$$f(x) = \sum_{n=0}^{\infty} \frac{q_n}{\sqrt{c_2}} H_n\left(\frac{x - c_1}{\sqrt{c_2}}\right) \varphi\left(\frac{x - c_1}{\sqrt{c_2}}\right),$$

where H_n are the probabilist's Hermite polynomials and φ the probability density function of the standard normal distribution, as well as $q_0 = 1$, $q_1 = q_2 = 0$, and for

$n \geq 3$

$$q_n = \frac{1}{n!}\mathbb{E}\left[H_n\left(\frac{Y-c_1}{\sqrt{c_2}}\right)\right] = \sum_{m=1}^{\lfloor \frac{n}{3} \rfloor} \sum_{\substack{k_1+\cdots+k_m=n \\ k_i \geq 3}} \frac{c_{k_1}\cdots c_{k_m}}{m!k_1!\cdots k_m!}\left(\frac{1}{\sqrt{c_2}}\right)^n.$$

2. For any $a \in \mathbb{R}$

$$\begin{aligned}\mathbb{E}\left[Y\mathbb{1}_{Y\geq a}\right] = {} & c_1\mathcal{N}\left(\frac{c_1-a}{\sqrt{c_2}}\right) + \sqrt{c_2}\varphi\left(\frac{c_1-a}{\sqrt{c_2}}\right) \\ & + \sum_{n=3}^{\infty}(-1)^{n-1}q_n\varphi\left(\frac{c_1-a}{\sqrt{c_2}}\right)\left[aH_{n-1}\left(\frac{c_1-a}{\sqrt{c_2}}\right) - \sqrt{c_2}H_{n-2}\left(\frac{c_1-a}{\sqrt{c_2}}\right)\right],\end{aligned}$$

where furthermore $\mathcal{N}$ denotes the cumulative distribution function of the standard normal distribution.

In particular, we have

$$q_3 = \frac{c_3}{3!c_2^{\frac{3}{2}}}, \qquad q_4 = \frac{c_4}{4!c_2^{\frac{4}{2}}}, \qquad q_5 = \frac{c_5}{5!c_2^{\frac{5}{2}}}, \qquad q_6 = \frac{c_6+10c_3^2}{6!c_2^{\frac{6}{2}}}, \qquad q_7 = \frac{c_7+35c_3c_4}{7!c_2^{\frac{7}{2}}}.$$

Therefore, all we have to do is determine the swap cumulants. This will be done in several steps: First, we will use the fact that cumulants can be computed from moments, see Appendix B.4. Second, we compute the so-called swap moments in Equation (3.4.5), which in turn are computed from so-called bond moments. Last but not least, in order to compute the bond moments, we need to derive a new system of Riccati equations in Equation (3.4.7).

3.4.3.1. Bond and Swap Moments

Since, cumulants can be expressed by moments, vice versa (see Appendix B.4), we will study the *Swap moments* in this section, which we denote by

$$M^m(t) := \mathbb{E}^{\mathbb{Q}^{T_0}}\left[(\mathrm{Swap}(T_0))^m\middle|\mathcal{F}_t\right] = \mathbb{E}^{\mathbb{Q}^{T_0}}\left[\left(\sum_{i=0}^{N}a_iP(T_0,T_i)\right)^m\middle|\mathcal{F}_t\right].$$

It can be shown by induction that the m-th power can be rewritten as

$$\left(\sum_{i=0}^{N}a_iP(T_0,T_i)\right)^m = \sum_{0\leq i_1,\ldots,i_m\leq N} a_{i_1}\cdots a_{i_m}\left(\prod_{k=1}^{m}P(T_0,T_{i_k})\right).$$

Now, notice that all a_i are $\mathcal{F}_t$ measurable and therefore

$$M^m(t) = \sum_{0\leq i_1,\ldots,i_m\leq N} a_{i_1}\cdots a_{i_m}\mathbb{E}^{\mathbb{Q}^{T_0}}\left[\prod_{k=1}^{m}P(T_0,T_{i_k})\middle|\mathcal{F}_t\right] \tag{3.4.5}$$

and we will call $\mathbb{E}^{\mathbb{Q}^{T_0}}\left[\prod_{k=1}^{m}P(T_0,T_{i_k})\middle|\mathcal{F}_t\right]$ the *bond moments.*

Similar to Cheng (2013): *pp. 44–46* we will reduce the problem to finding the bond moments for the short-rate model without a determinstic-shift extension by using (3.4.4)

$$\begin{aligned}
M^m(t) &= \sum_{0\leq i_1,\ldots,i_m\leq N} a_{i_1}\cdots a_{i_m}\mathbb{E}^{\mathbb{Q}^{T_0}}\left[\prod_{k=1}^m P(T_0,T_{i_k})\middle|\mathcal{F}_t\right]\\
&= \sum_{0\leq i_1,\ldots,i_m\leq N} a_{i_1}\cdots a_{i_m}\mathbb{E}^{\mathbb{Q}^{T_0}}\left[\prod_{k=1}^m \frac{P^M(0,T_{i_k})}{P^M(0,T_0)}\frac{P^{\text{CIR-}}(0,T_0)}{P^{\text{CIR-}}(0,T_{i_k})}P^{\text{CIR-}}(T_0,T_{i_k})\middle|\mathcal{F}_t\right]\\
&= \left(\frac{P^{\text{CIR-}}(0,T_0)}{P^M(0,T_0)}\right)^m\sum_{0\leq i_1,\ldots,i_m\leq N} a_{i_1}\cdots a_{i_m}\prod_{k=1}^m\frac{P^M(0,T_{i_k})}{P^{\text{CIR-}}(0,T_{i_k})}\mathbb{E}^{\mathbb{Q}^{T_0}}\left[\prod_{k=1}^m P^{\text{CIR-}}(T_0,T_{i_k})\middle|\mathcal{F}_t\right]\\
&= \left(\frac{P^{\text{CIR-}}(0,T_0)}{P^M(0,T_0)}\right)^m\sum_{0\leq i_1,\ldots,i_m\leq N} a^*_{i_1}\cdots a^*_{i_m}\mathbb{E}^{\mathbb{Q}^{T_0}}\left[\prod_{k=1}^m P^{\text{CIR-}}(T_0,T_{i_k})\middle|\mathcal{F}_t\right],
\end{aligned}$$

where $a^*_{i_k}=a_{i_k}\frac{P^M(0,T_{i_k})}{P^{\text{CIR-}}(0,T_{i_k})}$.

Thus, we only have to calculate the bond moments for the CIR- model.

For a numerical implementation, the m-fold sum over all permutations of i_k is unfavorable. Therefore, we rewrite it as follows: By definition, there will always be m coefficients a_{i_k} in the m-fold sum but it is possible to get e.g. a_{i_1} twice, etc. Hence, fixing indices for a_0 up to a_N we can sum over the powers of all occurrences, which have to sum up to m. However, the individual products of the coefficients can appear multiple times as well, e.g. for $m=2, N=2$ summing over all permutations would lead to two times the term a_0a_1, since we encounter $i_0=0,\ i_1=1$ and $i_1=1,\ i_0=0$. Finally, we derive similar to Cheng (2013): *p. 28 Remark 4.2.1* the following expression

$$\sum_{0\leq i_1,\ldots,i_m\leq N} a_{i_1}\cdots a_{i_m}\left(\prod_{k=1}^m P(T_0,T_{i_k})\right)=\sum_{\substack{0\leq k_0,\ldots,k_N\leq N\\ k_0+\cdots+k_N=m}}\frac{m!}{k_0!\cdots k_N!}a_0^{k_0}\cdots a_N^{k_N}\left(\prod_{j=0}^N P(T_0,T_j)^{k_j}\right).$$

Finding this set of indices is known as *subset sum problem*, which is NP-hard but can be solved by e.g. dynamical programming.[6] The interested reader is referred to Curtis and Sanches (2017) for recent developments using a GPU for large subset sum problems. In our case, m will be at most 7 and due to annual payments N will be at most equal to the maximal tenor plus one, i.e. 11, which is considered as a small subset sum problem for which we will utilize a simpler implementation. Even with semi-annual payments a simple implementation with dynamic programming is sufficient, since we will need to calculate the subset sum problems only once and pass it to the calibration procedure.

Now, let us derive the Riccati equation for the bond moments. First of all, notice that the

[6] More details about the implementation can be found on `https://github.com/kevinkamm/CIR--/blob/main/subsetSum.m`.

affine structure of $P(t,T)$ is preserved

$$\begin{aligned}
&\prod_{j=0}^{N} P^{\text{CIR-}}\left(T_0,T_j\right)^{k_j} \\
&\quad = \prod_{j=0}^{N} \left(A_x(T_0,T_j) e^{-B_x(T_0,T_j)x(T_0)} A_y(T_0,T_j) e^{B_y(T_0,T_j)y(T_0)} \right)^{k_j} \\
&\quad = \left(\prod_{j=0}^{N} A_x(T_0,T_j)^{k_j} \right) e^{-\sum_{j=0}^{N} k_j B_x(T_0,T_j)x(T_0)} \left(\prod_{j=0}^{N} A_y(T_0,T_j)^{k_j} \right) e^{\sum_{j=0}^{N} k_j B_y(T_0,T_j)y(T_0)} \\
&\quad =: A_x\left(T_0,\{k_0,\dots,k_N\}\right) e^{-B_x(T_0,\{k_0,\dots,k_N\})x(T_0)} A_y\left(T_0,\{k_0,\dots,k_N\}\right) e^{B_y(T_0,\{k_0,\dots,k_N\})y(T_0)}
\end{aligned} \tag{3.4.6}$$

By Proposition 3.1.7 we have also for $t \leq T_0$

$$\begin{aligned}
&\mathbb{E}^{\mathbb{Q}^{T_0}}\left[\prod_{j=0}^{N} P^{\text{CIR-}}\left(T_0,T_j\right)^{k_j} \middle| \mathcal{F}_t \right] \\
&\quad = \frac{1}{P^{\text{CIR-}}\left(t,T_0\right)} \mathbb{E}^{\mathbb{Q}}\Bigg[e^{-\int_t^{T_0} r^{\text{CIR-}}(s)ds} A_x\left(T_0,\{k_0,\dots,k_N\}\right) e^{-B_x(T_0,\{k_0,\dots,k_N\})x(T_0)} \\
&\qquad\quad A_y\left(T_0,\{k_0,\dots,k_N\}\right) e^{B_y(T_0,\{k_0,\dots,k_N\})y(T_0)} \Bigg| \mathcal{F}_t \Bigg] \\
&\quad \overset{!}{=} \frac{1}{P^{\text{CIR-}}\left(t,T_0\right)} M_x(t,T_0) e^{-N_x(t,T_0)x(t)} M_y(t,T_0) e^{N_y(t,T_0)y(t)}.
\end{aligned}$$

We notice that by martingale pricing the discounted price process $e^{-\int_0^t r^{\text{CIR-}}(s)ds} M_x(t,T_0) e^{-N_x(t,T_0)} M_y(t,T_0) e^{N_y(t,T_0)}$ has to be a martingale. Since it has an affine structure as well, it places us exactly in the same situation as in the derivation of Lemma 3.3.3 seen in Section 3.3 with the difference of variable terminal conditions.

Therefore, we have the same Riccati equation but different terminal values dependent on $k_0,\dots,k_N$.

For generic terminal values $a_z, b_z \in \mathbb{R}_{\geq 0}$ the explicit solution is given by

$$\begin{aligned}
M_z(t,T_0) &= a_z \left(\frac{\phi_1^z \exp\left(\phi_2^z (T_0-t)\right)}{\phi_1^z + \phi_2^z \left(\exp\left(\phi_1^z (T_0-t)\right) - 1\right)\left(1 + b_z\left(\phi_1^z - \phi_2^z\right)\right)} \right)^{\phi_3^z}, &\qquad M_z(T_0,T_0) &= a_z \\
N_z(t,T_0) &= \frac{b_z \phi_1^z + \left(\exp\left(\phi_1^z (T_0-t)\right) - 1\right)\left(1 + b_z\left(\phi_1^z - \phi_2^z\right)\right)}{\phi_1^z + \phi_2^z \left(\exp\left(\phi_1^z (T_0-t)\right) - 1\right)\left(1 + b_z\left(\phi_1^z - \phi_2^z\right)\right)}, &\qquad N_z(T_0,T_0) &= b_z.
\end{aligned} \tag{3.4.7}$$

As seen from our derivation in Equation (3.4.6), the terminal values $t = T_0$ are equal to

$$a_z = A_z\left(T_0,\{k_0,\dots,k_N\}\right), \quad b_z = B_z\left(T_0,\{k_0,\dots,k_N\}\right), \quad z \in \{x,y\},$$

and we can now compute the bond moments and therefore the swap moments for the Gram-Charlier expansion. Thus, using the one-to-one relationship between moments and cumulants in Section B.4, we have an explicit formula for the swap cumulants and we can apply Proposition 3.4.4, which is part of the next subsection.

3.4.3.2. Expansion Formula

As described in Tanaka, Yamada and Watanabe (2010), we can now use Proposition 3.4.4 to formulate the Gram-Charlier expansion formula:

$$\text{Swaption}(t; K, \zeta) = P(t, T_0)\left(C_1\mathcal{N}\left(\frac{C_1}{\sqrt{C_2}}\right) + \sqrt{C_2}\varphi\left(\frac{C_1}{\sqrt{C_2}}\right)\left(1 + \sum_{l=3}^{\infty}(-1)^l q_l H_{l-2}\right)\right)$$

where we replace the c_n in Proposition 3.4.4 by $C_n \coloneqq c_n(t)P(t, T_0)^n$ for $n \geq 1$ and the swap cumulants $c_n(t)$ are derived from the swap moments $M^m(t)$ using their one-to-one relationship shown in Section B.4.

In the following, we will denote by

$$\text{GC}\left(L; K, \zeta\right) \coloneqq P(t, T_0)\left(C_1\mathcal{N}\left(\frac{C_1}{\sqrt{C_2}}\right) + \sqrt{C_2}\varphi\left(\frac{C_1}{\sqrt{C_2}}\right)\left(1 + \sum_{l=3}^{L}(-1)^l q_l H_{l-2}\right)\right)$$

the L-th order of the Gram-Charlier expansion of the $T_0 \times (T_N - T_0)$ swaption with strikes K and swaption type ζ with annual payment dates .

3.4.4. Numerical tests

We will now perform some numerical experiments in our model.[7] In Section 3.4.4.1 we will briefly discuss the market data, which we will use to perform all numerical tests in the subsequent sections. Afterwards, we will describe the calibration procedure of our model in Section 3.4.4.2. This is followed by a short subsection on simulating the model with the Euler-Maruyama scheme in Section 3.4.4.3 and in Section 3.4.4.4 we investigate the par rates of constant maturity swaps (CMS). Last but not least, we compare the model Bermudan swaption prices to Bloomberg's Hull-White one factor model prices in Section 3.4.4.5.

We used for the calculations `Matlab 2022a` with the (Global) Optimization Toolbox running on Windows 10 Pro, on a machine with the following specifications: processor Intel(R) Core(TM) i7-8750H CPU @ 2.20 GHz and 2x32 GB (Dual Channel) Samsung SODIMM DDR4 RAM @ 2667 MHz. All calculations were sped-up by multiprocessing on a single CPU whenever possible.

3.4.4.1. Market Data

To obtain the market zero-coupon bond term-structure, we first build the EUR Euribor-swap curve which is created from the most liquid interest rate instruments available in the market and constructed as follows: We consider deposit rates and Euribor rates with maturity from one day to one year and par-swap rates versus six-month Euribor rates with maturity from two years to thirty years. Then the zero interest curve and the zero-coupon bond curve are calculated using a standard "bootstrapping" technique in conjunction with cubic spline

[7] More details about the implementation can be found on https://github.com/kevinkamm/CIR--/blob/main/main.m.

interpolation of the continuously compounded rate (cf. Miron and Swannell (1991) for more details).

We tested the model at two different dates 30/12/2019 and 30/11/2020, because at 30/12/2019 the market zero rates were partially negative (up to year six) and on 30/11/2020 they were completely negative. This enables us to test the compatibility of the model in the most relevant different scenarios. However, since the results on 30/12/2019 and 30/11/2020 are very similar, we decided to present only the results on 30/12/2019 and make all the data at 30/11/2020 available online to shorten the presentation.

As aforementioned, we will calibrate the model to swaption prices (Table B.6). They are computed by Bachelier's formula from normal volatilities quoted in the market (Table B.2) and the swaption strikes can be found in Table B.4.

After the calibration, we will assess the performance of the model by comparing its prediction of par CMS rates to Bloomberg's CMS rates in Section 3.4.4.4 and pricing Bermudan swaptions. The benchmark for Bermudan swaption prices will be Bloomberg's Hull-White one factor model alongside the corresponding strikes. The values are displayed in Table B.8, Table B.9 and Table B.10, respectively.

All data has been downloaded from Bloomberg and is used in the following subsections for our numerical experiments. We start in the next subsection with calibrating our model to the swaption surface.

3.4.4.2. Calibration

In this subsection, we will discuss how we use the Gram-Charlier expansion to calibrate our model to parts of the swaption surface in Table B.6. Since we are using a deterministic shift extension, a perfect fit to the market zero-coupon curve (see Table B.11) is always guaranteed. Let us denote the parameter vector by $\Pi := [\phi_1^x, \phi_2^x, \phi_3^x, \phi_1^y, \phi_2^y, \phi_3^y, x_0, y_0]^T \in \mathbb{R}^8_{>0}$. We will formulate the calibration procedure as a constraint minimization problem in $\mathbb{R}^8_{>0}$ for the parameters Π with objective function

$$f(\Pi) := \sum_{l \in \mathcal{L}} \sum_{T_0 \in \mathcal{M}} \sum_{T_N \in \mathcal{T}} \left(\frac{\mathrm{MarketSwaption}_{T_0}^{T_N}(K, \zeta)}{\mathrm{GC}_{T_0}^{T_N}(l, \Pi; K, \zeta)} - 1 \right)^2, \tag{3.4.8}$$

where $\mathcal{L} \subset \mathbb{N}$ is a set of natural numbers containing the orders of the Gram-Charlier expansion, $\mathcal{M}$ is a set of maturities and $\mathcal{T}$ a set of final times. We will go into further details how to choose these sets in Remark 3.4.5.

The objective function describes the relative square difference between the market swaption prices and the theoretical prices derived by the Gram-Charlier expansion using the short-rate model (3.4.3). We would like to note that one could also think of different objective functions, which might lead to slightly different results. For instance, we tested absolute squared errors but the results were similar.

The set of admissible parameters $\mathcal{A}$ will be the same as in Section 3.3.3.2, which we will recall briefly for the convenience of the reader: We set

$$A := \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 1 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -2 & 0 & 0 & 0 \end{bmatrix}$$

with boundary conditions $\Pi_i \geq 0$, $i = 1, \dots, 8$, and $\Pi_3 = \phi_3^x \geq 1$, as well as $\Pi_6 = \phi_3^y \geq 1$, leading to the set of admissible parameters

$$\mathcal{A} := \left\{ \Pi \in \mathbb{R}^8_{\geq 0}, \Pi_3, \Pi_6 \geq 1 : A \cdot \Pi \leq 0 \right\}. \tag{3.4.9}$$

Finally, a solution Π^* to the calibration problem is a minimizer of

$$\min_{\Pi \in \mathcal{A}} f\left(\Pi\right). \tag{3.4.10}$$

Before we present some results, we would like to make the following remark on the choices of $\mathcal{L}$, $\mathcal{M}$ and $\mathcal{T}$.

Remark 3.4.5. As always in calibration procedures with parametrized models, there is the notion of over- and underfitting to the data. Overfitting usually occurs when there are more parameters than independent values to calibrate to. For example, we saw a very good fit to a single swaption price. Underfitting on the other hand, occurs when the model is not able to fit to the whole data, e.g. fitting this model to the entire swaption surface.

In our experiments, we determined that 4 up to 6 values performed best with regards to the Bermudan swaption pricing (Section 3.4.4.5) and finding the CMS par rates (Section 3.4.4.4). This is not very surprising, since the model has in total 8 parameters but since the two CIR processes are independent and subtracted to deal with the negative interest rates it has essentially 4 parameters to model the data.

Therefore, we decided to perform tests on columns of the swaption surface and excluded short maturities. Additionally, removing the last maturity in the column from the calibration increased the speed of the optimization with usually the same accuracy. Moreover, we performed tests on several diagonals of the swaption surface with similar results and therefore decided to focus only on columns in this paper.

Another aspect of this calibration procedure is the question which orders to use of the Gram-Charlier expansion. Since it is an orthogonal expansion, there is no a-priori error estimate of the truncated expansion formula. This also means that increasing the order might not be beneficial for the accuracy. Through comparing the Gram-Charlier swaption prices with Monte-Carlo swaption prices (see Table 3.7) using the same parameters, we found both prices to be closer too each other if we were using the order three, five and seven in the calibration procedure. A non-rigorous and heuristic idea behind this reasoning is that if the three orders are close too each other then the expansion “converges” to the correct price of the swaption in a loose sense.

To conclude, to avoid over- and underfitting we will calibrate to columns of the swaption

Table 3.5.: Computational times and values of (3.4.8) using different initial points and different swaption columns in Table B.6 and corresponding strikes Table B.4 in the case of payer swaptions and maturities ranging from 5 to 15.

Tenor Method	1	2	5	7	10
`ga`	$f(\Pi)=$ 3.94e-2 in 76.2 s	$f(\Pi)=$ 7.12e-2 in 85.8 s	$f(\Pi)=$ 5.75e-2 in 100 s	$f(\Pi)=$ 2.27e-2 in 168 s	$f(\Pi)=$ 1.79e-2 in 891 s
`ga` & `fmincon`	$f(\Pi)=$ 3.94e-2 in 76.6 s	$f(\Pi)=$ 7.92e-2 in 87.6 s	$f(\Pi)=$ 6.61e-3 in 118.6 s	$f(\Pi)=$ 1.12e-3 in 206.6 s	$f(\Pi)=$ 8.04e-4 in 945.9 s
I_1 & `fmincon`	$f(\Pi)=$ 7.90e-2 in 0.9 s	$f(\Pi)=$ 4.78e-2 in 0.8 s	$f(\Pi)=$ 6.62e-3 in 2.47 s	$f(\Pi)=$ 1.10e-3 in 52 s	$f(\Pi)=$ 3.00e-4 in 181 s
I_2 & `fmincon`	$f(\Pi)=$ 8.62e-1 in 0.3 s	$f(\Pi)=$ 5.80e-1 in 1.35 s	$f(\Pi)=$ 6.55e-3 in 33.3 s	$f(\Pi)=$ 1.12e-3 in 49.9 s	$f(\Pi)=$ 6.95e-4 in 93.9 s

surface starting with maturity five and ending with maturity 15. Moreover, to have a "stable" Gram-Charlier swaption price we will use the orders three, five and seven in all experiments.

To solve (3.4.10) numerically, we would like to use `Matlab`'s function `fmincon` in the (Global) Optimization Toolbox. In order to use this function, we need an initial guess of the parameter Π and the computational time will depend on that choice.

Our experiments showed that initial guesses with small admissible values worked best for `fmincon`. Therefore, we use the following hand-made parameters as initial points for `fmincon`

$$I_1 := [0.1, 0.095, 0.3, 0.095, 0.1, 0.3, 0.01, 0.01]^T\,, \qquad\qquad I_2 := \frac{1}{2} I_1$$

and compare the performance to parameters found by `Matlab`'s function `ga`. For the algorithms, e.g. the interior point algorithm for `fmincon`, used by `Matlab` we refer the reader to Gilli, Maringer and Schumann (2011), in the context of financial mathematics.

In Table 3.5 we show the value of (3.4.8) after the calibration procedure and its computational time in seconds in the case of a payer swaption at 30/12/2019. We display four different choices of initial points, first of all only using `ga`, second `ga` as an initial point for `fmincon`, third I_1 as initial point for `fmincon` and last but not least I_2 for `fmincon`. We can see that the model fits the swaption values best using columns with larger tenor but the computational time increases as well for all methods. Also we can see that our choices I_1 and I_2 in conjunction with `fmincon` outperforms `ga` with respect to accuracy and it is significantly faster than the combination of `ga` and `fmincon`. Therefore, we will use in the following experiments only `fmincon` with I_1 or I_2 to present the results. In Table 3.6 we show the results of (3.4.10) with initial point I_1 using `fmincon` for reproducibility.

Table 3.6.: Calibrated parameters Π^* using I_1 with `fmincon` and different swaption columns in Table B.6 and corresponding strikes Table B.4 in the case of payer swaptions and maturities ranging from 5 to 15.

Π^* \ Tenor	1	2	5	7	10
ϕ_1^x	0.082	0.114	0.109	0.113	0.118
ϕ_2^x	0.0477	0.0947	0.0846	0.0899	0.092
ϕ_3^x	1.05	1.13	1.99	2	2
ϕ_1^y	0.155	0.0241	0.584	0.00192	0.00741
ϕ_2^y	0.165	0.0521	0.597	0.00851	0.00151
ϕ_3^y	1.33	1.19	1.26	1.78	1.73
x_0	0.000126	0.00147	0.00017	0.000107	0.00151
y_0	0.000128	0.0024	0.0021	0.0991	0.0988

Table 3.7.: Average absolute errors of Monte-Carlo prices compared to Gram-Charlier prices and market prices using the parameters shown in Table 3.6.

Methods \ Tenor	1	2	5	7	10
MC − GC3	4.96e-4	1.17e-3	5.48e-4	6.23e-4	6.18e-4
MC − GC5	2.90e-4	1.07e-4	9.54e-4	2.93e-4	3.57e-4
MC − GC7	7.65e-4	1.05e-3	2.14e-4	1.97e-4	2.86e-4
MC − Market	3.93e-4	8.99e-4	4.58e-4	3.85e-4	3.73e-4

3.4.4.3. Euler-Monte-Carlo Simulation

In order to forecast the future expected interest rate for e.g. pricing Bermudan swaptions in Section 3.4.4.5, we use the same Euler-Maruyama scheme as in Section 3.3.3.3 to simulate the instantaneous spot rate r (3.4.2).

In all of our experiments, we will use $M = 10000$ simulations and mesh size $\Delta = \frac{1}{256}$. On the one hand, looking at the fast calibration times using the Gram-Charlier approximation in Section 3.4.4.2, it is clear that Monte-Carlo methods cannot compete with respect to speed. On the other hand, since the Gram-Charlier expansion has no a-priori error bound let us now validate the calibration results by computing the Monte-Carlo prices with the parameters obtained by the Gram-Charlier expansion in Table 3.6. In Table 3.7 we compare the swaption prices obtained by selected orders of the Gram-Charlier expansion to the Monte-Carlo prices and also the Monte-Carlo prices to the market prices. To compare the prices, we will use an average absolute error, i.e. for $X, Y \in \mathbb{R}^{d_1,d_2}$

$$\|X - Y\| := \frac{1}{d_1 d_2} \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} |X_{ij} - Y_{ij}|.$$

The average absolute error between the Gram-Charlier orders and the Monte-Carlo prices are usually of order 10^{-4} and the Monte-Carlo prices compared to the market prices usually of order 10^{-4}, as well. It is important to note while reading this table that the prices themselves are usually of order 10^{-2}, therefore the accuracy is usually up to two significant

Table 3.8.: CMS rates computed with a calibration using I_2 and `fmincon` to the column with tenor 7 of the payer swaption surface with maturities ranging from 5 to 15.

Effective Date	Tenor	Index	Bloomberg's CMS Rate	Model CMS Rate	Abs Error
0	5	5	0.00145	0.00154	8.91e-05
0	10	5	0.00472	0.00499	0.000273
0	5	10	0.00465	0.0047	4.67e-05
0	10	10	0.00732	0.00738	6.06e-05
3	5	5	0.00562	0.00584	0.000226
3	5	10	0.00824	0.00825	8.32e-06
5	10	5	0.00999	0.01	3.64e-05
5	5	5	0.00958	0.00847	0.00112
5	5	10	0.011	0.0101	0.000912

orders. Hence, this validates the parameters obtained by the calibration with the Gram-Charlier expansion and we can proceed with finding CMS rates in the next subsection using Monte-Carlo techniques.

3.4.4.4. Pricing Constant Maturity Swaps (CMS)

From Section 3.1.3, we know the par CMS rates. In our case, we have $P(0,T) = P^M(0,T)$ by the deterministic shift extension and therefore we can calculate the par CMS rates as follows

$$\begin{aligned} K &= \frac{\mathbb{E}^{\mathbb{Q}}\left[\sum_{i=1}^{N} \alpha_i \exp\left(-\int_0^{T_{i-1}} r(s)ds\right) R_{i-1}^{i-1+c}(T_{i-1})\right]}{\sum_{i=1}^{N} \alpha_i P(0,T_{i-1})} \\ &= \frac{\mathbb{E}^{\mathbb{Q}}\left[\sum_{i=1}^{N} \alpha_i \exp\left(-\int_0^{T_{i-1}} r(s)ds\right) R_{i-1}^{i-1+c}(T_{i-1})\right]}{\sum_{i=1}^{N} \alpha_i P^M(0,T_{i-1})}. \end{aligned}$$

In our experiment, we will use Monte-Carlo simulation for the short-rate (3.4.2) and display the results in Table 3.8 using the initial parameters I_2 for `fmincon` in the case of payer swaptions. In the first column we see the effective date T_0, in the second the tenor T, such that $T_N = T_0 + T$ and in the third column the index c for the CMS. The next column shows Bloomberg's CMS rates, which is followed by the model CMS rates. In the last column we can see the absolute error of market and model rates. We can observe that the majority of CMS rates are very close to each other, telling us that the model performs well on average using just one column of the swaption data for the calibration. Using different columns in the calibration for all different CMS rates would improve the results further.

3.4.4.5. Pricing Bermudan Swaptions

For the definition of Bermudan swaptions and how to price them, we refer the reader to Section 3.1.5.

In Table 3.9 we can see the average absolute error of the Bermudan swaption prices in our model compared to Bloomberg's prices. We used as initial points I_1 and I_2 for `fmincon` in the case of receiver and payer swaptions with different tenors. We can see that the average

Table 3.9.: Average absolute errors of Monte-Carlo Bermudan swaption prices and Bloomberg's HW1 Bermudan swaption prices using the I_1 and I_2 as initial points for `fmincon`.

Tenor Methods	1	2	5	7	10
I_1 & `fmincon` (Payer)	0.0254	0.0129	0.0014	0.0073	0.203
I_2 & `fmincon` (Payer)	0.00196	0.00991	0.00269	0.00279	0.0088
I_1 & `fmincon` (Receiver)	0.948	0.0642	0.0036	0.0021	0.0102
I_2 & `fmincon` (Receiver)	0.00615	0.0149	0.0033	0.0021	0.0071

Table 3.10.: Absolute errors of Monte-Carlo Bermudan payer swaption prices and Bloomberg's HW1 Bermudan swaption prices using the I_1 as initial points for `fmincon` calibrated to the column with tenor equal to 5.

Tenor Maturity	2	5	7	10
1	1.295e-03	5.452e-04	6.337e-04	2.488e-03
3	1.026e-03	6.628e-04	9.348e-04	2.931e-03
5	1.284e-03	1.404e-03	1.629e-04	3.605e-03
7	9.416e-04	1.191e-03	4.314e-06	2.271e-03
10	1.267e-03	1.305e-03	1.603e-03	2.470e-03

errors are very sensitive with respect to the calibrated parameters by looking at the results of I_1 and I_2 for a fixed tenor. Additionally, we notice that usually the results are better, if we choose I_2 as an initial point. The best results on average are found while calibrating to the columns of the swaption surface with tenor 5 or 7. In Table 3.10 we show the absolute errors for the individual payer Bermudan swaptions using I_1 as initial point calibrated to the column with tenor 5 and see an overall good match. Particularly, the column with tenor 7 in Table 3.10 is very accurate.

We focused in this experiment on the average errors only and not on specific Bermundan swaptions. If one desires to do so, there might be better choices which swaption prices to use for the calibration. Usually, the so-called co-terminal swaption prices are used to achieve better results for a specific Bermudan swaption. Since we are satisfied with the average performance of the model, we will not perform these individual tests for the sake of brevity.

3.5. Conclusion

In this chapter, we studied two different short-rate models. First, we considered a simple endogenous model in Section 3.3. We saw a good fit to the market term structure but some difficulties to match the swaption surface. In Section 3.4, we extended this short-rate by applying the deterministic-shift extension. We derived the swaption prices by using the Gram-Charlier expansion in this model and calibrated it to columns of the market swaption surface. The calibration is fast and accurate. Using Monte-Carlo techniques, we obtained close CMS rates compared to Bloomberg's rates. Also compared to Bloomberg's Bermudan swaption prices via the HW1 model, our model performed very well.

At the very end of this thesis in Section 5.2.2, we will discuss some opportunities for future research.

4 Rating Transition Modelling

In this chapter, we will discuss rating transition modelling and apply it to compute credit and debit valuation adjustments (hereafter referred to as CVA and DVA) of a portfolio of trades between two parties having signed a collateral agreement dependent on ratings.

A rating is an indicator of the creditworthiness of an entity. A high rating associates less risk to an entity to not fulfill its financial obligations and a low rating a high risk. Ratings are usually denoted by letters **A**, **B**, . . . , **D**, where **A** denotes the best rating and **D** denotes the worst rating. The rating **D** is special. It means that an entity has defaulted, i.e. it can not fulfill its financial obligation towards a contracting party. In this chapter, we use the terms default and bankruptcy of an entity synonymous, implying that a defaulted company cannot recover from this state.

To keep this presentation as simple as possible in this chapter, we consider only four different ratings: **A**, **B**, **C**, **D** ordered from best to worst rating and identify them by integers $\{1, 2, \ldots, K\}$, whenever it is more convenient. But it is straightforward to use more ratings.

For our main application to collateralized XVA, it is important to model the rating changes of an individual entity or an entire sector on a continuous time scale. This can be done in two different ways. On the one hand, one can define a process X_t, which tells us at each time and trajectory the current rating of a company. The natural state-space of these processes is therefore discrete and the time axis is continuous. On the other hand, one can model the transition probabilities R_t of a sector at each point in time and derive a rating process using these transition probabilities. The state-space of this type of model is then a matrix whose entries are the probability of transitioning from one rating to another starting at an initial time t_0 (usually today) till a future time t. An example of such a $t - t_0$ rating matrix is given in Table 4.1. We can see that the individual rows sum up to one, meaning that all rows are valid probability distributions. These type of matrices are called stochastic for this reason. The last row corresponds to our idealized assumption that a defaulted entity cannot recover, i.e. the default state is absorbing. Rating agencies publish these type of matrices usually once a year for a few time frames. Short-term rating matrices are usually published with time frames of $1, 3, 6, 12$ months and long-term rating matrices with time frames of $1, 2, 3, 5, 10$ years. We see a lot of uncertainty in the historical data published by the agencies increasing with larger time frames, which we will discuss in more detail in Section 4.2.

From \ To	**A**	**B**	**C**	**D**
A	0.9395	0.0566	0.0037	2.7804e-04
B	0.0092	0.9680	0.0211	0.0017
C	6.2064e-04	0.0440	0.8154	0.1400
D	0	0	0	1

Table 4.1.: Example of a one year rating transition matrix.

We recognize two different approaches to rating modelling in the literature which is described in Bielecki, Jeanblanc and Rutkowski (2003): *p. 76 Section 4.12.1 Standing Assumptions* in more details. On the one hand, one can model ratings in an HJM-framework, independently proposed by Bielecki and Rutkowski (2000) and Schönbucher (2003). On the other hand, there are intensity-based models, introduced by the pioneering work of Jarrow, Lando and Turnbull (1997). As this chapter can also be viewed as an intensity approach, let us explain this in more details alongside a short illustration in Figure 4.1.

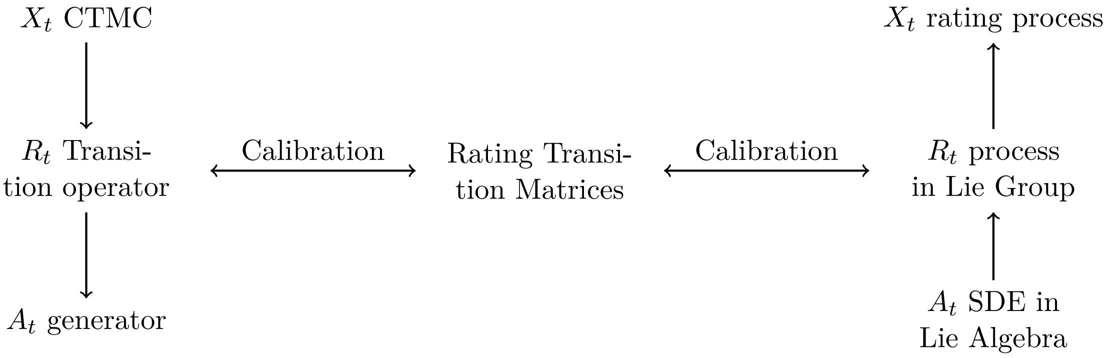


Figure 4.1.: Illustration how our approach compares to the literature.

In the intensity approach (left-hand side in Figure 4.1), usually the rating process X_t is modelled by a continuous-time Markov chain (CTMC). This seems quite natural, because its state space is discrete. Another feature of this approach is that due to the Markovianity, one can describe a CTMC fully by its transition operators R_t. Transition operators tell us for a given initial time and state the probability to transition to another state at a later time. This is exactly, what rating transition matrices describe. Assuming time-homogeneity of the CTMC, it is easy to derive a so-called generator A of the transition operator, which gives a full characterization of the CTMC. This leads to an analytically and numerically tractable model. We will discuss this approach for inhomogeneous CTMCs in Section 4.4 in more detail.

