



université
PARIS-SACLAY



Paris-Saclay University

Internship report

Application of Machine Learning in Risk Metrics Calculation

Author : Ms. Hadil SAADAOU
Academic tutor : Mr. Ahmed KEBAIER

Company : Natixis CIB
Internship supervisor : Mr. Nouredine LEHDILI

Internship from Monday April 11, 2023 to Friday September 29, 2023

Acknowledgments

Before starting the development of this report, I would like to thank in particular the people who have enabled me to carry out this work.

First of all, I sincerely thank my internship supervisor Mr. Nouredine LEHDILI for giving me the opportunity to work with him and whose expertise and knowledge was crucial in formulating this work.

I also thank Hoang Dung NGUYEN, quantitative analyst, for his very high availability and support. He has always been giving very good advice and providing constant follow-up of the subject, and I have acquired good technical skills from his expertise. And Nisrine MADHAR, for her support and listening, whether it is for a remark, an opinion or especially in front of an encountered problem.

I would also thank the entire Natixis Enterprise Risk Modelling team for accompanying me throughout this internship, for the trust they gave me in order to carry out my work and for providing considerable support.

I sincerely thank my family for their advice as well as for their unconditional support and encouragement, they never failed to support me financially and emotionally in every step of my life. I know that I would not be where I am today if it were not for them. And finally, I thank my friends for always motivating, supporting and raising me up throughout the lowest days.

Abstract

The main objective of the internship was to explore a numerical approximation that is able to ease the computational burden associated with the number of pricing needed to estimate the market and counterparty risk measures. The aim is then to determine prices of a portfolio of derivatives, with the least frequent use of the the pricing function, while maintaining satisfactory accuracy. To this end, we opted for an interpolation approach involving Chebyshev polynomials in combination with principal component analysis. Promising numerical results were obtained for a large class of assets and models, and showed that these interpolators met our needs both in terms of computational cost and accuracy.

Key words: value-at-risk, Expected Shortfall, Expected Exposure, Credit Valuation Adjustment, FRTB, Chebyshev interpolation, Principal Component Analysis, calculation's performance.

Table of contents

Introduction	7
Entreprise presentation	9
1 Risk Calculation in Banks	10
1.1 Market and Counterparty Risk Measures	10
1.1.1 Market Risk Measures	10
1.1.2 Counterparty Risk Measures	12
1.2 Evaluation of Financial Instruments	15
1.2.1 Interest Rate Swaps	15
1.2.2 European Swaptions	16
1.3 Diffusion Models of Risk Factors	17
1.3.1 LGM Zero Coupon Yield Curve Model	18
1.3.2 SABR Volatility Model	22
1.4 Risk Calculation Flow in Traditional Approach	25
2 Application of Chebyshev Interpolation in Risk Calculation	26
2.1 Review of Polynomial Interpolation Methods	27
2.1.1 Vandermonde Matrix Interpolation	27
2.1.2 Lagrange Interpolation	28
2.1.3 Runge's Phenomenon	28
2.2 Chebyshev Interpolation	29
2.2.1 Chebyshev Nodes and Polynomials	29
2.2.2 Evaluation of Chebyshev Interpolants	31
2.2.3 Chebyshev Interpolants in High Dimension	33
2.2.4 Principal Component Analysis for Dimensionality Reduction	34
2.2.5 Sliding Technique	35
3 Financial Trading Portfolios Case Study	39
3.1 Market Risk Calculation	39
3.1.1 Numerical Results of the Traditional Approach	39
3.1.2 Application of the Alternative Approach	40
3.2 Counterparty Risk	44
3.2.1 Numerical Results of the Traditional Approach	44
3.2.2 Approximation to the Valuation of a Swap	45
3.2.3 Numerical Results of the Alternative Approach	48
Conclusion	51
A LGM-1F	53

B	Change of Numeraire	56
C	Bond Options Pricing	58
C.1	Bond Option	58
C.2	Modified Bond Option	60

List of figures

1.1	Initial Zero Coupon yield curve : The table (<i>left</i>) shows the values of ZC rates for different maturities. These values are plotted (<i>right</i>) to form an initial yield curve. (<i>source</i> : Natixis).	21
1.2	SABR calibrated parameters: The matrices correspond to the parameters $(\sigma_{atm}, \beta, \rho, \nu)$ at different maturities and tenors (<i>source</i> : Natixis).	23
1.3	Initial implied volatility surface: three-dimensional surface plotted whereby the implied volatility (z -axis) for swaptions is plotted against the maturity (y -axis) and tenor (x -axis) in years	24
1.4	Risk calculation flow by MC approach (<i>source</i> : Ruiz and Zeron (2021))	25
2.1	Alternative risk flow (<i>source</i> : (Ruiz and Zeron, 2021))	26
2.2	Lagrange interpolation of the Runge's function by different number of equidistant points: (<i>Left</i>) 5 points, (<i>Center</i>) 10 points and (<i>Right</i>) 14 points.	29
2.3	Chebyshev points as projection result of projecting equidistant points on the upper half of the unitary circle onto the real line.	30
2.4	Interpolation of the Runge's function by different number of Chebyshev points: (<i>Left</i>) 5 points, (<i>Center</i>) 12 points and (<i>Right</i>) 18 points.	33
2.5	Two-dimensional Chebyshev grid points on $[-1, 1]^2$	34
2.6	Two-dimensional sliders grid points (<i>red</i>) vs. full Chebyshev grid points (<i>blue</i>)	37
2.7	Full Chebyshev grid points (<i>left</i>) VS sliders grid points (<i>right</i>).	37
3.1	Comparaison of PnL distribution at time $t = 10$ days: The true prices distribution (<i>left</i>) is the result of the full pricing approach whereas the approximated prices (<i>right</i>) were obtained by applying the Chebyshev interpolation.	41
3.2	Q-Q plot for comparing prices distribution (full pricing approach vs. Chebyshev interpolation): a point (x, y) on the plot corresponds to one of the quantiles of the true prices' distribution (x -axis) plotted against the same quantile of the Chebyshev prices' distribution (y -axis).	42
3.3	Comparaison of PnL distribution at time $t = 10$ days: The true prices's distribution (<i>left</i>) is the result of the full pricing approach whereas the approximative prices (<i>right</i>) were obtained by applying the sliding technique.	43
3.4	Q-Q plot for comparing prices distribution (full pricing approach vs. sliding technique): a point (x, y) on the plot corresponds to one of the quantiles of the true prices' distribution (x -axis) plotted against the same quantile of the slider prices' distribution (y -axis).	43
3.5	Expected Exposure profile of portfolio 2 on 8-year time horizon calculated with MC approach.	45
3.6	EE profile of a single interest rate swap by the normal dynamics approximation of swap rate vs. EE profile of the same swap by MC approach.	48
3.7	Chebyshev interpolation on time and risk factors domain.	49

3.8	EE profile for portfolio 2 by MC approach (<i>blue</i>) vs. Chebyshev interpolation (<i>red</i>).	49
3.9	Influence of the number of interpolation nodes on the EE profile. For each sub-plot, the number of grid points is indicated.	50

List of tables

1.1	Calibration error of the LGM 1F model.	21
3.1	Performace of the full pricing approach in VaR/ES calculation	40
3.2	Comparaison of performace in VaR/ES calculation	44
3.3	Performance of the full pricing approach in CVA calculation	45
3.4	Performace of the Chebyshev interpolation in CVA calculation	50

Introduction

Risk calculations has always been a core activity within the bank, and even more so since the 2008 financial crisis. Since then, there have been increasing questions about how risks are detected, quantified, reported and addressed. According to the new regulations, the internal methods used to measure these risks are all complex and potentially costly in terms of computational time.

For instance, the calculation of capital within the [Fundamental Review of the Trading Book](#) regulation (FRTB) requires, under Internal Model Approach (IMA), the daily calculation of market risk measures with different liquidity horizons. These calculations require the pricing of portfolios on hundreds if not thousands of scenarios. For the vast majority of these calculations, the main computational bottleneck comes in the evaluation of the portfolio pricer over all the risk factors scenarios. Hence, for a bank the real challenge is to be able to compute these risk measures while maintaining reasonable calculation times.

This remain equally true in the context of counterparty risk metrics computation, for which numerous future pricing runs are necessary to obtain exposures, the building blocks of many risk metrics. The accuracy of estimators is often achieved at the expense of the time required for calculation, which in some cases is exorbitant.

This internship attempts to address the following problem: how can we reduce the computational time required to calculate a set of prices, while maintaining satisfactory accuracy? The academic and professional literature [includes numerous methods to address this issue, often based on approximations of pricing functions](#). In particular, the application of machine learning in risk management represents a new avenue of research to build more complex risk models, at the initiative of regulators, while maintaining an acceptable computational time and accuracy level from a bank point of view. We refer to (Ruiz and Zeron, 2021) which introduces several techniques to sidestep the pricing problem in risk metrics calculation using machine learning algorithms, namely, neural networks and Chebyshev tensors.

In this work, we focus on a technique that is based on Chebyshev interpolants which have two unique mathematical properties. First, given an analytical target function, the approximated interpolant [converges](#) exponentially to the target. Second, the interpolation guarantees a fast and numerically [stable](#) evaluation. These two properties, detailed in subsequent sections, make Chebyshev interpolation a unique pricing function approximator as it massively reduces the computational burden associated with risk calculations that comes with pricing portfolios on multiple scenarios, [while keeping precision ranges well within what is required by regulators](#).

This report consists in three main chapters :

- The first chapter consists of theoretical introduction to the calculation of market and counterparty risk measures within the FRTB regulation. Furthermore, we present the essential building blocks for the practical implementation of numerical testing. This corresponds to the modeling of risk factors on the one hand, and the definition of pricing

methods associated with financial products constituting the study case portfolios on the other hand. Finally, we illustrate the computational bottleneck of the traditional approach in risk calculation.

- The second chapter comes to study interpolation theory and proposes an alternative method for risk metrics calculation. This approach consists in Chebyshev interpolation whose aspects will be detailed along with its pros and cons. We also introduce two dimensionality reduction technique (Principal Component Analysis, and the sliding technique) to bypass the exponential growth in Chebyshev grid points.
- In the last chapter, and in order to illustrate the first two chapters, several case studies involving different models and pricing methods of financial products, will be presented. We discuss the implementation of Chebyshev interpolants along with the numerical results of this alternative approach. Finally, We compare the precision and time performance of the new method against the traditional approach in risk metrics calculations.