However, in this setting the transition operators and generators are deterministic and in the special case of homogeneous CTMCs (the most common assumption in the literature), the generator is constant. While this makes it possible to calibrate the model directly to the published rating matrices, it limits the possibility for modelling time-dependent features or uncertainty.

In Section 4.5, we show how to model the rating transitions with a stochastic process (right-hand side in Figure 4.1) and notice that generators of CTMCs have values in a suitable subspace of the Lie algebra of stochastic matrices. This allows us to formulate Itô-SDEs taking values in $\mathbb{R}_{\geq 0}$ and to apply a basis transformation to the desired Lie algebra leading to a process A_t. The exponential map, i.e. the matrix exponential, maps the model in the Lie algebra to the proper Lie group of stochastic matrices resulting in a stochastic model R_t. For the calibration, we need to study the distribution of the time series of historical rating matrices, for which we use a TimeGAN and is referred to Section 4.3.

Before we start with modelling rating transitions, let us briefly review the relevant literature first.

4.1. Literature Review

As mentioned in the introduction, this chapter belongs to the intensity-based models. Therefore, let us briefly summarize the available literature on the intensity-based approach, focusing particularly on those results which provide context to our results.

The characteristic of this approach is that the historical and risk-neutral measure are given and the rating model itself is defined under a third measure, which is calibrated to the risk-neutral measure. Usually, Markov processes are used in this framework, in particular CTMCs, whose generator takes the role of the intensities known from default modelling. However, contrary to default modelling, one is not only interested in the first jump time of the process but in all transitions over time from one state to another.

In Jarrow, Lando and Turnbull (1997), the authors propose a continuous-time Markov chain model for the rating process. To tackle the problem of historical versus risk-neutral data, they start with a CTMC under the historical measure and assume that there exists a risk-neutral generator given by $A_t = \operatorname{diag}(\mu_1(t), \ldots, \mu_{K-1}(t), 1)\, A$, where A denotes the generator under the historical measure and $\mu_i(t)$ are positive integrable functions. In the main application the μ_i are assumed to be constants which amounts to assuming that the rating process is a time-homogeneous CTMC. The coefficients μ_i, which are calibrated to credit risky bonds, can be thought as risk-premia relating the historical measure to the risk-neutral measure, although the underlying change of measure is not described explicitly in the paper (a rigorous proof that such a change of measure exists can be found in Bielecki, Crépey and Herbertson (2011): *p. 12 Example 2.9.*).

In Bielecki, Cialenco and Iyigunler (2012), the authors are interested in bilateral CVA under rating triggers, as well, but are focusing on so-called *Close-outs*, namely clauses dictating the termination of the portfolio whenever a given credit rating is reached by one of the parties. To model the rating evolution, they use a Markov copulas for multivariate time-homogeneous Markov chains to include the possibility of calibrating the rating processes of different sectors in a consistent way. We will discuss this issue further in Section 4.7.

For the necessary change of measure they apply, in contrast to Jarrow, Lando and Turnbull (1997), an exponential change of measure technique proposed by Palmowski and Rolski (2002). We will show the implications of both variants in this chapter.

In more recent works of Bielecki, Jakubowski and Niewęgłowski (2015a, 2015b), conditional Markov chains and Markov copulas approaches are proposed and analysed in a more general mathematical framework.

A detailed overview of CTMC approaches can be found in Bielecki, Crépey and Herbertson (2011) in which they also discuss the time-inhomogeneous case briefly. Additionally, in Bielecki and Rutkowski (2004): *pp. 351 ff. Chapter 12 Markovian Models for Credit Migrations* is a detailed overview of further models for credit migrations.

We will compare our results of Section 4.4 mainly to Jarrow, Lando and Turnbull (1997) and Bielecki, Cialenco and Iyigunler (2012). Also, we will elaborate how the different change of measure techniques therein relate to each other.

Like Bielecki, Cialenco and Iyigunler (2012), we are using CTMCs in Section 4.4, but for simplicity we are not using a copula approach, which would be more realistic and is subject to future research. However, we are using a simple inhomogeneous extension instead of time-homogeneous CTMCs. For the calibration they are restricting the space of possible parameters, which will not be necessary in our case and allows for better calibration results.

To the best of our knowledge, the approach of modelling rating transitions starting from an SDE in a appropriate subspace of the Lie algebra of stochastic matrices is completely novel in rating transition modelling. We will discuss this approach in Section 4.5 in more detail. Additionally, the application of a Deep-Neural-Network (DNN) to learn the distribution of historical rating transition matrices seems entirely novel in this community, as well.

4.2. Historical and Market Data

In this section, we will discuss what kind of data is available for the calibration of a rating transition model. We will discuss two different sources of data: First, the historical data in form of so-called *rating matrices*, and second, default probabilities from Credit Default Swap quotes (CDS) in the risk-neutral world. The default probabilities can be considered as a subset of the rating transition data, since it will only represent the last column of the rating transition model after a proper change of measure.

Under the historical measure, we will discuss two different approaches to obtain rating matrices from historical rating data. One is called the *cohort method*, which is the most popular one among rating agencies, and the other method is called *Aalen-Johansen estimation*.

4.2.1. Historical Data

Rating agencies, such as S&P, Moody's and Fitch are required by "Rule 17g-7 of the Securities Exchange Act of 1934"[1] to publish the history of rating changes for some entities. The data set can be downloaded from the websites of the rating agencies and consists of rating histories of individual entities in different sectors, e.g. financial and corporate. We will use the data set from S&P with focus on the corporate sector. The data is structured like follows: for each entity it consists of a list of time stamps when a rating was changed or confirmed. Therefore, we can extract the historical ratings for each individual company for each day.

There are two major methods how to process this data, the Aalen-Johansen estimation and the cohort method, which is part of the next subsection.

4.2.1.1. Cohort Method

The *cohort method* is computed from the aforementioned rating histories as follows (cf. Lando and Skødeberg (2002): *pp.2ff. Equation 1*): Suppose, we have $N_i \in \mathbb{N}$ entities with a rating i at the beginning of the year, $s = 0$. Now, we look at time $t > s$, e.g. the

[1] Please visit https://www.sec.gov/structureddata/rocr-publication-guide.html for more details. Last accessed: 19.05.2022 12:23 CET.

end of the same year, how these entities have changed their rating. We denote by N_{ij} the number of entities, who transition from rating i to rating j and compute the corresponding transition probability as

$$p_{ij}(s,t) := \frac{N_{ij}}{N_i}.$$

However, in practice it can happen for various reasons that $\sum_j N_{ij} < N_i$, which implies that the transition matrices $P(s,t) := (p_{ij}(s,t))_{ij}$ do not have row sums equal to one. One reason why this is possible is the fact that companies can decide themselves that they do not want to be rated anymore by the rating agencies. We call this situation *withdrawal* of an entity.

Therefore, we see in the published rating matrices by the rating agencies that rows usually will not sum up to one and for large times only sum up to around 0.5. A natural question is how to repair these matrices, because for modelling rating transitions it is important that the rating matrices do not loose probability mass over time to use familiar concepts as Markov chains.

This problem is discussed in more details in Israel, Rosenthal and Wei (2001) and a simple reconstruction as in Algorithm 4.1 is recommended for small withdrawal rates.

```
Input  : $R_t^{\mathrm{H}} \in \mathbb{R}^{K,K}$ with row-sums less or equal to 1
Output: $R_t^{\mathrm{A}} \in \mathbb{R}^{K,K}$ with row-sums equal to 1

for $i \leftarrow 1$ to K do
    $\mathsf{wd} \leftarrow 1 - \sum_{j=1}^{\mathsf{K}} \left(R_t^{\mathrm{H}}\right)_{ij}$ ;
    if $\mathsf{wd} > 0$ then
        $\mathsf{y} \leftarrow \left(R_t^{\mathrm{H}}\right)_{i,j=1,\dots,\mathsf{K}}$;
        $\mathsf{y}\ (\mathsf{y} == 0) \leftarrow 1e^{-10}$;
        $\mathsf{b} \leftarrow \frac{\mathsf{y}}{\sum_{j=1}^{K} \mathsf{y}_j} \cdot \mathsf{wd}$;
        $\left(R_t^{\mathrm{A}}\right)_{i,j=1,\dots,\mathsf{K}} \leftarrow \left(R_t^{\mathrm{H}}\right)_{i,j=1,\dots,\mathsf{K}} + \mathsf{b}$
    else if $\mathsf{wd} < 0$ then
        $\left(R_t^{\mathrm{A}}\right)_{i,j=1,\dots,\mathsf{K}} \leftarrow \left(R_t^{\mathrm{H}}\right)_{i,j=1,\dots,\mathsf{K}}$;
        $\left(R_t^{\mathrm{A}}\right)_{i,i} \leftarrow \left(R_t^{\mathrm{A}}\right)_{i,i} + \mathsf{wd}$
    else
        $\left(R_t^{\mathrm{A}}\right)_{i,j=1,\dots,\mathsf{K}} \leftarrow \left(R_t^{\mathrm{H}}\right)_{i,j=1,\dots,\mathsf{K}}$
    end
end
```

Algorithm 4.1: Adjustment of the historical rating matrices.

They also note that in general the data, after repairing it, does not yield a valid generator, which is known as *embedding problem* in the literature. The concept of a Markov chain generator is introduced in Section 4.4 but for the moment the reader may view it as a matrix logarithm of a stochastic matrix.

In general, the embedding problem is still an open problem, which is deeply connected to root finding techniques. The paper by Lencastre et al. (2016) provides a nice overview of the known conditions on the existence of a Markov generator for both the homogeneous and inhomogeneous case. At this point in time, it has only been solved for 2 × 2 and 3 × 3 matrices in general. A condition for 4 × 4 can be found in Casanellas, Fernández-Sánchez and Roca-Lacostena (2021) with a good overview of the existing literature.

Therefore, an adjustment to either the rating matrices or the extracted generator is necessary, which we will come back to in Section 4.4.1.2. Another numerical approach to approximate a generator can be found in Kreinin and Sidelnikova (2001). They use a best approximation technique in a suitable space for transition matrices.

We will discuss at the very end in Chapter 5 another approach for repairing the rating matrices by using the techniques shown in this chapter.

4.2.1.2. Aalen-Johansen Estimator

After extracting these rating trajectories, we apply the so-called Aalen-Johansen estimator (cf. Lando and Skødeberg (2002)) to the processed data to compute the rating transition matrices with a given time span. Let us come back to the data set with the historical ratings and let us set our initial time to the first of January of a specific year and compute the rating transitions over one year to get an average rating transition matrix of one year in the corporate sector.

The *Aalen-Johansen estimator* (cf. Lando and Skødeberg (2002): *pp.8ff. Section 4*) is a non-parametric estimator of the transition probabilities of a time-inhomogeneous continuous-time Markov chain (ICTMC) and we will assume that the historical rating transition data can be modelled by an ICTMC. The rating transition probabilities starting at time s up to time t are then estimated by

$$P(s,t) := \prod_{k=1}^{m} (I + \Delta A(T_k)),$$

where T_k is the jump time in the interval $[s,t]$ and $m \in \mathbb{N}$ is the number of jumps, as well as the estimated generator

$$\Delta A(T_k) := \begin{pmatrix} -\frac{\Delta N_1(T_k)}{Y_1(T_k)} & \frac{\Delta N_{12}(T_k)}{Y_1(T_k)} & \frac{\Delta N_{13}(T_k)}{Y_1(T_k)} & \cdots & \frac{\Delta N_{1K}(T_k)}{Y_1(T_k)} \\ \frac{\Delta N_{21}(T_k)}{Y_2(T_k)} & -\frac{\Delta N_2(T_k)}{Y_2(T_k)} & \frac{\Delta N_{23}(T_k)}{Y_2(T_k)} & \cdots & \frac{\Delta N_{2K}(T_k)}{Y_2(T_k)} \\ \vdots & \vdots & \ddots & \cdots & \vdots \\ \frac{\Delta N_{K-1,1}(T_k)}{Y_{K-1}(T_k)} & \frac{\Delta N_{K-1,2}(T_k)}{Y_{K-1}(T_k)} & \cdots & -\frac{\Delta N_{K-1}(T_k)}{Y_{K-1}(T_k)} & \frac{\Delta N_{K-1,K}(T_k)}{Y_{K-1}(T_k)} \\ 0 & 0 & \cdots & \cdots & 0 \end{pmatrix}.$$

The jump process $\Delta N_{ij}(T_k)$ denotes the number of transitions from rating i to rating j at time T_k and $\Delta N_i(T_k)$ counts the total number of transitions away from rating i at time T_k. The jump process $Y_i(T_k)$ denotes the number of entities with rating i right before time T_k. The last row is zero, because we assume an absorbing default rating. So each time a rating changes in the underlying data, the estimated generator is updated accordingly.

We have seen in Section 4.2.1.1 that the cohort method is not guaranteed to produce stochastic matrices. The Aalen-Johansen estimator overcomes this problem naturally, by updating after each rating change. Therefore, this method ensures that rows sum up to one, which will be important later on.

However, we found that our results from both the cohort method and the Aalen-Johansen estimation differ from the rating matrices which are published by the agencies. We confirmed with S&P that they also use unpublished sensitive rating data and remove correlation structures from data, for which additional knowledge of the entities and their relation towards each other is necessary.

Therefore, the results presented in this chapter serve as an illustration how this methodology can be applied but the underlying data needs some work for an implementation in practice.

4.2.2. Market Data

In the market, Credit Default Swap (CDS) quotes can be found. This means we have some information of the default probabilities of an entity under the risk-neutral measure.

The default probabilities for the different initial ratings are obtained from market CDS spreads. Specifically, we considered the spreads for the the financial sector, without distinguishing by geography. To derive default probabilities from spreads, standard bootstrapping procedures (cf. Brigo and Mercurio (2006): *pp.764 ff. Chapter 22.3*) can be applied.

In particular, financial providers such as ICE[2] compute default probabilities for companies with different initial rating. Similar to the rating agencies, these probabilities are computed in a sector, such as the financial sector or the corporate sector.

Therefore, we can consider them as available market data and we test our numerical implementation with the three different synthetic data sets. The default probabilities in Table C.5 are designed to have slightly elevated probabilities of default compared to the historical ones.

In Table C.6, we keep the default probability for rating **C** but increase the ones for **B** and **C** a bit more. We consider this as the mild case of the three data sets.

Our last data set, Table C.7, serves as a test of robustness with unrealistic high probabilities of default.

4.3. Rating Properties and Rating Matrix Generation

In this section, we will explain how to generate synthetic rating transition matrices from historical data and discuss properties of short-term rating matrices.

In Section 4.3.1, we will give a brief introduction to the relevant Deep-Neural-Network (DNN) architectures, which we will use to generate synthetic rating matrices. This DNN is called TimeGAN and we will build a training data set for this DNN by applying the Aalen-Johansen estimator to the historical ratings seen in Section 4.2.1.

Afterwards, in Section 4.3.2 we will discuss some properties of rating matrices in the training data set and the synthetic data.

[2] Please visit `https://www.theice.com/index` for more details. Last accessed: 15.08.2022 16:28 CET.

4.3.1. TimeGAN

In this section, we show how one can use a generative adversarial network (GAN) for time series data to obtain fake rating transition matrices from paths of a Brownian motion.[3] In particular, we chose a network called TimeGAN by Yoon, Jarrett and Schaar (2019) to learn the rating distributions from the historical data.

The TimeGAN is supposed to learn a function

$$f(t_k, W_{t_k}(\omega)) = R_{t_k}(\omega)$$

mimicking the historical rating matrices $R^{\mathrm{H}}_{t_k}$ for $k = 1, \dots, n$, $n \in \mathbb{N}$. After the learning phase, we can use a path of the Brownian motion W to generate synthetic rating matrices at the points in time t_k.

Training data. We use the technique described in Section 4.2.1.2 to compute rating matrices with time spans of $1, 3, 6, 12$ months starting in 2011 till the end of 2019. For the one month rating matrices, we start at each month in a year and compute the transition probabilities with the Aalen-Johansen estimator till the next month. For the three month rating matrices we proceed similar but starting every three months and so on, such that data is not used twice for the rating matrices with respective time spans. After computing all these matrices we end up with 108 matrices for one month, 36 for three months, 18 for six months and nine for one year. After that, we build a set of time series data by considering all the permutations of the rating matrices leading to a data set of roughly 630000 different time sequences of rating matrices.

We are aware that this approach might raise some eyebrows but rating data is scarce and it is not unusual to assume independence of the rating events which justifies this approach. We will discuss the impact of this choice in Section 4.3.2 further, while studying properties of rating matrices.

Remark 4.3.1. One can alternatively use the rating matrices which are published by the rating agencies from e.g. the last 10 years. However, these are usually only available for long term rating matrices, i.e. 1 up to 10 years. However, as discussed in Section 4.2.1.1, these matrices have imperfections due to the withdrawal of entities. So one idea could be to repair them with an heuristic method and build up a training data set by again considering the permutations of the time series.

The TimeGAN combines an autoencoder with a generative adversarial network using recurrent neural networks linked by a supervising network. We would like to give a short intuition how these networks work together in our case and refer the reader to Yoon, Jarrett and Schaar (2019) for the details.

Autoencoder. For a detailed treatment of Variational Autoencoders (VAE) we refer the reader to Kingma and Welling (2019).

[3] More details about the implementation can be found on https://github.com/kevinkamm/RatingML/blob/main/DeepNeuralNetwork/main.py.

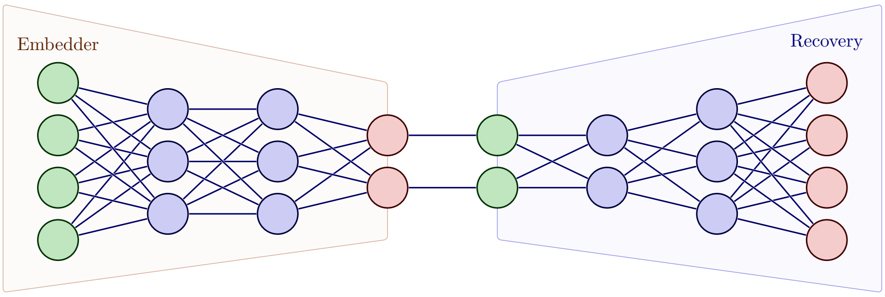


Figure 4.2.: Illustration of a VAE network.

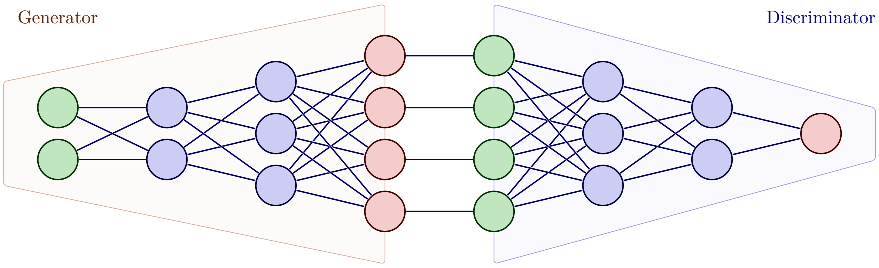


Figure 4.3.: Illustration of a GAN network.

The principle network architecture in an application without time series data is illustrated in Figure 4.2. There are two different networks linked to each other, one called *embedder* or *encoder* and the other one called *recovery* or *decoder*. The idea is to introduce a bottleneck between these networks. This forces the network to learn principle components of the data and helps with denoising as well as dimensionality reduction. For the training phase, the data is first embedded, recovered and afterwards compared to the original data to minimise the difference of both. After the training phase, the recovery network can be used to generate rating matrices from their embedded features. We will see how the generator network of the generative adversarial network can be used to generate fake features in the next two paragraphs.

Generative Adversarial Network. For a detailed treatment of Generative Adversarial Networks (GAN) we refer the reader to Goodfellow (2017).

The principle network architecture in an application without time series data is illustrated in Figure 4.3. There are two different networks linked to each other, one called *generator* and the other one called *discriminator*. The idea is to play these networks against each other. The generator network has a few random numbers as input and outputs fake data. The discriminator network will get the fake data from the generator as an input, as well as the real data. Then it is learning to distinguish between fake and real data by outputting a probability of the data being real. Since we know which of the input data is fake and which is real we can optimize the prediction of the discriminator network. The generator on the other hand is learning how to fool the discriminator, i.e. making it believe that the fake data point was real. After the learning phase and when the discriminator is not very confident anymore in distinguishing between fake and real, the generator network can be used to produce synthetic data.

Supervisor. The supervisor network does not have a special network architecture and it is placed in-between the embedder and recovery network, as well as between the generator and the discriminator network to establish a link between them. This makes it also possible in the training of the entire network to compare the supervised networks to the unsupervised networks. Another implication of this approach is that the generator network of the GAN is not generating the rating matrices directly but the features of the rating matrices. As aforementioned, combining the trained generator with the trained recovery network will enable us to generate synthetic rating matrices.

Recurrent networks. For a detailed treatment of Recurrent Neural Networks (RNN) and a comparison of Long-Short-Term-Memory (LSTM) to Gated-Recurrent-Units (GRU) we refer the reader to Chung et al. (2014).

So far, we have discussed how the supervised VAE and GAN can be used together at a single point in time to generate synthetic rating matrices. RNNs enable us to use time series data and all the aforementioned networks are augmented with GRUs in our implementation to take the time series of rating matrices into account. GRUs consist of two different gates. One is called the *update gate* and the other one is called *forget gate*. The update gate decides how much of the new temporal information is added to the time sequence. The forget gate has the possibility to forget the previous times in the time sequence, making the current point in time independent of the past.

Hyperparameters and network architecture. It is not the purpose of this paper to "over-optimize" the procedure, since it is a first step using these modern techniques for rating transitions. Additionally, for its next use case of rating triggers, an additional source of market data will be available and the current architecture might need some adjustments. We leave it up to the reader to change the hyperparameters and network architectures, because we are satisfied with the performance of the current setting, which is discussed in Section 4.3.2 in greater detail. We chose the following settings for our experiments.

1. We used 40 epochs in total and noticed that 10 epochs take roughly 1 hour in the training step.
2. We found that a batch size of 128 was a good middle-ground between speed and realistic rating matrices.
3. For the embedder we used three GRU layers. The first and last with 3 units and the second one with 2 units. The output dense layer has 4 units and a sigmoid activation function.
4. For the recovery we used three GRU layers. The first and last with 3 units and the second one with 2 units. The output dense layer has $K^2 = 16$ units and a sigmoid activation function.
5. For the supervisor we used two GRU layers, each with 4 units. The output dense layer has 4 units and a sigmoid activation function.

6. For the generator we used three GRU layers, each with 4 units. The output dense layer has 4 units and a sigmoid activation function. As an input we take the values of a Brownian path at $t = 1, 3, 6, 12$ months.
7. For the discriminator we used three GRU layers, each with 4 units. The output dense layer has a single unit and a sigmoid activation function.
8. All optimizers were Adam (cf. Kingma and Ba (2014)) with the standard learning rate $1e - 4$.

As aforementioned, for the training of the network we refer the reader to Yoon, Jarrett and Schaar (2019) and note that we used the standard loss functions indicated in this paper to obtain a purely data driven network.

4.3.2. Rating Properties

To estimate the quality of the TimeGAN, we observed from the historical data that short term rating matrices up to one year should have the following properties:

1. It is more likely to stay in the initial rating than changing to another: This means rating matrices are strongly diagonally dominant, i.e. for $i = 1, \dots, K$
$$[R_t(\omega)]_{ii} \geq \sum_{j \neq i} [R_t(\omega)]_{ij}. \tag{4.3.1}$$
2. Downgrading is more likely than upgrading: This means that the sum of the upper triangular matrix is bigger than the sum of the lower triangular matrix, i.e.
$$\sum_{i<j} [R_t(\omega)]_{ij} \geq \sum_{i>j} [R_t(\omega)]_{ij}. \tag{4.3.2}$$
3. Lower rated entities are more likely to default: This means that the default column is increasing from best starting rating to lowest, i.e.
$$[R_t(\omega)]_{1K} \leq [R_t(\omega)]_{2K} \leq \cdots \leq [R_t(\omega)]_{KK}. \tag{4.3.3}$$
4. The rating spreads more over time: We measure this by looking for decreasing diagonal elements, i.e. for all $s < t$ and all $i = 1, \dots, K$
$$[R_s(\omega)]_{ii} \geq [R_t(\omega)]_{ii}. \tag{4.3.4}$$

These properties are not strict in the sense that they can be violated on some occasions. Moreover, one might think of other properties for rating matrices. Also for long term rating matrices (more than 1 year) these properties might not hold true anymore. This makes it very hard to define rigorous conditions for rating matrices in general and is subject to future research and economical validation.

In Table 4.2, we can see a summary of the rating properties (4.3.1)–(4.3.4) for the training data set. The numbers represent the percentages of time-sequences satisfying the conditions averaged over all initial ratings. For the rating spreads over time, we consider time steps from 0 to 1 month, 1 to 3, 3 to 6 and 6 to 12 and write down the percentages for $t = 1, 3, 6, 12$ respectively. We can see that all of the rating matrices in the training data set were strongly diagonally dominant and nearly all had monotone increasing default columns.

The majority of the rating matrices put more emphasis on downgrading for time spans between one month and six months, while for one year all of them satisfied the condition.

For the increasing rating spread we see the biggest violations of the property. This is most likely due to the fact that we consider all permutations of the data. It might be beneficial to filter these sequences out of the training set.

Table 4.2.: Rating properties for training data. Average percentage of the time series fulfilling the conditions (4.3.1)–(4.3.4).

Time in months	Strongly diagonally dominant (4.3.1)	Downgrading is more likely (4.3.2)	Monotone default column (4.3.3)	Increasing rating spread (4.3.4)
1	100 %	87.96 %	100 %	100 %
3	100 %	97.22 %	99.9 %	85.81 %
6	100 %	94.44 %	100 %	83.18 %
12	100 %	100 %	100 %	90.53 %

In Table 4.3, we see exactly the same table for TimeGAN using $M = 12000$ synthetic time-sequences. Even though we did not impose any hard constraints, e.g. that rows must sum up to one, the DNN learned the conditions (4.3.1)–(4.3.4) very well, as well as that rows must sum to one. The only criterion which was not always satisfied was again (4.3.4) but less severe than for the training data. Since these properties are almost always satisfied, we did not optimize the hyperparameters or network architecture any further.

Table 4.3.: Rating properties for TimeGAN with $M = 12000$. Average percentage of the time series fulfilling the conditions (4.3.1)–(4.3.4) and average row sums.

Time in months	Strongly diagonally dominant (4.3.1)	Downgrading is more likely (4.3.2)	Monotone default column (4.3.3)	Increasing rating spread (4.3.4)	Average row sums
1	100 %	100 %	100 %	100 %	0.9999
3	100 %	100 %	100 %	100 %	0.9996
6	100 %	100 %	100 %	93.2 %	1.0002
12	100 %	100 %	100 %	93.33 %	1.0017

4.4. Deterministic Rating Transitions

We begin this section by describing the features making CTMCs a good choice for a rating model. First, it is a natural choice to consider a discrete state-space consisting of all ratings and a continuous-time framework. Moreover, rating transition probabilities have to be time- and state-dependent to reflect the different intensities associated with the different rating-changes over time. About this dependency, two aspects must be considered (see Schönbucher (2003): *Section 8.2* for more details): time homogeneity and the Markov property. Both assumptions are not completely realistic although time homogeneity can be seen as a stronger restriction than the Markovianity assumption.

As far as calibration is concerned, it is desirable for the model to be consistent with all the available data, namely both the historical rating transition matrices and the market quoted credit spreads (either coming from bonds or from credit default swaps). For this to be feasible, one needs a change of measure formula, preferably preserving some of the features of the model (for instance, Markovianity).

Finally, numerical and/or analytical tractability of the model is certainly a fundamental need. This led us to consider Markov processes for the rating process. CTMCs are then a natural choice in view of the above (continuous-time, a discrete state-space and Markovianity). As far as time dependence is concerned, we propose a simple non-homogeneous CTMC model, namely a piecewise-homogeneous CTMC (PHCTMC).

Let us briefly recall the relevant terminology of ICTMCs
(cf. Bielecki and Rutkowski (2004): *pp. 326 ff. Chapter 11 Markov Chains*). An ICTMC is a Markov process $(X_t)_{t\in[0,T]}$ with a discrete state space S. We will only consider the case of a finite state space, i.e. $S = \{1, \dots, K\}$. The functions $p_{ij}(s,t) := \mathbb{P}\left(X_t = j \middle| X_s = i\right)$ are called *transition functions* and their time-derivative from above is called *generator* of the ICTMC, i.e. $A_{ij}(t) := \lim_{h\downarrow 0} \frac{p_{ij}(t,t+h)-\delta_{ij}}{h}$, $i, j = 1, \dots, K$ and δ_{ij} is the Dirac-delta. We have the following immediate properties of generator: $A_{ij}(t) \geq 0$ for all $i \neq j$ and $A_{ii}(t) = -\sum_{j\neq i} A_{ij}(t)$.

For the change of measure we have the following result from Ding and Ning (2021): *pp. 13–15 Section 3 The Equivalent Martingale Measure* or Bielecki and Rutkowski (2004): *pp. 334 ff. Chapter 11.2.5 Change of Probability Measure*:

Theorem 4.4.1. *Let $(X_t)_{0\leq t\leq T}$ be an ICTMC on the probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t), \mathbb{P})$ taking values in $S := \{1, \dots, K\}$ with generator $A_t^{\mathbb{P}} := \left(A_{i,j}^{\mathbb{P}}(t)\right)_{i,j=1,\dots,K}$. Define*

$$L_t^\kappa := \exp\left(-M_t\right) \prod_{0<u\leq t} \left(1 + \sum_{i,j=1}^{K} \kappa_{ij}(u)\left(H_{ij}(u) - H_{ij}(u-)\right)\right),$$

where

1. *the stochastic processes $\kappa_{i,j}(t)$ are bounded, real-valued and $\mathbb{F}$-predictable, such that $\kappa_{ij}(t) > -1$ and $\kappa_{ii}(t) = 0$;*
2. *the number of jumps from rating i to j are $H_{ij}(t) := \sum_{0\leq u\leq t} \mathbb{1}_{X_{u-}=i} \mathbb{1}_{X_u=j}$;*
3. *$M_t := \int_0^t \sum_{i,j=1}^{K} \kappa_{ij}(u) A_{i,j}^{\mathbb{P}}(u) \mathbb{1}_{X_u=i} du$.*

Then L_t^κ is a strictly positive martingale under $\mathbb{P}$ satisfying $\mathbb{E}^{\mathbb{P}}\left[L_T\right] = 1$ and the equivalent probability measure $\mathbb{Q}^\kappa$ given by

$$\left.\frac{d\mathbb{Q}^\kappa}{d\mathbb{P}}\right|_{\mathcal{F}_t} = L_t^\kappa$$

$\mathbb{P}$-almost surely, is well-defined.

Furthermore, X_t is an ICTMC under $\mathbb{Q}^\kappa$ as well with generator

$$A_{i,i}^\kappa(t) = -\sum_{i\neq j} A_{i,j}^\kappa(t)$$
$$A_{i,j}^\kappa(t) = \left(1+\kappa_{i,j}(t)\right) A_{i,j}^{\mathbb{P}}(t).$$

We can see that for any admissible family $\kappa \coloneqq \left(\kappa_{ij}\right)_{i,j=1,\ldots,K}$ we get a valid change of measure and know the generator of the ICTMC after the change of measure as well.

As aforementioned, there are two major examples of this change of measure in the rating community.

Example 4.4.2. On the one hand, if we set $\kappa_{ij}(t) \coloneqq h_i(t)-1$ for strictly positive deterministic functions $h_i(t) > 0$, $i \neq j$, we recover the change of measure by Jarrow, Lando and Turnbull (1997), which we will call JLT change of measure from now on.

On the other hand, if we set $\kappa_{ij}(t) \coloneqq \frac{h_i(t)}{h_j(t)} - 1$ we get a time-inhomogeneous version of the exponential change of measure technique by Palmowski and Rolski (2002), which we will call exponential change of measure throughout the entire chapter.

The reason for choosing only K different functions h_i is due to the fact that after the change of measure we know only data with respect to the default column, and there seems to be no meaningful way to use an entire matrix κ_{ij}.

With this change of measure formula, we can calibrate the model (see Section 4.4.1.2) under the historical measure $\mathbb{P}$ and risk-neutral measure $\mathbb{Q}$ by finding the appropriate functions κ, such that $\mathbb{Q}^\kappa$ and $\mathbb{Q}$ are close with respect to the default probabilities.

4.4.1. Numerical Implementation of ICTMCs

After establishing the change of measure in Theorem 4.4.1 for the general case of ICTMCs, we will now consider piecewise homogeneous CTMCs (PHCTMC) X in this section.[4] To be more precise, let $T_0 \in [0,T]$ be the initial time and T_k, $k = 1,\ldots,n$ be the points in time when historical rating matrices—denoted by R_t^{H}—are available in an increasing order with $T_n = T$, then $X_\cdot$ is assumed to be homogeneous on each $[T_k, T_{k+1})$ $k = 0,\ldots,n-1$.

We used for the calculations `Matlab 2022a` with the (Global) Optimization Toolbox running on Windows 10 Pro, on a machine with the following specifications: processor Intel(R) Core(TM) i7-8750H CPU @ 2.20 GHz and 2x32 GB (Dual Channel) Samsung SODIMM DDR4 RAM @ 2667 MHz.

[4] More details about the implementation can be found on https://github.com/kevinkamm/LieRatingTriggers/blob/main/ICTMC/main.m.

This section is organized as follows: First, we have a look at the rating matrices generated by the TimeGAN and a minor adjustment to the data in Section 4.4.1.1. Then, we will calibrate a PHCTMC to the market data in Section 4.4.1.2 and we compare the exponential change of measure to the JLT change of measure in Section 4.4.2.

4.4.1.1. Historical and Market Data

Appendix C.1 contains the collection of the rating matrices under the historical measure and default probabilities under the risk-neutral measure, which we use in this section.

To be more precise, we will use the expected rating transition matrices generated by the TimeGAN explained in Section 4.3.1 as our historical data. For this, we generated 10000 rating matrices and took the mean. The results are reported in Table C.1–C.4 for the one, three, six and twelve month transition matrices.

Since these matrices have small imperfections in terms of row sums, which can be seen in Table 4.3, we apply Algorithm 4.1 to them to derive valid rating transition matrices. This adjustment can be considered as minor, since the imperfections are very small but it is important for extracting a valid generator.

For the default probabilities, we use artificial data to illustrate the behaviour of the model in a variety of different situations. In particular, the default probabilities in Table C.5 are designed to be close to the default probabilities under the historical measure, in Table C.6 we have larger default probabilities for the two best ratings and in Table C.7 we consider an unrealistic case of very high default probabilities to judge the robustness of the model.

4.4.1.2. Calibration

The calibration procedure has the following two steps: First, we have to extract the generator of the TimeGAN data and then calibrate the ICTMC using the previously obtained generators for the change of measure in Theorem 4.4.1 to the default probabilities Table C.5–C.7.

Let us first of all fix some notations throughout this section: R_t^{A} will be the adjusted rating matrix with row-sum equal to one and A_t will denote the approximation of the generator of R_t^{A}.

As discussed in Section 4.2.1.1, not all given datasets yield a valid generator or there are numerical errors, which will the case for our dataset. This means even though in theory there would exist a unique generator, which we compute by taking the matrix logarithm (cf. Al-Mohy and Higham (2012) and Al-Mohy, Higham and Relton (2013)) of R_t^{A}, the numerical scheme to compute a matrix logarithm can result in small negative off-diagonal values. Therefore, another repair is needed, which is described in Algorithm 4.2. First, we set the diagonal and any negative entries of the matrix logarithm of R_t^{A} to zero. For a justification of setting the negative entries to zero we refer to Israel, Rosenthal and Wei (2001): *p. 6 Section 3 The Non-Negativity Condition*. Then, we sum up the rows and set the new diagonal to the negative value of the sums.

A more sophisticated approach is presented in Kreinin and Sidelnikova (2001) by using a best-approximation approach in a suitable space.

```
Input   : $R_t^A \in \mathbb{R}^{K,K}$
Output: $A_t \in \mathbb{R}^{K,K}$ approximated generator of $R_t^A$
$\mathsf{A}_t \leftarrow \texttt{logm}\left(R_t^A\right)$;
$\mathsf{A}_t\left(\mathsf{A}_t < 0\right) \leftarrow 0$;
for $i \leftarrow 1$ to $\mathsf{K}$ do
    $\left(\mathsf{A}_t\right)_{ii} \leftarrow -\sum_{j \neq i}\left(\mathsf{A}_t\right)_{ij}$;
end
$A_t \leftarrow \mathsf{A}_t$;
```

Algorithm 4.2: Approximation of the historical generators.

For our particular dataset obtained by the TimeGAN the first adjustment in Algorithm 4.1 and the second adjustment in Algorithm 4.2 are negligible. As aforementioned, the first one only corrects small errors of the DNN and the second one only corrects the numerical errors of the matrix logarithm in our case.

These techniques could also be applied to rating matrices published by rating agencies. First, Algorithm 4.1 can be applied to remove the withdrawal column and Algorithm 4.2 ensures a valid generator. The larger the withdrawal resulting from the cohort method, the larger the impact of those two adjustments. We performed some tests on short term rating matrices published by Fitch in Kamm (2022) and discussed this issue in more details.

Calibrating the PHCTMC Now, we want to calibrate the PHCTMC simultaneously under historical measure to the adjusted rating data R_t^{A} and under the risk-neutral measure to the default probabilities $\mathrm{PD}(t) \in \mathbb{R}^{K,1}$ for each initial rating.

Let us start under the historical measure $\mathbb{P}$.

As aforementioned, let $T_0 \in [0,T]$ be the initial point and T_k, $k = 1,\dots,n$ be the points in time, when rating matrices are available in an increasing order with $T_n = T$, then $X.$ is assumed to be homogeneous on each $[T_k, T_{k+1})$ $k = 0,\dots,n-1$.

Now, by the Chapman-Kolmogorov equation we get

$$R^{\mathbb{P}}_{T_0,T} = R^{\mathbb{P}}_{T_0,T_1} \cdot R^{\mathbb{P}}_{T_1,T_2} \cdots R^{\mathbb{P}}_{T_{n-1},T_n} = \prod_{k=1}^{n} R^{\mathbb{P}}_{T_{k-1},T_k}. \tag{4.4.1}$$

By homogeneity on each sub-interval we know that the evolution system will reduce to a semigroup and its generator will be time-constant with an explicit formula

$$R^{\mathbb{P}}_{T_{k-1},t} = R^{\mathbb{P}}_{t-T_{k-1}} = \exp\left(A^{\mathbb{P}}_k\left(t - T_{k-1}\right)\right), \quad t \in [T_{k-1}, T_k). \tag{4.4.2}$$

Hence, to extract these generators from the TimeGAN data, we solve

$$R^{\mathbb{P}}_{T_0,T_k} \overset{!}{=} R^{\mathrm{A}}_{T_k},$$

which is by (4.4.2) under the assumption that $R^{\mathbb{P}}_{T_0,T_{k-1}}$ is invertible and $\left(R^{\mathbb{P}}_{T_0,T_{k-1}}\right)^{-1} \cdot R^{\mathrm{A}}_{T_k}$

has a matrix logarithm equivalent to

$$A_k^{\mathbb{P}} = \frac{\log\left(\left(R^{\mathbb{P}}_{T_0,T_{k-1}}\right)^{-1} \cdot R^{\mathrm{A}}_{T_k}\right)}{T_k - T_{k-1}}, \qquad R^{\mathbb{P}}_{T_0,T_{k-1}} = \prod_{l=1}^{k-1} R^{\mathbb{P}}_{T_{l-1},T_l}.$$

This yields an iterative scheme on the $\mathbb{P}$-side and we will use the same procedure described in Algorithm 4.2 to retrieve valid generators at each time.

On the risk-neutral side we proceed similarly. The generator on the $\mathbb{Q}$-side is assumed to have the same properties as the one on the $\mathbb{P}$-side, hence the change of measure formula in Theorem 4.4.1 will be piecewise constant on each sub-interval, as well, such that we solve a minimization problem at each time T_k by finding the appropriate values $1 + \kappa_{ij} \in \mathbb{R}_{>0}$.

Additionally, let us assume that the default state is absorbing, i.e. we have $e_K^T R_t^{\mathrm{A}} e_i = \delta_{Ki}$, $i = 1, \dots, K$, $t \geq 0$, and therefore $A_t e_i = 0$. Hence, we can choose the values corresponding to the last row $\left(\kappa_{K,j}\right)_j$ freely in each interval and will assume $\left(\kappa_{K,j}\right)_j \equiv 0$ without altering the following minimization problem for finding the right change of measure:

$$\min_{\substack{1+\kappa_{ij}\in\mathbb{R}_{>0},\\ \left(\kappa_{Kj}\right)_j=0,\\ A_k\in\mathcal{A}}} \left\| R^{\mathbb{Q}^\kappa}_{T_0,T_{k-1}} \cdot \exp\left(A_k^\kappa \left(T_k - T_{k-1}\right)\right) \cdot e_K - \mathrm{PD}\left(T_k\right) \right\|_{\mu_{\mathbb{Q}}} + \left\| A_k - A_k^{\mathbb{P}} \right\|_{\mu_{\mathbb{P}}}, \tag{4.4.3}$$

$$\mathcal{A} := \left\{ A \in \mathbb{R}^{K,K} : \text{ for all } i,j = 1,\dots,K \;\; A_{K,j} = 0,\;\; A_{i,j} \geq 0 \text{ for } i \neq j \text{ and } A_{i,i} \leq 0 \right\}.$$

$$A_k^\kappa := \begin{cases} \left(A_k^\kappa\right)_{i,i} & = -\sum_{i\neq j} \left(A_k^\kappa\right)_{i,j} \\ \left(A_k^\kappa\right)_{i,j} & = \left(1+\kappa_{i,j}\right)\left(A_k^\kappa\right)_{i,j}. \end{cases}$$

$$R^{\mathbb{Q}^\kappa}_{T_0,T_{k-1}} = \prod_{l=1}^{k-1} R^{\mathbb{Q}^\kappa}_{T_{l-1},T_l}.$$

The norms $\left\|\cdot\right\|_{\mu_{\mathbb{Q}}}, \left\|\cdot\right\|_{\mu_{\mathbb{P}}}$ are weighted norms and defined as follows: Let $X \in \mathbb{R}^{K,K}$, $x \in \mathbb{R}^{K,1}$ and $M^{\mathbb{P}} \in \mathbb{R}^{K,K}_{\geq 0}$, as well as $m^{\mathbb{Q}} \in \mathbb{R}^{K,1}_{\geq 0}$, we set

$$\begin{aligned} \left\|X\right\|_{\mu_P} &:= \left\| M^{\mathbb{P}} \odot X \right\|_F, \\ \left\|x\right\|_{\mu_Q} &:= \left\| m^{\mathbb{Q}} \odot x \right\|_2, \end{aligned}$$

where $\odot$ denotes the elementwise or Hadamard product. The idea of using weighted sums in the calibration procedure is explained in Remark 4.4.3 in more detail.

Moreover, after this iterated calibration, the whole right-generator of the ICTMC is then defined as $A_t^P = \sum_{k=1}^n A_k^P \mathbb{1}_{[T_{k-1},T_k)}(t)$, $P = \mathbb{P}, \mathbb{Q}$ and to evaluate the evolution system one has either to use (4.4.1) or solve the Kolmogorov forward equation with the inhomogeneous generator A_t^P.

Remark 4.4.3. Let us explain why we decided to include the generator under the measure $\mathbb{P}$ in the calibration procedure.

Having now two different components in the objective function, we added the possibility to add weights corresponding to the user's trust in the data, e.g. they may depend on liquidity and the size of the data set from which the data is obtained. If the data is assumed to be trustworthy, one can choose the individual weight parameter to be large, which ensures a better fit in this particular entry.

Another aspect for using this penalized calibration procedure, which is explained in more detail in KAMM (2022), is the following: If we would use rating matrices by the rating agencies directly and adjust them with Algorithm 4.1 and Algorithm 4.2, then this adjustment has been done in an arbitrary way and impacts the final results. In the case, where $M^{\mathbb{P}} \equiv +\infty$, i.e. the calibration is without the generator under $\mathbb{P}$, we got in KAMM (2022) peculiar results after the change of measure. Sometimes entire columns of the resulting rating matrices were zero, which does not make sense, since companies with the best rating usually have the highest probability to stay in their rating. Therefore, we added the generator under $\mathbb{P}$ to the calibration procedure to mitigate the effects of the initial choice of data reconstruction leading to a significant improvement and making the model feasible for a direct application to historical data.

Remark 4.4.4. We tried different versions of (4.4.3). In particular, we adjusted A_k by Algorithm 4.2 in the objective function but this led to a decrease in accuracy and increase in computational time. Moreover, we added the condition that the sum of each row in the generator is supposed to be zero to the constraints, which led to badly conditioned matrices and unpredictable behaviour in the calibration algorithm.

Therefore, we decided to leave it unconstrained but keep the bounds on the matrix entries and adjust the outcome of the calibration for the generator under $\mathbb{P}$ by Algorithm 4.2, which leads to the presented results.

To make this more precise, we did not add the constraint to $\mathcal{A}$ that all rows must sum to zero. Therefore, the matrix A_k as an output of the calibration procedure is not necessarily a valid generator but we use Algorithm 4.2 after the calibration procedure to repair it. Let us denote this adjustment $\widetilde{A}_k$ for the moment.

The average error of this final adjustment $(\frac{1}{K^2}\sum_{k,i,j}\left|\widetilde{A}_k - A_k\right|)$ after the calibration procedure was of magnitude 10^{-4}, when summing the absolute values of the differences of all entries and dividing by the number of entries, i.e. the error of this adjustment is negligible and makes the calibration algorithm more robust and justifies why we did not include the "sum to zero"-constraint in $\mathcal{A}$.

In particular, these choices make it possible to implement the calibration procedure as a weighted non-linear least squares problem.

We used `Matlab 2022a`'s function `fmincon` to solve the minimization problem with bounds (4.4.3) and display the corresponding errors in Table 4.4. For the errors we used the Frobenius norm to compare the resulting rating matrices under $\mathbb{P}$ to the adjusted rating matrices and the Euclidean norm to compare the resulting probabilities of default with the exponential change of measure to the market data under $\mathbb{Q}$. We divided both norms by the number of entries, i.e. K^2 for the Frobenius norm and K for the Euclidean norm. The total computational time for the calibration was 1.38 seconds and we can see a good fit to the data under both measures.

Table 4.4.: Calibration errors using rating matrices Table C.1–C.4. First row is the error of `fmincon`, second the mean error of the model rating transitions and adjusted market rating transitions and the third row contains the errors of the model and probabilities of default in Table C.6 using the exponential change of measure.

Time / Error	$t=\frac{1}{12}$	$t=\frac{3}{12}$	$t=\frac{6}{12}$	$t=1$
`fmincon`	$1.7e-06$	$1.21e-06$	$2.95e-06$	$2.94e-06$
$\frac{1}{K^2}\left\|R_t^{\mathbb{P}}-R_t^A\right\|_{\mathbb{R}^{K,K}}$	$7.17e-06$	$2.55e-06$	0.000144	0.00087
$\frac{1}{K}\left\|R_t^{\mathbb{Q}}e_K-\mathrm{PD}(t)\right\|_{\mathbb{R}^K}$	$4.89e-05$	$1.89e-05$	$1.35e-05$	$8.76e-05$

For the other probabilities of default and the JLT change of measure, we saw similar good results and discuss the model performance in more details in Section 4.4.2.

4.4.2. Comparison of Exponential and JLT Change of Measure

We already discussed in Section 4.4.1.2 that the calibration errors seen in Table 4.4 were very good in all cases.

However, to judge the performance of the model and the chosen change of measure, it is important to look at the entire evolution of the rating matrices under the risk-neutral measure. This we can see in Figure 4.4–4.6. Each figure contains sub figures of the entries in the rating transition matrices. For example, the sub figure in the upper left corner of each figure describes the transition from **A** to **A** from today till $T = 1$. The last row of the rating matrices is excluded, since its constant. All the figures contain the two different changes of measure. The blue bold line corresponds to the exponential change of measure and the red dashed line to the JLT change of measure.

Let us focus on Figure 4.4. This is the case of similar probabilities of default under the historical and risk-neutral measure. We expect that nothing extreme happens, meaning that the rating matrices under the risk-neutral measure should look similar to the ones under the historical measure in Table C.1–C.4. If we compare the value in the upper left corner at $T = 1$ to the rating matrix under $\mathbb{P}$ in Table C.4, we can see that the exponential change of measure performs much better than the JLT change of measure. It seems that the JLT change of measure overestimates the impact on the rating transitions apart from the default column. However, both changes of measures are very close to each other except for the transition from **A** to **A** and look like valid options for a change of measure.

In Figure 4.5 we can see that in the row, both changes of measure behave very different from each other. As in Figure 4.4, the JLT change of measure alters the rating matrix apart from the default column very much and the values of the transitions do not match the ones under the measure $\mathbb{P}$ at all.

Let us focus on the third sub figure in the second row for the moment. We can see the transition from **B** to **C**. Both changes of measure are showing a large deviation from the values under the measure $\mathbb{P}$. This leads to less entities staying in their starting rating.

However, for this case, where we increased the default probability quite a lot compared to

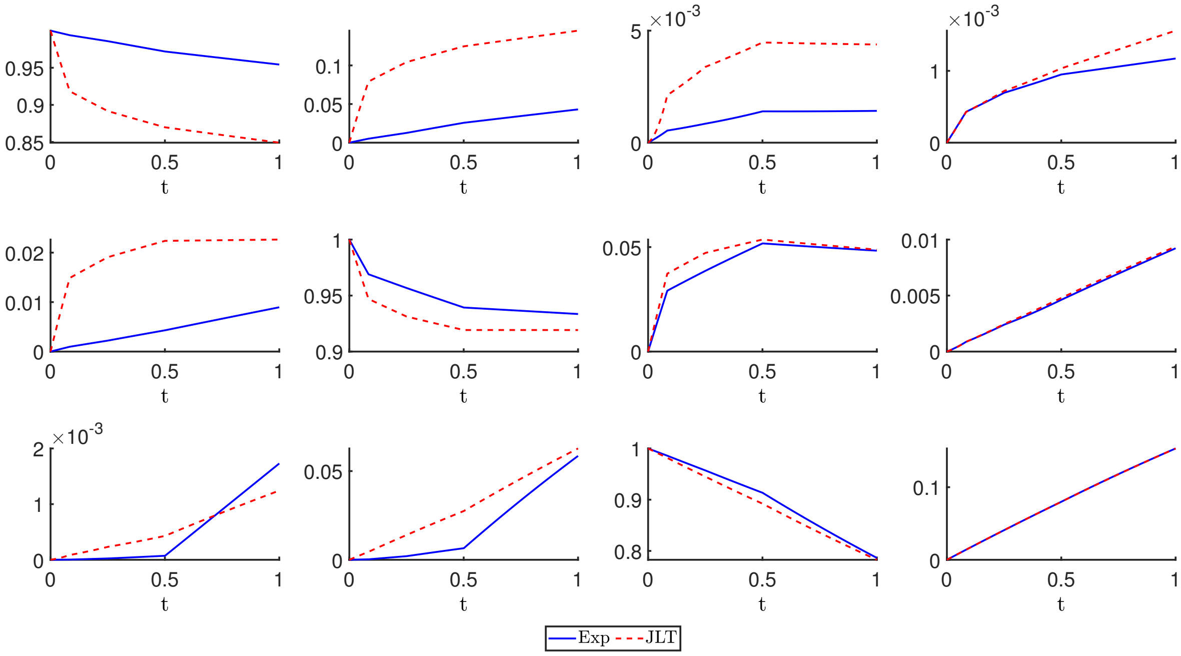


Figure 4.4.: Comparison of JLT and exponential change of measure for default data set one (Table C.5) under the risk-neutral measure.

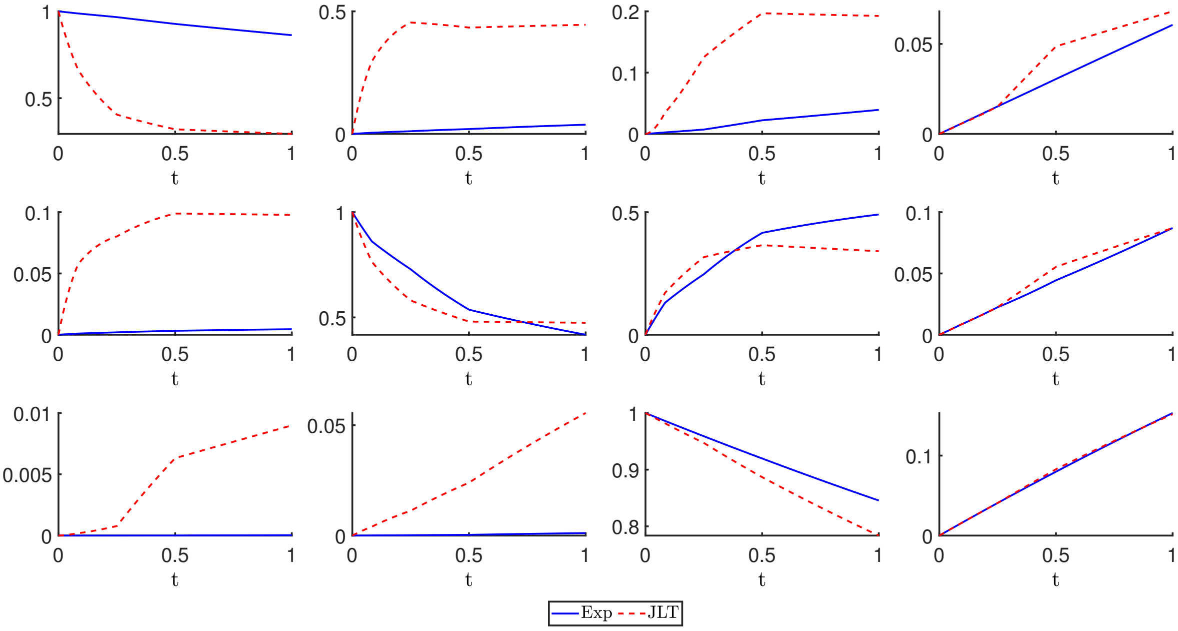


Figure 4.5.: Comparison of JLT and exponential change of measure for default data set two (Table C.6) under the risk-neutral measure.

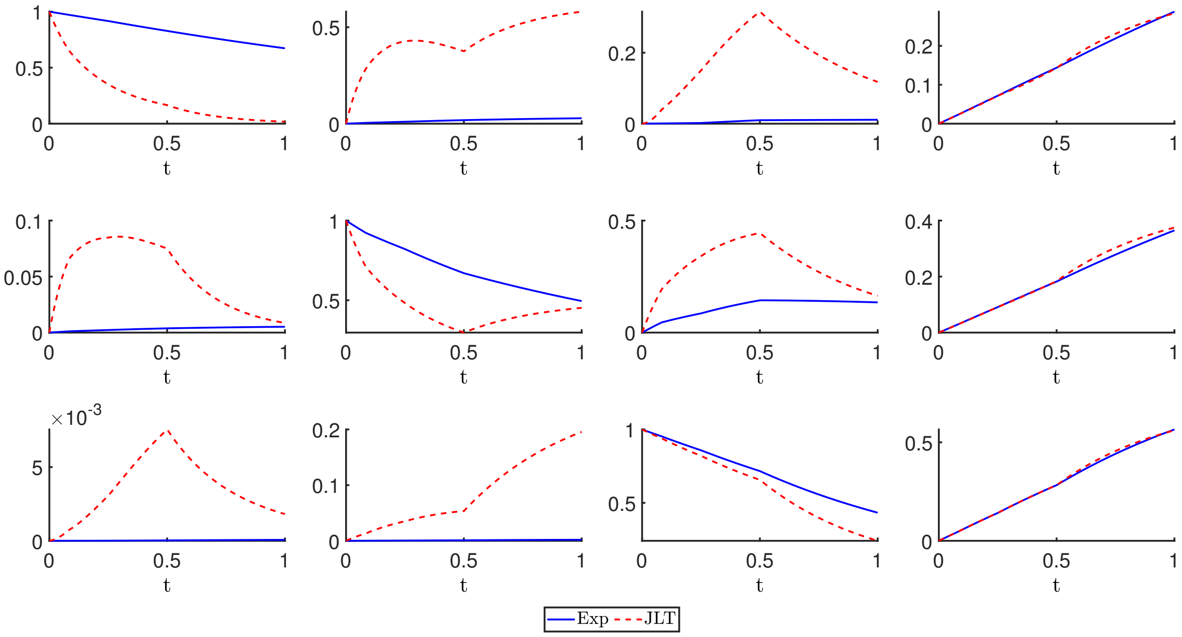


Figure 4.6.: Comparison of JLT and exponential change of measure for default data set three (Table C.7) under the risk-neutral measure.

Table C.5, this behaviour is expectable for the second row.

We would suggest to use the exponential change of measure in this case, because the first row seems to be a lot more consistent than in the JLT case.

For Figure 4.6, we used an extreme and unrealistic case of default probabilities under the risk-neutral measure (Table C.7). The exponential change of measure seems to be more well-behaved than the JLT change of measure, which exhibits strange curves for the transition probabilities in almost all entries.

For example, in the sub figure corresponding to the change from **B** to **B**, it does not make much sense that under the JLT change of measure, the probability to stay at **B** has its minimum at $t = 0.5$ and increases afterwards. We would expect a monotone decrease like we see for the exponential change of measure.

All in all, we suggest to use the exponential change of measure, since the JLT change of measure seems to overestimate the impact on the rating transition matrix apart from the default column.

Rating properties. Now, let us discuss the rating properties (4.3.1)–(4.3.4). Under the historical measure, all the properties were satisfied perfectly in all cases.

Having a closer look at Figure 4.4–4.6 for the exponential change of measure, we can already judge the properties (4.3.1)–(4.3.4).

All the properties were perfectly satisfied in all cases, except for the strong diagonal dominance (4.3.1) of the matrices.

As we can see in Figure 4.6, where we use the extreme case Table C.7 as the market default probabilities, (4.3.1) is satisfied till $t = 0.5$ but after that the second and third row do not satisfy it anymore. This is to be expected for this extreme case and the model behaves reasonable.

However, if we have a look at Figure 4.5, where we use the mild case Table C.6 as the market default probabilities, (4.3.1) is satisfied everywhere except starting from $t = 0.5$ in the second row. This behaviour is unexpected and requires further investigation. Also, under the JLT change of measure this issue remains unchanged.

We will see in Section 4.5.2.4 a different model, which will not have this problem in the mild case Table C.6.

4.5. Stochastic Rating Transitions

This section is structured as follows: In Section 4.5.1, we give a gentle introduction to matrix Lie groups and notice that the stochastic matrices form a subgroup of a matrix Lie group. We will see, how simple SDEs in $\mathbb{R}$ will lead to a stochastic and fully inhomogeneous model for rating transitions. In this framework, the change of measure is obtained by the standard Girsanov theorem.

Afterwards, we perform some numerical experiments in Section 4.5.2. The first step is to calibrate the rating process to the distributions learned by the TimeGAN, which is subject of Section 4.5.2.1. Then, in Section 4.5.2.2 and Section 4.5.2.3 we perform one test for each of the two methods proposed in Section 4.5.1 and assess their quality. In Section 4.5.2.4, we show how to calibrate the model to market default probabilities by using an analogue of the JLT and exponential change of measure seen in Section 4.4.1.2.

4.5.1. SDEs on the Lie Group of Stochastic Matrices

In this section, we show how an SDE can help to interpolate the generated rating matrices in time. This is a desirable feature for several applications, because it gives access to rating matrices of any time span or can help to forecast transition matrices with larger time spans.

To guarantee that our SDE will produce stochastic matrices, we noticed that this is a special kind of geometry and the proper tools are readily available in the matrix Lie group literature. We will recall all the necessary results first.

We consider the group $G = \{R \in \mathrm{GL}(K) : R\mathbf{1} = \mathbf{1}\}$, $\mathbf{1} = [1, \ldots, 1]^\top \in \mathbb{R}^K$, which is a matrix Lie group according to Coletti, Carneiro and Yepes (2020), i.e. a subgroup of

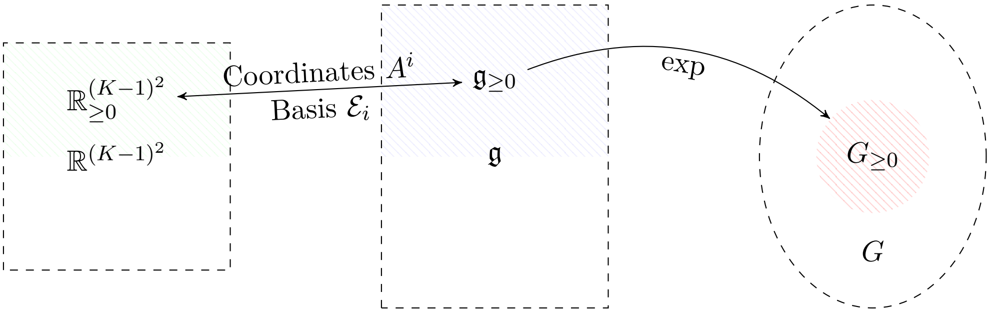


Figure 4.7.: Illustration of the relationship between $\mathbb{R}^{(K-1)^2}_{\geq 0}$, $\mathfrak{g}_{\geq 0}$ and $G_{\geq 0}$.

the general linear group $\mathrm{GL}(K)$ which is a differentiable manifold and for which the product is a differentiable mapping $G \times G \to G$. The tangent space at the identity of a Lie group is called the Lie algebra and is in this case given by $\mathfrak{g} = T_I G = \{A \in \mathbb{R}^{K \times K} : A\mathbf{1} = \mathbf{0}\}$. The Lie algebra $\mathfrak{g}$ is a vector space with $\dim(\mathfrak{g}) = K(K-1)$ since basis matrices for $\mathfrak{g}$ can be formulated as $E_{ij} - E_{ii}$ for $i, j = 1, \dots, K$ with $i \neq j$, where E_{ij} are elementary matrices. This makes the Lie algebra $\mathfrak{g}$ together with the matrix commutator, $[\cdot,\cdot] \colon \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$, $[A_1, A_2] = A_1 A_2 - A_2 A_1$, isomorphic to $\mathbb{R}^{K(K-1)}$. The matrix exponential $\exp\colon \mathfrak{g} \to G$, $\exp(A) = \sum_{k=0}^{\infty} A^k / A!$, maps elements from the Lie algebra to the Lie group and is a local diffeomorphism in a neighbourhood of $A = 0$. Analogue to Section 2.2, the directional derivative of the matrix exponential along an arbitrary matrix $H \in \mathfrak{g}$ is given by

$$\left(\frac{d}{dA}\exp(A)\right)(H) = \exp(A)\mathcal{L}_{-A}(H) \quad \text{with} \quad \mathcal{L}_{-A}(H) = \sum_{k=0}^{\infty} \frac{1}{(k+1)!}\mathrm{ad}_{-A}^{k}(H).$$

where $\mathrm{ad}_A \colon \mathfrak{g} \to \mathfrak{g}$, $\mathrm{ad}_A(H) = [A, H]$ denotes the adjoint operator, which is used iteratively,

$$\mathrm{ad}_A^0(H) = H, \quad \mathrm{ad}_A^k(H) = \mathrm{ad}_A\big(\mathrm{ad}_A^{k-1}(H)\big) = [A, \mathrm{ad}_A^{k-1}(H)]$$

for $k \geq 1$. For more details on Lie groups and Lie algebras we refer the interested reader to Hall (2003).

Consider the following SDE in the Lie algebra $\mathfrak{g}$

$$dA_t = B(t, A_t)dt + \sum_{i=1}^{(K-1)^2} C^j(t, A_t)dS_t^i, \quad L_0 = 0, \tag{4.5.1}$$

where $B, C \in \mathfrak{g}$ and S_t^i are one-dimensional general semimartingales. Applying a numerical scheme, e.g. the Euler-Maruyama scheme, to get an approximation $A_{t_{k+1}}$ of (4.5.1) after one time step[5] and computing $R_{t_{k+1}} = R_{t_k}\exp(A_{t_{k+1}})$ would result in a numerical method for solving

$$\begin{aligned} dR_t = R_t \left(\mathcal{L}_{-A_t}\left(B(t, A_t)\right) + \frac{1}{2}\sum_{i=1}^{(K-1)^2} \mathcal{Q}_{-A_t}\Big(C^i\left(t, A_t\right), C^i\left(t, A_t\right)\Big)\right) dt \\ + \sum_{i=1}^{(K-1)^2} R_t \mathcal{L}_{-A_t}\Big(C^i(t, A_t)\Big)\, dW_t^i, \quad R_0 = I, \end{aligned} \tag{4.5.2}$$

which can be easily verified by applying Proposition 2.2.2 to $R_t = R_0 \exp(A_t) \in G$ in the case $S_t^i = W_t^i$ a Brownian motion. As this approach preserves the geometry of the Lie group G opposed to applying the Euler-Maruyama scheme directly to (4.5.2), this method was called the *geometric Euler-Maruyama* scheme in Marjanovic and Solo (2018). Higher order schemes based on this approach can be found in Muniz et al. (2022a) and Muniz et al. (2022b).

[5] More details about the implementation can be found on https://github.com/kevinkamm/LieRatingTriggers/blob/main/Lie/gEMP.m.

Since we are interested in stochastic matrices that are elements of $G_{\geq 0} \coloneqq \{R \in G \colon R_{ij} \in [0,1], i,j = 1, \dots, K\}$, we now consider a subset of the Lie algebra $\mathfrak{g}$, namely $\mathfrak{g}_{\geq 0} \coloneqq \{A \in \mathfrak{g} : A_{ij} \geq 0, i \neq j, A_{ii} \leq 0, A_{Kj} = 0, i,j = 1, \dots, K\}$. Note that additional to the usual properties of generator matrices we choose the last line of matrices $A \in \mathfrak{g}_{\geq 0}$ to be zero because applying the matrix exponential exp to these matrices will generate matrices that have the last unit vector in the last line. This choice is in accordance with our assumption that the default state is absorbing. With this assumption the dimension of $\mathfrak{g}_{\geq 0}$ is now $\dim(\mathfrak{g}_{\geq 0}) = (K-1)^2$ because as before basis matrices can be denoted by $E_{ij} - E_{ii}$ but for $i = 1, \dots, K-1$, $j = 1, \dots, K$ and $i \neq j$. Similarly to before, there exists an isomorphism between $\mathfrak{g}_{\geq 0}$ and $\mathbb{R}^{(K-1)^2}$, which is illustrated on the left-hand side in Figure 4.7. We will denote the basis for $\mathfrak{g}_{\geq 0}$ by $\mathcal{E}_i$, $i = 1, \dots, (K-1)^2$. The fact that for any $A \in \mathfrak{g}_{\geq 0}$ we have $\exp(A) \in G_{\geq 0}$ is well-known and a proof can be found in Stroock (2005): *pp. 86ff. Chapter 4.2.5: Solving Kolmogorov's Equation.*

Direct exponential mapping. For the interpolation of the generated rating matrices we consider the SDE (4.5.1) again and discuss some conditions for the solution A_t to be evolving in $\mathfrak{g}_{\geq 0}$ such that $\exp(A_t) \in G_{\geq 0}$. Therefore, we make the assumption that the equation is decoupled in the following sense:

$$dA_t = B(t, A_t)dt + \sum_{i=1}^{(K-1)^2} C^i(t, A_t)dS_t^i = \sum_{i=1}^{(K-1)^2} \left(b_i(t, A_t^i)dt + c_i(t, A_t^i)dS_t^i \right) \mathcal{E}_i, \tag{4.5.3}$$

where $\mathcal{E}_i$ denotes the basis vectors of $\mathfrak{g}_{\geq 0}$. If the solution A_t^i of $dA_t^i = b_i(t, A_t^i)dt + c_i(t, A_t^i)dS_t^i$ is $\mathbb{P}$-almost surely positive for all $t \geq 0$ and for all i then $A_t \in \mathfrak{g}_{\geq 0}$ and $R_t^{\mathrm{SDE}} \coloneqq \exp(A_t) \in G_{\geq 0}$.

Let us show two examples:

1. Let $b_i(t,x) \equiv b_i \in \mathbb{R}_{\geq 0}$, $c_i(t,x) \equiv c_i \in \mathbb{R}_{\geq 0}$: In this case, $A_t^i = b_i t + c_i S_t^i$ and the condition $A_t^i \geq 0$ leads to $b_i t + c_i S_t^i \geq 0$ for all t $\mathbb{P}$-almost surely. Further assuming $S_t^i \geq 0$ would be one example.
2. A_t^i are CIR-processes, i.e. $S_t^i = W_t^i$ and $dA_t^i = a_i \left(b_i - A_t^i\right) dt + \sigma_i \sqrt{A_t^i} dW_t^i$.

For this simple approach there is a price to pay, namely R_t^{SDE} cannot be viewed as an evolution system of a Markovian rating process, since the Chapman-Kolmogorov equation is not necessarily satisfied. Or in other words, the associated rating process will not be memoryless and it is difficult to sample it.

Geometric Euler-Maruyama. In order to preserve the Chapman-Kolmogorov equation, one could use the aforementioned geometric Euler-Maruyama scheme and define $R_t^{\mathrm{SDE}} = R_0 \exp(A_t)$. However, to ensure that $R_t^{\mathrm{SDE}} \in G_{\geq 0}$, which is equivalent to ensuring that the approximation for L_t is in $\mathfrak{g}_{\geq 0}$, we need an additional assumption. For the Euler-Maruyama scheme to have results in $\mathfrak{g}_{\geq 0}$ it would be necessary that all increments $\Delta A_{t_k} \geq 0$, i.e. $A_t \geq 0$ must have monotonically increasing paths in time, as well.

A class of processes satisfying this condition easily, would be all jump processes with positive jumps only. Another possibility could involve processes with stochastic coefficients of the form

$$dA_t^i = a_i(t, Y_t^i)dt, \quad a_i(t, y) \geq 0 \tag{4.5.4}$$

$$dY_t^i = b_i(t, Y_t^i)dt + c_i(t, Y_t^i)dS_t^i, \quad Y_0^i = y_0^i \tag{4.5.5}$$

In this case, A_t^i are positive, pathwise-increasing, continuous stochastic processes for any semimartingale S_t^i.

Remark 4.5.1. Let us note that decoupling the SDE in the Lie algebra does not mean that the SDE in the Lie group will be decoupled as well. On the contrary, one can see by the definition of the matrix exponential and the matrix multiplication therein that the resulting SDE will be fully coupled.

From a computational point of view, the decoupling in the Lie algebra is very advantageous, because all SDEs can be solved in parallel. Since we want to calibrate the SDE in the Lie group to historical rating matrices, it will be very important that the SDEs in the Lie algebra can be solved very fast.

From an analytical point of view, this approach translates the problem of defining an SDE with values in the space of stochastic matrices to simple SDEs taking values in $\mathbb{R}$, where a vast amount of literature and standard analytical tools are available.

Girsanov's theorem. As in Section 4.4, we need to derive the dynamics of our model (4.5.2) under a second measure $\mathbb{Q}$. For this we apply the standard Girsanov theorem to our SDEs in $\mathbb{R}$. Let us show this in a special case of the decoupled SDE (4.5.3).

Let the SDE in the Lie algebra satisfy the dynamics under $\mathbb{P}$

$$dA_t^{\mathbb{P}} = \sum_{i=1}^{(K-1)^2} \left(b_i(t, A_t^{\mathbb{P},i})dt + c_i(t, A_t^{\mathbb{P},i})dW_t^i \right) \mathcal{E}_i,$$

where W_t^i are independent standard Brownian motions. Denote $W_t := \left(W_t^1, \ldots, W_t^{(K-1)^2} \right)$ and let $\mathcal{F}_t$ be the natural filtration of W_t. Furthermore, assume that the process

$$L_t := \exp \left(\int_0^t \kappa_s \cdot dW_s - \frac{1}{2} \int_0^t |\kappa_s|^2 \, ds \right)$$

is a $\mathbb{P}$ martingale satisfying $\mathbb{E}^{\mathbb{P}} [L_t] = 1$ for adapted, measurable and square integrable processes $\kappa_s \in \mathbb{R}^{(K-1)^2}$. The measure $\mathbb{Q}^\kappa$ given by

$$\left. \frac{d\mathbb{Q}^\kappa}{d\mathbb{P}} \right|_{\mathcal{F}_t} := L_t$$

is well-defined and

$$W_t^\kappa := W_t - \int_0^t \kappa_s ds$$

is a Brownian motion under the measure $\mathbb{Q}^\kappa$.

Moreover, assume that the processes κ are such that the dynamics of the SDE in the Lie algebra $\mathfrak{g}_{\geq 0}$ denoted by A_t^κ are well defined. Then we have under the new measure

$$
\begin{aligned}
dA_t^\kappa &= \sum_{i=1}^{(K-1)^2} \left(b_i(t, A_t^{\kappa,i}) + c_i(t, A_t^{\kappa,i})\kappa_t^i dt + c_i(t, A_t^{\kappa,i}) dW_t^{\kappa,i} \right) \mathcal{E}_i \\
&=: B^\kappa(t, A_t^\kappa) dt + \sum_{i=1}^{(K-1)^2} C^i(t, A_t^\kappa) dW_t^{\kappa,i},
\end{aligned}
$$

where $B^\kappa(t, A_t^\kappa) := \sum_{i=1}^{(K-1)^2} \left(b_i(t, A_t^{\kappa,i}) + c_i(t, A_t^{\kappa,i})\kappa_t^i \right) \mathcal{E}_i$ and $C^i(t, A_t^\kappa) := c_i(t, A_t^{\kappa,i})\mathcal{E}_i$.