Company presentation

Natixis is a french institution specialized in asset and wealth management, corporate and investment banking, insurance and payments. Created in 2006, Natixis is a subsidiary of the BPCE group (France's second largest banking group), with offices in 30 countries and around 13,600 employees. It is a banking partner of companies and institutions, for which it develops five complementary fields of expertise around the world:

- Asset and wealth management
- Corporate and investment banking
- Financial services
- Insurance
- Corporate banking

The internship took place within the Risk Management Department, in the **Enterprise Risk Management** (ERM) department. The mission of the Risk Division is to have a cross-functional view of all the risks to which Natixis is exposed in these activities and to monitor and control them.

The Enterprise Risk Management (ERM) department aims to organise and design the various risk assessment models and methodologies (market, counterparty, credit, operational, etc.). The goal is to extract the best decisions to make in order to find a compromise between performance and risk exposure. Its main tasks will be:

- the identification, modelling and quantification of the set of risks to which Natixis is exposed,
- the responsibility for exposure to Natixis risks,
- the definition of the Natixis stress-testing framework for all risks.

This department is then divided into four teams, each dealing with a different risk category (market, credit, stress test and operational).

Chapter 1

Risk Calculation in Banks

1.1 Market and Counterparty Risk Measures

The Basel Framework is the full set of standards of the Basel Committee on Banking Supervision (BCBS), which is the primary global standard setter for the prudential regulation of banks.

Basel III is an internationally agreed set of measures developed by the Basel Committee on Banking Supervision in response to the financial crisis of 2007-09. The measures aim to strengthen the regulation, supervision and risk management of banks. Like all Basel Committee standards, Basel III standards are minimum requirements which apply to internationally active banks. Members are committed to implementing and applying standards in their jurisdictions within the time frame established by the Committee (Basel Committee on Banking Supervision, 2017).

1.1.1 Market Risk Measures

According to Basel Committee on Banking Supervision, Market risk is defined as "the risk of losses in on- and off-balance sheet risk positions arising from movements in market prices" (Basel Committee on Banking Supervision, 2020).

In other words, market risk can be defined as the risk of loss arising from unpredictable fluctuations in market prices or interest rates. Value at risk and expected shortfall are two measures used to quantify market risk.

In this section, we detail the main of the market risk measures calculation. We focus on the value-at-risk and Expected Shortfall and their numerical computing method.

1.1.1.1 Value-at-Risk

Value at risk, denoted by VaR, is a measure used in risk management to assess the worst potential loss on a portfolio of instruments resulting from market movements over a given time horizon and a pre-defined confidence level (Basel Committee on Banking Supervision, 2020).

In other terms, the value-at-risk can be defined as the maximum potential loss that a portfolio may incur over a given time horizon with a given probability. It is thus characterized by the following parameters:

- The time horizon h , corresponds to the time horizon over which portfolio variations are measured.

- The confidence level α , gives the probability that the observed loss will be less than the calculated VaR.
- The value of the portfolio concerned by the VaR calculation.

This risk measure allows the bank to identify the minimum capital required to hedge market risk. For instance, if a bank's VaR over a 10-day horizon is 20 million euros with a probability of 99%, this means that there is a 1% chance that the loss will surpass 20 million euros in the next 10 days. Subsequently, the bank must preserve part of its capital to meet this potential loss.

From a mathematical point of view, let V_t denotes the mark-to-market (MtM) value at time t of a given portfolio. We define the loss $L_t(h)$ over time horizon h as follows:

$$L_t(h) = -(V_{t+h} - V_t). \quad (1.1.1)$$

We can thus define the VaR over the time interval h at a confidence level $(1 - \alpha)$ as follows:

$$\begin{aligned} \text{VaR}_\alpha(h) &= -\inf \{v \mid \mathbb{P}(\Delta V_t \leq v) > \alpha\} \\ &= -\inf \{v \mid \mathbb{P}(L_t(h) \geq -v) > \alpha\}, \end{aligned} \quad (1.1.2)$$

where $\Delta V_t = V_{t+h} - V_t = -L_t(h)$ represents the PnL (Profit and Loss) of the portfolio over the time horizon h . (Lehdili et al., 2019)

From a statistical point of view, VaR is the quantile of the loss for probability α :

$$\mathbb{P}(L_t(h) \leq \text{VaR}_\alpha(h)) = \alpha, \quad (1.1.3)$$

which implies

$$\text{VaR}_\alpha(h) = F_L^{-1}(\alpha). \quad (1.1.4)$$

However, even if the Value at Risk is a widely used measure of risk, it has its limitations. Primarily, it ignores the tail of the left-hand distribution of PnL distribution beyond the confidence level. To overcome this problem, another risk measure has been introduced, namely Expected Shortfall.

1.1.1.2 Expected Shortfall

The market risk framework of the Basel accord consists of an internal models' approach (IMA) and a standardised approach (SA). To capture tail risk better, the revised framework also saw a shift in the measure of risk under stress from the value-at-risk to Expected Shortfall. In addition the Basel committee proposes that the confidence level is to be changed from 99% to 97.5% for ES. Two daily VaR's for both confidence level 99% and 97.5% to be calculated and a daily Expected Shortfall for confidence level 97.5% (Lehdili et al., 2019).

Expected shortfall (ES) quantifies the average of potential losses exceeding the VaR at a given confidence level (Basel Committee on Banking Supervision, 2020), it is defined as follows

$$\text{ES}_\alpha(h) = \mathbb{E}[L_t(h) \mid L_t(h) \geq \text{VaR}_\alpha(h)] = \frac{1}{1 - \alpha} \int_0^{1-\alpha} \text{VaR}_\beta(h) d\beta, \quad (1.1.5)$$

where VaR_β is the value-at-risk for confidence level β , which changes from 0 to β (Lehdili et al., 2019).

1.1.1.3 Estimation of the VaR/ES

There exist 3 approaches to estimate the VaR/ES of a portfolio (Lehdili et al., 2019):

- The historical method which assumes that all future scenarios are entirely represented by recent past data.
- The parametric method involves using statistical models or distributions to estimate the VaR.
- Monte-Carlo approach is based on the diffusion of the risk factors of the portfolio over the desired time horizon.

In this work, we are only interested in the Monte-Carlo (MC) approach. In fact, this method is the most known method for developing accurate VaR estimation. For each trajectory of the diffusion, the entire portfolio is re-evaluated, allowing to take into account all potential changes in the risk factors. We then obtain a distribution of the portfolio value at the chosen time horizon, corresponding to each market trajectory. We thus deduce the VaR by taking the α -quantile of this distribution.

The main advantage of MC method is that it can be applied to a wide spectrum of cases, compared to other methods that fail the task due the defined assumptions. It allows a high degree of freedom in the choice of the risk factors' diffusion and it presents better accuracy. However, its implementation can become complex and very expensive in time and memory due to the reevaluation of the portfolio on a substantial number of scenarios.

1.1.2 Counterparty Risk Measures



Counterparty risk is mainly present on the OTC market. It is most often a risk of default between two parties bound by a derivative contract. Since the introduction of the Basel Committee's requirements for calculating credit risk exposures, many banks have had to reform their pricing infrastructure.

Calculating risk indicators such as expected exposure (EE) or potential future exposure (PFE) is essential for assessing the security of a bank's positions. However, from a pricing perspective, these calculations require considerable computational effort. The EE calculation requires hypothetical scenarios using calibrated financial market models. These scenarios represent potential movements in risk factors in the future. Consequently, to measure the bank's exposure, portfolios have to be evaluated for many future scenarios, which is a very intensive task, especially for large portfolios involving thousands of transactions that are subject to different risk factors. (Basel Committee on Banking Supervision, 2023)

1.1.2.1 Expected Exposure

In the event of a counterparty default, the bank determines the values of the transactions in question. If this value is

- negative, the bank is indebted to the defaulting counterparty and is still required to pay the amount,
- positive, the defaulting counterparty is unable to meet its commitments, and the bank will have a claim on the assets of the defaulting party. It can hope to recover a certain fraction of its debt.

The exposure corresponds to the value of all the contracts of the bank that it has in common with a counterparty, which it will lose if the latter defaults before maturity. However, in the event

of default the bank may recover a collateral, which refers to all assets, securities or liquidity, guaranteed by the counterparty to the bank in the event of default. We then have

$$E(t) = \max(V(t) - C(t), 0), \quad (1.1.6)$$

where $E(t)$ is the value of the bank's exposure at time t , $V(t)$ denotes the value of the bank's portfolio with the counterparty at time t and $C(t)$ is the collateral value at time t . In the remainder, we assume that the collateral is equal to 0.

The exposure to a derivatives counterparty can vary significantly over the life of a transaction, with the nature of the derivative product being an important factor in understanding how much exposure is potentially at risk should the counterparty default. Therefore, it is those exposures that hold a positive mark-to-market (MtM) that are relevant. It can be thought of as a probability weighted average of all possible derivative valuations at a point in time, where negative valuations are treated as zero (as there is no obligation due from the counterparty for the scenarios where the derivative has a negative MtM to the bank)(Allan Clark, 2019).

Computing the exposure is an essential part of measuring and capitalizing counterparty risk. The approach we adopt consists of valuing the portfolio to determine its average exposure in the event of default. The first step is to model the risk factors (interest rates, exchange rates, implied volatilities, etc.). These risk factors are used to model the financial products, which are then valued (by simulation, closed formulas, etc.). At this point we obtain the MtM of the portfolio taken into account when calculating the exposure.

In practice, the exposures are often subject to a **netting** agreement enabling the various portfolio transactions to be aggregated. For instance, in the case of several transactions with the same counterparty, position netting makes it possible to aggregate all the positions between the two parties before calculating the exposure.

In case an institution is allowed to use the IMM to calculate the exposure value, it shall measure the exposure value of those transactions at the level of the netting set European Banking Authority (European Banking Authority).