Moreover, the dynamics of R_t under the measure $\mathbb{Q}^\kappa$ are given by

$$
\begin{aligned}
dR_t^\kappa = R_t^\kappa &\left(\mathcal{L}_{-A_t^\kappa} \left(B^\kappa(t, A_t^\kappa) \right) + \frac{1}{2} \sum_{i=1}^{(K-1)^2} \mathcal{Q}_{-A_t^\kappa} \left(C^i \left(t, A_t^\kappa\right), C^i \left(t, A_t^\kappa\right) \right) \right) dt \\
&+ \sum_{i=1}^{(K-1)^2} R_t^\kappa \mathcal{L}_{-A_t^\kappa} \left(C^i(t, A_t) \right) dW_t^{\kappa,i}.
\end{aligned}
$$

One sufficient condition for this change of measure to be valid is the positivity of $A_t^{\kappa,i}$ $\mathbb{P}$-almost surely for all $t \geq 0$ and $i = 1, \ldots, (K-1)^2$.

We will see in Section 4.5.2.4, how to apply this for the calibration to the market default probabilities.

4.5.2. Numerical Tests

In this section, we conduct two experiments, one for the direct exponential mapping and one for the geometric Euler approach. We calibrate the resulting rating models R_t^{SDE} to R_t^{GAN} at $t = 1$, i.e. 1 year, by matching the first four moments. This is described in Section 4.5.2.1 in more details. In Section 4.5.2.2, we show one example for the direct method using CIR processes on $\mathfrak{g}_{\geq 0}$ and in Section 4.5.2.3 we show another example for the geometric Euler approach using a constant drift and volatility. In both sections, we will discuss the fit to the TimeGAN rating matrices by looking at their corresponding distributions at $1, 3, 6, 12$ months and study the properties of rating matrices introduced in Section 4.3.

After that, in Section 4.5.2.4 we calibrate R_t^{gEM} under the risk-neutral measure to default probabilities using an analogue of the JLT and exponential change of measure. We will discuss their impact on the distribution of the rating matrices and on the rating properties for three different cases of market default probabilities.

We used for the calibration of the rating SDE `Matlab 2022a` with the (Global) Optimization Toolbox and for the training of the TimeGAN `(Intel-)Python 3.9` with `Tensorflow 2.8.0` running on Windows 10 Pro, on a machine with the following specifications: processor Intel(R) Core(TM) i7-8750H CPU @ 2.20 GHz and 2x32 GB (Dual Channel) Samsung SODIMM DDR4 RAM @ 2667 MHz, and a NVIDIA GeForce RTX 2070 with Max-Q Design (8 GB GDDR6 RAM).

Contrary to Section 4.4.1.2, we are calibrating in this section the rating model separately under both measures. This means we are first calibrating the model parameters under the historical measure in Section 4.5.2.1 and afterwards in Section 4.5.2.4 we calibrate the change of measure variables to match the market default probabilities.

4.5.2.1. Calibration of the Rating SDE under the Historical Measure

Before we start to explain how we calibrate R_t^{SDE} to R_t^{GAN}, let us explain why we do not calibrate directly to the historical data. Suppose that we select one specific time series of historical rating matrices and try to fit our model in a least-square sense in expectation. Then, the randomness should be eliminated by the optimizer since we want to fit all the different trajectories to one time sequence. This is not the way to go, if we desire a stochastic model for the rating transitions. Another approach would be considering all of the training data set, sample as many trajectories, and calibrate again in a least-square sense. There is no reason why each of the random trajectories should match the particular rating matrix where it is subtracted from, maybe it would match another one perfectly. So comparing trajectories does not make much sense either.

Hence, it makes more sense to compare distributions or moments of the data and the model. Now, the problem with using the historical rating matrices directly in this approach would be that at each specific point in time, we only have a few available matrices. Take for example the one year rating matrices, we only have 9 different matrices. Discussing a distribution of such a sample size is not very insightful.

Therefore, we rely on the ability of the TimeGAN to learn the behaviour of the time series of rating matrices. As aforementioned, considering the time series allows us to artificially inflate the data set by using all the permutations in time for the training. After the learning phase, we can sample fake time series data, getting an arbitrary number of different rating matrices at each point in time. Now, it makes sense to compare the moments of the fake rating matrices to the ones obtained at each point in time from R_t^{SDE}.

To be more precise, we use the standard estimators for mean, variance and moments of higher order in our experiments, i.e. for $k = 3, \dots, n$, $n \in \mathbb{N}$,

$$
\begin{aligned}
[\mu_1(t)]_{ij} &:= \frac{1}{M} \sum_{w=1}^{M} [R_t(w)]_{ij}\,, \\
[\mu_2(t)]_{ij} &:= \frac{1}{M-1} \sum_{w=1}^{M} \Big([R_t(w)]_{ij} - [\mu_1(t)]_{ij} \Big)^2\,, \\
[\mu_k(t)]_{ij} &:= \frac{1}{M} \sum_{w=1}^{M} \Big([R_t(w)]_{ij} - [\mu_1(t)]_{ij} \Big)^k\,.
\end{aligned}
$$

Let Π denote the parameter set. Then, our objective function $f^n \colon \Pi \to \mathbb{R}^{n\cdot(K-1)\cdot K}$ is given by

$$
\begin{aligned}
&f_k \colon \Pi \to \mathbb{R}^{(K-1)\cdot K}, \quad f_k(p) := \operatorname{vec}\Big(\mu_k^{\mathrm{SDE}}(t;p) - \mu_k^{\mathrm{GAN}}(t) \Big) \\
&f^n(p) := [w_1 \cdot f_1(p), \dots, w_n \cdot f_n(p)]^T\,,
\end{aligned}
$$

where $w_k \in \mathbb{R}_{\geq 0}$ are weights and our minimisation problem can be formulated as a non-linear least square problem

$$\min_{p \in \Pi} \|f^n(p)\|_2^2 . \tag{4.5.6}$$

Of course, this procedure can be generalized by considering multiple points in time. Since this minimisation problem is very dependent on the performance of the DNN and its ability to learn the distribution of rating transition matrices from the historical data, one can also think of a penalized version of (4.5.6). For example, one can add another least-square term for the most recent time series, i.e.

$$\min_{p \in \Pi} \lambda_1 \|f^n(p)\|_2^2 + \lambda_2 \frac{1}{M} \sum_{w=1}^{M} \sum_{k=1}^{n} \left\| R_{t_k}^{\mathrm{SDE}}(w; p) - R_{t_k}^{\mathrm{H}} \right\|_F^2,$$

where $\|\cdot\|_F$ denotes the Frobenius norm and $\lambda_1, \lambda_2 \geq 0$ are weights. We will make the code publicly available and leave this experiment for the reader.

Remark 4.5.2. As aforementioned, using rating matrices with more than four ratings is straightforward in this approach. Since the SDEs in the Lie algebra are decoupled and can be computed in parallel, solving them will not lead to a major performance bottleneck compared to fewer ratings. The more relevant issue is that the number of parameters in the calibration increases quadratically, making it more and more important to use some principle component analysis to make the calibration more efficient. A possibility to use the autoencoder of the TimeGAN comes to mind, this is however subject to future research.

Also, it is straightforward to remove the condition that the default rating is absorbing. In this case, we would need $(K-1) \cdot K$ decoupled SDEs in the Lie algebra.

4.5.2.2. The Case of Direct Exponential Mapping.

Let us now consider $R_t^{\mathrm{CIR}} := \exp(A_t)$, where

$$dA_t^i = a_i \left(b_i - A_t^i \right) dt + \sigma_i \sqrt{L_t^i} dW_t^i .$$

Each of the SDEs have a parameter for the mean-reversion b_i, mean-reversion speed a_i and volatility σ_i, which are all assumed to be positive, and their own standard Brownian motion W_t^i.[6] The Brownian motions are assumed to be mutually independent. During our calibration procedure we allow the Feller-condition to be violated for simplicity. The parameter set is therefore given by positive real numbers $\Pi^{\mathrm{CIR}} := \mathbb{R}_{\geq 0}^{3 \cdot (K-1)^2}$ by stacking the individual parameters below each other. We found during our experiments that values between zero and one worked best. We calibrated R_t^{CIR} for $t = 1$, i.e. for the 12 month rating transitions, by matching the moments up to order 4. For the variance, we added a weight $w_2 = 10$ and set $w_1 = w_3 = w_4 = 1$ to put more emphasis on the variance. The corresponding parameters after the calibration procedure with $M = 1000$ trajectories for R_t^{CIR} and $M = 10000$ trajectories

[6] More details about the implementation can be found on https://github.com/kevinkamm/RatingML/blob/main/LieSDE/Direct_CIR/main.m.

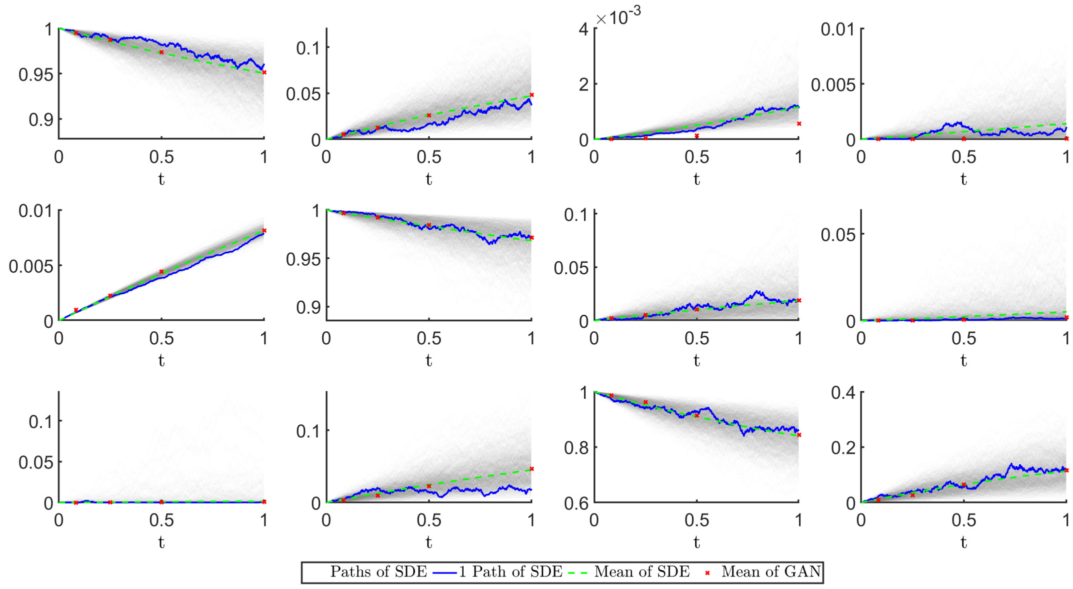


Figure 4.8.: Trajectories of calibrated R_t^{CIR} with parameters as in Table 4.5.

for R_t^{GAN} can be found in Table 4.5. The first column explains to which basis element the coefficients belong. To be more precise, 2-3 means that the initial rating is 2 and at $t = 1$ we transition to rating 3. The minimisation error (4.5.6) in this case was $1.1764e-04$, telling us that the moments up to order 4 match very well and it took roughly 15.6214 seconds using `lsqnonlin` with the Trust-Region-Reflective algorithm.

Table 4.5.: Parameters of R^{CIR} after calibration at $t = 1$ to R^{GAN} using $n = 4$ moments.

From-To	a	b	sigma
1-2	1.01e-01	8.78e-02	6.46e-03
1-3	4.66e-02	3.66e-02	2.89e-01
1-4	2.65e-01	2.14e-01	1.40e-01
2-1	2.71e-01	2.13e-01	1.76e-01
2-3	3.07e-02	2.56e-02	9.98e-03
2-4	1.59e-01	1.40e-01	1.53e-01
3-1	3.79e-02	3.45e-02	5.14e-02
3-2	6.60e-02	6.17e-02	1.69e-01
3-4	4.02e-01	3.83e-01	2.75e-01

In Figure 4.8, we can see the trajectories of R_t^{CIR} over time for each entry in the rating matrix except for the last row. The upper left corner are the transition probabilities from **A** to **A**, right next to it from **A** to **B** and so on. The grey lines are a cloud of 1000 trajectories of R_t^{CIR} and the blue line is one trajectory. The green dashed line is the mean at each time of the process and the red dots are the means of R_t^{GAN} at $t = 1, 3, 6, 12$ months. We can see that the paths are rough and the mean-reversion of the CIR processes is apparent as well, since the blue line tends to come back to the green dashed line illustrating its mean. Also, we see again a good fit over time to R_t^{GAN} by comparing how close the mean of R_t^{CIR} is compared to the mean of R_t^{GAN}.

Remark 4.5.3. We modelled the rating transition by starting with an SDE on the positive half-space of the Lie algebra of stochastic matrices. Another approach could involve modelling the SDE on the appropriate half-space of the Lie group directly. To do this, it would be necessary to use SDEs respecting the underlying geometry, i.e. Stratonovich-SDEs, since they obey the chain rule, or the Itô counterpart by Itô-Stratonovich conversion.

In this line of research, numerical methods such as Runge-Kutta-Munthe-Kaas (RKMK) (cf. Muniz et al. (2022b)) or the Magnus expansion are available.

The advantage of studying these SDEs directly on the Lie group are that one can check more easily if the SDE will satisfy the rating matrix properties.

Remark 4.5.4. We would like to point out that similar to Section 4.4.1.2, we could calibrate the Lie model directly on the level of generator. For this, we would compute the matrix logarithm of the generated matrices by the TimeGAN and, if necessary, obtain valid elements of the Lie algebra by applying Algorithm 4.2.

This will lead to a significant performance improvement and we expect similar results in terms of accuracy. However, we decided in this section to show that it is numerically feasible to circumvent the issues arising from the embedding problem by computing the rating transitions in each step of the calibration. Also, for the PHCTMC, we could compute the transition matrices in each step of the calibration with the same drawback of reduced performance.

Analysis of the rating distributions and properties Since we expect that downgrades are more likely than upgrades, we expect that the rating distributions should be skewed with one tail being fatter than the other. We can see this in both Figure 4.9 ($t = 0.5$) and Figure 4.10 ($t = 1$). Each of the figures are ordered as the entries for the rating matrices excluding the last row. This means that the upper left sub figure shows the transitions for **A** to **A**, the one right next to it **A** to **B** and so on. The red columns are the histogram of R_t^{GAN} and the blue columns illustrate the histogram of R_t^{CIR}. We fitted beta distributions to the histograms. The red solid line is the according beta distribution of R_t^{GAN} and the blue dashed line the beta distribution of R_t^{CIR}.

Let us focus for the moment on Figure 4.10, i.e. the rating transitions for one year. The distributions using R^{GAN} look like they have two modes and suggest a mixture Gaussian model. Therefore, the beta distributions do not describe the data very well. However, we have no intuition why the rating transitions should have two modes and consider it as subject for further investigation.

For R_t^{CIR} we see a close match of the beta distribution to the histograms and match our initial intuition that the model should have one tail being fatter than the other.

In Figure 4.9, we see in most of the sub figures a good match of the shapes of the beta distributions of R^{GAN} and R^{CIR} even though the CIR processes have constant coefficients and are calibrated to the moments of R^{GAN} at $t = 1$. We saw the same for $t = 1, 3$ months and therefore decided not to put the figures to shorten the presentation.

Let us now assess the quality of the model rating matrices as for the training data set and TimeGAN by (4.3.1)–(4.3.4). Table 4.6 is structured exactly like Table 4.2. We can

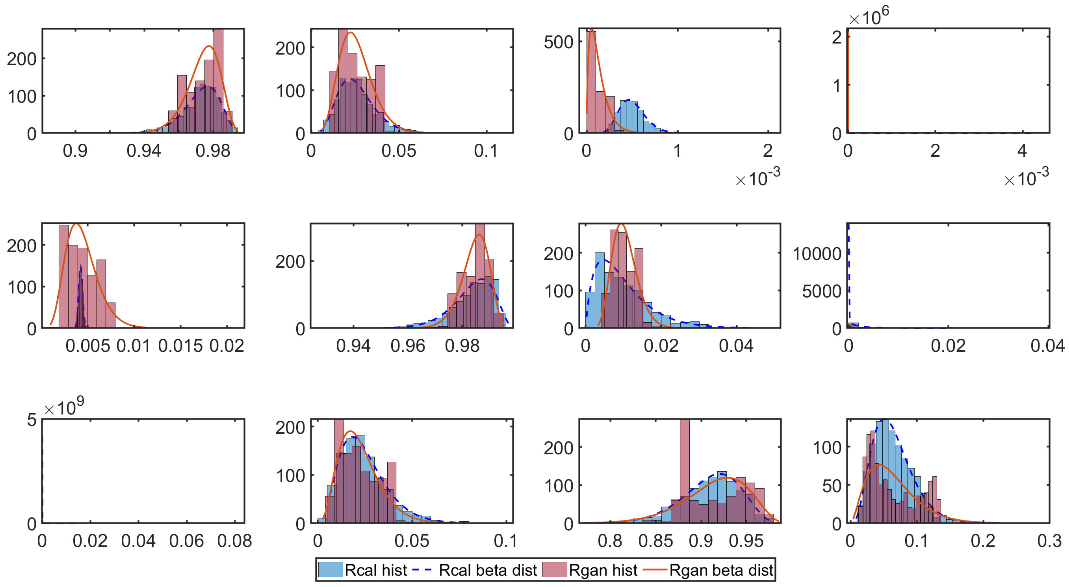


Figure 4.9.: Histograms of rating transition probabilities at 6 months.

see similar results to Table 4.3. Almost all the conditions are satisfied perfectly except for (4.3.4), where only 7 % violated the condition at $t = 6, 12$ months. Another downside of this method can be seen in Figure 4.8 by focusing on the blue trajectory in the default-column. It seems possible that the default is not absorbing because the trajectories are not monotonically increasing, only the mean is increasing. This could be viable if we allow companies to recover from default over time provided that they were not bankrupt to begin with. In fact, this would be more realistic, because otherwise either every entity would eventually default or at some point no entity would default anymore. Also it could be interesting to study conditions in this setting to ensure monotone increasing paths in the default column, which is subject to future research. We will see in the next section that the geometric Euler approach will not suffer from this problem.

Table 4.6.: Rating properties for R^{CIR}. Average percentage of the time series fulfilling the conditions (4.3.1)–(4.3.4).

Time in months	Strongly diagonally dominant (4.3.1)	Downgrading is more likely (4.3.2)	Monotone default column (4.3.3)	Increasing rating spread (4.3.4)
1	100 %	100 %	100 %	100 %
3	100 %	100 %	100 %	100 %
6	100 %	100 %	100 %	93.22 %
12	100 %	100 %	100 %	93.34 %

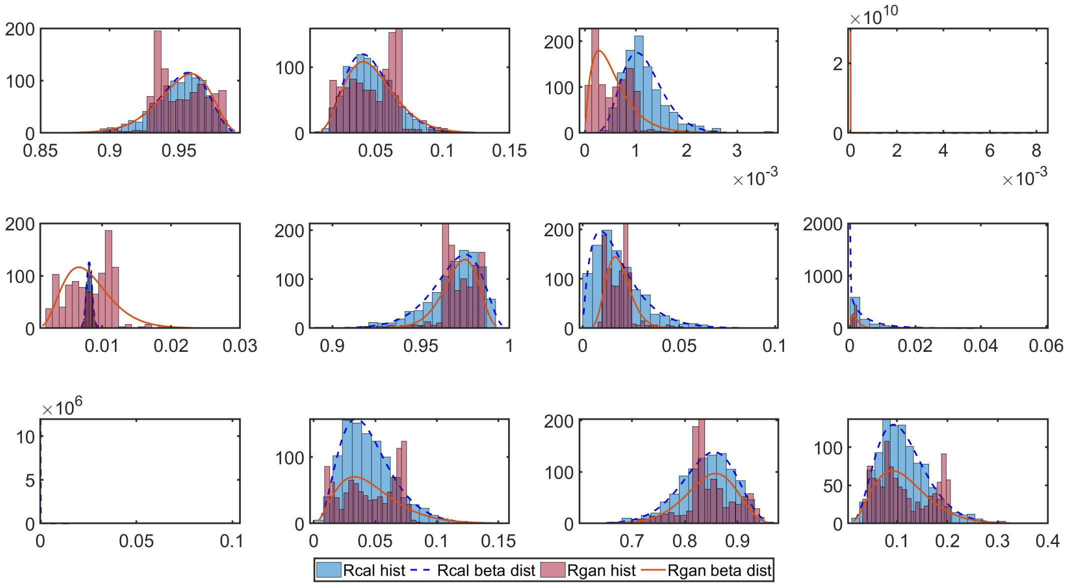


Figure 4.10.: Histograms of rating transition probabilities at 12 months.

4.5.2.3. The Case of the Geometric Euler Maruyama Scheme.

Let us now consider R_t^{gEM} and assume that each of the SDEs are given by

$$\begin{aligned} dA_t^i &= \left|Y_t^i\right|^{a_i} dt \\ dY_t^i &= b_i dt + \sigma_i dW_t^i, \quad Y_0^i = 0. \end{aligned}$$

They have a parameter for a constant drift b_i, power a_i and volatility σ_i, which are all assumed to be positive, as well as mutually independent Brownian motions W_t^i.[7] The parameter set is therefore given by positive real numbers $\Pi^{\mathrm{gEM}} := \mathbb{R}_{\geq 0}^{3\cdot(K-1)^2}$ by stacking the individual parameters below each other. We found during our experiments that values between zero and two worked best. We calibrated R_t^{gEM} for $t = 1$, i.e. for the 12 month rating transitions, by matching the moments up to order 4. For the variance, we added a weight $w_2 = 10$ and set $w_1 = w_3 = w_4 = 1$ to put more emphasis on the variance. The corresponding parameters after the calibration procedure with $M = 1000$ trajectories for R_t^{gEM} and $M = 10000$ trajectories for R_t^{GAN} can be found in Table 4.7. The first column explains to which basis element the coefficients belong. To be more precise, $2 - 3$ means starting rating is 2 and at $t = 1$ we transition to rating 3. The minimisation error (4.5.6) in this case was $4.56342e - 05.$, telling us that the moments up to order 4 match very well and it took roughly 983.269 seconds using `lsqnonlin` with the Trust-Region-Reflective algorithm.

In Figure 4.11, we can see the trajectories of R_t^{gEM} over time for each entry in the rating matrix except for the last row. The upper left corner are the transition probabilities from **A** to **A**, right next to it from **A** to **B** and so on. The grey lines are a cloud of 1000 trajectories of R_t^{gEM} and the blue line is one trajectory. The green dashed line is the mean at each time

[7] More details about the implementation can be found on https://github.com/kevinkamm/LieRatingTriggers/blob/main/Lie/main.m.

Table 4.7.: Parameters of R^{gEM} after calibration at $t = 1$ to R^{GAN} using $n = 4$ moments.

From-To	a	b	sigma
1-2	1.47e+00	6.10e-02	3.52e-02
1-3	2.00e+00	1.84e-04	1.69e-04
1-4	1.05e+00	1.11e-01	4.02e-02
2-1	1.25e+00	1.75e-01	4.25e-02
2-3	1.92e+00	9.68e-03	7.33e-03
2-4	1.26e+00	8.41e-02	3.09e-02
3-1	1.96e+00	1.04e-02	1.04e-02
3-2	1.89e+00	2.70e-02	2.68e-02
3-4	7.72e-01	6.61e-02	1.32e-01

of the process and the red dots are the means of R_t^{GAN} at $t = 1, 3, 6, 12$ months. We can see that the paths are much smoother compared to Figure 4.8. Also, we see again a good fit at the terminal time to R_t^{GAN} by comparing how close the mean of R_t^{gEM} is compared to the mean of R_t^{GAN}. For $t = 1, 3, 6$ months, we see a slight deviation of their corresponding means, suggesting that we should either use time-dependent parameters or different SDEs.

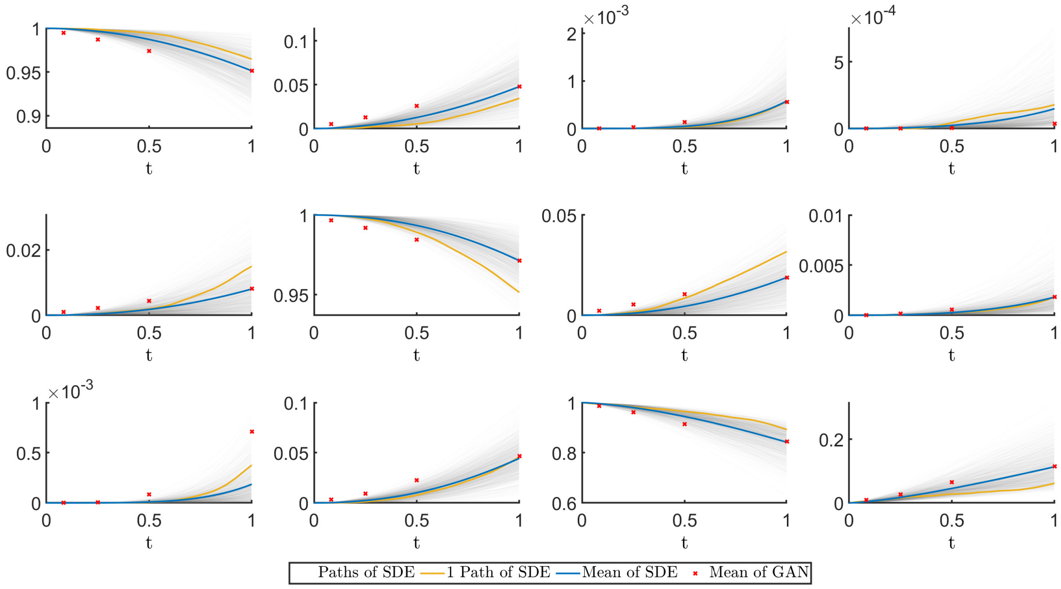


Figure 4.11.: Trajectories of calibrated R_t^{gEM} with parameters as in Table 4.7.

Analysis of the rating distributions and properties In Figure 4.12 and Figure 4.13, we can see the analogue of Figure 4.9 and Figure 4.10 from Section 4.5.2.2. We used the same trajectories of R_t^{GAN} in these plots to be able to compare both methods amongst each other.

Let us focus for the moment on Figure 4.13, i.e. the rating transitions for one year. For R_t^{gEM} we see a close match of the beta distribution to the histograms as well. Also, we see a very good fit of the beta distributions of R^{GAN} and R^{gEM} towards each other. This fit looks even closer than in Figure 4.10 for R^{GAN} and R^{CIR}.

In Figure 4.12, the six month rating transitions, we see a worse fit to the data than we saw in Figure 4.9 using R_t^{CIR}. This suggests that one should either use a different underlying SDE for Y_t^i or introduce time-dependent parameters.

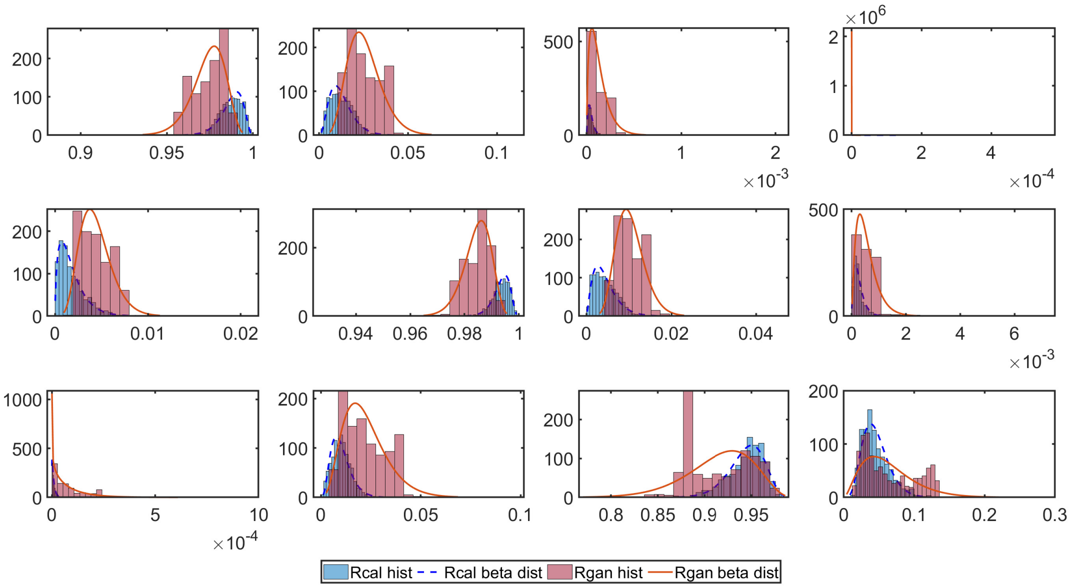


Figure 4.12.: Histograms of rating transition probabilities at 6 months.

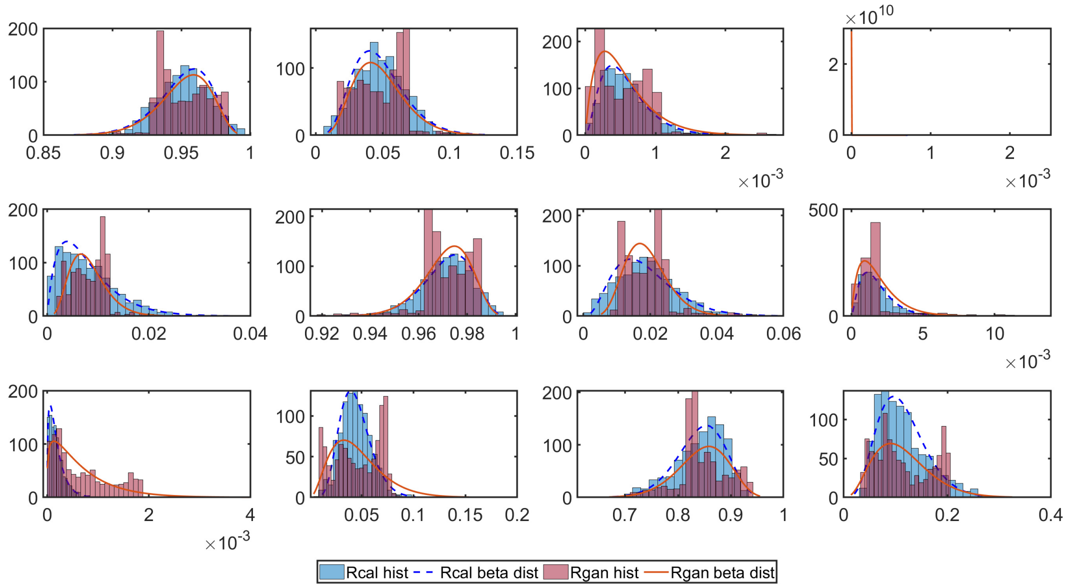


Figure 4.13.: Histograms of rating transition probabilities at 12 months.

Most remarkably all the conditions (4.3.1)–(4.3.4) were satisfied perfectly for this model.

4.5.2.4. Calibration of the Rating SDE under the Risk-Neutral Measure

In this subsection, we will consider again R_t^{gEM} and assume that each of the SDEs are given by

$$
\begin{aligned}
dA_t^i &= \left|Y_t^i\right|^{a_i} dt \\
dY_t^i &= b_i dt + \sigma_i dW_t^i, \quad Y_0^i = 0.
\end{aligned}
$$

Let us assume that the processes $\kappa_t \equiv \kappa \in \mathbb{R}^{(K-1)^2}$ are constant, then the dynamics of A_t^i are given by

$$
\begin{aligned}
dA_t^{\kappa,i} &= \left|Y_t^{\kappa,i}\right|^{a_i} dt \\
dY_t^{\kappa,i} &= \left(b_i + \sigma_i \kappa_i\right) dt + \sigma_i dW_t^{\kappa,i}.
\end{aligned}
$$

In this case, no further conditions on κ are required to ensure a valid change of measure such that $A_t \in \mathfrak{g}_{\geq 0}$. The process $R_t^{\mathrm{gEM},\kappa}$ can now again be computed by the geometric Euler method.

Like in Example 4.4.2, we will consider an analogy of the JLT change of measure and exponential change of measure in this setting.

To see how this can be done analogue to Example 4.4.2, let us define

$$
D := \sum_{k=1}^{(K-1)^2} \kappa_k \mathcal{E}_k \in \mathbb{R}^{K \times K}.
$$

and denote the entries of D by d_{ij}, $i, j = 1, \dots, K$. Let us consider a vector $h \in \mathbb{R}^K$, such that $h_K = 1$.[8]

Now, for the JLT change of measure each row of D has the same parameter κ_k. To make this more precise, set κ_k, $k = 1, \dots, (K-1)^2$, such that $d_{ij} = h_i$ for all $j = 1, \dots, K$, $i = 1, \dots, K-1$ with $i \neq j$.

Similarly, for the exponential change of measure we set κ_k, $k = 1, \dots, (K-1)^2$, such that $d_{ij} = \frac{h_i}{h_j}$ for all $j = 1, \dots, K$, $i = 1, \dots, K-1$ with $i \neq j$ and furthermore assume $h_i \neq 0$.

For the calibration under $\mathbb{Q}$, we assume that a_i, b_i, σ_i are already known from the calibration under the historical measure as seen in Section 4.5.2.1. Therefore, we are looking for the values of h, such that the default probabilities of our model, i.e. the last column of $R_t^{\mathrm{gEM},h}$, are close to the market default probabilities $\mathrm{PD}(t)$. We will consider here for simplicity the case, where we calibrate at the terminal time $T = 1$ year.

To be more precise, we are looking for a solution to the minimization problem

$$
\min_{h \in \mathbb{R}^{K-1} \setminus \{0\}} \left\| R_T^{\mathrm{gEM},h} e_K - \mathrm{PD}(T) \right\|_2^2. \tag{4.5.7}
$$

As in Section 4.4.1.2, we are considering the three different cases of the market default

[8] More details about the implementation can be found on https://github.com/kevinkamm/LieRatingTriggers/blob/main/Lie/gEMQ.m.

Table 4.8.: Parameters of R^{gEM} after calibration at $t = 1$ to market default probabilities using the coefficients Table 4.7.

PD(T)	Exp. change of measure		JLT change of measure	
	error (4.5.7)	h	error (4.5.7)	h
Table C.5	9.831e-09	[1.976, 3.086, 1.018]	6.868e-08	[1.362, 3.122, 1.060]
Table C.6	2.102e-11	[38.750, 16.853, 1.002]	3.007e-08	[34.514, 17.273, 1.006]
Table C.7	6.054e-12	[94.27, 41.80, 2.30]	5.395e-11	[72.74, 41.21, 13.01]

probabilities in Table C.5–C.5 using the JLT and exponential change of measure.

The computational times for the calibration using `lsqnonlin` with the Trust-Region-Reflective algorithm were in all cases around 120 seconds.

In Table 4.8, we can see the results of the calibration (4.5.7). The errors were all excellent and usually the fit to the market default probabilities was a bit smaller for the exponential change of measure than for the JLT change of measure. However, for these small errors the difference is negligible. We can also see the calibrated parameters h corresponding to the different changes of measure. We did not print the last value of h_K, since it is equal to one. For each row, we notice that the values of h are not too different for the different changes of measure. This can be explained by the fact that in the JLT change of measure each row is multiplied by h_i, $i = 1, \dots, K$, and for the exponential change of measure the default column is multiplied by $\frac{h_i}{h_K} = \frac{h_i}{1} = h_i$ as well. Since this is the column on which we calibrate, we can expect similar values.

However, since the structure of the change of measure differs a lot from each other, we can expect a very different impact on the rest of the rating transition matrix. Similar to Figure 4.4–Figure 4.6, we can see in Figure 4.14–4.16 the evolution of the mean over $M = 1000$ trajectories from $t = 0$ up to $t = 1$ year corresponding to the market default probabilities Table C.5–C.7, respectively. The yellow dashed line corresponds to the mean under the historical measure to compare the impact of the individual change of measure. The blue bold line depicts the JLT change of measure and the dark red bold line the exponential change of measure. The bright red crosses are the market default probabilities at $t = 1$. Each figure is divided into a matrix of sub figures corresponding to the entries in the rating matrix, i.e. the upper left corner describes the transition probabilities from **A** to **A** at each time t and the one right next to it from **A** to **B**, and so on. The last row of the rating matrix is excluded, since its constant.

Aside from the difference that we are only calibrating at the terminal time instead of at $1, 3, 6, 12$ months, the results look similar to those for the ICTMC. In Figure 4.4, where the default probabilities under the risk-neutral measure were close to the ones under the historical measure, we see that both changes of measure impact the entire rating matrix very similar. This changes for Figure 4.5 and Figure 4.6. The JLT change of measure overestimates the impact on the rating matrix apart from the default column.