If we note V_i the value of position i in a portfolio consisting of N financial instruments, the exposure of the bank would be, without netting,

$$E(t) = \sum_i \max(0, V_i(t)),$$

and with netting,

$$E(t) = \max\left(0, \sum_i V_i(t)\right).$$

At this point, we can calculate the expected exposure EE as follows

$$EE(t) = \mathbb{E}[E(t)D(t)|\tau = t], \quad (1.1.7)$$

where $D(t)$ is discounting factor at time t , $D(t) = \frac{B(0,T)}{B(t,T)}$ and τ is the default time of the counterparty.

1.1.2.2 Credit Valuation Adjustment

CVA (Credit Valuation Adjustment) is an adjustment to the fair value (or price) of derivative instruments to account for counterparty credit risk (CCR) (on Banking Supervision, 2015).

In other words, the CVA is an estimate of the loss related to the default of the counterparty before the maturity of the common contracts. It is necessary for the bank to quantify this risk and include it in the value of the contract.

The CVA is characterized by

- R the recovery rate, it corresponds to a fraction of the exposure that the bank receives in case of default of the counterparty,
- $\mathbb{P}(\tau \leq t)$ the probability that the counterparty will default before time t ,
- τ the default date of the counterparty,
- T the maximum maturity of the common contracts with the counterparty.

We can then determine the expression of the CVA, which corresponds to the expectation of the losses of the bank due to the default of the counterparty

$$\text{CVA} = (1 - R) \int_0^T \text{EE}(t) d\mathbb{P}(\tau \leq t). \quad (1.1.8)$$

The presence of the term $(1 - R)$ means that the bank is going to recover a fraction of the exposure in case of default.

1.1.2.2.1 Loss Given Default Let $LGD = 1 - R$ denote the Loss Given Default. When a counterparty defaults, the loss to creditors is not necessarily the full exposure as some portion might be recovered from the defaulting counterparty. The size of the recovered exposure relative to the total exposure is referred to as the recovery rate. The balance is the amount expected to be lost in case of default, and this is referred to as the Loss Given Default (Allan Clark, 2019).

1.1.2.2.2 Calculating Probability of Default The probability of default represents the likelihood of a counterparty being unable to meet its obligations. One must consider, however, that the probability of default can be different depending on the time horizon being analysed. This is important because projected counterparty exposure might change over different time periods, and thus assigning the correct probability of default to the correct exposure is necessary. In particular, a term structure of default probabilities is required in order to calculate CVA.

One approach to calculating default probabilities is using CDS. Note that a CDS is a type of credit derivative that provides the buyer with protection against default and other risks. The CDS market provides a market price of counterparty risk that can be observed for different contract tenors. Typically, there are traded CDS for large bank counterparties available in various tenors from 6M to 10Y, allowing a term structure of default probabilities to be implied (Allan Clark, 2019).

In this work, we choose to assume that the probability of the survival of the counterparty follows an exponential law. Thus, with

$$\mathbb{P}(\tau \geq t) = e^{-\lambda t},$$

we deduce the expression of the probability of default

$$\mathbb{P}(\tau \leq t) = 1 - \mathbb{P}(\tau \geq t) = 1 - e^{-\lambda t}, \quad (1.1.9)$$

and

$$d\mathbb{P}(\tau \leq t) = \lambda e^{-\lambda t} dt. \quad (1.1.10)$$

1.1.2.3 Monte-Carlo estimation of EE and CVA

In what follows, we are going to explain how to determine the EE and to compute the CVA by Monte-Carlo approach:

1. Generate M risk factors scenarios on n points of the time grid $0 = t_0, \dots, t_n = T$.
2. For each scenario $j = 1 \dots M$ and for each date t_k , calculate the value of the portfolio $V^{(j)}(t_k)$.
3. Compute the exposure of the bank for the scenario j at time t_k

$$E^{(j)}(t_k) = \max(V^{(j)}(t_k), 0).$$

4. Compute the expected exposure of the bank at each time t_k

$$EE(t_k) = \frac{1}{M} \sum_{j=1}^M D^{(j)} E^{(j)}(t_k).$$

5. Compute the CVA (at date 0)

$$CVA_0 = (1 - R) \sum_{k=1}^n EE(t_k) [P(t_k) - P(t_{k-1})].$$

1.2 Evaluation of Financial Instruments

In this section, we focus on the evaluation of the financial products that constitute our portfolio. The values of our portfolio are obtained by conventional methods taking into account different scenarios of the risk factors. In other words, for each combination of risk factors, we need to compute the theoretical value of our portfolio using pricers. We consider interest rate swaps and European swaptions to conduct the study. In what follows, we will detail the pricers used for each of these products.

Considering a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ filtered by the natural filtration $(\mathcal{F}_t)_{t \geq 0}$, \mathbb{Q} denotes the risk-neutral probability measure and \mathbb{Q}^T the T -forward measure (see Appendix B for detail).

1.2.1 Interest Rate Swaps

An interest rate swap is a contract that exchanges payments between two differently indexed legs. We note T_0, \dots, T_{n-1} the set of fixing dates. At every instant T_i in a prespecified set of payments dates T_1, \dots, T_n the fixed leg pays out the amount

$$N \tau_i K,$$

corresponding to a fixed interest rate K , a nominal N and a year fraction τ_i between T_{i-1} and T_i , whereas the floating leg pays the amount

$$N \tau_i L(T_{i-1}, T_i),$$

corresponding to the floating interest rate $L(T_{i-1}, T_i)$ (e.g. Libor rate) resetting at the previous instant T_{i-1} for the maturity given by the current payment instant T_i .

- **Payer interest rate swap:** when the fixed leg is paid and the floating leg is received.

The discounted payoff is then expressed as

$$\sum_{i=1}^n NB(t, T_i) \tau_i (L(T_{i-1}, T_i) - K).$$

- **Receiver interest rate swap:** when the floating leg is paid and the fixed leg is received. The discounted payoff is then expressed as

$$\sum_{i=1}^n B(t, T_i) N \tau_i (K - L(T_{i-1}, T_i)).$$

We define the swap rate as the rate for which the swap valuation is zero at the initial time. We then have the following expression for the swap rate

$$S_t = \frac{\sum_{i=1}^n \tau_i B(t, T_i) L(T_{i-1}, T_i)}{\sum_{i=1}^n \tau_i B(t, T_i)}. \quad (1.2.1)$$

We derive the final expression for the value of the swap

$$P_{swap}(t) = NA(t)(S_t - K), \quad (1.2.2)$$

with $A(t) = \sum_{i=1}^n \tau_i B(t, T_i)$ is the swap annuity.

We can also view the swap contract as a portfolio of Forward Rate Agreements (FRAs) and obtain the following expression for the price of a receiver forward-start swap (Brigo and Mercurio, 2001, Page 13-16)

$$-NB(t, T_0) + NB(t, T_n) + N \sum_{i=1}^n \tau_i K B(t, T_i), \quad (1.2.3)$$

and the forward swap rate at time t is defined as the rate in the fixed leg of the above interest rate swap that makes the swap a fair contract at the present time (Brigo and Mercurio, 2001, Page 13-16)

$$S(t, T_0, T_n) = \frac{B(t, T_0) - B(t, T_n)}{\sum_{i=1}^n \tau_i B(t, T_i)}, \quad (1.2.4)$$

where $B(t, T_i)$ is the zero-coupon bond of maturity T_i , and τ_i is the year fraction between T_{i-1} and T_i .

Remark 1.2.1. The swap price is fully determined by the ZC bond prices. Thus, the main drivers of the swap price are the ZC yield curves.

1.2.2 European Swaptions

An European swaption of maturity T_e and tenor T_n , is an option giving its owner the right but not the obligation to enter, at time T_e , in an interest swap rate contract of tenor T_n . In other words, the underlying instrument of an interest rate swaption is an interest rate swap contract.

Let T_0 denote the date of inception of the swap contract, which is usually equal to $T_e + 2$ days. The fixed swap rate defined at date T_e is

$$S(T_e, T_0, T_n) = \frac{B(T_e, T_0) - B(T_e, T_n)}{\sum_{i=1}^n \tau_i B(T_e, T_i)}. \quad (1.2.5)$$

A payer swaption of strike K written on the above swap is an option maturing in T_e with payoff

$$\pi_{swaption}(K, T_e, T_0, T_n) = \left(B(T_e, T_0) - B(T_e, T_n) - \sum_{i=1}^n \tau_i K B(T_e, T_i) \right)^+. \quad (1.2.6)$$

Considering that $A(t) = \sum_{i=1}^n \tau_i B(t, T_i)$ is the annuity of the swaption of nominal N , the price of this latter can be computed as the expectation of the payoff $\pi_{swaption}$ under the T_e -forward measure \mathbb{Q}^{T_e} as follows

$$\begin{aligned} P_{swaption}(t, T_e, T_0, T_n) &= N \times A(t) \mathbb{E}^{\mathbb{Q}^{T_e}} \left[\left(\frac{B(t, T_0) - B(t, T_n)}{\sum_{i=1}^n \tau_i B(t, T_i)} - K \right)^+ \right] \\ &= N \times A(t) \mathbb{E}^{\mathbb{Q}^{T_e}} [(S(t, T_0, T_n) - K)^+]. \end{aligned} \quad (1.2.7)$$

The advantage of this change of numeraire is that it allows us to get a simpler expression for the swaption's price that is easier to calculate. The chosen numeraire is the annuity $A(t)$ which corresponds to the sum of zero coupon bonds. Note that the discounted value of this term is an \mathcal{F}_t martingale under the risk neutral measure. For details of the change of numeraire in order to obtain this expression of the price, see appendix B.

Thus, the price of a payer swaption can be viewed as a put option with underlying the swap rate $S(t, T_0, T_n)$ and can be obtained using the Black and Scholes formula as follows

$$P_{swaption}(t, T_e, T_0, T_n) = N \times A(t) (S(t, T_0, T_n) \Phi(d_1) - K \Phi(d_2)), \quad (1.2.8)$$

where

$$\begin{aligned} d_1 &= \frac{\ln\left(\frac{S(t, T_0, T_n)}{K}\right) + \frac{\sigma^2}{2}(T_e - t)}{\sigma^{imp} \sqrt{T_e - t}}, \\ d_2 &= d_1 - \sigma^{imp} \sqrt{T_e - t}, \end{aligned}$$

with σ^{imp} is the swaption's implied volatility and Φ is the cumulative distribution function of the standard normal variable

Remark 1.2.2. All the entities that constitute the formula above are deterministic except the implied volatility σ^{imp} . Along with the ZC bond prices, the implied volatilities are considered as risk factors for swaptions. Therefore, one must consider the diffusion of both ZC yield curves and implied volatility surfaces in order to price European swaptions, unlike the swap price value which depends only on ZC yield curves.

1.3 Diffusion Models of Risk Factors

Risk position refers to the extent to which the current value of an instrument is vulnerable to losses caused by fluctuations in a risk factor. For instance, if a bond is denominated in a currency different from the reporting currency of a bank, it carries risk positions related to general interest rate risk, credit spread risk, and foreign exchange (FX) risk. In this context, risk positions represent the potential losses in the instrument's current value that may arise from changes in the underlying risk factors such as interest rates, credit spreads, or exchange rates (Basel Committee on Banking Supervision, 2020).

The risk factors are the variables governing the evolution of the value of our portfolio. These are the ones that will feed our pricing functions. The choice of the number of risk factors is a problem on its own. One of our main objectives is to try to get closer to what it is done in practice by considering more risk factors modelling the market. Therefore, considering that we will study interest rate products, namely swaps and swaptions, we consider the following risk factors: interest rate curves and implied volatilities surface.

1.3.1 LGM Zero Coupon Yield Curve Model

The diffusion of Zero Coupon yield curves is carried out using an LGM-1F model (the One Factor Linear Gauss Markov). This model comes within the framework of Heath-Jarrow-Morton which assumes that under the risk neutral probability \mathbb{Q} , the dynamics of the zero-coupon bond of maturity T is given by

$$\frac{dB(t, T)}{B(t, T)} = r_t dt + \Gamma(t, T) dW_t^{\mathbb{Q}}. \quad (1.3.1)$$

The instantaneous forward rate of maturity T is defined by

$$f(t, T) = \partial_T \ln(B(t, T)),$$

and gives the corresponding dynamics of $f(t, T)$ under \mathbb{Q}

$$df(t, T) = -\gamma(t, T)\Gamma(t, T)dt + \gamma(t, T)dW_t^{\mathbb{Q}},$$

where $\gamma(t, T) = \partial_T \Gamma(t, T)$.

The LGM-1F verifies the following properties

- Gaussian : the volatilities $\gamma(t, T)$ and $\Gamma(t, T)$ are deterministic.
- Linear : the volatility of instantaneous forward rates takes the form $\gamma(t, T) = \sigma(t) \exp \left\{ -\int_t^T \lambda(s) ds \right\}$ where σ and λ are deterministic functions.
- Markovian : the dynamics of the model for date $s > t$ can be entirely deduced from its observation at date t .

In our work, we make the following assumptions :

- $\lambda(\cdot)$, the mean reversion parameter, is independant of time $\lambda(s) = \lambda$.
- $\sigma(\cdot)$ is piecewise constant.

Consequently, we can write

$$\gamma(t, T) = \sigma(t)e^{-\lambda(T-t)}, \quad \Gamma(t, T) = \frac{\sigma(t)}{\lambda}(e^{-\lambda(T-t)} - 1).$$

Under these assumptions, the dynamics of the short rate process is

$$dr_t = \left[\lambda(f(0, t) - r_t) + \partial_T f(0, t) + \int_0^t \sigma^2(s)e^{-2\lambda(t-s)} ds \right] dt + \sigma(t)dW_t^{\mathbb{Q}}.$$

The above equation can be written as follows :

$$dX_t = [\phi(t) - \lambda X_t]dt + \sigma(t)dW_t, \quad (1.3.2)$$

where $X_t = r_t - f(0, t)$, and $\phi(t) = \int_0^t \sigma^2(s)e^{-2\lambda(t-s)} ds$.

We can then determine the expression of the zero coupon bond $B(t, T)$ in terms of X_t (see Appendix A for the demonstration),

$$B(t, T) = \frac{B(0, T)}{B(0, t)} \exp \left(-\frac{1}{2} \beta^2(t, T) \phi(t) - \beta(t, T) X_t \right), \quad (1.3.3)$$

with $\beta(t, T) = \frac{1-e^{-\lambda(T-t)}}{\lambda} = \int_t^T e^{-\lambda(u-t)} du$.

The rationale behind our selection to this model is the fact that it only has one stochastic driver (one factor) X_t which is a gaussian process. Also, the LGM model is mathematically equivalent to the Hull-White model but offers significant improvements in calibration stability and accuracy. It is also more accurate.

1.3.1.1 Calibration Using Market Data

In this part, we present a method for the calibration of the piecewise constant function $\sigma(\cdot)$ defined in the previous LGM-1F framework. The mean reversion parameter is assumed to be constant and equal to 0.01. The volatility σ is calibrated using the market prices of a diagonal european swaptions.

Consider a set of n calibration swaptions expiring at dates $T_1^e < T_2^e < \dots < T_n^e$. We note by j the index of the swaptions $1 \leq j \leq n$. We set the strikes K of the diagonal swaptions equal to the forward swap rate. We aim at finding a piecewise constant function σ such that the LGM-1F model reconstructs the market prices of the predefined swaptions. Note that the price of a european swaption of expiry T_j^e depends on σ only in the interval $[0, T_j^e]$ (Chaix, 2003):

$$\forall j \in \{1 \dots n\}, \forall t \in [T_{j-1}^e, T_j^e], \sigma(t) = \sigma_j.$$

In the LGM-1F framework, the price of a european swaption with constant nominal N , is given by a close formula. The payoff of a payer swaption is given by the equation (1.2.2). Thus, the price of this payer swaption writes as

$$\begin{aligned} PS(T_j^e, T_{n(j)}, T_n) &= B(t, T_j^e) \mathbb{E}^{Q^{T_j^e}} \left[\left(B(T_j^e, T_{n(j)}) - B(T_j^e, T_n) - \sum_{i=1}^n \delta_i K B(T_j^e, T_i) \right)^+ \right] \\ &= B(t, T_j^e) \mathbb{E}^{Q^{T_{n(j)}}} \left[\left(1 - \frac{B(T_j^e, T_n)}{B(T_j^e, T_{n(j)})} - K \sum_{i=1}^n \delta_i \frac{B(T_j^e, T_i)}{B(T_j^e, T_{n(j)})} \right)^+ \right] \\ &= B(t, T_j^e) \mathbb{E}^{Q^{T_{n(j)}}} \left[\left(1 - \sum_{i=1}^n c_i \frac{B(T_j^e, T_i)}{B(T_j^e, T_{n(j)})} \right)^+ \right] \text{ with } c_i = K\delta_i + 1 \end{aligned} \quad (1.3.4)$$

where $T_{n(j)}$ is the fixing date which comes after the exercise date T_j^e ($T_j^e \leq T_{n(j)} \leq T_n$). If T_j^e is a fixing date, then $T_j^e = T_{n(j)}$ (Chaix, 2003).

Our swaption can be reviewed as a modified put bond option. Details on pricing bond options and modified bond options can be found in Appendix C. We consider the notation from Appendix C and take into account the following parameters

$$\sum_{i=1}^n \frac{B_{T_j^e, T_i}(x_0)}{B_{T_j^e, T_0}(x_0)} = K \text{ and } K_i = \frac{B_{T_j^e, T_i}(x_0)}{B_{T_j^e, T_0}(x_0)}. \quad (1.3.5)$$

This results in

$$\begin{aligned} PS(T_j^e, T_{n(j)}, T_n, K) &= B(t, T_{n(j)}) \sum_{i=n(j)}^n c_i \mathbb{E}_t^{Q^{T_{n(j)}}} \left[\left(K_i - \frac{B(T_j^e, T_i)}{B(T_j^e, T_0)} \right)^+ \right] \\ &= \sum_{i=n(j)}^n c_i BSPut \left(\frac{B(t, T_i)}{B(t, T_{n(j)})}, t, \sigma_{ji}^{bs}, K_i, T_j^e \right), \end{aligned} \quad (1.3.6)$$

where σ_j^{bs} verify

$$\begin{aligned} \left(\sigma_{ji}^{bs}\right)^2 T_j^e &= \int_0^{T_j^e} |\Gamma(s, T_i) - \Gamma(s, T_n)|^2 ds \\ &= \left(\frac{\exp\left(-\lambda(T_n - T_j^e)\right) - \exp\left(-\lambda(T_i - T_j^e)\right)}{\lambda} \right)^2 \phi(T_j^e). \end{aligned} \quad (1.3.7)$$

We can see that the price of the payer swaption can be written as a linear combination of Black-Scholes put option prices. The dependency of these prices on σ_{ji}^{bs} corresponds to a dependency on $\phi(T_e)$.

The calibration algorithm can now be formulated. Given the value of $\phi(T_{j-1}^e)$, we seek to determine $\phi(T_j^e)$ and σ_j

1. Calculate the characteristic elements of the exercise date T_j^e , in order to find the value of the antecedent x_0 (cf. (1.3.5)) using a Newton-Raphson-type algorithm.
2. We calculate the swaptions' prices of maturity T_j^e according to the LGM form formula and solve a minimisation problem of the difference between LGM prices and market prices using a second Newton-Raphson-type algorithm in order to determine $\phi(T_j^e)$.
3. Finally, we find the value of σ_j , using the definition of $\phi(T_j^e)$:

$$\begin{aligned} \phi(T_j^e) &= \int_0^{T_j^e} \sigma^2(s) e^{-2\lambda(T_j^e - s)} ds \\ \phi(T_j^e) &= \int_{T_{j-1}^e}^{T_j^e} \sigma_j^2 e^{-2\lambda(T_j^e - s)} ds + \int_0^{T_{j-1}^e} \sigma^2(s) e^{-2\lambda(T_j^e - s)} ds \\ \phi(T_j^e) &= \sigma_j^2 \frac{1 - e^{2\lambda(T_{j-1}^e - T_j^e)}}{2\lambda} + e^{-2\lambda(T_j^e - T_{j-1}^e)} \int_0^{T_{j-1}^e} \sigma^2(s) e^{-2\lambda(T_{j-1}^e - s)} ds \\ \phi(T_j^e) &= \sigma_j^2 \frac{1 - e^{2\lambda(T_{j-1}^e - T_j^e)}}{2\lambda} + e^{-2\lambda(T_j^e - T_{j-1}^e)} \phi(T_{j-1}^e). \end{aligned}$$

This results in

$$\sigma_j = \sqrt{\frac{2\lambda \left(\phi(T_j^e) - e^{-2\lambda(T_j^e - T_{j-1}^e)} \phi(T_{j-1}^e) \right)}{1 - e^{2\lambda(T_{j-1}^e - T_j^e)}}}. \quad (1.3.8)$$

We initialize these iterations using the fact that for the first exercise date T_1^e and the calculation of $\phi(T_1^e)$, we have $\phi(T_0^e) = \phi(0) = 0$.