Let us now compare the sub figure for the transition from **B** to **C** in Figure 4.5 to Figure 4.15. As a reminder, in the ICTMC case, we saw large transition probabilities up to

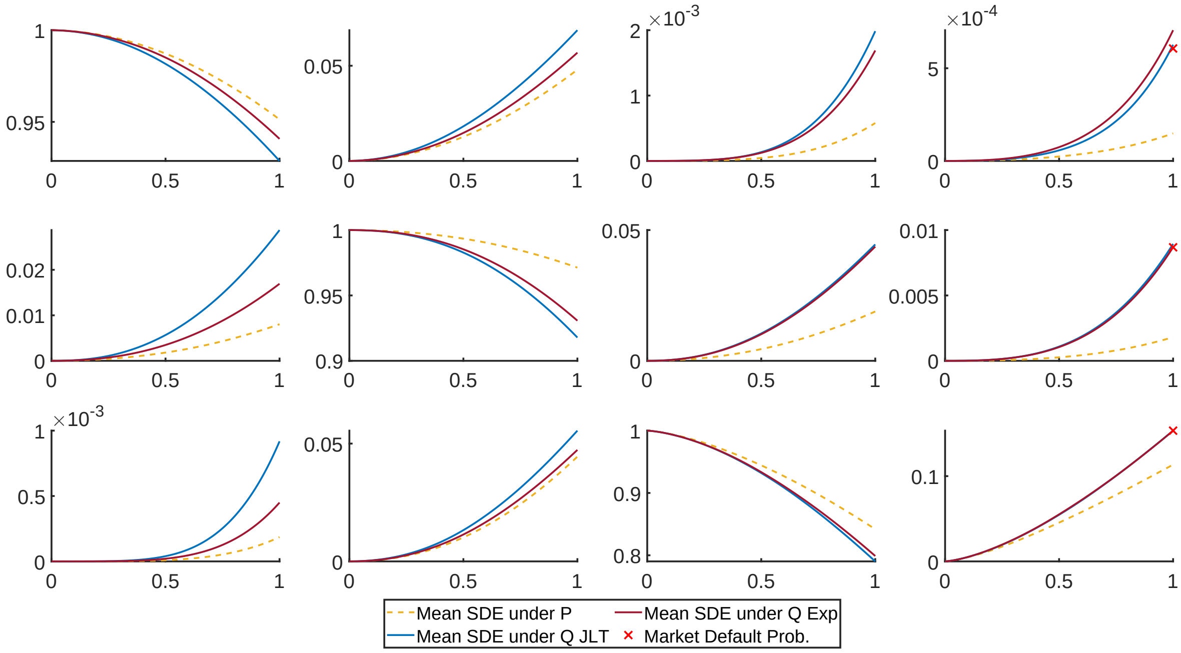


Figure 4.14.: Comparison of JLT and exponential change of measure of the mean over all trajectories of R_t^{gEM} using the market default probabilities Table C.5.

50 %, which were far from the ones under the historical measure. In this model, this is far more well-behaved with transition probabilities reaching only 20 % at maximum. This is why we suggest even if the potential user is only interested in deterministic rating transitions to consider the mean of R_t^{gEM} as a possible candidate.

All in all, we come to the same conclusion as for the ICTMC model. The exponential change of measure seems to be closer to the transition probabilities under the historical measure, which we deem to be more consistent. Therefore, we suggest to use the exponential change of measure and will exclude further tests for the JLT change of measure.

Analysis of the rating distributions and properties. Let us now have a closer look at the distributions and rating properties for the exponential change of measure.

For the rating distributions, we consider the second market default probabilities in Table C.6, which are slightly more elevated compared to the historical default probabilities. The other cases show similar results. In Figure 4.17, the histogram of the distribution of R_t^{gEM} under measure $\mathbb{Q}^h$ is corresponding to the light blue columns and a fitted beta distribution to this histogram is illustrated by the dark blue dashed line. In purple, we show the fitted beta distribution under the measure $\mathbb{P}$. The top picture shows the distributions at six months and the bottom picture at one year. Both pictures contain again sub figures corresponding to the entries in the rating matrix, where the last row is again excluded. We can see that the change of measure impacts both, the mean and the spread of the distribution under the new measure. The larger the difference between the risk-neutral and historical default probabilities, the greater the impact. This can be seen by comparing the first and second row to the third row. In the third row, we kept a close default probability under both measures and the distributions both look similar at $t = 0.5$ and $t = 1$ years.

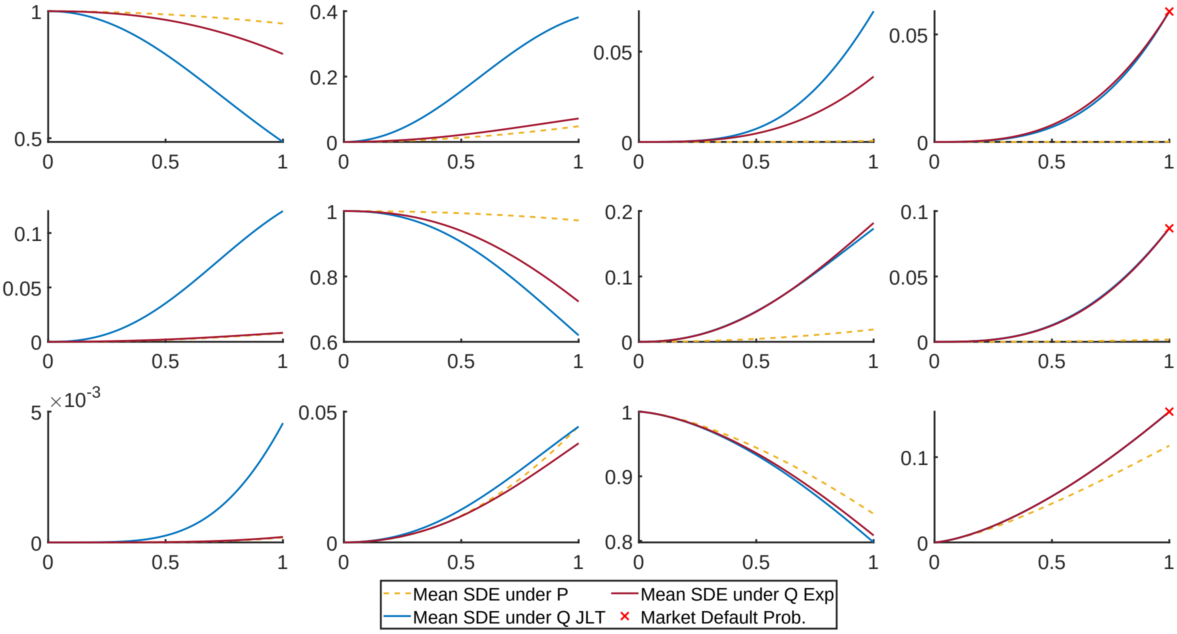


Figure 4.15.: Comparison of JLT and exponential change of measure of the mean over all trajectories of R_t^{gEM} using the market default probabilities Table C.6.

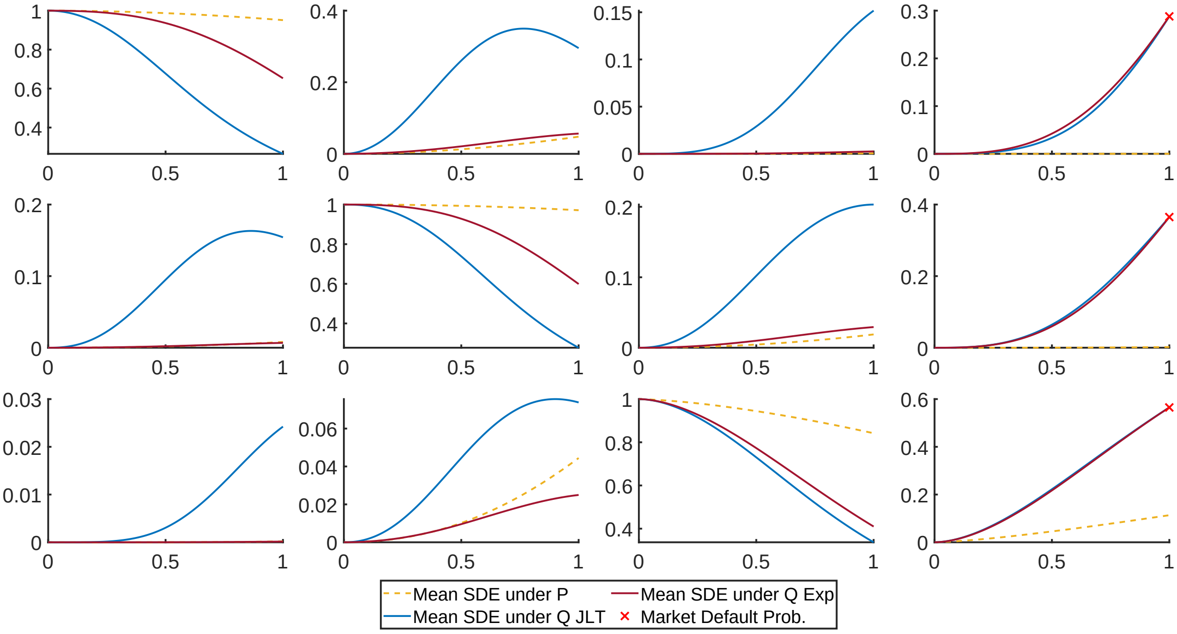


Figure 4.16.: Comparison of JLT and exponential change of measure of the mean over all trajectories of R_t^{gEM} using the market default probabilities Table C.7.

Table 4.9.: Rating properties for R^{gEM} using the exponential change of measure calibrated to the market default probabilities Table C.5. Average percentage of the time series fulfilling the conditions (4.3.1)–(4.3.4).

Time in months	Strongly diagonally dominant (4.3.1)	Downgrading is more likely (4.3.2)	Monotone default column (4.3.3)	Increasing rating spread (4.3.4)
1	100 %	100 %	98.3 %	100 %
3	100 %	100 %	99.36 %	100 %
6	100 %	100 %	99.96 %	100 %
12	100 %	100 %	100 %	100 %

In the first and second row, we see the greatest difference for the the default probabilities and the diagonal elements. The other distributions are slightly shifted but more similar to the historical transitions, which matches the observation we made in Figure 4.14–4.16.

Now, we have a closer look at the rating properties (4.3.1)–(4.3.4). Table 4.9–4.11 contain the summaries of the rating properties using Table C.5–C.7 as the default probabilities, respectively. The numbers represent the percentages of time-sequences satisfying the conditions averaged over all initial ratings. For the rating spreads over time, we consider time steps from 0 to 1 month, 1 to 3, 3 to 6 and 6 to 12 and write down the percentages for $t = 1, 3, 6, 12$ respectively.

We can see in all three tables that almost all the properties are satisfied after the change of measure. For the monotone default column, we see in all three tables slight violations and we checked that the violation of this property for almost all the trajectories were between **A** and **B** with a small magnitude.

In Table 4.11, we can see for the first time a violation of the strongly diagonal dominant property. With these huge changes of the probability of default, it is not surprising and we confirmed that almost no trajectory resulted in strong diagonal dominance in the third row, which we can also see in Figure 4.16. The other rows still remained diagonal dominant explaining the 75 %. Since this situation with so large probabilities of default is unrealistic and only serves as a test of robustness, we find that the behaviour of the model is as expected.

We conclude that the model can deal with small, medium and large probabilities of default under the risk-neutral measure using the exponential change of measure in a satisfactory manner. Also, we find again evidence that the conditions (4.3.1)–(4.3.4) describe a rating model well for realistic data, and violations occur for unrealistic cases as expected.

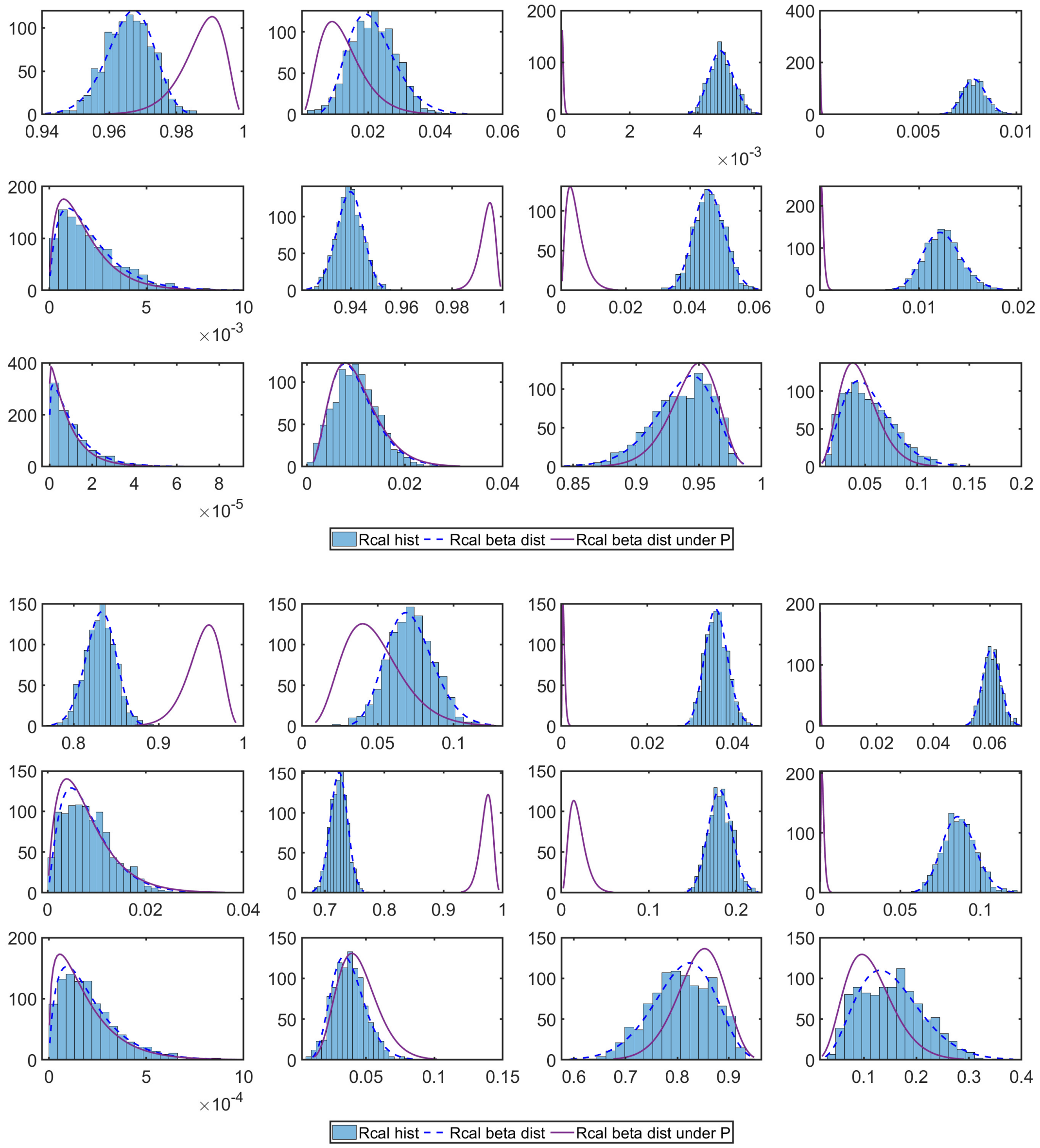


Figure 4.17.: Histograms of rating transition probabilities at 6 months (top figure) and 12 months (bottom figure) of R_t^{gEM} using the market default probabilities Table C.5.

Table 4.10.: Rating properties for R^{gEM} using the exponential change of measure calibrated to the market default probabilities Table C.6. Average percentage of the time series fulfilling the conditions (4.3.1)–(4.3.4).

Time in months	Strongly diagonally dominant (4.3.1)	Downgrading is more likely (4.3.2)	Monotone default column (4.3.3)	Increasing rating spread (4.3.4)
1	100 %	100 %	95.4 %	100 %
3	100 %	100 %	98.96 %	100 %
6	100 %	100 %	99.63 %	100 %
12	100 %	100 %	95.46 %	100 %

Table 4.11.: Rating properties for R^{gEM} using the exponential change of measure calibrated to the market default probabilities Table C.7. Average percentage of the time series fulfilling the conditions (4.3.1)–(4.3.4).

Time in months	Strongly diagonally dominant (4.3.1)	Downgrading is more likely (4.3.2)	Monotone default column (4.3.3)	Increasing rating spread (4.3.4)
1	100 %	100 %	99.63 %	100 %
3	100 %	100 %	100 %	100 %
6	100 %	100 %	100 %	100 %
12	75.02 %	100 %	100 %	100 %

4.6. Simulation of the Rating Process

In this section, we show how to simulate the PHCTMC from Section 4.4 and the doubly stochastic rating model with R_t^{gEM} as stochastic transition operators from Section 4.5.

We will first show, how to use the so-called *stochastic simulation algorithm* (SSA) for our PHCTMC in Section 4.6.1 and use the same algorithm to approximate the Lie model by an PHCTMC for each trajectory.

4.6.1. Simulation of the PHCTMC

There are several techniques in the literature concerning the efficient simulation of ICTMCs and its forward equation. For a detailed discussion, we refer to Li, Lin and Zio (2012) and Arns, Buchholz and Panchenko (2010) among many others. However, for our PHCTMC described in Section 4.4.1.2, we will use the Gillespie Stochastic Simulation Algorithm (SSA) (cf. Gillespie (2007)), which is also called *Kinetic Monte Carlo (KME)* method, on each sub-interval, where the PHCTMC is homogeneous. It turns out that this approach is very fast, because our state space has few states.[9] The algorithm is displayed in Algorithm 4.3 and works as follows:

Remember that our PHCTMC is homogeneous on each interval $[T_{k-1}, T_k]$, $k = 1, \ldots, n$, and now iterate over those intervals, i.e. assume that we are already at $t = T_{k-1}$ with current rating i. On each sub-interval we proceed as follows:

1. If $t \leq T_k$ and $(A_k)_{ii} \neq 0$ draw two uniform random numbers r_1, r_2, otherwise end and set $R_t^{i_0} = i$ on $[t, T_k]$;
2. Retrieve the exponentially distributed transition waiting time with parameter $-(A_k)_{ii}$ as
$$\tau = \frac{-\log(r_1)}{-(A_k)_{ii}} = \frac{\log(r_1)}{(A_k)_{ii}}.$$
If $t+\tau \geq T_k$ set $R^{i_0} = i$ and go to the next interval, starting with step (i), else continue to calculate the next state;
3. Now, sample from the discrete state transition distribution $\left[\frac{(A_k)_{ij}}{-(A_k)_{ii}}\right]_{j\neq i}$. This can be done by choosing the first integer j, such that $\sum_{k=1,k\neq i}^{j} \frac{(A_k)_{ij}}{-(A_k)_{ii}} > r_2$, which is equivalent to
$$\min_j \sum_{l=1,l\neq i}^{j} (A_k)_{il} > -(A_k)_{ii}\, r_2.$$
Now, go back to (i) with $R_{t+\tau}^{i_0} = j$.

[9] More details about the implementation can be found on https://github.com/kevinkamm/LieRatingTriggers/blob/main/ICTMC/ssa.m.

```
Input  : $A \in \mathbb{R}^{n,K,K}$ generator, $i_0 \in \{1, \dots, K\}$ initial state, $M \in \mathbb{N}$ number of
         trajectories
Output: $R_t^{i_0} \in [0, T] \times \mathbb{R}^{1,K,M}$ trajectories of the simulated PHCTMC starting in $i_0$
for $m \leftarrow 1$ to $M$ do
    $i \leftarrow i_0$;
    for $k \leftarrow 1$ to $n$ do
        while $t < T_k$ do
            if $A_{k,i,i} == 0$ then
                \\Absorbing state
                $t \leftarrow T_k$;
            else
                \\Calculate waiting time
                $r_1 \leftarrow$ Uniform $(0, 1)$;
                $\tau \leftarrow \frac{\log(r_1)}{A_{k,i,i}}$;
                \\Update time
                if $t + \tau \geq T_k$ then
                    $t \leftarrow T_k$;
                    break;
                else
                    $t \leftarrow t + \tau$;
                end
                \\Calculate state transition;
                temp $\leftarrow$ cumsum $\left(\frac{A_{k,i,j=1,\dots,i-1,i+1,\dots,K}}{-A(k,i,i)}\right)$;
                $r_2 \leftarrow$ Uniform $(0, 1)$;
                $j \leftarrow$ findFirst (temp $\geq r_2$);
                if $j < i$ then
                    $i \leftarrow j$;
                else
                    $i \leftarrow j + 1$;
                end
            end
            update $\left(R_t^{i_0}\right)$;
        end
        update $\left(R_t^{i_0}\right)$;
    end
end
```

Algorithm 4.3: Iterative SSA for the PHCMTC starting in $\{1, \dots, K\}$.

Let us now judge the accuracy of this simulation technique and let us have a look at Table 4.12. The errors were computed by first calculating the transition matrices from the simulated rating processes $R_t^{i,P}$, $P = \mathbb{P}, \mathbb{Q}$, by counting how many trajectories are at each state and dividing by the total amount of trajectories. The result of this is denoted by $R_t^{\mathrm{Sim},P}$

Table 4.12.: Simulation errors of the PHCTMC using Table C.6 as market default probabilities. First row is the mean error of the model rating transitions and simulated rating transitions under the historical measure. The second row contains the errors under the risk-neutral measure of the model and simulated rating transitions.

Time / Error	$t = \frac{1}{12}$	$t = \frac{3}{12}$	$t = \frac{6}{12}$	$t = 1$
$\frac{1}{K^2} \left\| R_t^{\mathbb{P}} - R_t^{\text{Sim},\mathbb{P}} \right\|_{\mathbb{R}^{K,K}}$	0.000513	0.000536	0.000648	0.0014
$\frac{1}{K^2} \left\| R_t^{\mathbb{Q}} - R_t^{\text{Sim},\mathbb{Q}} \right\|_{\mathbb{R}^{K,K}}$	0.000667	0.000891	0.000852	0.00123

and we used the Frobenius norm divided by the squared number of ratings to evaluate the error. We can see in all cases a close fit to the calibrated transition matrices.

We calculated the SSAs for each initial rating in parallel on a CPU and sampling $M = 10000$ trajectories took roughly 1 second.

In Figure 4.18, we can see an example of the simulated PHCTMC X_t under the historical measure in the top figure and under the risk-neutral measure in the bottom picture. We used the exponential change of measure with the mild probabilities of default in Table C.6. The grey lines illustrate $M = 10000$ different paths of the PHCTMC X_t and the highlighted paths in different colors are some particular examples of trajectories. Comparing the transitions under the risk-neutral measure to the historical ones, reveal a huge difference. We can see by the deep grey areas in the bottom picture that a lot more transitions occurred under the risk-neutral measure. Consistently to Figure 4.5, we can see at the right-hand side the probability distribution of the ratings at $t = 1$ year and recognize the large probability to transition from **B** to **C**. The model presents another unexpected behaviour. By the gradient of the grey areas in time from a dark grey to a light grey color, we can judge that the number of transitions reduce over time. We believe that this should be reversed for a more realistic model. And will show in the next subsection that R_t^{gEM} exhibits this desired behaviour.

4.6.2. Simulation of the Lie Model

By construction R_t^{gEM} yields pathwise valid transition operators of an ICTMC, since the Chapman-Kolmogorov identity is satisfied. We will approximate this fully inhomogeneous CTMC by an PHCTMC conditional on a path of the transition operator.

Like in a nested Monte-Carlo simulation for doubly stochastic processes, we will first sample $M_1 \in \mathbb{N}$ generators R_t^{gEM} and simulate conditioned on each path $M_2 \in \mathbb{N}$ trajectories of the rating model with Algorithm 4.3. In total, we will have $M := M_1 \cdot M_2 \in \mathbb{N}$ paths.

We calculated the nested SSA[10] for each initial rating in parallel on a CPU and sampling $M = M_1 \cdot M_2$ with $M_1 = 1000$ different rating transition operators simulated by R_t^{gEM} and $M_2 = 10000$ SSA trajectories took roughly 64 minutes using a machine with 64 CPU cores.

Let us now judge the accuracy of this simulation technique in this setting. Therefore, let us have a look at Table 4.13. The errors were computed by first calculating the transition

[10] More details about the implementation can be found on https://github.com/kevinkamm/LieRatingTriggers/blob/main/Lie/nestedSSA.m.

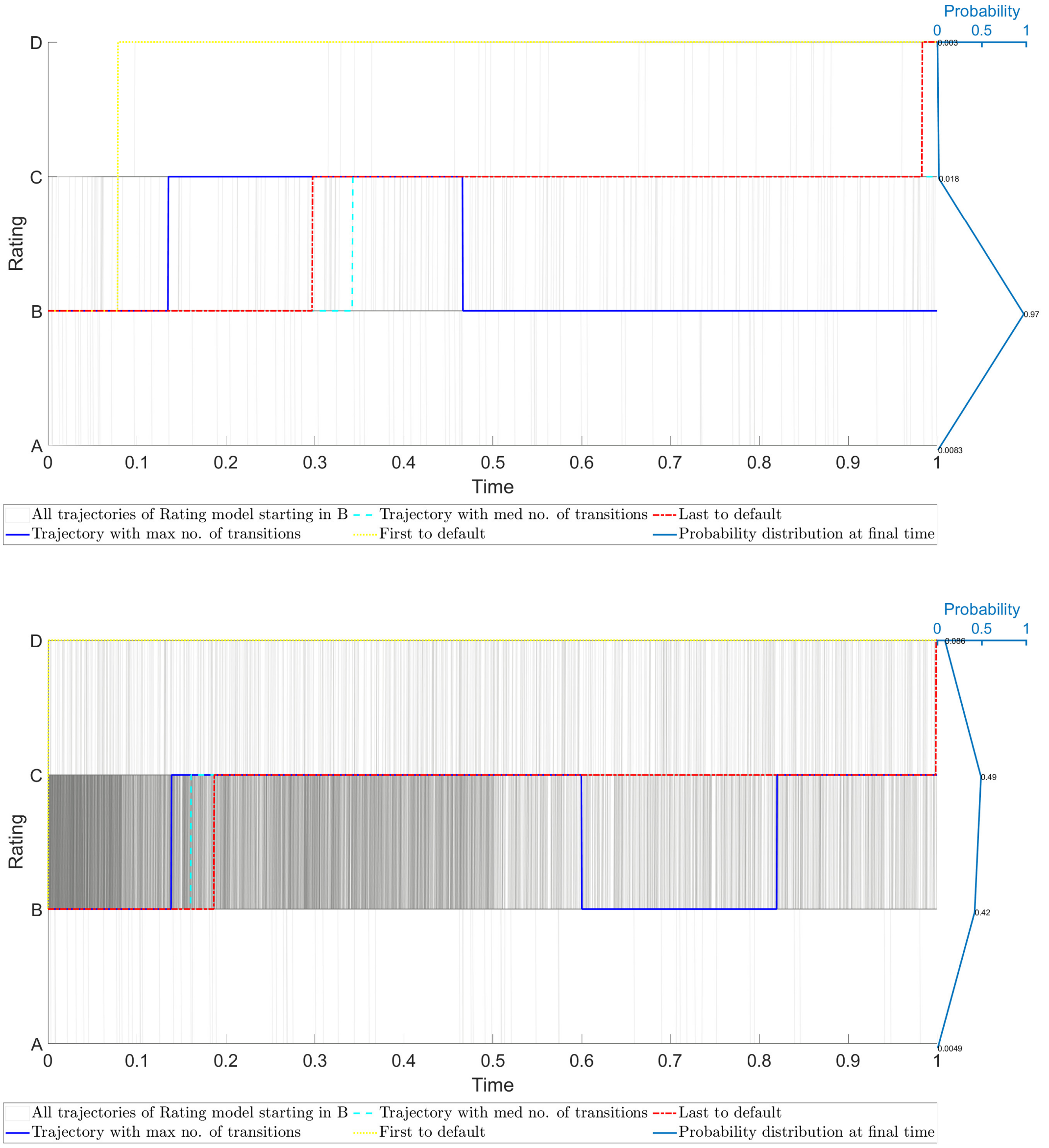


Figure 4.18.: Simulated trajectories of PHCTMC X_t calibrated to TimeGAN's data set starting in rating **B**. The top picture is under measure $\mathbb{P}$ and the bottom picture is under measure $\mathbb{Q}$ using Table C.6 as market default probabilities.

Table 4.13.: Simulation errors of R_t^{gEM} using Table C.6 as market default probabilities. First row is the mean error of the model rating transitions and simulated rating transitions under the historical measure. The second row contains the errors under the risk-neutral measure of the model and simulated rating transitions.

Time Error	$t = \frac{1}{12}$	$t = \frac{3}{12}$	$t = \frac{6}{12}$	$t = 1$
$\frac{1}{K^2} \left\| R_t^{\mathbb{P}} - R_t^{\text{Sim},\mathbb{P}} \right\|_{\mathbb{R}^{K,K}}$	7.7545e-04	0.0017	0.0030	0.0052
$\frac{1}{K^2} \left\| R_t^{\mathbb{Q}} - R_t^{\text{Sim},\mathbb{Q}} \right\|_{\mathbb{R}^{K,K}}$	8.3978e-04	0.0023	0.0044	0.0083

matrices from the simulated rating processes $R_t^{i,P}$, $P = \mathbb{P}, \mathbb{Q}$, by counting how many trajectories are at each state and dividing by the total amount of trajectories. The result of this is denoted by $R_t^{\mathrm{Sim},P}$ and we used the Frobenius norm divided by the squared number of ratings and the mean over all trajectories to evaluate the error. We can see slightly worse errors than in Table 4.12, which is to be expected, since the model is more complicated. However, the errors are still satisfactorily small.

Simulating this model takes a lot longer than simulating a PHCTMC with one deterministic generator, which is expected, since we are sampling 1000-times more paths. As a side note, sampling only $100 \cdot 100 = 10000$ paths takes roughly 16 seconds with 6 CPU cores. In our experiments, we found that the inner simulations for the SSA M_2 should at least be 1000 to guarantee a small simulation error.

In Figure 4.19, we can see an example of the simulated ratings X_t using R_t^{gEM} under the historical measure in the top figure and under the risk-neutral measure in the bottom picture. We used the exponential change of measure with the mild probabilities of default in Table C.6. The grey lines illustrate $M = 100 \cdot 100 = 10000$ different paths of X_t and the highlighted paths in different colors are some particular examples of trajectories. Similar to Figure 4.18, we can see that the transitions under the risk-neutral measure compared to the historical ones, reveal a huge difference. We can see by the deep grey areas in the bottom picture that a lot more transitions occurred under the risk-neutral measure. Consistently to Figure 4.15, we can see at the right-hand side the probability distribution of the ratings at $t = 1$ year and recognize the large probability to transition from **B** to **C**. Compared to the PHCTMC in the previous section, we see a much smoother behaviour and the number of transitions seem to increase the more we progress in time, which seems to be more realistic.

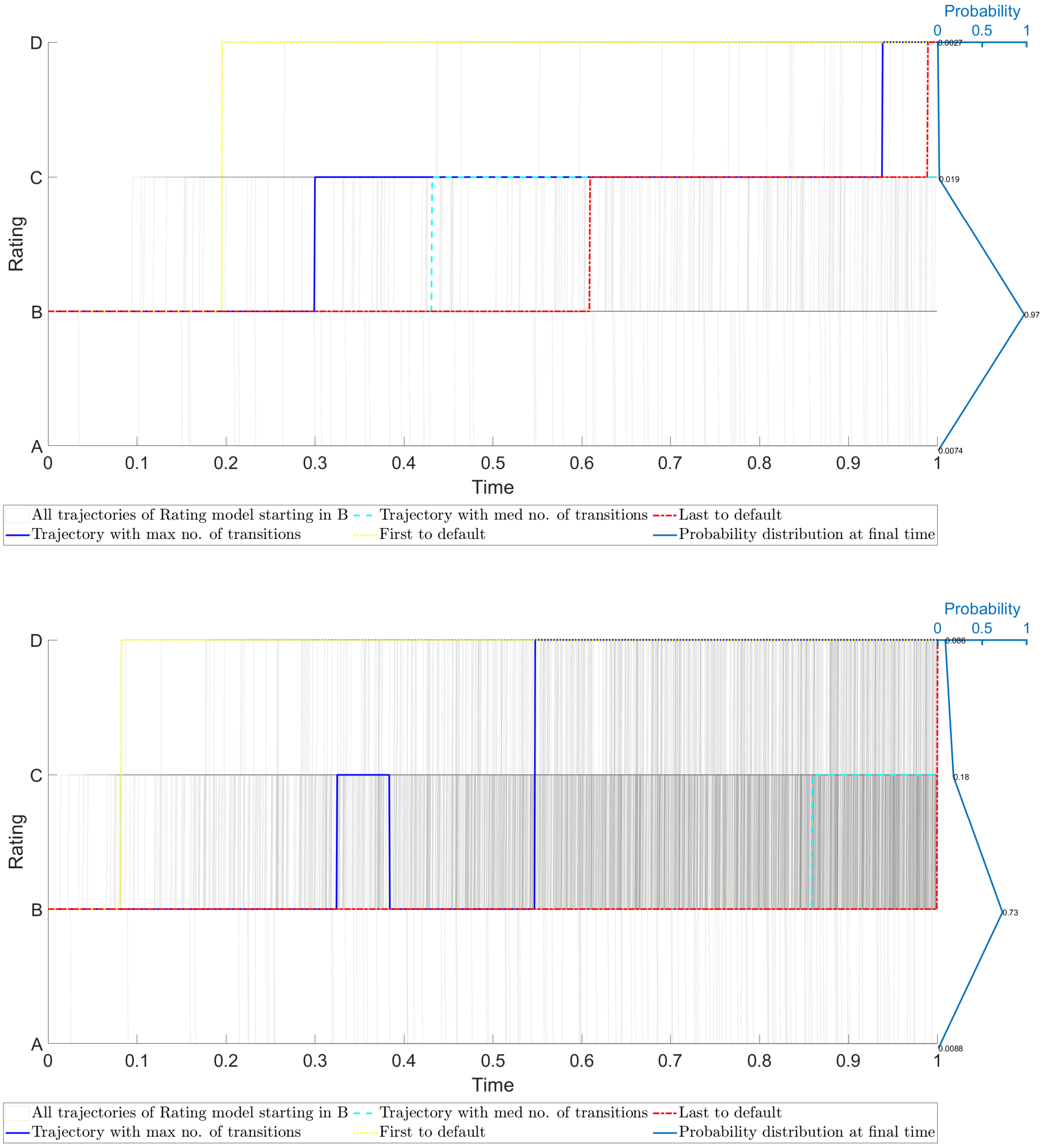


Figure 4.19.: Simulated trajectories X_t using the transition operators from R_t^{gEM} starting in rating **B**. The top picture is under measure $\mathbb{P}$ and the bottom picture is under measure $\mathbb{Q}$ using Table C.6 as market default probabilities.

4.7. Application to Rating Triggers for Collateral-Inclusive Bilateral Valuation Adjustments

In this section, we study bilateral credit and debit valuation adjustments (hereafter referred to as CVA and DVA) of a portfolio of trades between two parties having signed a collateral agreement dependent on ratings. The main references for the general theory of XVA are Oosterlee and Grzelak (2019): *pp. 375 ff. Chapter 12.3 Credit Valuation Adjustment and Risk Management* and Brigo, Morini and Pallavicini (2013): *pp. 305 ff. Chapter 13 Collateral, Netting, Close-Out and Re-Hypothecation*. Recall that CVA and DVA are adjustment to the fair price of a financial portfolio accounting for the potential loss in case of default of the counterparty and the owner, respectively. These are usually defined as an average of the exposure (positive and negative, respectively) weighted by the probability of default. The two parties usually sign a so-called netting agreement, so as to consider the exposure at portfolio level (as opposed to trade-wise). Attached to the netting agreement, one often has a Credit Support Annex (CSA) by which each of the two parties further agrees to interchange securities (referred to as collateral) to reduce the exposure of the other party. In the case of bankruptcy, the collateral account can be used to mitigate the losses of the non-defaulting party, although collateral is often non-segregated and therefore also at risk. Since posting collateral is another expense for an entity, it is desirable to keep the postings as small as possible while simultaneously keeping the losses due to a default event small as well. To this goal, more and more CSAs specify thresholds of permitted unsecured exposure in terms of the credit quality of the parties: the higher the credit quality of a party, the smaller the amount collateral it has to post (and the larger the unsecured exposure of the other party).

A customary way to measure the credit quality of an entity is to use credit ratings. A high rating means that the entity is very likely to fulfill its financial obligations towards its contracting party, while a low rating associates an increasing risk for meeting the aforementioned obligations. In this line of thought, the default can be viewed as the worst possible rating. CSAs dependent on ratings are said to have rating triggers: a change of rating of one of the parties triggers a change of threshold of that party.

Since the exposure depends on the amount of collateral posted or received, to compute CVA and DVA in presence of a CSA with rating triggers, it is necessary to model the rating processes of the contracting parties. For this, we proposed two different models in Section 4.4 and Section 4.5 by considering deterministic and stochastic rating transition models, respectively.

In this section, we will use R_t^{gEM} as our rating transition model and refer to Kamm (2022) for similar tests in the PHCTMC model seen in Section 4.4.

Since the doubly-stochastic rating process X_t corresponding to R_t^{gEM} is pathwise Markovian, we would like to point out that the memoryless property of such processes make them somewhat unrealistic as models for the rating process, since an entity with a history of successive downgrades is more likely to be considered risky than a competitor with long-time constant rating. We have already seen one possibility to calibrate a non-Markovian model by using R_t^{CIR} as a rating transition model but we will leave further studies of this model for future research. Also self-exciting processes, such as Hawkes processes, could be explored to account for the increasing riskiness for fast successive downgrades.

In most of the literature, time-homogeneous models for rating transitions are considered. However, it is empirically evident that a time homogeneity assumption is often violated. We have already seen in our previous experiments that we can calibrate our fully inhomogeneous model in a meaningful way to the data, which is an improvement to the existing models in the literature.

For a more detailed discussion of the Markovianity and time-homogeneity we refer the reader to Lencastre et al. (2014).

Throughout this section, we are taking the point of view of a bank having a portfolio of deals with a counterparty. The two parties have signed a netting set agreement with a CSA having rating-dependent thresholds. The stochastic process representing the future mark-to-market of the portfolio is denoted by V_t. We will assume that both contracting parties are subject to default and the default time will be denoted by τ_B and τ_C for the bank and the counterparty, respectively. Additionally, we will suppose that the same rating matrices apply to both, i.e. they are in the same industrial sector and the consideration of two different sectors is discussed in Remark 4.7.1. For our illustration, we will assume that the bank has the highest rating today, whereas the counterparty has a mid-range rating today, since we expect a bank to default less likely than the majority of companies. The evolution of their ratings over time will be denoted by X_t^B and X_t^C, respectively, and we will set $X_t := \left(X_t^B, X_t^C\right)$ to shorten notation.