Calibration results

Table 1.1 shows the prices obtained by applying the LGM-1F pricing formula with the calibrated σ_j , of 10 diagonal swaptions used for calibration, and the market prices of the same swaptions. Market prices were extracted from Natixis's data.

LGM-1F Prices	BS Prices	Absolute Error
24.42	24.41	0.01
28.79	28.80	0.01
32.34	32.33	0.01
38.74	38.74	0.00
42.25	42.24	0.01
42.58	42.55	0.03
50.43	50.43	0.00
54.42	54.42	0.00
59.15	59.13	0.02
63.46	63.45	0.01

Table 1.1: Calibration error of the LGM 1F model.

1.3.1.2 Reconstruction of ZC Yield Curves

Recalling the expression of the ZC bonds using the LGM-1F diffusion

$$B(t, T) = \frac{B(0, T)}{B(0, t)} \exp \left(-\frac{1}{2} \beta^2(t, T) \phi(t) - \beta(t, T) X_t \right).$$

The input for the diffusion of ZC curves is only the initial curve displayed in Figure 1.1. It corresponds to different ZC rates for different maturities.

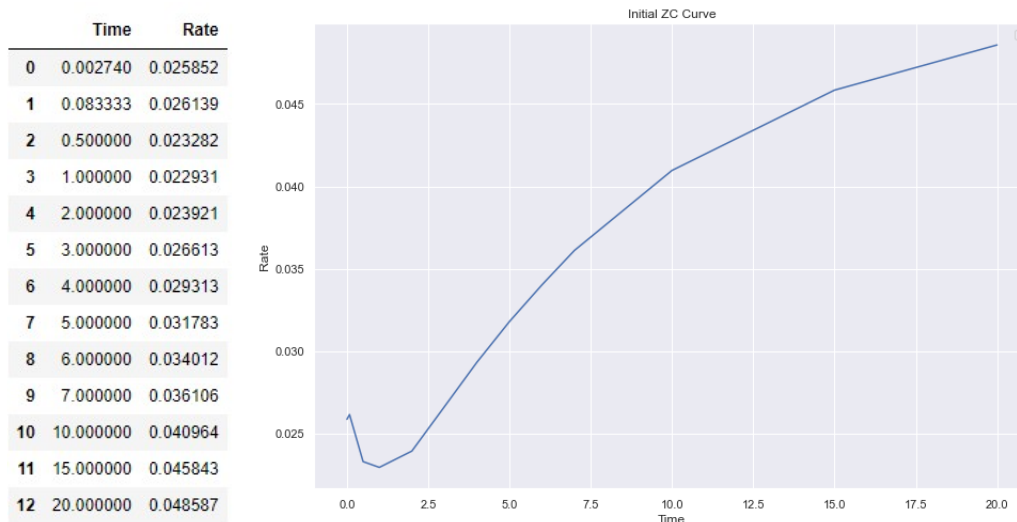


Figure 1.1: Initial Zero Coupon yield curve : The table (*left*) shows the values of ZC rates for different maturities. These values are plotted (*right*) to form an initial yield curve. (*source*: Natixis).

We furnish the ZC rate values of Figure 1.1 to serve as real market data, allowing us to align our diffusion results closely with the actual market movements of ZC interest rates. This ensures

that our simulations closely reflect the real-world evolution of ZC interest rates.

We only need the diffusion of X_t , what amounts to diffusion of the short rate r_t , in order to get the ZC bonds. Then, we can compute the ZC rates of the yield curve for each tenor T , and thus shock the initial curve.

1.3.2 SABR Volatility Model

The implied volatility is defined as the parameter which, substituted in the Black-Scholes valuation formula, leads to the price of the option observed on the market.

Indeed, the Black and Scholes model assumes that the volatility of an asset is constant over the trading period and returns a flat volatility surface. However, if we use the fact that implied volatility can be obtained by matching the observed option prices from the market with the theoretical value under BS model, we get that options with different strikes require different volatilities to match their market prices which suggests that volatility is not a constant and it depends on the strike, and it forms the so called volatility smile.

Therefore, the BS assumption fails to capture the asset volatility dynamics which is important in the cases of pricing complex derivatives. It is a matter of fact that local volatility models capture the volatility smile but not the price dynamics. Therefore we opt for stochastic volatility models, namely SABR which provides a close forms solution for implied volatility that captures correctly the smile and price dynamics.

The stochastic differential equations (SDE) for the SABR volatility model

$$\begin{aligned} dF_t &= \alpha_t (F_t)^\beta dW_t, \\ d\alpha_t &= \nu \alpha_t dZ_t, \quad \alpha_0 = \sigma_{atm} F_0^{1-\beta} \\ d\langle W_t, Z_t \rangle &= \rho dt, \end{aligned} \tag{1.3.9}$$

where F_t is the forward rate, α_t is its volatility, W_t and Z_t are two correlated standard brownian motions of correlation ρ , ν is the volatility of the volatility process (known as vol-of-vol), β in $[0, 1]$ determines how the at-the-money volatility changes when the forward price changes and σ_{atm} is the At-The-Money volatility.

We were given four matrices for the calibrated values of the SABR parameters that serve as inputs to our diffusion. These matrices are illustrated in Figure 1.2

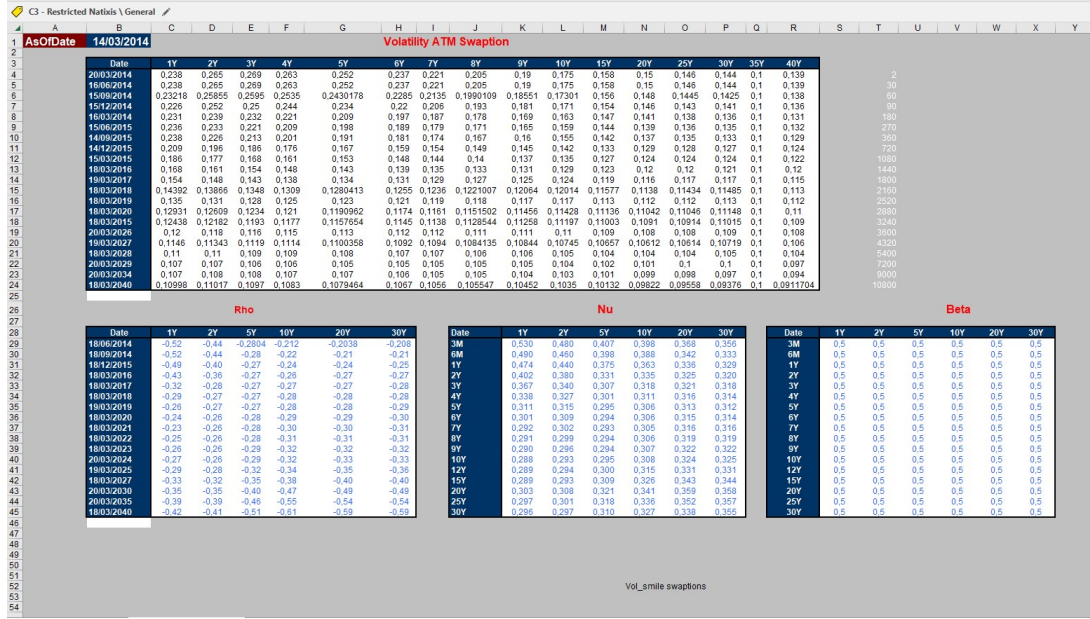


Figure 1.2: SABR calibrated parameters: The matrices correspond to the parameters $(\sigma_{atm}, \beta, \rho, \nu)$ at different maturities and tenors (*source*: Natixis).

The main advantage of the SABR model is the closed form solution that allows to get an analytical expression from which we can obtain the implied volatility and then plug it into the Black-Scholes pricing formula.

In order to simplify the formula of the SABR implied volatility, we denote by $F_0 = F$ and $\alpha_0 = \alpha$ the initial values of the forward rate and its volatility. We also have separated its expression to the following terms

$$C = (FK)^{(1-\beta)/2},$$

$$A = 1 + \left(\frac{(1-\beta)^2 \alpha^2}{24C^2} + \frac{\alpha\beta\nu\rho}{4C} + \frac{\nu^2(2-3\rho^2)}{24} \right) T,$$

$$B = 1 + \frac{((1-\beta) \ln(F/K))^2}{24} + \frac{((1-\beta) \ln(F/K))^4}{1920},$$

$$z = \frac{\nu}{\alpha} C \ln\left(\frac{F}{K}\right), \quad x = \ln\left(\frac{\sqrt{1-2\rho z + z^2} + z - \rho}{1-\rho}\right).$$

There is only left to compute the implied volatility corresponding to the following 2 cases

$$\sigma = \begin{cases} \frac{\alpha}{C} A & \text{if } F = K \\ \frac{\nu \ln(F/K) A}{xB} & \text{otherwise.} \end{cases} \quad (1.3.10)$$

After extracting the calibrated parameters (1.2), we can insert them in the SABR model function in order to get the SABR implied volatilities. We plot the entire initial implied volatility surface for all tenors and maturities in Figure 1.3.

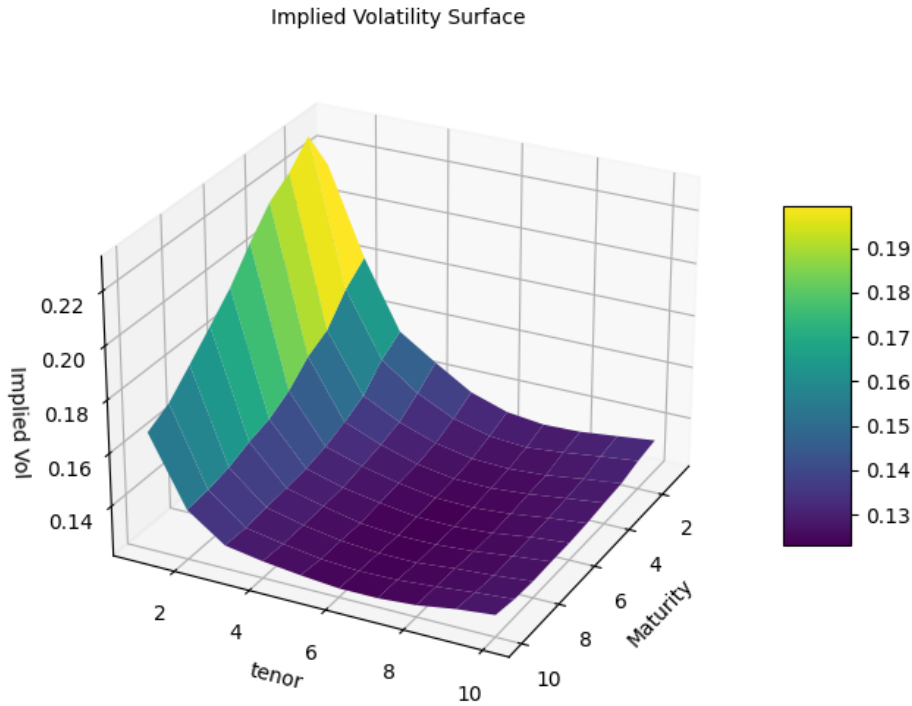


Figure 1.3: Initial implied volatility surface: three-dimensional surface plotted whereby the implied volatility (z -axis) for swaptions is plotted against the maturity (y -axis) and tenor (x -axis) in years

We observe in Figure 1.3 that the implied volatility surface simultaneously shows both volatility smile and term structure of volatility.

Implied volatilities are very important because they are embedded in option prices, and option prices reflect future market expectations. This implies that implied volatility is a forward estimate of the option volatility.

We diffuse the implied volatility surfaces using the SABR dynamics. In fact, we take the SABR calibrated parameters on different maturities and tenors. We diffuse the forward rate and its instantaneous volatility using the dynamics above, and then calculate the implied volatility with the same closed formula while maintaining the values of (β, ρ, ν) constant. We notice that the diffusion of F_t and α_t is equivalent to shocking the *ATM* volatilities and subsequently, shocking the implied volatility surfaces.

Remark 1.3.1. It is worth noting that unlike spot rates, ZC rates and volatility is not always diffused stochastically. It is either set as a constant throughout the simulation, or a non-stochastic term structure is used or it is diffused fully stochastically.

- If the volatility is set as a constant, it is not included in the input variables domain of the pricing function. It is not considered as a risk factor since it has no effect on the price of the portfolio.
- If a constant term structure volatility model is used, the volatility changes from time point to time point, but it is constant at each time point. Here, the volatility is excluded of the

domain of risk factors of the pricing function.

- If a local volatility model is used, the implied volatility changes from scenario to another within the same time point. However such volatility depends fully on the value of the spot and it is thus discarded from the domain of risk factors of the pricing function.
- The only case when the volatility needs to be considered as a risk factor for the pricing function is when it is diffused stochastically throughout the MC simulation. This is the case which we are interested in.

1.4 Risk Calculation Flow in Traditional Approach

We illustrate through Figure 1.4 a typical flow of risk calculation. We determine all risk factors that affect the considered trading portfolio. Then, a large number of scenarios are generated (step 1), for instance we take 10,000 scenarios. The pricing function of each portfolio trade is called on each scenario (step 2) to obtain the distribution of the portfolio values (step 3). From this distribution, all kinds of risk measures can be computed (step 4).

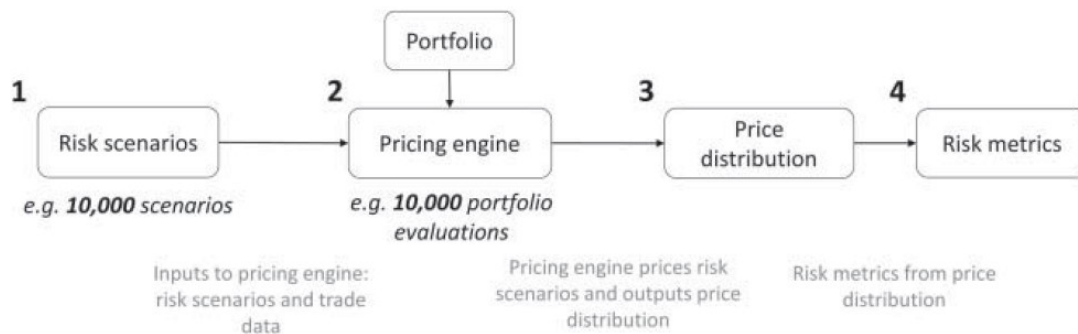


Figure 1.4: Risk calculation flow by MC approach (*source: Ruiz and Zeron (2021)*)

The repeated revaluation of a large portfolio under different scenarios of risk factors and for different time horizons in the vast majority of risk calculations, presents a bottleneck because it can turn out computationally expensive and time-consuming. For this, financial institutions must be aware of the computing time optimisation aspects and they have to find alternatives to improve the trade-off between speed and accuracy. The next chapter shows that this is possible by using Chebyshev interpolation.

Chapter 2

Application of Chebyshev Interpolation in Risk Calculation

The banking crisis in 2008 brought up the emergency to make profound changes in the derivatives industry. For instance, computing the risks became primary and the banks were racing to find accurate and time/cost-effective approaches to calculate risk measures. Despite the achievements that has been made through time, the computational needs did not cease to increase due to market demand and the regulations which introduced higher standards for risk management.

The pricing problem figuring in the classical approach can be mitigated with approximation methods. In practice, we aim at adding an interpolation proxy in the risk calculation engine to surpass this computational bottleneck.

All risk factors are generated the same way as in the previous approach. The objective is to build proxies for the pricing functions and use them to evaluate all risk scenarios. Therefore, in step (2a) of Figure 2.1, a set of grid points in the space of the risk factors scenarios is determined. These are the points to which the pricing functions will be called to build the proxy function in step (2b). If the proxy is built appropriately, the level of accuracy will be such that the price distribution obtained in step (2c) will be very close to the one of the previous method.

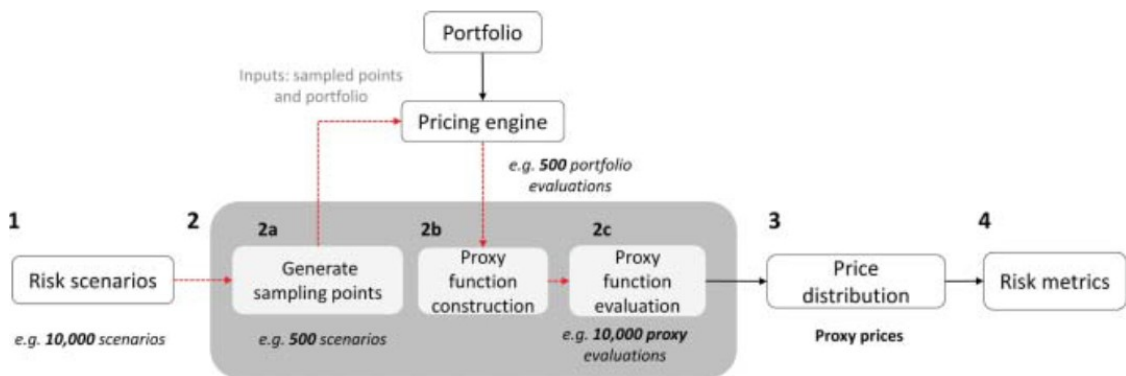


Figure 2.1: Alternative risk flow (*source: (Ruiz and Zeron, 2021)*)

In this chapter, we propose an interpolation method with the aim of reducing the computational effort to calculate risk measures, while maintaining the required accuracy.

For all the theorems, definitions, demonstrations and all further details we refer to (Ruiz and Zeron, 2021).

2.1 Review of Polynomial Interpolation Methods

A polynomial interpolation consists in finding a polynomial that fits a discrete set of known data points, enabling the construction of new data points within the range of the data. Formally, a polynomial $p(x)$ interpolate the data $(x_0, \nu_0), \dots, (x_n, \nu_n)$ if $p(x_i) = \nu_i$ for all i in $0, \dots, n$.

Polynomial functions are used in various aspects in mathematics. One of their major uses has been to approximate arbitrary continuous functions.

Given a function f and a set of points x_0, \dots, x_n on its domain, consider the tensor T defined by x_0, \dots, x_n and the values $f(x_0), \dots, f(x_n)$. This tensor has a unique polynomial interpolant of degree at most n that can potentially be used as a proxy for f .

Note that as the number of grid points x_0, \dots, x_n increases, a sequence of polynomial interpolants p_n is defined. We are interested in determining whether the sequence p_n converges to f , and at which rate. For this purpose, we consider looking at the impact of the geometry of the grid points x_0, \dots, x_n as well as the interpolation method.

Several theories converge on the idea that any continuous function can be approximated, as closely as needed, by a polynomial. For instance, we cite **Weierstass's Approximation Theorem**.

Theorem 2.1.1. Let f be a continuous function on $[a, b]$ and let ϵ be an arbitrary real value greater than zero. There is a polynomial p such that, for any $x \in [a, b]$,

$$|f(x) - p(x)| < \epsilon, \forall \epsilon > 0.$$

In what follows, we introduce two polynomial interpolation methods that can serve to approximate the function in question. These methods, consider equidistant interpolation grid points. We will prove later on the unstability of these techniques and their unsuitability to meet our objectives.