Let C_t be the stochastic process representing the value of the collateral account. In particular $C_t > 0$ if the collateral is received by the bank. We will assume for simplicity that C_t depends on V only through the value V_t: more precisely we will suppose that $C_t := f\left(V_t, X_t\right)$. In particular to avoid path dependencies, we assume there are no minimum transfer amounts, the impact of this assumption being not material for our purposes.

We will discuss the following three scenarios of collateral agreements:

1. *uncollateralized*, i.e. no collateral is interchanged and $f \equiv 0$;
2. *perfectly collateralized*, i.e. collateral is posted instantaneously at a discrete set of times, e.g. daily, and is equal to the mark-to-market $(V_t = C_t)$, $f(v, r) = v$;
3. *rating-trigger dependent*, more precisely we will focus on the case of thresholds depending on rating (see below for a description of f in this case).

The relation of the bank to the counterparty is illustrated in Figure 4.20 and reads as follows[11]:

To illustrate the impact of collateral in risk mitigation, let us assume instantaneous posting and no rehypothecation of collateral for simplicity. Assume the counterparty (but not the bank) defaults at time τ. Then on the one hand we have the value of the portfolio V_τ and on the other hand we have the value of the collateral account C_τ. We distinguish four cases:

- $V_\tau \geq 0, C_\tau \geq 0$: the portfolio generates a positive exposure for the bank but this is

[11] We will use the same conventions as in Brigo, Morini and Pallavicini (2013): *pp. 310 ff. Chapter 13.2 Bilateral CVA Formula under Collateralization*, in particular $X^+ = \max(X, 0)$ and $X^- = \min(X, 0)$.

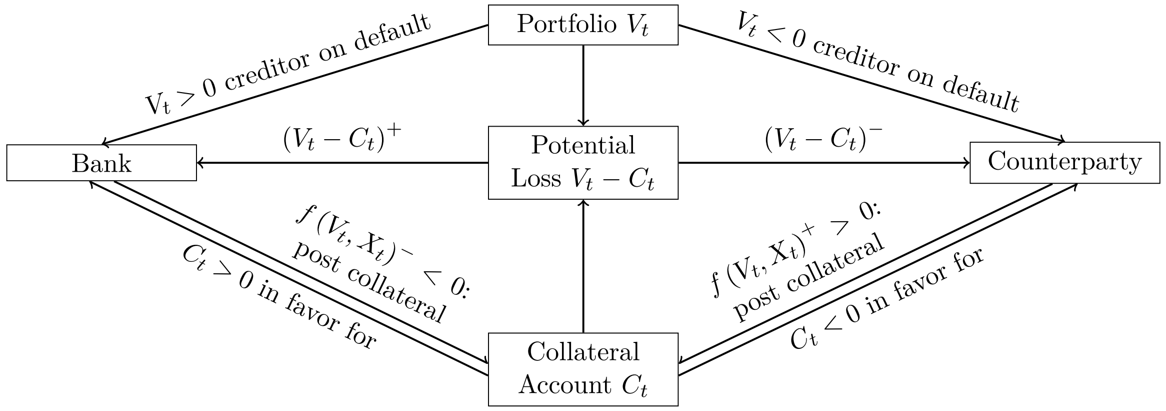

Portfolio V_t
$V_t > 0$ creditor on default
$V_t < 0$ creditor on default
Bank
$(V_t - C_t)^+$
Potential Loss $V_t - C_t$
$(V_t - C_t)^-$
Counterparty
$f(V_t, X_t)^- < 0$: post collateral
$C_t > 0$ in favor for
$f(V_t, X_t)^+ > 0$: post collateral
$C_t < 0$ in favor for
Collateral Account C_t

Figure 4.20.: Illustration of bank and counterparty relations in terms of exposure and collateral agreements.

mitigated by the collateral (which can be fully retrieved by the bank because of no rehypotecation). Therefore the outstanding claim is $V_\tau - C_\tau$.

- $V_\tau \geq 0, C_\tau \leq 0$: although the portfolio generates a positive exposure for the bank, the bank had posted collateral just before default. Because of no rehypothecation, the bank can fully get back its collateral and the outstanding claim is therefore V_τ.

- $V_\tau \leq 0, C_\tau \geq 0$: the counterparty gets back the collateral posted to the bank and also gets the value of the portfolio $|V_\tau|$.

- $V_\tau \leq 0, C_\tau \leq 0$: the counterparty keeps the collateral posted by the bank and also gets the remaining value of the portfolio $|V_\tau - C_\tau|$.

The behaviour in case of default of the bank is symmetrical.

Additionally, the individual collateral postings depending on the collateral agreement are depicted by $f(V_t, X_t)^-$, meaning that the bank has to post collateral if this value is greater than zero and its analogue for the counterparty is given by $f(V_t, X_t)^+$.

A comprehensive explanation of all default events in this bilateral setup can be found in Brigo, Morini and Pallavicini (2013): *pp. 311–312 Chapter 13.2.1 Collection of CVA Contributions*.

Next, we will discuss the impact of rating triggers compared to the aforementioned scenarios of collateral agreements on collateral-inclusive CVA, DVA and BVA, followed by a discussion on the pre-default distribution of the rating processes in Section 4.7.2.

4.7.1. XVA with Different Collateral Agreements

We are interested in the impact of rating triggers on BVA, CVA and DVA (Collateralized Bilateral, Credit, Debit Valuation Adjustments) without the possibility of rehypothecation and zero interest rate at mid-market to simplify the investigation.[12]

[12] More details about the implementation can be found on https://github.com/kevinkamm/LieRatingTriggers/blob/main/Lie/main.m.

Before we dive into this topic, let us first of all discuss our benchmark portfolio. Since our main purpose is to analyse the general behaviour of the rating model, we are not interested in setting up an accurate model for the computation of V_t. In particular, the V_t that we consider does not represent the value of a portfolio of real deals. Rather we decided to simulate V_t using a number of independent Brownian motions with different volatilities and life-times l^i to account for the cash-flows of the portfolio. To be more precise[13]

$$V_t := V_0 + \sigma_0 W_t^0 + \sum_{i=1}^{n} \sigma_i W_t^i \mathbb{1}_{t \leq l^i},$$

where $V_0 \in \mathbb{R}_{\geq 0}$ is the initial value, W^i, $i = 0, \ldots, n$, $n \in \mathbb{N}$, are independent Brownian motions, $\sigma_i \in \mathbb{R}$ are volatilities and $l^i \in [0, T]$ are uniformly distributed random variables describing the different life-times of the cash-flows. In the experiment we use $V_0 = 0$, $n = 24$ and σ_i are the standard normal random variables multiplied by 10 for scaling and its sign indicates a positive or negative cash-flow (from the bank perspective). Also, notice that we designed the portfolio in such a way that at least one cash-flow survives till T by not adding a finite life-time to W_t^0.

Now, let us briefly recall the relevant definitions of XVA (cf. Brigo, Morini and Pallavicini (2013): *p. 314 Equation 13.4, p. 316 Equation 13.10*) without re-hypothecation

$$\begin{aligned} \mathrm{BVA}(t, T, \mathrm{C}) &:= \mathrm{DVA}(t, T, \mathrm{C}) - \mathrm{CVA}(t, T, \mathrm{C}), && (4.7.1)\\ \mathrm{DVA}(t, T, \mathrm{C}) &:= -\mathbb{E}^{\mathbb{Q}}\left[\mathbb{1}_{\tau=\tau_B<T} \, \mathrm{LGD}_B \, \left(V_\tau^- - C_\tau^-\right)^- \middle| \mathcal{G}_t \right], && (4.7.2)\\ \mathrm{CVA}(t, T, \mathrm{C}) &:= \mathbb{E}^{\mathbb{Q}}\left[\mathbb{1}_{\tau=\tau_C<T} \, \mathrm{LGD}_C \, \left(V_\tau^+ - C_\tau^+\right)^+ \middle| \mathcal{G}_t \right], && (4.7.3) \end{aligned}$$

where $\mathcal{G}_t$ is the filtration containing all the default-free market information plus default monitoring. These values are calculated under a risk-neutral measure $\mathbb{Q}$, which explains why we were interested in changing the measure of our rating model from the historical probabilities $\mathbb{P}$ to the risk-neutral measure in the first place.

The evaluation of the collateral account at the exact time of the default event, i.e. C_τ, might seem confusing. We could imagine a scenario in which bonds or stocks could be used as collateral, making it necessary to evaluate the collateral account at the default event. In our case, we will assume that the collateral account will be a pure cash account, meaning that upon a default event the value will not be updated from its previous value $C_{\tau-}$. Therefore, it is very important to study the distribution of ratings prior to default, which is subject to Section 4.7.2.

We now describe the function $f(\mathrm{V}_t, X_t)$ in the case of rating-triggers dependent agreements, following Brigo, Morini and Pallavicini (2013): *pp. 316ff. Chapter 13.5.2 Collateralization Through Margining*. Let $r_i^x \geq 0$, $x \in \{B, C\}$, $i = 1, \ldots, K$ denote the threshold for the party x in case x has rating i: this means that the maximum unsecured exposure of the

[13] More details about the implementation can be found on https://github.com/kevinkamm/LieRatingTriggers/blob/main/Lie/portfolio.m.

other party will be at most r_i^x.

Now, we introduce the rating triggers ρ^x with corresponding thresholds r_i^x as

$$\rho^x(i) \coloneqq \sum_{j=1}^{K} r_j^x \mathbb{1}_j(i).$$

As a small example, setting for all $i = 1, \dots, K$ and $x = \{B, C\}$ the thresholds $r_i^x = +\infty$ lead to the uncollateralized scenario and $r_i^x = 0$ to the perfectly collateralized scenario.

The amount of collateral to be posted by the bank at time t_j is then

$$\left(\mathrm{V}_{t_j} + \rho^B(X_{t_j}^B)\right)^- - C_{t_j-}^-.$$

For the counterparty we have analogously

$$\left(\mathrm{V}_{t_j} - \rho^C(X_{t_j}^C)\right)^+ - C_{t_j-}^+.$$

As aforementioned, we assume for simplicity that the value C_{t_j} of the collateral account at time t_j is equal to $C_{\beta(t_j)}$ where $\beta(u)$ is the last collateral posting date before u. In particular, we assume there is no remuneration on the collateral account. We then have

$$C_{t_0} \coloneqq 0, \quad C_{t_n} \coloneqq 0, \quad C_{u-} \coloneqq C_{\beta(u)}.$$

$$C_{t_j} \coloneqq C_{t_j-} + \left(\left(\mathrm{V}_{t_j} + \rho^B(X_{t_j}^B)\right)^- - C_{t_j-}^-\right) + \left(\left(\mathrm{V}_{t_j} - \rho^C(X_{t_j}^C)\right)^+ - C_{t_j-}^+\right).$$

This can be rewritten as

$$C_{t_j} = \left(\mathrm{V}_{t_j} + \rho^B(X_{t_j}^B)\right)^- + \left(\mathrm{V}_{t_j} - \rho^C(X_{t_j}^C)\right)^+ =: f\left(\mathrm{V}_{t_j}, X_{t_j}\right).$$

We will use in all experiments 365 posting dates per year. In Figure 4.21, one can see one trajectory of the portfolio, collateral account and individual postings by both counterparties in the top picture. The picture in the middle indicates the ratings of both counterparties over time for this particular trajectory and the bottom picture shows the corresponding threshold for each point in time. The orange boxes are magnifications of the indicated sections in the figure.

One can see that for this choice of trajectory, the bank has no rating transition and the counterparty has many, ranging through all thresholds defined in Table 4.14.

At the section “Zoom A”, one can see that neither the bank nor the counterparty has to post collateral. In the region, where the portfolio is negative, the bank does not have to post collateral, since the allowed threshold is not exceeded. Same for the counterparty in the region, where the portfolio is positive. In section “Zoom B”, the portfolio gets too negative and the threshold for the bank is exceeded, such that the bank has to post collateral (green dots). After that the portfolio gets positive and the counterparty has to post collateral instead of the bank. At “Zoom C”, the rating of the counterparty is improving and we can

Table 4.14.: Rating thresholds for both counterparties. Values in Euro.

A	**B**	**C**	**D**
Bank			
$10 \cdot 10^6$	$5 \cdot 10^6$	0	0
Counterparty			
$10 \cdot 10^6$	$5 \cdot 10^6$	0	0

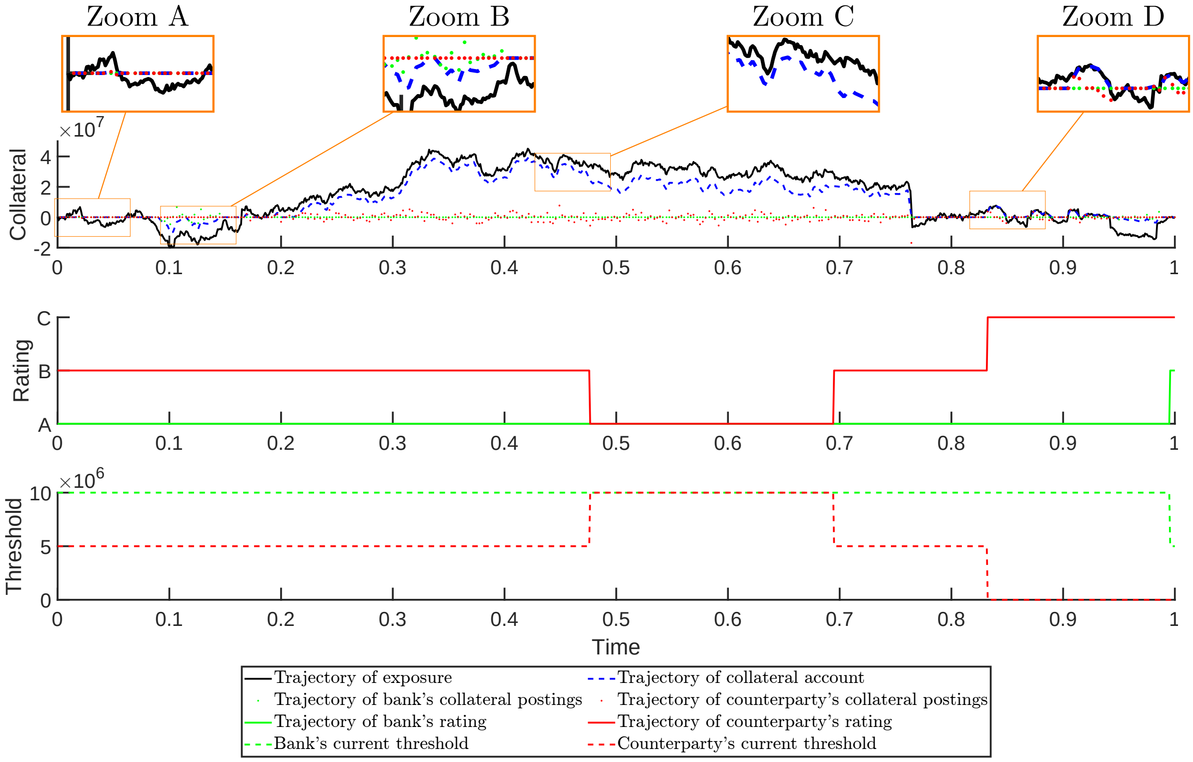


Figure 4.21.: One trajectory of a collateral agreement with rating triggers. The top picture shows the collateral account and portfolio over time, the middle one the rating evolution and the bottom one the corresponding rating thresholds.

see that it has to post less collateral than beforehand, which is apparent from the distance between the black bold (the portfolio) and blue dashed line (the collateral account). After that at around $t = 0.7$ the ratings drops back to **B** and the distance between the exposure and collateral gets smaller again. At "Zoom D", the rating of the counterparty drops further to **C**, forcing the perfectly collateralized scenario, since the threshold is zero for this rating. Therefore, the dashed blue line follows the black bold line perfectly, whenever its above zero. At the very end we can see that the rating of the bank drops to **B** without any impact.

In Table 4.15 are the values of DVA (4.7.2), CVA (4.7.3) and BVA (4.7.1) using the Loss-Given-Default $\text{LGD}_B = 0.6$, $\text{LGD}_C = 0.6$ and $M = 10000$ simulations for the three collateral agreements: no collateralization, perfect collateralization and collateralization with rating triggers.

One can see that the collateralization with rating triggers lies in between the values of the uncollateralized case and the perfectly collateralized case, which is the expected behaviour,

Table 4.15.: XVA with the different collateral agreements (no, perfectly and rating triggers) using Table C.6 as market default probabilities and $\mathrm{LGD}_B = 0.6$, as well as $\mathrm{LGD}_C = 0.6$ with $M = 10000$ simulations and thresholds defined in Table 4.14.

XVA	Uncollateralized	Rating Triggers	Perfectly collateralized
DVA	1015922	587335	351276
CVA	896413	376938	271492
BVA	119509	210397	79784

because as illustrated in Figure 4.21 one has a possible transition from unsecured money to the perfectly collateralized scenario, where rating thresholds are zero. The difference to the perfectly collateralized case is that there can be transitions from high ratings to default in one instant, which will be subject of the next subsection.

Remark 4.7.1. In this framework of rating transition modelling, it is straightforward to include the possibility of counterparties in two different sectors, e.g. financial and corporate.

Suppose we are in the setting of Section 4.5.2.4 and define two independent processes in the Lie algebra, whose components are

$$dA_t^{1,i} = \left|A_t^{1,i}\right|^{a_i^1} dt, \quad dY_t^{1,i} = b_i^1 dt + \sigma_i^1 dW_t^{1,i},$$
$$dA_t^{2,i} = \left|A_t^{2,i}\right|^{a_i^2} dt, \quad dY_t^{2,i} = b_i^2 dt + \sigma_i^2 dW_t^{2,i}.$$

The Brownian motions $W_t^{j,i}$, $i = 1, \dots, (K-1)^2$, $j = 1, 2$, are assumed to be mutually independent.

Now, instead of defining two individual changes of measures for both processes, we will define $\kappa := (\kappa^1, \kappa^2) \in \mathbb{R}^{2(K-1)^2}$ and $W_t := \left(W_t^{1,1}, \dots, W_t^{1,(K-1)^2}, W_t^{2,1}, \dots, W_t^{2,(K-1)^2}\right)^\top$. The Girsanov transform now takes again the form

$$L_t := \exp\left(\int_0^t \kappa_s \cdot dW_s - \frac{1}{2}\int_0^t |\kappa_s|^2 \, ds\right)$$

and the corresponding measure is given by

$$\left.\frac{d\mathbb{Q}^\kappa}{d\mathbb{P}}\right|_{\mathcal{F}_t} := L_t,$$

where $\mathcal{F}_t := \sigma(W_t)$ with $\mathbb{Q}^\kappa$ Brownian motion $W_t^\kappa := W_t - \int_0^t \kappa_s ds$.

The dynamics of A_t^1 and A_t^2 under this new measure are given by

$$dA_t^{\kappa,1,i} = \left|A_t^{\kappa,1,i}\right|^{a_i^1} dt, \quad dY_t^{\kappa,1,i} = \left(b_i^1 + \sigma_i^1 \kappa_s^{1,i}\right) dt + \sigma_i^1 dW_t^{\kappa,1,i},$$
$$dA_t^{\kappa,2,i} = \left|A_t^{\kappa,2,i}\right|^{a_i^2} dt, \quad dY_t^{\kappa,2,i} = \left(b_i^2 + \sigma_i^2 \kappa_s^{2,i}\right) dt + \sigma_i^2 dW_t^{\kappa,2,i}.$$

This means that we can repeat the calibration procedure described in Section 4.5.2.1 and Section 4.5.2.4 simultaneously for both processes using data from two different sectors, while ensuring that they have dynamics under the same risk-neutral measure $\mathbb{Q}^\kappa$.

This trick is not possible for the ICTMC model and copula methods as in Bielecki, Cialenco and Iyigunler (2012) have to be considered, which is subject to future research in our model.

4.7.2. Pre-Default Rating Distribution

As it is apparent from the definition of the rating thresholds, it is important to study the distribution of the rating process one time-instant prior to default, because this will determine the unsecured amount of money at the default event. We will call this henceforth pre-default distribution and will also compare the distribution under $\mathbb{P}$ to the one under $\mathbb{Q}$ with the help of Figure 4.22, which were obtained by Monte-Carlo simulation.

Now, let us have a closer look at Figure 4.22. First of all, one can see the pre-default distribution under the measure $\mathbb{P}$ in the top picture and under the measure $\mathbb{Q}$ in the bottom picture. Disregarding the individual colors, the probability of being in a certain rating prior to default is given by the total height of the column. The composition of the individual colors of each column indicates the contribution of each starting rating, e.g. in the third column we can see that the most prominent contributions are resulting from the initial rating **C**, but there are contributions of the other ratings as well.

In the market, it can be observed that the default probabilities in the risk-neutral world are usually higher than the default probabilities quoted under the historical measure in the rating matrices. This phenomenon has an impact in our model on all other ratings as well, which can already be seen in Figure 4.19 by the spread of the grey lines indicating all simulated trajectories. In the risk-neutral world there seem to be more transitions than in the historical world causing this spread of grey lines. The reason for this is that the calibration of this model has essentially one parameter for each rating because $h_i \in \mathbb{R}^K$. Therefore, the higher probability of default under the measure $\mathbb{Q}$ compared to the one under $\mathbb{P}$ has a significant impact on the other ratings as well.

We can see that under the measure $\mathbb{P}$, the top picture, almost all the defaults had a prior rating of C, while under the measure $\mathbb{Q}$, this is still the most prominent case but significantly smaller. It is more likely under the measure $\mathbb{Q}$ that a company starting with a high rating defaults and this without transitioning to the rating prior to default first, which is indicated by the different heights of the each individual color for each rating.

It is yet an open question and needs thorough economical investigation whether this behaviour makes sense or not, because it has a significant impact on the performance of collateralization with rating triggers. To be more precise, the more likely it is that a company starting in a good rating defaults without first transitioning to a rating, where a low threshold is defined, the more unsecured money we have at a default event.

In Kamm (2022), we show similar results using the PHCTMC from Section 4.4 with the exponential change of measure.

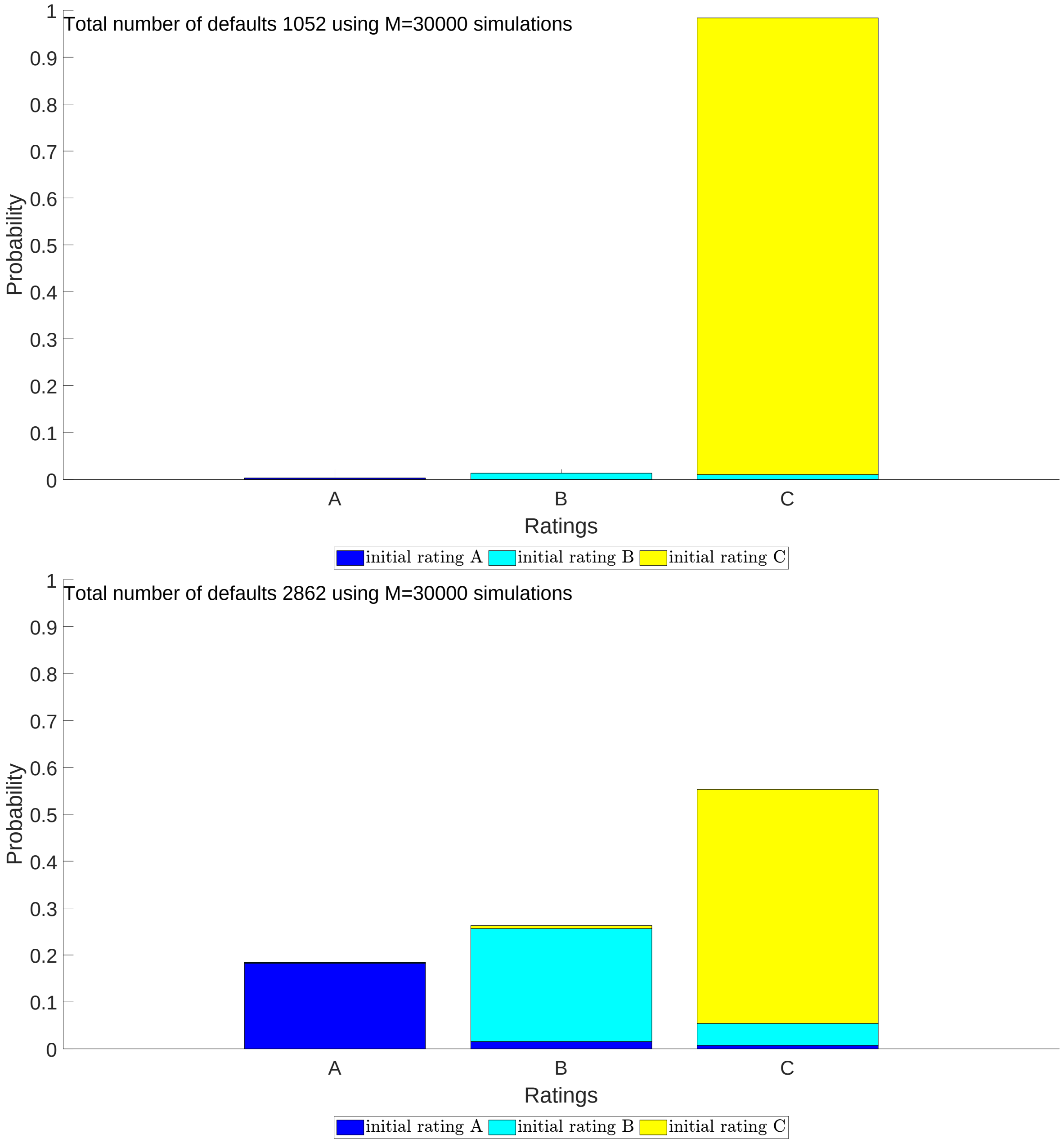


Figure 4.22.: Pre-default distribution using R_t^{gEM}. The upper picture is under the historical measure and the lower plot under the risk-neutral measure using the exponential change of measure with Table C.6 as market default probabilities.

4.8. Conclusion

In this chapter, we have seen two methods to compute rating transition matrices from historical rating data of companies, namely the cohort method (Section 4.2.1.1) and the Aalen-Johansen estimation (Section 4.2.1.2). We have discussed the issue of the withdrawal of companies and demonstrated how to use a TimeGAN for a time-series analysis of the historical Aalen-Johansen rating transition matrices. We used the generative abilities of the DNN to get information about the distribution of the historical data and calibrated two different models to it under the historical measure.

The first model was a piecewise homogeneous continuous-time Markov chain (Section 4.4) with deterministic transition operators. We have demonstrated how the most popular change of measure techniques perform on our dataset and found an improvement by introducing a penalized calibration procedure.

For the second model, we formulated rating transitions as processes on Lie groups by using its relation to its Lie algebra and imposing SDEs there. We showed two different approaches, first the direct exponential mapping in Section 4.5.2.2 and showed numerical results using CIR processes in the Lie algebra. Afterwards, we demonstrated, how the geometric Euler method can be applied to preserve the Chapman-Kolmogorov equations in Section 4.5.2.3. In Table 4.16, we compare the two methods and their features.

R_t^{CIR}	R_t^{gEM}
Simple method with fast calibration	More complex with slower calibration
Needs only L_t^i to be positive	Requires that L_t^i has monotonically increasing paths
Satisfies all rating properties well	Satisfies all rating properties perfectly
Does not satisfy the Chapman-Kolmogorov equations	Satisfies Chapman-Kolmogorov equations
Default column is not absorbing	Default column is absorbing

Table 4.16.: Comparison of R_t^{CIR} and R_t^{gEM}.

In the case of R_t^{gEM}, we showed similar to Section 4.4, how to apply the Jarrow-Lando-Turnbull and exponential change of measure technique to account for the CDS quotes (Section 4.5.2.4). We found, as in the case of ICTMCs, that the exponential change of measure performed very well.

Afterwards, we showed in Section 4.6 how to simulate the PHCTMC model and the Lie model using their generator and process in the Lie algebra by using a (nested) stochastic simulation algorithm.

In the end, in Section 4.7.1 we used the rating process X_t obtained from the SSA to compute CVA and DVA. We saw, as expected, an improvement using rating triggers compared to the uncollateralized case and discussed the influence of the pre-default distribution.

For future research, we could try to learn the historical generators instead of the rating transitions. In this case, we would be able to calibrate the SDE on the Lie algebra to the fake generators. Also novel neural network architectures called DeepONets (cf. Lu, Jin and

Karniadakis (2019)) could be thought of in this framework.

Another line of research could involve adding an additional network to the TimeGAN, which outputs the calibrated parameters of the target SDE directly. It would be beneficial to link the Autoencoder or Supervisor network to this new network to exploit dimensionality reductions, especially in the case of larger rating transition matrices.

Conclusion and Future Research 5

In this thesis, we studied three different topics. In Chapter 2, we have seen how to derive the Itô-stochastic Magnus expansion for general linear matrix-valued SDEs, provided a convergence result up to a strictly positive stopping time, and gave an asymptotic error bound for the cumulative distribution function of the stopping time. We have demonstrated that the numerical scheme arising from the truncation of the Magnus series expansion can be applied to solve SPDEs with one and two spatial dimensions accurately and efficiently. In particular, we have shown detailed experiments in the case of the stochastic heat equation and the stochastic Langevin equation.

This was followed by a model for negative interest rates based on the difference of two independent CIR processes in Chapter 3. We have seen that in this affine setting the analytical tractability of the original CIR process for short-rate modelling is preserved and gave evidence in Section 3.3 that the model can reproduce the market term-structure with negative interest rates. We extended the model in Section 3.4 by a deterministic shift, making it exogenous, and calibrated it with the aid of the Gram-Charlier expansion to the market swaption surface. The model resulted in a good fit to Bermudan swaption prices compared to Bloomberg's Hull-White one factor model and a good accuracy for constant maturity swaps.

In Chapter 4, we studied rating transition modelling in detail. We have discussed problems arising from the market data and shown how to use a TimeGAN to analyse the historical rating matrices by applying the Aalen-Johansen estimator to the historical rating time series data of individual entities. We took two different perspectives of modelling rating transitions. In Section 4.4, we started from the point of view of a rating model for an entity and impose that it evolves like an inhomogeneous continuous-time Markov chain. This lead to deterministic transition operators and generators, which we calibrated with a penalized method simultaneously to rating matrices under the historical measure and to market default probabilities under the risk-neutral measure. We have shown the impact of the two most popular choices of change of measure on the rating transition evolution under the risk-neutral measure and concluded that the exponential change of measure behaves better than the Jarrow-Lando-Turnbull change of measure.

In Section 4.5, we began our investigation of rating transition models starting from SDEs taking values in $\mathbb{R}$, used a basis transformation to translate the dynamics to a positive half-space of a Lie algebra and applied the exponential map to define a fully inhomogeneous and stochastic rating transition model. We discussed its calibration and showed a different technique for the calibration procedure by splitting the individual calibrations under the historical measure and the risk-neutral measure. For the calibration under the historical measure, we used the moments of the distributions of rating matrices learned by the TimeGAN and demonstrated in three different cases of market default probabilities the behaviour of the model. We confirmed our conclusion from the ICTMC model that the exponential change of measure leads to more reasonable results than the JLT change of measure.

Last but not least, we applied our results from Section 4.5 in the case of R_t^{gEM} to compute CVA and DVA with rating triggers. We have shown that the rating dependent thresholds of unsecured money balances the issue of minimizing the potential loss at default under the constraint of small collateral postings. In Section 4.7.2, we discussed that the large jumps from high ratings to default exclude the possibility of the rating trigger model to reach the same value of XVA as in the perfectly collateralized case.

5.1. A Unified Model for XVA

For the evaluation of collateral-inclusive bilateral Credit-/Debit Valuation adjustments in markets with possible negative interest rates, we propose the following model:

Use R_t^{gEM} as the rating transition model with the exponential change of measure technique to unify default and rating models under the risk-neutral measure consistent with historical data.

For the discount factor, we suggest to use the difference of two independent CIR processes with deterministic shift extension calibrated to the market swaption surface.

The short-rate model may be correlated to the Brownian motions appearing in R_t^{gEM}, whose impact is left for future research. More opportunities for future research are discussed in the next section.

5.2. Future Research

There are a lot of opportunities for future research. We will discuss briefly some possibilities in the next subsections starting with the Magnus expansion. This is followed by opportunities for future research in the negative interest rate framework in Section 5.2.2. Afterwards, we will discuss applications and extensions of the rating transition model in Section 5.2.3–5.2.6.

5.2.1. Magnus Expansion

For the Magnus expansion there are many possibilities for future research. On the one hand, we could try to extend the results to a non-linear setting. Attempts in this direction have been made in Wang et al. (2020) and Yang et al. (2021).

On the other hand, it would be interesting to study the Magnus expansion for matrix-valued SDEs with jumps or with respect to the fractional Brownian motion.

This would yield a novel numerical scheme for SPDEs with jumps or rough SPDEs with maybe the same benefits as we have seen in Section 2.4.3 and Section 2.4.4.

Another possibility could involve a Magnus expansion for Backward SDEs (BSDEs).

5.2.2. Negative Interest Rate Model

Also, in this framework, there are multiple opportunities for further research. One possibility could be to use multiple risk-factors, possibly unsymmetrical for the negative and positive component. To be more precise, we could decide to use two positive CIR processes and one negative, i.e. $r(t) = x^1(t) + x^2(t) - y(t)$. Adding correlations and time-dependent features are also another line of research. Also other methods than the Gram-Charlier expansion for the calibration to the swaption surface could be explored.

5.2.3. Default Model from Rating Transitions

Instead of using doubly stochastic processes (Cox processes) for hazard rates to model default probabilities, we could use a rating model instead. Since the last column of the rating matrices contains the probabilities of default, we can consider rating models as a superset of default models. With our consideration of stochastic rating transitions, we have an analogue to the Cox processes used for hazard rate models.

This could lead to a new calibration procedure, since we could price Credit-Default-Swaps with a rating model. In this case, a risk-neutral measure could be derived in the rating model instead of taking the approach of calibrating the change of measure parameters by matching default probabilities. Also the volatilities of the rating model could be calibrated in this way, as we have seen for the interest rate models.

5.2.4. Characterization of Rating Transition Models

In Section 4.3, we started to investigate properties of a rating model. In the literature, this aspect is not much explored. Currently, we try to find a set of properties, which is fully characterizing short-term rating transition models.

We would like to use these properties to train neural networks with soft-constraints instead of fully data driven ones like the TimeGAN.

Also, this is important for studying different SDEs in the Lie algebra. We showed in this thesis only one example for the geometric Euler-Maruyama scheme but gave no justification for the structure of Y_t in (4.5.5) apart from good numerical results.

With a full characterization of short-term rating models, we can discuss the class of SDEs ensuring these properties. Also we hinted at other properties, such as self-excitement as a possible feature, which can be added. Additionally, jump-diffusions could be investigated to take rare events into account, such as the Covid pandemic.

5.2.5. Stochastic Reconstruction of Rating Matrices

We have discussed an issue with the cohort method for computing rating matrices from historical data. Since this is the method employed by the majority of rating agencies, it is necessary to study defect rating matrices in more detail.

We are currently investigating the possibility to repair the defect rating matrices in a stochastic way by training a DNN. One idea could be to use the Discriminator network of the trained TimeGAN combined with its Autoencoder to have a data-driven judge of what makes a rating matrix a rating matrix. In this case, we would like to train another generator network using a defect rating matrix as input and stochastically adding values to it, which are judged by the TimeGAN discriminator. Training this *conditional generator* could lead to

Table 5.1.: Information mismatch under the historical and risk-neutral measure.

	Historical data	Risk-neutral data
Entity	(unobserved)	observed
Sector	observed	(unobserved)

a feasible reconstruction of the defect rating matrices. Then, we can apply all the techniques from Chapter 4 to these reconstructed rating matrices.

A justification for a stochastic repairing method is the fact that the withdrawal of an entity adds a stochastic uncertainty to the defect rating matrix, since we do not know, in which rating it would have been at the end of the evaluation period.

Instead of using the TimeGAN discriminator as a judge of a good rating matrix, we also would like to use a characterization of a rating matrix discussed in Section 5.2.4.

5.2.6. Information Mismatch and Filtering

In this thesis, we have modelled rating transition for an entire market sector, such as the financial or corporate sector, since rating matrices are only computable for entire sectors with both the Aalen-Johansen estimator and the cohort method. They were fundamental for the calibration under the historical measure. For the risk-neutral calibration, we explained in Section 4.2.2 that we consider sector default probabilities instead of entity data.

It would be more realistic to calibrate the rating triggers for computing CVA and DVA based on entity data instead of sector data.

In Table 5.1, we summarized this issue, i.e. direct entity data is only known under the risk-neutral measure and sector data is only directly known under the historical measure. However, we additionally know historical ratings of an entity and can compute default probabilities under the risk-neutral measure, indicated by the brackets in the table. This gives us some hope to use a filtering approach to define rating triggers which are customized for specific counterparties. We noticed in (4.5.5) a close connection to the Langevin dynamics. If we disregard for the moment the fact that A_t^i has to be positive and set $a^i(t, y) \equiv y$ with $dY_t^i = \sigma_i dW_t^i$, we are exactly arriving at the prototype leading to the stochastic Langevin equation arising in filtering problems. This is explained in more detail in Pascucci and Pesce (2022b). We would like to exploit this close connection and are currently studying the filtering problem.