2.1.1 Vandermonde Matrix Interpolation

If we have $n + 1$ data points we can create the same number of equations by fitting them to a n^{th} degree polynomial

$$p_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n,$$

This corresponds to a polynomial interpolation when $f(x_i) = p_n(x_i)$ for all $(x_i, f(x_i))_{i=1,2,\dots,n+1}$. Substituting these interpolation equations, we get a system of linear equations in the coefficients a_i , which reads in matrix form as follows:

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n+1} & x_{n+1}^2 & \dots & x_{n+1}^n \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f(x_0) \\ \vdots \\ f(x_n) \end{bmatrix}. \quad (2.1.1)$$

An interpolant $p_n(x)$ corresponds to a solution $a = [a_0, \dots, a_n]^T$ of the above matrix equation $Va = f$. The matrix V is a Vandermonde matrix whose determinant is known to be non zero and thus ensuring that the matrix is invertible.

This may be useful in polynomial interpolation, since inverting the Vandermonde matrix allows expressing the coefficients of the polynomial. In other words, given $n + 1$ distinct nodes with coordinates $(x_i, p_n(x_i))$, there exists a unique polynomial $p_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$ of degree n which interpolates them and its coefficients can be computed by solving

$$a = V^{-1}f.$$

2.1.2 Lagrange Interpolation

In numerical analysis, the Lagrange interpolating polynomial is the unique polynomial of lowest degree that interpolates a given set of data.

Given a set of n equidistant data points x_1, \dots, x_n with their corresponding responds (y_1, \dots, y_n) , the Lagrange interpolation polynomial is the following

$$p(x) = \sum_{i=1}^n y_i l_i(x),$$

where $l_i(x)$ are the Lagrange basis polynomials

$$l_i(x) = \prod_{j=1, j \neq i}^n \frac{x - x_j}{x_i - x_j} = \frac{x - x_1}{x_i - x_1} \dots \frac{x - x_{i-1}}{x_i - x_{i-1}} \cdot \frac{x - x_{i+1}}{x_i - x_{i+1}} \dots \frac{x - x_n}{x_i - x_n}$$

$$\text{or simply } l_i(x) = \frac{\prod_{j=1, j \neq i}^n (x - x_j)}{\prod_{j=1, j \neq i}^n (x_i - x_j)}.$$

The most important property of these basis polynomials is

$$l_{j \neq i}(x_i) = 0$$

$$l_i(x_i) = 1$$

So, we assure that $L(x_i) = y_i$, which indeed interpolates the data.

Remark 2.1.1. One would expect that when we increase the number of interpolation equidistant nodes, the interpolation polynomials presented above will converge uniformly to f . However, this is not the case for the following Runge's Phenomenon.

2.1.3 Runge's Phenomenon

Runge's phenomenon is a problem of oscillation of polynomials at the edges of the interval that occurs when using polynomial interpolation with polynomials of high degree over a set of equispaced interpolation points.

The Runge's Phenomenon is a real example of a one-dimensional analytic function for which equidistant interpolation diverges exponentially. Illustrations of this phenomenon are provided in Figure 2.2.

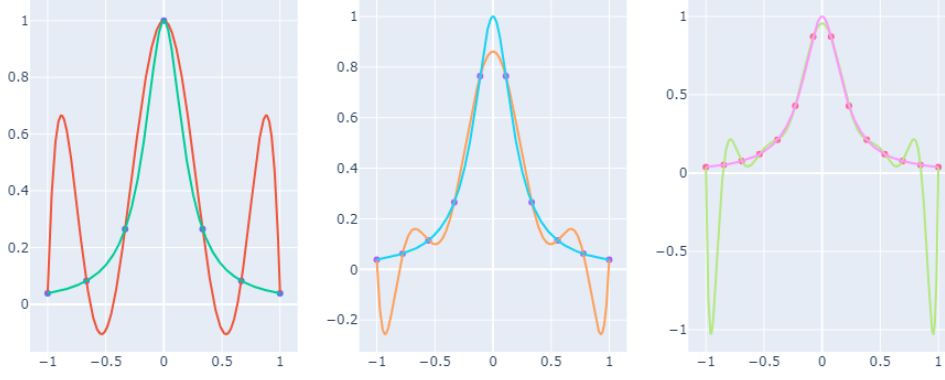


Figure 2.2: Lagrange interpolation of the Runge's function by different number of equidistant points: (Left) 5 points, (Center) 10 points and (Right) 14 points.

Hence, interpolation, especially on equidistant points, had a negative reputation historically mainly due to

- Exponential divergence for some analytic functions: Runge's Phenomenon.
- There is no interpolation scheme that works for all continuous functions.

The theorem affirms the existence of p but does not give an explicit formula of it, nor specifies a proper way to derive p .

It is also worth noting that the choice of the polynomial p to use as a proxy must take into account the accuracy of the proxy and the feasibility and efficiency when evaluating the proxy.

2.2 Chebyshev Interpolation

In this section, we present the Chebyshev interpolation and explain why we opted for this technique to approximate pricing functions.

2.2.1 Chebyshev Nodes and Polynomials

The most natural way to choose the interpolation nodes is to choose them uniformly distributed in the interval $[-1, 1]$. This amounts to considering the methods presented previously, which is not adequate, as we proved, in certain cases. Another reason for not choosing equidistant points is the following inequality: if f is C^n on the interval $[-1, 1]$ and p_n is the interpolation polynomial on x_1, x_2, \dots, x_n , then

$$\forall x \in [-1, 1], |f(x) - p_n(x)| \leq \sup_{x \in [-1, 1]} (x - x_1)(x - x_2) \cdots (x - x_n) \times \frac{\sup_{x \in [-1, 1]} |f^{(n)}|}{n!}$$

is the interpolation error formula.

The idea is then to choose x_i reducing the error of the polynomial interpolation, they are the solutions to the following minimization problem

$$\arg \min_{(x_1, \dots, x_n)} \left(\sup_{x \in [-1, 1]} (x - x_1)(x - x_2) \cdots (x - x_n) \right).$$

This is achieved when the x_i are the roots of the Chebyshev polynomial of degree n , which are defined by

$$x_i = \cos\left(\frac{(2i-1)\pi}{2n}\right), i = 1, \dots, n.$$

By using these points we reduce the numerator of the interpolation error formula

$$(x - x_1) \cdots (x - x_n) = \frac{1}{2^{n-1}} T_n(x),$$

where $T_k(x) = \cos(k \cos^{-1}(x))$ is the k -th Chebyshev polynomial.

$$\begin{aligned} T_0(x) &= 1, \\ T_1(x) &= x, \\ &\dots \\ T_{n+1}(x) &= 2xT_n(x) - T_{n-1}(x). \end{aligned}$$

These points are the result of projecting equidistant points on the upper half of the unitary circle onto the real line as seen in Figure 2.3

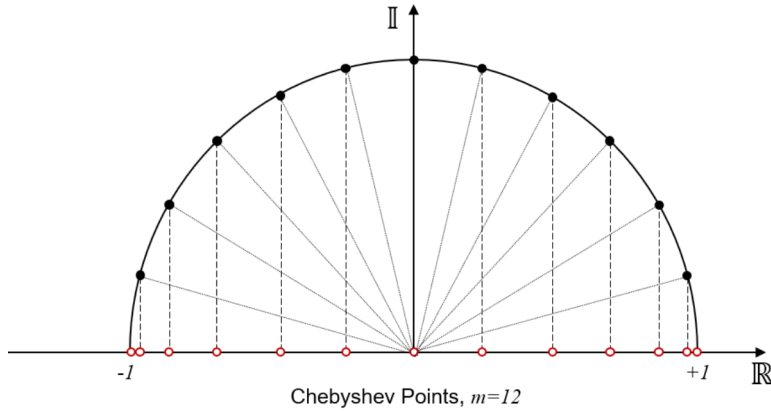


Figure 2.3: Chebyshev points as projection result of projecting equidistant points on the upper half of the unitary circle onto the real line.

This definition can be extended to any interval $[a, b]$ by mapping $[-1, 1]$ to $[a, b]$ with the help of a linear transformation followed by

$$x \mapsto a + 0.5(b - a)(x + 1). \quad (2.2.1)$$

Remark 2.2.1. If the function f enjoys specific smoothness conditions (e.g. Lipschitz continuous), then the Chebyshev series converges uniformly and absolutely (Ruiz and Zeron, 2021).

Theorem 2.2.1. If f is Lipschitz continuous on $[-1, 1]$, it has a unique representation as a Chebyshev series

$$f_n(x) = \sum_{k=0}^n a_k T_k(x),$$

which is absolutely and uniformly convergent. The coefficients a_k are given for $k \leq 1$ by the formula

$$a_k = \frac{2}{\pi} \int_{-1}^1 \frac{f(x)T_k(x)}{\sqrt{1-x^2}}, \quad (2.2.2)$$

if $k = 0$, the above formula is divided by 2.

Here, the above representation offers a very attractive way of approximating Lipschitz continuous functions. However, the coefficients presented in this projection involve solving an integral which usually corresponds to approximating $f(x)$ several times.

The following theorem affirms that for any analytic function f , the convergence of the Chebyshev series is exponential. Note that a function is analytic if for every point x in its domain, the Taylor Series of f at x exists and converges to the value $f(x)$.

Theorem 2.2.2. Let f be an analytic function on the interval $[-1, 1]$. Consider its analytical continuation to the open Bernstein ellipse E_ρ of radius ρ , where it satisfies $|f(x)| \leq M$, for some M (Ruiz and Zeron, 2021). Then for each $n \geq 0$,

$$\sup_{x \in [-1, 1]} |f(x) - f_n(x)| \leq \frac{2M\rho^{-n}}{\rho - 1},$$

where f_n is the Chebyshev projection of degree n .

In addition, polynomial interpolation of Chebyshev points has the same convergence properties as Chebyshev series. Thus, the polynomial interpolant p_n to a continuous function f of the first $n + 1$ Chebyshev points can be expressed as a linear combination of the first $n + 1$ Chebyshev polynomials as follows

$$p_n(x) = \sum_{k=0}^n c_k T_k(x). \quad (2.2.3)$$

The coefficients c_k can be computed from the images of f on Chebyshev points by using the Fast Fourier Transform (FFT). Another theorem states that the information carried by the coefficients a_k of the Chebyshev series.