Since the Magnus expansion yields an efficient scheme for solving the stochastic Langevin equation, we even have access to more complicated models, which in turn would be feasible for an implementation in practice.

Another idea is inspired by Giles and Reisinger (2012). We could imagine that a model for a single entity is an analogue to a particle of a larger system. Each particle obeys an SDE and the entire system evolves like an SPDE. If we have enough particles this could lead to an SPDE on the Lie group instead of an SDE.

Bibliography

Al-Mohy, Awad, Nicholas Higham and Samuel Relton (July 2013). “Computing the Fréchet Derivative of the Matrix Logarithm and Estimating the Condition Number”. In: *SIAM Journal on Scientific Computing* 35, C394–C410. doi: 10.1137/120885991.

Al-Mohy, Awad H. and Nicholas J. Higham (2011). “Computing the action of the matrix exponential, with an application to exponential integrators”. In: *SIAM J. Sci. Comput.* 33(2), pp. 488–511. issn: 1064-8275. doi: 10.1137/100788860. url: https://doi.org/10.1137/100788860.

Al-Mohy, Awad H. and Nicholas J. Higham (2012). “Improved Inverse Scaling and Squaring Algorithms for the Matrix Logarithm”. In: *SIAM Journal on Scientific Computing* 34(4), pp. C153–C169. doi: 10.1137/110852553. eprint: https://doi.org/10.1137/110852553. url: https://doi.org/10.1137/110852553.

Andersen, Leif B.G., Peter Jäckel and Christian Kahl (2010). “Simulation of Square-Root Processes”. In: *Encyclopedia of Quantitative Finance*. American Cancer Society. isbn: 9780470061602. doi: https://doi.org/10.1002/9780470061602.eqf13009. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470061602.eqf13009. url: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470061602.eqf13009.

Arns, M., P. Buchholz and A. Panchenko (2010). “On the Numerical Analysis of Inhomogeneous Continuous-Time Markov Chains”. In: *INFORMS Journal on Computing* 22(3), pp. 416–432. doi: 10.1287/ijoc.1090.0357. eprint: https://doi.org/10.1287/ijoc.1090.0357. url: https://doi.org/10.1287/ijoc.1090.0357.

Azencott, Robert (1982). “Formule de Taylor stochastique et développement asymptotique d’intégrales de Feynman”. In: *Seminar on Probability, XVI, Supplement*. Vol. 921. Lecture Notes in Math. Springer, Berlin-New York, pp. 237–285.

Baker, H.K. et al. (1996). *Stochastic Mean and Stochastic Volatility: A Three-factor Model of the Term Structure of Interest Rates and Its Applications in Derivatives Pricing and Risk Management*. Competitive Trading of NYSE Listed Stocks: Measurement and Interpretation of Trading Costs Bd. 5. Blackwell Publishers. url: https://books.google.de/books?id=ci1SMQAACAAJ.

Barrera, G., M.A. Högele and J.C. Pardo (2022). “Cutoff stability of multivariate geometric Brownian motion”. In: Preprint arXiv: 2207.01666. Preprint arXiv: 2207.01666 (math.PR).

Ben Arous, Gérard (1989). “Flots et séries de Taylor stochastiques”. In: *Probab. Theory Related Fields* 81(1), pp. 29–77. issn: 0178-8051. doi: 10.1007/BF00343737. url: https://doi.org/10.1007/BF00343737.

Bielecki, T., M. Jeanblanc and M. Rutkowski (2003). “Modelling and Valuation of Credit Risk”. In: *CIME-EMS Summer School on Stochastic Methods in Finance*.

Bielecki, Tomasz, Igor Cialenco and Ismail Iyigunler (May 2012). “Collateralized CVA Valuation with Rating Triggers and Credit Migrations”. In: *International Journal of Theoretical and Applied Finance* 16. doi: 10.1142/S021902491350009X.

Bielecki, Tomasz and Marek Rutkowski (Apr. 2000). “Multiple Ratings Model of Defaultable Term Structure”. In: *Mathematical Finance* 10, pp. 125–139. doi: 10.1111/1467-9965.00085.

Bielecki, Tomasz R., Stéphane Crépey and Alexander Herbertson (2011). “Markov Chain Models of Portfolio Credit Risk”. In: Oxford University Press. Chap. 3, 328 ff. isbn: 978-0-199-54678-7.

Bielecki, Tomasz R., Jacek Jakubowski and Mariusz Niewęgłowski (2015a). *Conditional Markov Chains Part II: Consistency and Copulae*. arXiv: 1501.05535 [math.PR].

Bielecki, Tomasz R., Jacek Jakubowski and Mariusz Niewęgłowski (Jan. 2015b). *Conditional Markov Chains Revisited Part I: Construction and properties*. arXiv: 1501.05531 [math.PR].

Bielecki, Tomasz R. and Marek Rutkowski (2004). *Credit Risk: Modeling, Valuation and Hedging*. Vol. 1. Springer Finance. Springer Berlin, Heidelberg, pp. XVIII, 501. isbn: 978-3-540-67593-8. doi: 10.1007/978-3-662-04821-4. url: https://doi.org/10.1007/978-3-662-04821-4.

Björk, T. (2004). *Arbitrage Theory in Continuous Time*. Oxford Finance Series. Oxford University Press, Incorporated. isbn: 9780191533846. url: https://books.google.de/books?id=TJgjgJATeVIC.

BLANES, S. et al. (2009). "The Magnus expansion and some of its applications". In: *Phys. Rep.* 470(5-6), pp. 151–238. ISSN: 0370-1573. DOI: 10.1016/j.physrep.2008.11.001. URL: https://doi.org/10.1016/j.physrep.2008.11.001.

BREZIS, HAIM (2011). *Functional analysis, Sobolev spaces and partial differential equations.* Universitext. Springer, New York, pp. xiv+599. ISBN: 978-0-387-70913-0.

BRIGO, D. and NAOUFEL EL-BACHIR (2006). "Credit Derivatives Pricing with a Smile-Extended Jump Stochastic Intensity Model". In: *Derivatives.* URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=950208.

BRIGO, D., M. MORINI and A. PALLAVICINI (2013). *Counterparty Credit Risk, Collateral and Funding: With Pricing Cases For All Asset Classes.* The Wiley Finance Series. Wiley. ISBN: 9780470661789. URL: https://books.google.de/books?id=Y-X9zqVcOFkC.

BRIGO, DAMIANO and F MERCURIO (2001a). "A deterministic-shift extension of analytically tractable and time-homogeneous short rate models". English. In: *Finance and Stochastics* 5, pp. 369 –388. ISSN: 0949-2984.

BRIGO, DAMIANO and FABIO MERCURIO (2001b). "A deterministic–shift extension of analytically–tractable and time–homogeneous short–rate models". In: *Finance and Stochastics* 5(3), pp. 369–387. ISSN: 0949-2984. DOI: 10.1007/PL00013541. URL: https://doi.org/10.1007/PL00013541.

BRIGO, DAMIANO and FABIO MERCURIO (2006). *Interest rate models: theory and practice: with smile, inflation and credit.* Springer. ISBN: 978-36-625-1743-7.

BURRAGE, KEVIN and PM BURRAGE (1999). "High strong order methods for non-commutative stochastic ordinary differential equation systems and the Magnus formula". In: *Physica D: Nonlinear Phenomena* 133(1-4), pp. 34–48.

CAMPANA, LORENZO (2022). *Stochastic modelling of non-spherical particles in turbulence.* PhD Thesis Université Côte d'Azur et de INRIA.

CASANELLAS, MARTA, JESÚS FERNÁNDEZ-SÁNCHEZ and JORDI ROCA-LACOSTENA (2021). "The embedding problem for Markov matrices". In: arXiv: 2005.00818 [math.PR].

CASTELL, FABIENNE (1993). "Asymptotic expansion of stochastic flows". In: *Probab. Theory Related Fields* 96(2), pp. 225–239. ISSN: 0178-8051. DOI: 10.1007/BF01192134. URL: https://doi.org/10.1007/BF01192134.

CASTELL, FABIENNE and JESSICA GAINES (1995). "An efficient approximation method for stochastic differential equations by means of the exponential Lie series". In: *Math. Comput. Simulation* 38(1-3). Probabilités numériques (Paris, 1992), pp. 13–19. ISSN: 0378-4754. DOI: 10.1016/0378-4754(93)E0062-A. URL: https://doi.org/10.1016/0378-4754(93)E0062-A.

CASTELL, FABIENNE and JESSICA GAINES (1996). "The ordinary differential equation approach to asymptotically efficient schemes for solution of stochastic differential equations". In: *Ann. Inst. H. Poincaré Probab. Statist.* 32(2), pp. 231–250. ISSN: 0246-0203. URL: http://www.numdam.org/item?id=AIHPB_1996__32_2_231_0.

CERCIGNANI, C. (1988). *The Boltzmann equation and its applications.* Springer-Verlag: New York, pp. xii+455. ISBN: 0-387-96637-4.

CHENG, YIN HEI (2013). "Pricing Derivatives by Gram-Charlier Expansions". In: URL: http://hdl.handle.net/10012/7413.

CHOW, PAO-LIU (2015). *Stochastic partial differential equations.* Second. Advances in Applied Mathematics. CRC Press, Boca Raton, FL, pp. xvi+317. ISBN: 978-1-4665-7955-2.

CHUNG, JUNYOUNG et al. (2014). *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling.* DOI: 10.48550/ARXIV.1412.3555. URL: https://arxiv.org/abs/1412.3555.

COLETTI, CRISTIAN, RAFAEL CARNEIRO and SANDRA YEPES (Feb. 2020). "Some geometric properties of stochastic matrices". In: *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics* 7(1). DOI: 10.5540/03.2020.007.01.0439.

CORREALES, ALVARO and CARLOS ESCUDERO (2018). *Ito vs Stratonovich in the presence of absorbing states.* arXiv: 1811.06750 [math.PR].

Cox, John C., Jonathan E. Ingersoll and Stephen A. Ross (1981). "A Re-Examination of Traditional Hypotheses about the Term Structure of Interest Rates". In: *The Journal of Finance* 36(4), pp. 769–799. issn: 00221082, 15406261. url: http://www.jstor.org/stable/2327547.

Cox, John C., Jonathan E. Ingersoll and Stephen A. Ross (1985). "A Theory of the Term Structure of Interest Rates". In: *Econometrica* 53(2), pp. 385–407. url: https://EconPapers.repec.org/RePEc:ecm:emetrp:v:53:y:1985:i:2:p:385-407.

Curtis, V.V. and C.A.A. Sanches (2017). "A Low-Space Algorithm for the Subset-Sum Problem on GPU". In: 83(C), 120–124. issn: 0305-0548. doi: 10.1016/j.cor.2017.02.006. url: https://doi.org/10.1016/j.cor.2017.02.006.

Dereich, Steffen, Andreas Neuenkirch and Lukasz Szpruch (2012). "An Euler-type method for the strong approximation of the Cox-Ingersoll-Ross process". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 468(2140), pp. 1105–1115. doi: 10.1098/rspa.2011.0505. eprint: https://royalsocietypublishing.org/doi/pdf/10.1098/rspa.2011.0505. url: https://royalsocietypublishing.org/doi/abs/10.1098/rspa.2011.0505.

Desvillettes, L. and C. Villani (2001). "On the trend to global equilibrium in spatially inhomogeneous entropy-dissipating systems: the linear Fokker-Planck equation". In: *Comm. Pure Appl. Math.* 54(1), pp. 1–42. issn: 0010-3640.

Di Francesco, Marco (Sept. 2012). "A General Gaussian Interest Rate Model Consistent with the Current Term Structure". In: *ISRN Probability and Statistics* 2012. doi: 10.5402/2012/673607.

Di Francesco, Marco and Kevin Kamm (2021). "How to handle negative interest rates in a CIR framework". In: *SeMA Journal*. issn: 2281-7875. doi: 10.1007/s40324-021-00267-w. url: https://doi.org/10.1007/s40324-021-00267-w.

Di Francesco, Marco and Kevin Kamm (2022). "On the Deterministic-Shift Extended CIR Model in a Negative Interest Rate Framework". In: *International Journal of Financial Studies* 10(2). issn: 2227-7072. doi: 10.3390/ijfs10020038. url: https://www.mdpi.com/2227-7072/10/2/38.

Di Francesco, Marco and Andrea Pascucci (2004). "On the complete model with stochastic volatility by Hobson and Rogers". In: *Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.* 460(2051), pp. 3327–3338. issn: 1364-5021. doi: 10.1098/rspa.2004.1370. url: https://doi.org/10.1098/rspa.2004.1370.

Ding, Kailin and Ning Ning (2021). "Markov chain approximation and measure change for time-inhomogeneous stochastic processes". In: *Applied Mathematics and Computation* 392, p. 125732. issn: 0096-3003. doi: https://doi.org/10.1016/j.amc.2020.125732. url: https://www.sciencedirect.com/science/article/pii/S0096300320306858.

Doss, Halim (1977). "Liens entre équations différentielles stochastiques et ordinaires". In: *Ann. Inst. H. Poincaré Sect. B (N.S.)* 13(2), pp. 99–125.

Duan, Jinqiao and Jia-an Yan (2008). *General Matrix-Valued Inhomogeneous Linear Stochastic Differential Equations and Applications*. doi: 10.48550/ARXIV.0808.1112. url: https://arxiv.org/abs/0808.1112.

Fliess, Michel and Dorothée Normand-Cyrot (1982). "Algèbres de Lie nilpotentes, formule de Baker-Campbell-Hausdorff et intégrales itérées de K. T. Chen". In: *Seminar on Probability, XVI*. Vol. 920. Lecture Notes in Math. Springer, Berlin-New York, pp. 257–267.

Friz, Peter K., Paul P. Hager and Nikolas Tapia (2022). "Unified signature cumulants and generalized Magnus expansions". In: *Forum Math. Sigma* 10, Paper No. e42, 60. doi: 10.1017/fms.2022.20. url: https://doi.org/10.1017/fms.2022.20.

Gatarek, Dariusz and Juliusz Jabłecki (2021). "Between Scylla and Charybdis: The Bermudan Swaptions Pricing Odyssey". In: *Mathematics* 9(2). issn: 2227-7390. doi: 10.3390/math9020112. url: https://www.mdpi.com/2227-7390/9/2/112.

Gikhman, Ilya I. (2011). "A Short Remark on Feller's Square Root Condition". In: *SSRN Electronic Journal*. doi: http://dx.doi.org/10.2139/ssrn.1756450. url: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1756450#.

Giles, Michael B. and Christoph Reisinger (2012). *Stochastic finite differences and multilevel Monte Carlo for a class of SPDEs in finance*. doi: 10.48550/ARXIV.1204.1442. url: https://arxiv.org/abs/1204.1442.

Gillespie, Daniel T. (2007). "Stochastic Simulation of Chemical Kinetics". In: *Annual Review of Physical Chemistry* 58(1). PMID: 17037977, pp. 35–55. doi: 10.1146/annurev.physchem.58.032806.104637. eprint: https://doi.org/10.1146/annurev.physchem.58.032806.104637. url: https://doi.org/10.1146/annurev.physchem.58.032806.104637.

Gilli, Manfred, Dietmar Maringer and Enrico Schumann (2011). *Numerical Methods and Optimization in Finance*. 1st ed. Elsevier. url: https://EconPapers.repec.org/RePEc:eee:monogr:9780123756626.

Glasserman, Paul (2004). *Monte Carlo methods in financial engineering*. Springer: New York. isbn: 0387004513 9780387004518 1441918221 9781441918222. url: http://www.amazon.com/Financial-Engineering-Stochastic-Modelling-Probability/dp/0387004513/ref=pd_sim_b_68?ie=UTF8&refRID=1AN8JXSDGMEV2RPHFC2A.

Goodfellow, Ian (2017). *NIPS 2016 Tutorial: Generative Adversarial Networks*. doi: 10.48550/ARXIV.1701.00160. url: https://arxiv.org/abs/1701.00160.

Hall, B.C. (2003). *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*. Graduate Texts in Mathematics. Springer. isbn: 9780387401225. url: https://books.google.de/books?id=m1VQi8HmEwcC.

Higham, Nicholas J. (2008). *Functions of Matrices*. Society for Industrial and Applied Mathematics. doi: 10.1137/1.9780898717778. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9780898717778. url: https://epubs.siam.org/doi/abs/10.1137/1.9780898717778.

Ho, Thomas S. Y. and Sang-Bin Lee (1986). "Term Structure Movements and Pricing Interest Rate Contingent Claims". In: *The Journal of Finance* 41(5), pp. 1011–1029. doi: https://doi.org/10.1111/j.1540-6261.1986.tb02528.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1540-6261.1986.tb02528.x. url: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.1986.tb02528.x.

Hull, J. and A. White (1990). "Pricing Interest-Rate-Derivative Securities". In: *Review of Financial Studies* 3, pp. 573–592.

Hull, John C. (2006). *Options, futures, and other derivatives*. 8. ed., Pearson internat. ed. Pearson Prentice Hall.

Ibáñez, Javier et al. (2022). "Two Taylor Algorithms for Computing the Action of the Matrix Exponential on a Vector". In: *Algorithms* 15(2). issn: 1999-4893. doi: 10.3390/a15020048. url: https://www.mdpi.com/1999-4893/15/2/48.

Israel, Robert B., Jeffrey S. Rosenthal and Jason Z. Wei (2001). "Finding Generators for Markov Chains via Empirical Transition Matrices, with Applications to Credit Ratings". In: *Mathematical Finance* 11(2), pp. 245–265. doi: https://doi.org/10.1111/1467-9965.00114. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/1467-9965.00114. url: https://onlinelibrary.wiley.com/doi/abs/10.1111/1467-9965.00114.

Jamshidian, Farshid (1990). "An analysis of American options". In: *Review of futures markets*.

Jarrow, Robert, David Lando and Stuart M Turnbull (1997). "A Markov Model for the Term Structure of Credit Risk Spreads". In: *Review of Financial Studies* 10(2), pp. 481–523. url: https://EconPapers.repec.org/RePEc:oup:rfinst:v:10:y:1997:i:2:p:481-523.

Jeanblanc, M., M. Yor and M. Chesney (2009). *Mathematical Methods for Financial Markets*. Springer Finance. Springer London. isbn: 9781852333768. url: https://books.google.de/books?id=ZhbROxoQ-ZMC.

Kamm, Kevin (2022). "An introduction to rating triggers for collateral-inclusive XVA in an ICTMC framework". In: doi: 10.48550/ARXIV.2207.03883. url: https://arxiv.org/abs/2207.03883.

Kamm, Kevin and Michelle Muniz (2022). "A novel approach to rating transition modelling via Machine Learning and SDEs on Lie groups". In: doi: 10.48550/ARXIV.2205.15699. url: https://arxiv.org/abs/2205.15699.

Kamm, Kevin, Stefano Pagliarani and Andrea Pascucci (2021). "On the Stochastic Magnus Expansion and Its Application to SPDEs". In: *Journal of Scientific Computing* 89(3), p. 56. issn: 1573-7691. doi: 10.1007/s10915-021-01633-6. url: https://doi.org/10.1007/s10915-021-01633-6.

Kamm, Kevin, Stefano Pagliarani and Andrea Pascucci (2022). “Numerical solution of kinetic SPDEs via stochastic Magnus expansion”. In: doi: 10.48550/ARXIV.2207.09776. url: https://arxiv.org/abs/2207.09776.

Kingma, Diederik P. and Jimmy Ba (2014). *Adam: A Method for Stochastic Optimization.* doi: 10.48550/ARXIV.1412.6980. url: https://arxiv.org/abs/1412.6980.

Kingma, Diederik P. and Max Welling (2019). “An Introduction to Variational Autoencoders”. In: *Foundations and Trends® in Machine Learning* 12(4), pp. 307–392. doi: 10.1561/2200000056. url: https://doi.org/10.1561%2F2200000056.

Kloeden, Peter E. and Eckhard Platen (1992). *Numerical solution of stochastic differential equations.* Vol. 23. Applications of Mathematics (New York). Springer-Verlag, Berlin, pp. xxxvi+632. isbn: 3-540-54062-8. doi: 10.1007/978-3-662-12616-5. url: https://doi.org/10.1007/978-3-662-12616-5.

Kreinin, Alexander and Marina Sidelnikova (Jan. 2001). “Regularization Algorithms for Transition Matrices”. In: *Algo Research Quarterly* 4, pp. 23–40.

Krylov, N. V. and B. L. Rozovskii (1977). “The Cauchy problem for linear stochastic partial differential equations”. In: *Izv. Akad. Nauk SSSR Ser. Mat.* 41(6), pp. 1329–1347, 1448. issn: 0373-2436.

Krylov, Nikolaj V. (2008). *Controlled diffusion processes.* Vol. 14. Springer Science & Business Media.

Kunita, Hiroshi (1980). “On the representation of solutions of stochastic differential equations”. In: *Seminar on Probability, XIV (Paris, 1978/1979) (French).* Vol. 784. Lecture Notes in Math. Springer, Berlin, pp. 282–304.

Kunita, Hiroshi (2019). *Stochastic flows and jump-diffusions.* Vol. 92. Probability Theory and Stochastic Modelling. Springer, Singapore, pp. xvii+352. isbn: 978-981-13-3800-7; 978-981-13-3801-4. doi: 10.1007/978-981-13-3801-4. url: https://doi.org/10.1007/978-981-13-3801-4.

Kuo, Hui-Hsiung (2006). *Introduction to stochastic integration.* Universitext. Springer, New York, pp. xiv+278. isbn: 978-0387-28720-1; 0-387-28720-5.

Lando, David and Torben M. Skødeberg (2002). “Analyzing rating transitions and rating drift with continuous observations”. In: *Journal of Banking & Finance* 26(2), pp. 423–444. issn: 0378-4266. doi: https://doi.org/10.1016/S0378-4266(01)00228-X. url: https://www.sciencedirect.com/science/article/pii/S037842660100228X.

Lencastre, Pedro et al. (2014). *Are credit ratings time-homogeneous and Markov?* arXiv: 1403.8018 [q-fin.RM].

Lencastre, Pedro et al. (2016). “From empirical data to time-inhomogeneous continuous Markov processes”. In: *Phys. Rev. E* 93 (3), p. 032135. doi: 10.1103/PhysRevE.93.032135. url: https://link.aps.org/doi/10.1103/PhysRevE.93.032135.

Li, Yan-Fu, Yan-Hui Lin and Enrico Zio (May 2012). “Stochastic Modeling by Inhomogeneous Continuous Time Markov Chains”. In: url: https://hal-supelec.archives-ouvertes.fr/hal-00737094.

Liao, Jiyuan (2018). “The Existence and Uniqueness of Solutions for Mean-Reverting β-Process”. In: *Open Journal of Statistics.* doi: 10.4236/ojs.2018.82021. url: https://www.scirp.org/journal/paperinformation.aspx?paperid=83975.

Lions, P.-L. (1994). “On Boltzmann and Landau equations”. In: *Philos. Trans. Roy. Soc. London Ser. A* 346(1679), pp. 191–204. issn: 0962-8428.

Longstaff, Francis A. and Eduardo S. Schwartz (2001). “Valuing American Options by Simulation: A Simple Least-Squares Approach”. In: *The Finance.*

Lord, Gabriel, Simon J. A. Malham and Anke Wiese (2008). “Efficient strong integrators for linear stochastic systems”. In: *SIAM J. Numer. Anal.* 46(6), pp. 2892–2919. issn: 0036-1429. doi: 10.1137/060656486. url: https://doi.org/10.1137/060656486.

Lu, Lu, Pengzhan Jin and George Em Karniadakis (2019). “DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators”. In: *CoRR* abs/1910.03193. arXiv: 1910.03193. url: http://arxiv.org/abs/1910.03193.

Maghsoodi, Yoosef (1996). “SOLUTION OF THE EXTENDED CIR TERM STRUCTURE AND BOND OPTION VALUATION”. In: *Mathematical Finance* 6(1), pp. 89–109. doi: https://doi.org/10.1111/j.1467-9965.1996.tb00113.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-

9965.1996.tb00113.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9965.1996.tb00113.x.

Magnus, Wilhelm (1954). “On the exponential solution of differential equations for a linear operator”. In: *Comm. Pure Appl. Math.* 7, pp. 649–673. ISSN: 0010-3640. DOI: 10.1002/cpa.3160070404. URL: https://doi.org/10.1002/cpa.3160070404.

Marjanovic, Goran and Victor Solo (2018). “Numerical methods for stochastic differential equations in matrix Lie groups made simple”. In: *IEEE Trans. Automat. Control* 63(12), pp. 4035–4050. ISSN: 0018-9286. DOI: 10.1109/tac.2018.2798703. URL: https://doi.org/10.1109/tac.2018.2798703.

Mercurio, F. and A. Pallavicini (2005). “Mixing Gaussian Models to Price Cms Derivatives”. In: *SSRN Electronic Journal.*

Mikulevicius, R. (2000). “On the Cauchy problem for parabolic SPDEs in Hölder classes”. In: *Ann. Probab.* 28(1), pp. 74–103. ISSN: 0091-1798. URL: https://doi.org/10.1214/aop/1019160112.

Miron, P. and P. Swannell (1991). *Pricing and Hedging Swaps.* Euromoney books. Euromoney Books. ISBN: 9781855640528. URL: https://books.google.de/books?id=4Z2DQgAACAAJ.

Moan, Per Christian and Jitse Niesen (2008). “Convergence of the Magnus series”. In: *Found. Comput. Math.* 8(3), pp. 291–301. ISSN: 1615-3375. DOI: 10.1007/s10208-007-9010-0. URL: https://doi.org/10.1007/s10208-007-9010-0.

Muniz, Michelle et al. (2022a). “Higher strong order methods for linear Itô SDEs on matrix Lie groups”. In: *To appear in BIT Numer Math*, https://doi.org/10.1007/s10543–022–00911–5. URL: https://doi.org/10.1007/s10543-022-00911-5.

Muniz, Michelle et al. (June 2022b). *Strong stochastic Runge-Kutta-Munthe-Kaas methods for nonlinear Itô SDEs on manifolds.*

Newton, Nigel J. (1991). “Asymptotically efficient Runge-Kutta methods for a class of Itô and Stratonovich equations”. In: *SIAM J. Appl. Math.* 51(2), pp. 542–567. ISSN: 0036-1399. DOI: 10.1137/0151028. URL: https://doi.org/10.1137/0151028.

Oosterlee, C.W. and L.A. Grzelak (2019). *Mathematical Modeling And Computation In Finance: With Exercises And Python And Matlab Computer Codes.* World Scientific Publishing Company. ISBN: 9781786347961. URL: https://books.google.de/books?id=TsPKDwAAQBAJ.

Orlando, Giuseppe, Rosa Maria Mininni and Michele Bufalo (2019a). “A new approach to forecast market interest rates through the CIR model”. In: *Studies in Economics and Finance* 37(2), pp. 267–292. URL: https://ideas.repec.org/a/eme/sefpps/sef-03-2019-0116.html.

Orlando, Giuseppe, Rosa Maria Mininni and Michele Bufalo (2019b). “Interest rates calibration with a CIR model”. In: *Journal of Risk Finance* 20(4), pp. 370–387. URL: https://EconPapers.repec.org/RePEc:eme:jrfpps:jrf-05-2019-0080.

Orlando, Giuseppe, Rosa Maria Mininni and Michele Bufalo (2020). “Forecasting interest rates through Vasicek and CIR models: A partitioning approach”. In: *Journal of Forecasting* 39(4), pp. 569–579. DOI: https://doi.org/10.1002/for.2642. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/for.2642. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/for.2642.

Palmowski, Zbigniew and Tomasz Rolski (2002). “A technique for exponential change of measure for Markov processes”. In: *Bernoulli* 8(6), pp. 767 –785. DOI: bj/1076364805. URL: https://doi.org/.

Pascucci, Andrea (2011). *PDE and martingale methods in option pricing.* Vol. 2. Bocconi & Springer Series. Springer, Milan; Bocconi University Press, Milan, pp. xviii+719. ISBN: 978-88-470-1780-1. DOI: 10.1007/978-88-470-1781-8. URL: https://doi.org/10.1007/978-88-470-1781-8.

Pascucci, Andrea and Antonello Pesce (2020). “The parametrix method for parabolic SPDEs”. In: *Stochastic Processes and their Applications* 130(10), pp. 6226–6245. ISSN: 0304-4149. DOI: https://doi.org/10.1016/j.spa.2020.05.008. URL: https://www.sciencedirect.com/science/article/pii/S0304414918302138.

Pascucci, Andrea and Antonello Pesce (2022a). “Backward and forward filtering under the weak Hörmander condition”. In: *To appear in Stoch. Partial Differ. Equ. Anal. Comput.* arXiv: 2006.13325 [math.PR].

Pascucci, Andrea and Antonello Pesce (2022b). “On stochastic Langevin and Fokker-Planck equations: the two-dimensional case”. In: *J. Differential Equations* 310, pp. 443–483. issn: 0022-0396. doi: 10.1016/j.jde.2021.11.004. url: https://doi.org/10.1016/j.jde.2021.11.004.

Russo, Vincenzo and Gabriele Torri (2019). “Calibration of one-factor and two-factor Hull–White models using swaptions”. In: *Computational Management Science* 16(1), pp. 275–295. issn: 1619-6988. doi: 10.1007/s10287-018-0323-z. url: https://doi.org/10.1007/s10287-018-0323-z.

Schönbucher, P.J. (2003). *Credit Derivatives Pricing Models: Models, Pricing and Implementation*. The Wiley Finance Series. Wiley. isbn: 9780470842911. url: https://books.google.de/books?id=s-sJAQAAMAAJ.

Schrager, D. and A. Pelsser (2006). “Pricing Swaptions and Coupon Bond Options in Affine Term Structure Models”. In: *Capital Markets: Asset Pricing & Valuation eJournal*.

Smith, Peter J. (1995). “A Recursive Formulation of the Old Problem of Obtaining Moments from Cumulants and Vice Versa”. In: *The American Statistician* 49(2), pp. 217–218. issn: 00031305. url: http://www.jstor.org/stable/2684642.

Stroock, Daniel W. (2005). *An Introduction to Markov Processes*. 1st ed. Graduate Texts in Mathematics. Springer: Heidelberg. isbn: 3-540-23499-3.

Sussmann, H. J. (1988). “Product expansions of exponential Lie series and the discretization of stochastic differential equations”. In: *Stochastic differential systems, stochastic control theory and applications (Minneapolis, Minn., 1986)*. Vol. 10. IMA Vol. Math. Appl. Springer, New York, pp. 563–582. doi: 10.1007/978-1-4613-8762-6_32. url: https://doi.org/10.1007/978-1-4613-8762-6_32.

Tanaka, Keiichi, Takeshi Yamada and Toshiaki Watanabe (2010). “Applications of Gram–Charlier expansion and bond moments for pricing of interest rates and credit risk”. In: *Quantitative Finance* 10(6), pp. 645–662. doi: 10.1080/14697680903193371. eprint: https://doi.org/10.1080/14697680903193371. url: https://doi.org/10.1080/14697680903193371.

Vasicek, Oldrich (1977). “An equilibrium characterization of the term structure”. In: *Journal of Financial Economics* 5(2), pp. 177–188. issn: 0304-405X. doi: https://doi.org/10.1016/0304-405X(77)90016-2. url: https://www.sciencedirect.com/science/article/pii/0304405X77900162.

Wang, Zhenyu et al. (2020). “The Magnus Expansion for Stochastic Differential Equations”. In: *J. Nonlinear Sci.* 30(1), pp. 419–447. issn: 0938-8974. doi: 10.1007/s00332-019-09578-9. url: https://doi.org/10.1007/s00332-019-09578-9.

Yamato, Yuiti (1979). “Stochastic differential equations and nilpotent Lie algebras”. In: *Z. Wahrsch. Verw. Gebiete* 47(2), pp. 213–229. issn: 0044-3719. doi: 10.1007/BF00535284. url: https://doi.org/10.1007/BF00535284.

Yang, Guoguo et al. (2021). “A class of new Magnus-type methods for semi-linear non-commutative Itô stochastic differential equations”. In: *Numerical Algorithms*. issn: 1572-9265. doi: 10.1007/s11075-021-01089-7. url: https://doi.org/10.1007/s11075-021-01089-7.

Yoon, Jinsung, Daniel Jarrett and Mihaela van der Schaar (2019). “Time-series Generative Adversarial Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc. url: https://proceedings.neurips.cc/paper/2019/file/c9efe5f26cd17ba6216bbe2a7d26d490-Paper.pdf.

The Itô-Stochastic Magnus Expansion

A

A.1. Auxiliary Results

The next lemma is useful, if one wants to express stochastic integrals as Lebesgue integrals and is based on Itô's formula.

Lemma A.1.1. *For* $p, p_1, p_2, q, q_1, q_2 \in \mathbb{N}_0$ *we have*

$$\int_0^t s^p W_s^q dW_s = \frac{1}{q+1}\left(t^p W_t^{q+1} - \int_0^t \left[\frac{q(q+1)}{2} s^p W_s^{q-1} + p W_s^{q+1} s^{p-1}\right] ds\right) \tag{A.1.1}$$

$$\int_0^t s^{p_1} \int_0^s r^{p_2} W_r^q dr ds = \frac{1}{1+p_1}\left(t^{1+p_1} \int_0^t s^{p_2} W_s^q ds - \int_0^t s^{1+p_1+p_2} W_s^q ds\right) \tag{A.1.2}$$

$$\begin{aligned}\int_0^t s^{p_1} W_s^{q_1} \int_0^s r^{p_2} W_r^{q_2} dr dW_s = \frac{1}{q_1+1}\Bigg(& t^{p_1} W_t^{q_1+1} \int_0^t s^{p_2} W_s^{q_2} ds \\ & - \int_0^t s^{p_1+p_2} W_s^{q_1+q_2+1} ds \\ & - \frac{q_1(q_1+1)}{2} \int_0^t s^{p_1} W_s^{q_1-1} \int_0^s r^{p_2} W_r^{q_2} dr ds \\ & - p_1 \int_0^t s^{p_1-1} W_s^{q_1+1} \int_0^s r^{p_2} W_r^{q_2} dr ds \Bigg).\end{aligned} \tag{A.1.3}$$

Proof. Note that (A.1.1) is a special case of (A.1.3) by setting $p = p_1 + 1$, $q = q_1$ and $p_2 = q_2 = 0$.

Now, we show (A.1.2). With Itô's product rule we get

$$\begin{aligned} d\left(s^{p_1} \int_0^s r^{p_2} W_r^q dr \; s\right) &= s^{p_1} \int_0^s r^{p_2} W_r^q dr ds + s \; d\left(s^{p_1} \int_0^s r^{p_2} W_r^q dr\right) + 0 \\ &= s^{p_1} \int_0^s r^{p_2} W_r^q dr ds + s \; d\left(s^{p_1} s^{p_2} W_s^q ds + \int_0^s r^{p_2} W_r^q dr p_1 s^{p_1-1} ds\right) \\ &= (1+p_1)\, s^{p_1} \int_0^s r^{p_2} W_r^q dr ds + s^{1+p_1+p_2} W_s^q ds. \end{aligned}$$

Rearranging the equation yields the claim.

Next, we show (A.1.3). With Itô's product rule we get

$$\begin{aligned} d\left(s^{p_1} W_s^{q_1} \int_0^s r^{p_2} W_r^{q_2} dr W_s\right) = {} & s^{p_1} W_s^{q_1} \int_0^s r^{p_2} W_r^{q_2} dr dW_s + W_s d\left(s^{p_1} W_s^{q_1} \int_0^s r^{p_2} W_r^{q_2} dr\right) \\ & + d\left\langle \cdot^{p_1} W_\cdot^{q_1} \int_0^\cdot r^{p_2} W_r^{q_2} dr, W_\cdot \right\rangle_s . \end{aligned}$$

Now, use Itô's product rule and Itô's formula on the the following term

$$\begin{aligned}
d\left(s^{p_1}W_s^{q_1}\int_0^s r^{p_2}W_r^{q_2}dr\right) &= s^{p_1}W_s^{q_1}d\left(\int_0^s r^{p_2}W_r^{q_2}dr\right) + \int_0^s r^{p_2}W_r^{q_2}drd\left(s^{p_1}W_s^{q_1}\right) + 0 \\
&= s^{p_1+p_2}W_s^{q_1+q_2}ds + \int_0^s r^{p_2}W_r^{q_2}dr\Bigg(s^{p_1}\left(q_1W_s^{q_1-1}dW_s + \frac{q_1(q_1-1)}{2}W_s^{q_1-2}ds\right) \\
&\qquad + W_s^{q_1}\left(p_1s^{p_1-1}ds\right) + 0\Bigg) \\
&= \int_0^s r^{p_2}W_r^{q_2}drs^{p_1}q_1W_s^{q_1-1}dW_s + \Bigg[s^{p_1+p_2}W_s^{q_1+q_2} + \int_0^s r^{p_2}W_r^{q_2}drs^{p_1}\frac{q_1(q_1-1)}{2}W_s^{q_1-2} \\
&\qquad + \int_0^s r^{p_2}W_r^{q_2}drW_s^{q_1}p_1s^{p_1-1}\Bigg]ds
\end{aligned}$$

For the quadratic variation from above we have

$$d\left\langle \cdot^{p_1}W_\cdot^{q_1}\int_0^\cdot r^{p_2}W_r^{q_2}dr, W_\cdot\right\rangle_s = \int_0^s r^{p_2}W_r^{q_2}drs^{p_1}q_1W_s^{q_1-1}ds.$$

In total, we have

$$\begin{aligned}
&d\left(s^{p_1}W_s^{q_1}\int_0^s r^{p_2}W_r^{q_2}drW_s\right) \\
&\quad = (1+q_1)s^{p_1}W_s^{q_1}\int_0^s r^{p_2}W_r^{q_2}drdW_s + \Bigg[s^{p_1+p_2}W_s^{q_1+q_2+1} + \frac{q_1(q_1+1)}{2}s^{p_1}W_s^{q_1-1}\int_0^s r^{p_2}W_r^{q_2}dr \\
&\qquad + p_1\int_0^s r^{p_2}W_r^{q_2}drW_s^{q_1+1}s^{p_1-1}\Bigg]ds.
\end{aligned}$$

Rearranging the equation yields the claim. □

B The Cox-Ingersoll-Ross Model in a Negative Interest Rate Framework

B.1. Derivation of the Riccati Equations and Coefficients

Let everything be as in Lemma 3.3.3. In particular, let

$$P(t,T) \overset{!}{=} A_x(t,T)\exp\left(-B_x(t,T)x(t)\right)A_y(t,T)\exp\left(B_y(t,T)y(t)\right).$$

To derive the Riccati equations (3.3.10) we use the fact, that we are modelling under the martingale measure $\mathbb{Q}$, therefore the discounted price process $\exp\left(-\int_0^t r(s)ds\right)P(t,T)$ needs to be a martingale.

By independence of x and y, as well as Itô's formula we derive after some algebra

$$\begin{aligned}
&d\left[\exp\left(-\int_0^t r(s)ds\right)A_x(t,T)\exp\left(-B_x(t,T)x(t)\right)A_y(t,T)\exp\left(B_y(t,T)y(t)\right)\right]\\
&\quad= A_x(t,T)\exp\left(-\int_0^t x(s)ds - B_x(t,T)x(t)\right)\Bigg[\exp\left(\int_0^t y(s)ds + B_y(t,T)y(t)\right)\Bigg[\\
&\qquad\quad A_y(t,T)\Big[y(t)dt + B_y(t,T)dy(t) + y(t)\left(\partial_t B_y\right)(t,T)dt + \frac{1}{2}B_y^2(t,T)d\left\langle y\right\rangle_t\Big]\\
&\qquad\quad + \left(\partial_t A_y\right)(t,T)dt\Bigg]\Bigg]\\
&\qquad + A_y(t,T)\exp\left(\int_0^t y(s)ds + B_y(t,T)y(t)\right)\Bigg[\exp\left(-\int_0^t x(s)ds + B_x(t,T)x(t)\right)\Bigg[\\
&\qquad\qquad A_x(t,T)\Big[-x(t)dt - B_x(t,T)dx(t) - x(t)\left(\partial_t B_x\right)(t,T)dt + \frac{1}{2}B_x^2(t,T)d\left\langle x\right\rangle_t\Big]\\
&\qquad\qquad + \left(\partial_t A_x\right)(t,T)dt\Bigg]\Bigg].
\end{aligned}$$

Now, in order to be a martingale the parts of bounded variation have to vanish, which leads us after rearranging the terms to

$$\begin{aligned}
0 \overset{!}{=}\ & y(t)\Bigg[A_x(t,T)A_y(t,T)\Big[1 + B_y(t,T)\lambda_y(t) + \left(\partial_t B_y\right)(t,T) + \frac{1}{2}B_y^2(t,T)\gamma_y(t)\Big]\Bigg]\\
&+ x(t)\Bigg[A_y(t,T)A_x(t,T)\Big[-1 - B_x(t,T)\lambda_x(t) - \left(\partial_t B_x\right)(t,T) + \frac{1}{2}B_x^2(t,T)\gamma_x(t)\Big]\Bigg]\\
&+ A_x(t,T)A_y(t,T)\Bigg[B_y(t,T)\eta_y(t) + \frac{1}{2}B_y^2(t,T)\delta_y(t) - B_x(t,T)\eta_x(t) + \frac{1}{2}B_x^2(t,T)\delta_x(t)\Bigg]\\
&\quad + A_x(t,T)\left(\partial_t A_y\right)(t,T) + A_y(t,T)\left(\partial_t A_x\right)(t,T).
\end{aligned}$$

Thus, we derive the following Riccati System

$$\begin{aligned}
1 + B_y(t,T)\lambda_y(t) + \left(\partial_t B_y\right)(t,T) + \frac{1}{2}B_y^2(t,T)\gamma_y(t) &= 0, \qquad B_y(T,T) = 0,\\
-1 - B_x(t,T)\lambda_x(t) - \left(\partial_t B_x\right)(t,T) + \frac{1}{2}B_x^2(t,T)\gamma_x(t) &= 0, \qquad B_x(T,T) = 0,\\
B_y(t,T)\eta_y(t) + \frac{1}{2}B_y^2(t,T)\delta_y(t) - B_x(t,T)\eta_x(t) + \frac{1}{2}B_x^2(t,T)\delta_x(t) &\\
+ \partial_t\left(\log A_y\right)(t,T) + \partial_t\left(\log A_x\right)(t,T) &= 0.
\end{aligned}$$

A solution to the last equation can be found by further assuming that the individual x and y parts will be zero, leading to two separate equations

$$\begin{aligned}
B_y(t,T)\eta_y(t) + \frac{1}{2}B_y^2(t,T)\delta_y(t) + \partial_t\left(\log A_y\right)(t,T) &= 0, \quad A_y(T,T) = 1,\\
-B_x(t,T)\eta_x(t) + \frac{1}{2}B_x^2(t,T)\delta_x(t) + \partial_t\left(\log A_x\right)(t,T) &= 0, \quad A_x(T,T) = 1.
\end{aligned}$$

We will now turn to the special case of the CIR processes (3.3.4). We see immediately that the equations for x are in the usual form and defining $\lambda_x(t) \equiv -k_x, \eta_x(t) \equiv k_x\theta_x, \gamma_x(t) \equiv \sigma_x^2, \delta_x(t) \equiv 0$ yields the explicit solution from the literature (cf. Brigo and Mercurio (2006) p. 66 equation (3.25)).

Concerning the y terms, we make the following educated guess and verify, that it solves the equation:

$$\begin{aligned}
A(t,T) &= \left(\frac{2h\exp\left((k+h)(T-t)/2\right)}{2h + (k+h)\left(\exp\left((T-t)h\right) - 1\right)},\right)^{\frac{2k\theta}{\sigma^2}}\\
B(t,T) &= \frac{2\left(\exp\left((T-t)h\right) - 1\right)}{2h + (k+h)\left(\exp\left((T-t)h\right) - 1\right)},\\
h &= \sqrt{k^2 - 2\sigma^2},
\end{aligned}$$

where we will drop the index for indicating that we are considering the y coefficients for readability and assume that $k^2 \geq 2\sigma^2$.

Verification for B We will first check the formula for the Riccati equation in B:

We will now simplify the nominator and the denominator of $\partial_t B + \frac{1}{2}\sigma^2 B^2 - kB$, which is given by

$$
\begin{aligned}
&\partial_t B + \frac{1}{2}\sigma^2 B^2 - kB = \\
&\quad \frac{\sigma^2 \left(2\,\mathrm{e}^{\sqrt{k^2-2\,\sigma^2}\,(T-t)} - 2\right)^2}{2\left(\left(\mathrm{e}^{\sqrt{k^2-2\,\sigma^2}\,(T-t)} - 1\right)\left(k + \sqrt{k^2-2\,\sigma^2}\right) + 2\sqrt{k^2-2\,\sigma^2}\right)^2} \\
&\quad - \frac{2\,\mathrm{e}^{\sqrt{k^2-2\,\sigma^2}\,(T-t)}\sqrt{k^2-2\,\sigma^2}}{\left(\mathrm{e}^{\sqrt{k^2-2\,\sigma^2}\,(T-t)} - 1\right)\left(k + \sqrt{k^2-2\,\sigma^2}\right) + 2\sqrt{k^2-2\,\sigma^2}} \\
&\quad - \frac{k\left(2\,\mathrm{e}^{\sqrt{k^2-2\,\sigma^2}\,(T-t)} - 2\right)}{\left(\mathrm{e}^{\sqrt{k^2-2\,\sigma^2}\,(T-t)} - 1\right)\left(k + \sqrt{k^2-2\,\sigma^2}\right) + 2\sqrt{k^2-2\,\sigma^2}} \\
&\quad + \frac{\mathrm{e}^{\sqrt{k^2-2\,\sigma^2}\,(T-t)}\sqrt{k^2-2\,\sigma^2}\left(k + \sqrt{k^2-2\,\sigma^2}\right)\left(2\,\mathrm{e}^{\sqrt{k^2-2\,\sigma^2}\,(T-t)} - 2\right)}{\left(\left(\mathrm{e}^{\sqrt{k^2-2\,\sigma^2}\,(T-t)} - 1\right)\left(k + \sqrt{k^2-2\,\sigma^2}\right) + 2\sqrt{k^2-2\,\sigma^2}\right)^2}.
\end{aligned}
$$

After bringing the terms to the common denominator, we consider now the nominator of this transformation

$$
\begin{aligned}
&\frac{1}{2}\sigma^2\left(2\exp\left(\sqrt{k^2-2\sigma^2}\tau\right) - 2\right)^2 \\
&\quad - \left(2\sqrt{k^2-2\sigma^2}\exp\left(\sqrt{k^2-2\sigma^2}\tau\right) + k\left(2\exp\left(\sqrt{k^2-2\sigma^2}\tau\right) - 2\right)\right) \\
&\qquad \left(\left(\exp\left(\sqrt{k^2-2\sigma^2}\tau\right) - 1\right)\left(k + \sqrt{k^2-2\sigma^2}\right) + 2\sqrt{k^2-2\sigma^2}\right) \\
&\quad + \exp\left(\sqrt{k^2-2\sigma^2}\tau\right)\sqrt{k^2-2\sigma^2}\left(k + \sqrt{k^2-2\sigma^2}\right)\left(2\exp\left(\sqrt{k^2-2\sigma^2}\tau\right) - 2\right) \\
&\quad = 2\,h\,k - 2\,k^2 - 2\,k^2\,\mathrm{e}^{2\,h\,\tau} + 2\,\sigma^2 + 4\,\sigma^2\,\mathrm{e}^{h\,\tau} + 2\,\sigma^2\,\mathrm{e}^{2\,h\,\tau} - 2\,h\,k\,\mathrm{e}^{2\,h\,\tau},
\end{aligned}
$$

where we substituted $\tau := T - t$ and $h := \sqrt{k^2 - 2\sigma^2}$. The denominator can be simplified in the same way, leading to

$$
2\,k^2 - 2\,h\,k + 2\,k^2\,\mathrm{e}^{2\,h\,\tau} - 2\,\sigma^2 - 4\,\sigma^2\,\mathrm{e}^{h\,\tau} - 2\,\sigma^2\,\mathrm{e}^{2\,h\,\tau} + 2\,h\,k\,\mathrm{e}^{2\,h\,\tau}.
$$

In total, we see that the denominator differs only by a sign, hence $\partial_t B + \frac{1}{2}\sigma^2 B^2 - kB = -1$, which yields the claim.

Verification for A The formula can be derived by just integrating and taking the exponential.

$$
\left(\frac{2h\mathrm{e}^{\frac{1}{2}\tau(k+h)}}{\left(\mathrm{e}^{\tau\,h} - 1\right)(h+k) + 2h}\right)^{\frac{2\,k\,\theta}{\sigma^2}},
$$

where $h := \sqrt{k^2 - 2\,\sigma^2}$ and $\tau := T - t$.

Let us just take the logarithm and derivative to verify this formula:

$$\log\left(A(\tau)\right) = \frac{\tau\ \left(2\,k^2\,\theta + 2\,k\,\theta\,h\right)}{2\,\sigma^2} - \frac{2\,k\,\theta\ \ln\left(\mathrm{e}^{\tau\,h}\,h - k + h + k\,\mathrm{e}^{\tau\,h}\right)}{\sigma^2} + \frac{2\,k\,\theta\ \ln\left(2\,\sqrt{k^2 - 2\,\sigma^2}\right)}{\sigma^2}.$$

Now, taking the derivative yields

$$\partial_\tau\left(\log\left(A\left(\tau\right)\right)\right) = \frac{2\,k\,\theta\ \left(\mathrm{e}^{\tau\,h} - 1\right)}{\mathrm{e}^{\tau\,h}\,h - k + h + k\,\mathrm{e}^{\tau\,h}} = k\theta\frac{2\left(\mathrm{e}^{\tau\,h} - 1\right)}{\left(\mathrm{e}^{\tau\,h} - 1\right)\left(h + k\right) + 2h}.$$

After undoing the substitution for τ this is equal to $k\theta B$, which yields the claim.

B.2. Instantaneous Forward Rate

The definition of the instantaneous forward rate (cf. Brigo and Mercurio (2006) p. 13 equation (1.23)) is given by

$$f(t,T) := -\partial_T \log\left(P\left(t,T\right)\right).$$

By (3.3.5) we therefore have

$$\begin{aligned} f(t,T) &= -\partial_T\left(\log\left(A_x(t,T)e^{-B_x(t,T)x(t)}A_y(t,T)e^{B_y(t,T)y(t)}\right)\right) \\ &= -\partial_T\left(\log\left(A_x(t,T)\right) - B_x(t,T)x(t)\right) - \partial_T\left(\log\left(A_y(t,T)\right) + B_y(t,T)y(t)\right) \\ &= -\frac{\partial_T\left(A_x(t,T)\right)}{A_x(t,T)} + \partial_T\left(B_x(t,T)\right)x(t) - \frac{\partial_T\left(A_y(t,T)\right)}{A_y(t,T)} - \partial_T\left(B_y(t,T)\right)y(t). \end{aligned}$$

Let $z \in \{x,y\}$ and consider the case of the CIR model (3.3.4). Then those derivatives are given by the following expressions: Let us calculate the derivative of A_z first

$$\begin{aligned} &\partial_T\left(A_z(t,T)\right) \\ &= \phi_z^3\left(\frac{\phi_z^1\phi_z^2 e^{\phi_z^2(T-t)}}{\phi_z^1 + \phi_z^2\left(e^{\phi_z^1(T-t)} - 1\right)} - \frac{\left(\phi_z^1\right)^2\phi_z^2 e^{\phi_z^1(T-t)}e^{\phi_z^2(T-t)}}{\left(\phi_z^1 + \phi_z^2\left(e^{\phi_z^1(T-t)} - 1\right)\right)^2}\right)\left(\frac{\phi_z^1 e^{\phi_z^2(T-t)}}{\phi_z^1 + \phi_z^2\left(e^{\phi_z^1(T-t)} - 1\right)}\right)^{\phi_z^3 - 1}. \end{aligned}$$

Hence, we get

$$-\frac{\partial_T\left(A_x(t,T)\right)}{A_x(t,T)} = \frac{\phi_z^2\phi_z^3\left(\phi_z^1 - \phi_z^2\right)\left(e^{(T-t)\phi_z^1} - 1\right)}{\phi_z^1 + \phi_z^2\left(e^{(T-t)\phi_z^1} - 1\right)}.$$

Now, we compute the derivative of B_z

$$\partial_T\left(B_z(t,T)\right) = \frac{\left(\phi_z^1\right)^2 e^{(T-t)\phi_z^1}}{\left(\phi_z^1 + \phi_z^2\left(e^{(T-t)\phi_z^1} - 1\right)\right)^2}.$$

B.3. Hermite Polynomials

In this short section we briefly recall the probabilist's Hermite polynomials, which are key to the Gram-Charlier expansion.

Definition B.3.1. The *(probabilist's) Hermite polynomials* $H_n(x)$ are defined as $H_0(x) \equiv 1$ and for $n \geq 1$

$$(-1)^n \left(\varphi(x)\right)^{-1} \left(\frac{d^n}{dx^n}\varphi\right)(x),$$

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

Notice, that they are orthogonal with respect to the Gaussian measure, i.e.

$$\int_{\mathbb{R}} H_m(x)H_n(x)\varphi(x)dx = \delta_{nm}n!.$$

In particular,

$$H_1(x) = x, H_2(x) = x^2 - 1, H_3(x) = x^3 - 3x, H_4(x) = x^4 - 6x^2 + 3,$$
$$H_5(x) = x^5 - 10x^3 + 15x, H_6(x) = x^6 - 15x^4 + 45x^2 - 15, H_7(x) = x^7 - 21x^5 + 105x^3 - 105x.$$

B.4. Cumulants and Moments

Let us denote by μ_i the moments and by c_i the cumulants. Their relationship towards each other is determined by the moment generating function (cf. Smith (1995)) like follows

$$M(t) = 1 + \sum_{i=1}^{\infty} \mu_i \frac{t^i}{i!} = \exp\left(\sum_{i=1}^{\infty} c_i \frac{t^i}{i!}\right) = \exp\left(K(t)\right).$$

Therefore, assuming that the moments μ_i are known we can compute the cumulants c_i by differentiating the formula from above

$$c_i = \frac{d^i}{dt^i} \log\left(M(t)\right)\bigg|_{t=0}.$$

Since, we only need a few of them, we can compute the formulas and implement them directly, leading to

$$c_1 = \mu_1, \quad c_2 = \mu_2 - \mu_1^2, \quad c_3 = 2\mu_1^3 - 3\mu_2\mu_1 + \mu_3, \quad c_4 = -6\mu_1^4 + 12\mu_2\mu_1^2 - 4\mu_3\mu_1 - 3\mu_2^2 + \mu_4,$$
$$c_5 = 24\mu_1^5 - 60\mu_2\mu_1^3 + 20\mu_3\mu_1^2 + 30\mu_2^2\mu_1 - 5\mu_4\mu_1 - 10\mu_2\mu_3 + \mu_5$$
$$c_6 = -120\mu_1^6 + 360\mu_2\mu_1^4 - 120\mu_3\mu_1^3 - 270\mu_2^2\mu_1^2 + 30\mu_4\mu_1^2 + 120\mu_2\mu_3\mu_1 - 6\mu_5\mu_1 + 30\mu_2^3 - 10\mu_3^2 - 15\mu_2\mu_4 + \mu_6$$
$$c_7 = 720\mu_1^7 - 2520\mu_2\mu_1^5 + 840\mu_3\mu_1^4 + 2520\mu_2^2\mu_1^3 - 210\mu_4\mu_1^3 - 1260\mu_2\mu_3\mu_1^2 + 42\mu_5\mu_1^2 - 630\mu_2^3\mu_1 + 140\mu_3^2\mu_1 + 210\mu_2\mu_4\mu_1 - 7\mu_6\mu_1 + 210\mu_2^2\mu_3 - 35\mu_3\mu_4 - 21\mu_2\mu_5 + \mu_7.$$

B.5. Least Square Monte Carlo Method (LSMC)

In this section, we will demonstrate how to approximate the conditional expectation via LSMC. Let us first of all recall the following facts about the conditional expectation (cf. Pascucci (2011): *pp. 654 ff.*):

Let $X \in L^2(\Omega, \mathcal{F}, \mathbb{Q})$ and $\mathcal{A} \subseteq \mathcal{F}$ be a sub-σ-algebra.

1. Then the conditional expectation is the L^2-best approximation, i.e.

$$\mathbb{E}^{\mathbb{Q}}\left[\left(X - \mathbb{E}^{\mathbb{Q}}\left[X|\,\mathcal{A}\right]\right)^2\right] \leq \mathbb{E}^{\mathbb{Q}}\left[(X-Y)^2\right]$$

for all $Y \in L^2(\Omega, \mathcal{A}, \mathbb{Q})$.

2. Furthermore, the factorization Lemma tells us that there exists a function u, such that

$$\mathbb{E}^{\mathbb{Q}}\left[Y|\,R\right] = u(R)$$

and combined with the argument above

$$u(R) = \arg\min_{v(\cdot)} \mathbb{E}^{\mathbb{Q}}\left[|v(R) - Y|^2\right]$$

where $v(\cdot)$ runs over all measurable functions.

The idea is now to approximate the function $u(x)$. Therefore, fix a basis $(b_i(x))_{i=1,\ldots,n}$ and set $b^n(\cdot) \coloneqq [b_1(\cdot), \ldots, b_n(\cdot)]$. Then, we approximate u by $u(x) \approx \lambda^T b^n(x)$ where λ solves the least square problem

$$\lambda = \arg\min_{\alpha \in \mathbb{R}^n} \mathbb{E}^{\mathbb{Q}}\left[\left|\alpha^T b^n(R) - Y\right|^2\right].$$

The problem we encounter is that in this least square problem we have random variables. So we can numerically deal with this problem by simulating those random variables, if it is possible, and view this least square problem as finding a linear regression for data points introduced by the realizations of the random variables.

Thus, let y_i be realizations of Y and set $y = [y_1, \ldots, y_m]^T$. Additionally, let $b_{ij} = b_i(r_j)$, where r_j is a realization of R, and define the matrix $b = [b_{ij}]_{i=1,\ldots,n;j=1,\ldots,m}$.

Then the above least square problem reads

$$\lambda = \arg\min_{\alpha \in \mathbb{R}^n} |b\alpha - y|^2.$$

This is known as ordinary least square problem and the optimal solution is given by

$$\lambda = \left(b^T b\right)^{-1} b^T y.$$

This tells us how to approximate the conditional expectation via a Monte Carlo linear regression approach.

B.6. Mean Error of Discount Factors

Table B.1.: Mean error of discount factors (DF) of our model with parameters given in Table 3.2 at 30/12/2019 and 30/11/2020.

Maturity (in years)	Mean Error of DF at 30/12/2019	Mean Error of DF at 30/11/2020
0.0833333	0.000409963	0.000560815
0.25	0.00112268	0.00119113
0.5	0.00161849	0.00144972
0.75	0.00137477	0.00113393
1	0.000814101	0.000550498
1.25	0.000328491	0.000398677
1.5	$1.4362e-05$	0.000571146
1.75	$5.62967e-05$	0.000805009
2	0.000402274	0.00125292
2.25	0.000824884	0.00191315
2.5	0.0011063	0.00288173
2.75	0.00133005	0.00356734
3	0.00143482	0.00401817
3.25	0.0018223	0.00441028
3.5	0.00242705	0.00477583
3.75	0.00303645	0.00512345
4	0.00354305	0.00537247
4.25	0.00409665	0.0051572
4.5	0.00461513	0.00473785
4.75	0.00491301	0.00449665
5	0.00502117	0.00423046
5.25	0.00485501	0.00398605
5.5	0.0048752	0.00410309
5.75	0.00549422	0.00442951
6	0.00642713	0.00477755
6.25	0.00743045	0.00495872
6.5	0.00841358	0.00514248
6.75	0.00945159	0.00546137
7	0.010154	0.00596334
7.25	0.0105884	0.006623
7.5	0.0111652	0.00753078
7.75	0.0116295	0.00803137
8	0.0120698	0.00842589
8.25	0.0125788	0.00889094
8.5	0.0132799	0.00938679
8.75	0.0138392	0.00983092
9	0.0146873	0.0100521
9.25	0.0156794	0.0103367
9.5	0.0166502	0.0109601
9.75	0.0174248	0.0117957
10	0.0180652	0.0126498
15	0.0251373	0.0314676
20	0.0331911	0.0233781
25	0.0160001	0.0174282
30	0.00094742	0.00846645

B.7. Market Data

Table B.2.: Market data containing the volatility surface for the swaption pricing at 30/12/2019 in bps.

Tenor Maturity	1	2	3	4	5	7	10
1	17.5	21.8	26.8	31.4	35.2	40.2	45.6
2	25.4	29.3	33.5	36.4	39.5	43.5	47.5
3	34	36.7	39.2	41.1	43.2	46.2	49.3
4	40	41.5	43.4	44.8	46.2	48.4	50.9
5	43.7	44.6	45.8	47	48.4	50.1	52.3
7	49.7	49.8	50.5	51.4	52.1	53.1	54.4
10	54.6	54.4	54.7	54.9	55.1	55.2	55.6
15	54.8	54.4	54.5	54.4	54.2	54.2	54.4
20	53.6	53.2	53.4	53	52.9	52.8	52.5

Table B.3.: Market data containing the volatility surface for the swaption pricing at 30/11/2020 in bps.

Tenor Maturity	1	2	3	4	5	7	10
1	16.4	18.7	21.6	24.4	26.9	31.5	36.5
2	21.2	24.2	27.4	29.9	31.7	35.9	40.6
3	27.3	29.9	32.4	34.3	36	39.5	43.3
4	32.4	34.6	36.6	38.2	39.5	42	45.5
5	36.7	38.5	39.9	41	42.2	44.4	47.3
7	43.4	44.2	45	45.8	46.6	47.9	49.7
10	48.5	48.6	49.2	49.7	50.1	50.6	51.7
15	50.3	50.1	50.6	50.7	50.7	51	51.3
20	49.8	49.7	50	50	50	49.9	49.6

Table B.4.: Market data containing the swaption strikes at 30/12/2019.

Tenor Maturity	1	2	5	7	10
1	−0.260793 %	−0.195187 %	−0.011405 %	0.140129 %	0.330514 %
2	−0.129665 %	−0.0782444 %	0.139932 %	0.273273 %	0.449172 %
5	0.268095 %	0.38307 %	0.556996 %	0.655339 %	0.757978 %
7	0.547079 %	0.611571 %	0.76683 %	0.830788 %	0.891069 %
10	0.880582 %	0.907944 %	0.967521 %	0.988131 %	0.992003 %
15	1.04232 %	1.04153 %	1.01776 %	0.985317 %	0.924744 %
20	0.925377 %	0.901441 %	0.827386 %	0.778437 %	0.721445 %

Table B.5.: Market data containing the swaption strikes at 30/11/2020.

Tenor Maturity	1	2	5	7	10
1	−0.558066 %	−0.544838 %	−0.455765 %	−0.37221 %	−0.238803 %
2	−0.531679 %	−0.502856 %	−0.386908 %	−0.294521 %	−0.162606 %
5	−0.315638 %	−0.264729 %	−0.117094 %	−0.0324645 %	0.0536401 %
7	−0.117189 %	−0.0652544 %	0.0603589 %	0.1157 %	0.150538 %
10	0.150213 %	0.179568 %	0.225372 %	0.223805 %	0.196761 %
15	0.234862 %	0.219855 %	0.16784 %	0.12791 %	0.0641018 %
20	0.0500327 %	0.0277677 %	−0.0398808 %	−0.0837531 %	−0.134806 %

Table B.6.: Market data containing the swaption prices at 30/12/2019.

Tenor Maturity	1	2	5	7	10
1	0.000702236	0.00175071	0.00706456	0.0112631	0.0181169
2	0.0014433	0.00333027	0.0111956	0.017189	0.0265694
5	0.00391314	0.00796766	0.0214221	0.0308074	0.0453508
7	0.00521117	0.0104082	0.0268942	0.0380283	0.0548627
10	0.00668368	0.0132567	0.0330802	0.045932	0.0651091
15	0.00781681	0.0154396	0.0378811	0.0525334	0.0743464
20	0.00840243	0.0166069	0.0407885	0.0565876	0.0795953

Table B.8.: Bloomberg's Hull-White one factor prices of receiver Bermudan swaptions at 30/12/2019.

Tenor Maturity	2	5	7	10
1	0.21%	1.06%	1.85%	3.28%
3	0.57%	1.83%	2.86%	4.63%
5	0.87%	2.48%	3.71%	5.72%
7	1.11%	3.03%	4.43%	6.65%
10	1.4%	3.62%	5.2%	7.59%

Table B.9.: Bloomberg's Hull-White one factor prices of payer Bermudan swaptions at 30/12/2019.

Tenor Maturity	2	5	7	10
1	0.25%	1.4%	2.55%	4.76%
3	0.6%	2.08%	3.42%	5.74%
5	0.9%	2.7%	4.16%	6.59%
7	1.13%	3.2%	4.75%	7.18%
10	1.41%	3.72%	5.33%	7.91%

Table B.7.: Market data containing the swaption prices at 30/11/2020.

Tenor Maturity	1	2	5	7	10
1	0.000661505	0.00151259	0.00547626	0.00900521	0.0149394
2	0.00121578	0.00278226	0.00916199	0.0145571	0.0235389
5	0.00336776	0.00707345	0.0194074	0.0285692	0.0433793
7	0.0047279	0.00963075	0.0253482	0.0364087	0.0537919
10	0.00630168	0.0126162	0.0323949	0.0456958	0.0665005
15	0.00790807	0.0157371	0.0397201	0.0558865	0.0802858
20	0.00898181	0.017927	0.0451252	0.0631302	0.0898938

Table B.10.: Market data containing the Bermudan swaption strikes at 30/12/2019.

Tenor Maturity	2	5	7	10
1	−0.194%	0.00912%	0.14%	0.33%
3	0.0789%	0.274%	0.432%	0.561%
5	0.335%	0.534%	0.644%	0.767%
7	0.612%	0.771%	0.84%	0.894%
10	0.926%	1.01%	0.994%	1.02%

Table B.11.: Market data containing the zero rate curve and zero coupon curve at 30/12/2019.

Maturity (in years)	Zero rate (in %)	Zero-coupon price
0.083333	−0.469999993219972	1.0004001991529
0.25	−0.388000020757318	1.00096969387991
0.5	−0.324999983422458	1.00163343819125
0.75	−0.314333918504417	1.00237481461989
1	−0.322000007145107	1.00323926670136
1.25	−0.323286440253412	1.00405360258242
1.5	−0.316161320131414	1.00476558980205
1.75	−0.303842297803669	1.00535001652119
2	−0.289547047577798	1.00582418019158
2.25	−0.275860329135469	1.00623288634409
2.5	−0.262835313503729	1.006604855007
2.75	−0.249892233800608	1.00691299093433
3	−0.236451346427202	1.00713375064174
3.25	−0.222084053437044	1.00725039326453
3.5	−0.20696636298112	1.00728054250496
3.75	−0.191425434683623	1.00721781901104
4	−0.175788428168744	1.00706740209126
4.25	−0.160311330630236	1.00684531811395
4.5	−0.144965462482105	1.00655553463348
4.75	−0.129650957156002	1.00618948972951
5	−0.114267959725112	1.00573933685071
5.25	−0.0987154224631581	1.00520062530541
5.5	−0.0828875612342017	1.00457454544122
5.75	−0.0666773874613114	1.00384671986489
6	−0.0499779242090881	1.00300667524933
6.25	−0.0327643402378897	1.00205088034181
6.5	−0.0153403983915723	1.00099833086134
6.75	0.00190798987986796	0.999871102605028
7	0.0185949131264351	0.998698306220564
7.25	0.0344518735623467	0.997505079039002
7.5	0.0496800311054812	0.996279818846146
7.75	0.0645979575189415	0.995003816465917
8	0.0795242260210216	0.993656440330286
8.25	0.0947347900819295	0.992214008696662
8.5	0.110335148849572	0.990662992494919
8.75	0.126388167535652	0.988997743889118
9	0.142956722993404	0.987213788328959
9.25	0.160050573928316	0.985308478446392
9.5	0.177466994199449	0.983284710270437
9.75	0.194950156980411	0.981173005874126
10	0.212244223803282	0.979004189945635
15	0.473523046821356	0.931543316237289
20	0.611338950693607	0.885166902653398
25	0.652327481657267	0.849865688031976
30	0.640345783904195	0.825611308910539

Table B.12.: Market data containing the zero rate curve and zero coupon curve at 30/11/2020.

Maturity (in years)	Zero rate (in %)	Zero-coupon price
0.083333	−0.499999988824129	1.00041207460911
0.24444	−0.526142632588744	1.00130160930003
0.5	−0.507755391299725	1.00252751322004
0.75	−0.503638433292508	1.00378359716281
1	−0.517199980095029	1.00519888845098
1.24444	−0.524928161568994	1.00658242962185
1.5	−0.525975602238304	1.00791998180754
1.75	−0.522984338103072	1.00920759910672
2	−0.518596358597279	1.01045317135578
2.24444	−0.514924664329897	1.01166557509739
2.5	−0.511966253088758	1.0128933501334
2.75	−0.509189122195153	1.01412683679008
3	−0.50606126897037	1.01533680279481
3.24722	−0.502153361016866	1.0164921683478
3.5	−0.497446840398652	1.01760034898714
3.75	−0.492025761253245	1.01867206863544
4	−0.485974224284291	1.01969106459461
4.24444	−0.479376664922526	1.02062910472473
4.5	−0.47231881015648	1.02152663951854
4.75	−0.464886791550967	1.02238363928715
5	−0.457166694104671	1.02318805666289
5.24444	−0.449221897941854	1.02391569856637
5.5	−0.441024916645461	1.0246017102629
5.75	−0.432525547282481	1.02524040701367
6	−0.423673586919904	1.02581359580938
6.24444	−0.414435069675712	1.02629288605375
6.5	−0.404840884313273	1.02671307570551
6.75	−0.394938214854612	1.02707383363012
7	−0.384774198755622	1.02736441987847
7.24722	−0.37439006592308	1.02757527517123
7.5	−0.363803462582268	1.02771717281163
7.75	−0.353026057560157	1.0277994267812
8	−0.342069566249847	1.02781095336332
8.24444	−0.330958580803831	1.02773379298185
8.5	−0.319769247063562	1.02760240180053
8.75	−0.308590599271419	1.02742329257773
9	−0.297511671669781	1.02719551414522
9.24444	−0.286599750659389	1.02691014728746
9.5	−0.275835084599585	1.02659173861864
9.75	−0.265176203111905	1.02623638422587
10	−0.254581612534821	1.02583261454546
15	−0.0622837862465531	1.00939445284171
20	0.0184025324415416	0.996324084241296
25	0.0234601888223551	0.994148968990786
30	−0.00393075206375215	1.00118069913068

Rating Transition Modelling

C

C.1. TimeGAN and Default Data

In this appendix, we list the rating matrices used for Section 4.4.

Table C.1.: Expected rating matrix under measure ℙ for one month using TimeGAN.

To From	**A**	**B**	**C**	**D**
A	99.481 %	0.523 %	0.000 %	0.000 %
B	0.097 %	99.665 %	0.235 %	0.003 %
C	0.000 %	0.326 %	98.626 %	0.983 %
D	0.000 %	0.000 %	0.000 %	100.000 %

Table C.2.: Expected rating matrix under measure ℙ for three months using TimeGAN.

To From	**A**	**B**	**C**	**D**
A	98.720 %	1.278 %	0.003 %	0.000 %
B	0.226 %	99.201 %	0.544 %	0.019 %
C	0.001 %	0.937 %	96.198 %	2.697 %
D	0.000 %	0.000 %	0.000 %	100.000 %

Table C.3.: Expected rating matrix under measure ℙ for six months using TimeGAN.

To From	**A**	**B**	**C**	**D**
A	97.381 %	2.601 %	0.014 %	0.000 %
B	0.444 %	98.439 %	1.047 %	0.057 %
C	0.008 %	2.262 %	91.357 %	6.485 %
D	0.000 %	0.000 %	0.000 %	100.000 %

Table C.4.: Expected rating matrix under measure $\mathbb{P}$ for twelve months using TimeGAN.

To From	**A**	**B**	**C**	**D**
A	95.142 %	4.801 %	0.056 %	0.004 %
B	0.812 %	97.142 %	1.883 %	0.184 %
C	0.071 %	4.648 %	84.422 %	11.532 %
D	0.000 %	0.000 %	0.000 %	100.000 %

Table C.5.: Dataset one of default probabilities under measure $\mathbb{Q}$ for $1, 3, 6, 12$ months.

To From	**D** $\left(t = \frac{1}{12}\right)$	**D** $\left(t = \frac{3}{12}\right)$	**D** $\left(t = \frac{6}{12}\right)$	**D** $\left(t = \frac{12}{12}\right)$
A	0.005 %	0.015 %	0.031 %	0.061 %
B	0.075 %	0.225 %	0.444 %	0.868 %
C	1.378 %	4.077 %	7.987 %	15.336 %
D	100.000 %	100.000 %	100.000 %	100.000 %

Table C.6.: Dataset two of default probabilities under measure $\mathbb{Q}$ for $1, 3, 6, 12$ months.

To From	**D** $\left(t = \frac{1}{12}\right)$	**D** $\left(t = \frac{3}{12}\right)$	**D** $\left(t = \frac{6}{12}\right)$	**D** $\left(t = \frac{12}{12}\right)$
A	0.512 %	1.531 %	3.053 %	6.065 %
B	0.754 %	2.245 %	4.440 %	8.682 %
C	1.378 %	4.077 %	7.987 %	15.336 %
D	100.000 %	100.000 %	100.000 %	100.000 %

Table C.7.: Dataset three of default probabilities under measure $\mathbb{Q}$ for $1, 3, 6, 12$ months.

To From	**D** $\left(t = \frac{1}{12}\right)$	**D** $\left(t = \frac{3}{12}\right)$	**D** $\left(t = \frac{6}{12}\right)$	**D** $\left(t = \frac{12}{12}\right)$
A	2.400 %	7.200 %	14.400 %	28.800 %
B	3.042 %	9.125 %	18.250 %	36.500 %
C	4.708 %	14.125 %	28.250 %	56.500 %
D	100.000 %	100.000 %	100.000 %	100.000 %