Theorem 2.2.3. Let f be Lipschitz continuous on $[-1, 1]$ and let p_n be its Chebyshev interpolant. Let c_k and a_k be the Chebyshev coefficients of f_n and p_n , respectively. Then

$$\begin{aligned} c_0 &= a_0 + a_{2n} + a_{4n} + \dots \\ c_n &= a_n + a_{3n} + a_{5n} + \dots \end{aligned}$$

and for $1 \leq k \leq n - 1$

$$c_k = a_k + (a_{2n+k} + a_{4n+k} + \dots) + (a_{2n-k} + a_{4n-k} + \dots).$$

One can affirm that polynomial interpolation on Chebyshev points offers an explicit expression of the interpolator that we can obtain by evaluating the function f on Chebyshev points and applying the Fast Fourier Transform and high accuracy when increasing the number of interpolating points.

2.2.2 Evaluation of Chebyshev Interpolants

We have seen the advantages of the expression 2.2.3 for Chebyshev interpolants. However, there a more efficient way to evaluate these interpolants without the need to apply Fast Fourier Transform.

The barycentric formula presented in this section is based on the expression of polynomial interpolants in terms of Lagrange polynomials. As we have seen previously, the j^{th} Lagrange polynomial is expressed as

$$l_j(x) = \frac{\prod_{i \neq j} (x - x_i)}{\prod_{i \neq j} (x_j - x_i)}.$$

Then, for the points of interpolation (x_0, \dots, x_n) and the values of f on them (ν_0, \dots, ν_n) , the unique polynomial p_n to f of degree at most n can be expressed as

$$p_n(x) = \sum_{i=0}^n \nu_i l_i(x),$$

This expression has an evaluation cost of $O(n^2)$. Luckily, Jacobi noted that some of the products present in $\prod_{i \neq j} (x - x_i)$ are repeated many times, and consequently modified the expression to obtain another that has linear evaluation cost $O(n)$:

$$p_n(x) = l(x) \sum_{j=0}^n \frac{\lambda_j}{x - x_j} \nu_j,$$

where $l(x)$ is the node polynomial given by $l(x) = \prod_{k=0}^n (x - x_k)$ and $\lambda_j = \frac{1}{\prod_{i \neq j} (x_j - x_i)}$.

This expression is known as the modified Lagrange formula (barycentric interpolation formula of type 1) and it does not assume the particularity that can be represented by the interpolation points. Later on, M. Dupuy. (1948) came with a new expression known as the barycentric interpolation formula of type 2, which applies when the interpolation points are Chebyshev points (Ruiz and Zeron, 2021),

$$p_n(x) = \sum_{j=0}^n \frac{\lambda_j \nu_j}{x - x_j} / \sum_{j=0}^n \frac{\lambda_j}{x - x_j}. \quad (2.2.4)$$

A generalization of the formula 2.2.4 is given by the following theorem.

Theorem 2.2.4. Let (x_0, \dots, x_n) be the first $n + 1$ Chebyshev points and let (ν_0, \dots, ν_n) be the values of f on these points. Then the polynomial interpolant p_n to f on (x_0, \dots, x_n) is given by

$$p_n(x) = \sum_{j=0}^n \prime \frac{(-1)^j \nu_j}{x - x_j} / \sum_{j=0}^n \prime \frac{(-1)^j}{x - x_j}, \quad (2.2.5)$$

where $p_n(x) = \nu_j$ if $x = x_j$. Note that the symbol \prime means the terms in the summation are multiplied by 0.5 when $j = 0, n$.

Hence, Chebyshev interpolants can be evaluated in linear time and in a numerically stable manner using Lagrange polynomials and the barycentric interpolation formula, offering numerous advantages:

- Numerical stability.
- Evaluation in linear time $O(n)$.
- Only need grid points and values on grid points.
- Bypassing the intermediate FFT step.
- Scale invariant (possibility to map to general interval of the form $[a, b]$).

So far, it has been proven that Chebyshev interpolants might be an efficient tool to approximate pricing functions in order to calculate risk measures.

What makes Chebyshev points so special?

- Chebyshev interpolation succeeds at mitigating the Runge's phenomenon. Interpolation on Chebyshev nodes reproduces more stable and accurate results unlike interpolation methods on equidistant grid points which present an excessive oscillation especially with high degree polynomials.

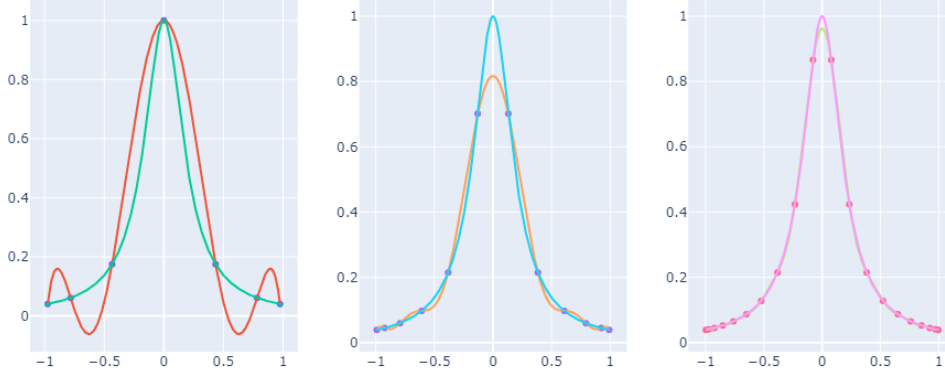


Figure 2.4: Interpolation of the Runge's function by different number of Chebyshev points: (Left) 5 points, (Center) 12 points and (Right) 18 points.

- Using polynomial interpolation on Chebyshev points, The convergence of the interpolant is exponential compared to the convergence rate of interpolation on equidistant nodes. It also allows us to use fewer points while achieving high level of accuracy.

2.2.3 Chebyshev Interpolants in High Dimension

Most pricing functions depend on a multitude of risk factors, for example, a swaption depends on interest rate curves and implied volatility surfaces. Therefore in the applications of our interest, which consist of using Chebyshev interpolants within risk calculations, we need to extend the definition presented so far to higher dimensions.

Definition 2.2.1. Consider the hyper-rectangle $[-1, 1]^d$ in \mathbb{R}^d . For each dimension consider $n_i + 1$ Chebyshev points, where $1 \leq i \leq d$. Let $f_{(k_1, \dots, k_d)}$ be the value of a function f on the Chebyshev point $p_{(k_1, \dots, k_d)}$, where $0 \leq k_i \leq n_i$. Then, the Chebyshev interpolant to f evaluated at p , where $p = (p_1, \dots, p_d) \in [-1, 1]^d$, is

$$\sum_{j \in J} c_j T_j(p). \quad (2.2.6)$$

The index j is an element of the set $J = \{(j_1, \dots, j_d) | 0 \leq j_i \leq n_i\}$ and the functions T_j are defined as

$$T_j(p_1, \dots, p_d) = \prod_{i=1}^d T_{j_i}(p_i).$$

The coefficients c_j are given by

$$c_j = \left(\prod_{i=1}^d \frac{2^{1(\{0 < j_i < n_i\})}}{n_i} \right) \sum_{k_1=0}^{n_1} \prime \cdots \sum_{k_d=0}^{n_d} \prime f_{(k_1, \dots, k_d)} \prod_{i=1}^d \cos \left(j_i \pi \frac{k_i}{n_i} \right), \quad (2.2.7)$$

where the symbole \prime means the terms in the summation are multiplied by 0.5 when $k_i = 0, n_i$.

This evaluation relies on the calculation of the coefficients c_j , the cost of which increases considerably as d increases. Similarly to the univariate case, when the function f is analytic, the Chebyshev interpolants converge with a high degree of precision. Figure 2.5 represents an example of a 2-dimensional grid of 10×10 Chebyshev points on $[-1, 1]$

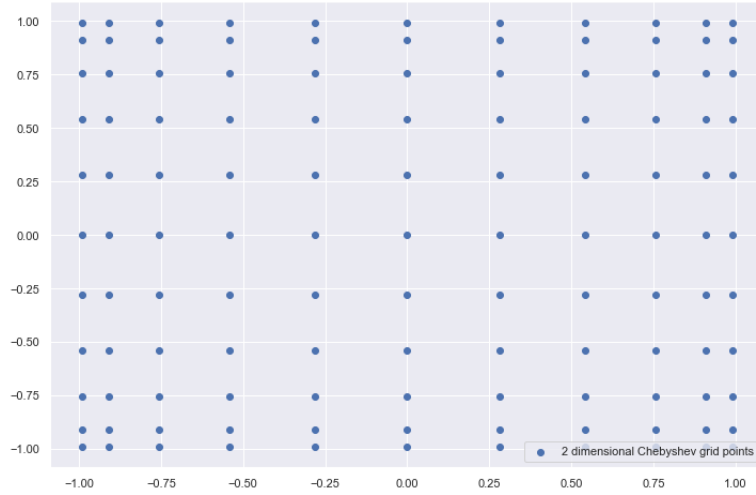


Figure 2.5: Two-dimensional Chebyshev grid points on $[-1, 1]^2$.

Theorem 2.2.5. Let f be a d -dimensional analytic function defined on $[-1, 1]^d$. Consider its analytical continuation to a generalised Bernstein ellipse E_p , where it satisfies $\sup_{x \in [-1, 1]^d} |f(x)| \leq M$, for some M . Then, there exists a constant $C > 0$, such that

$$\sup_{x \in [-1, 1]^d} |f(x) - p_n(x)| \leq C \rho^{-m},$$

where $\rho = \min_{(1 \leq i \leq d)} \rho_i$, and $m = \min_{(1 \leq i \leq d)} m_i$. The collection of values ρ_i define the radius of the generalised Bernstein ellipse E_p , and the values m_i define the size of the Chebyshev mesh (Zeron and Ruiz, 2021).

2.2.4 Principal Component Analysis for Dimensionality Reduction

As seen previously, when the dimensionality of the interpolant tensor increases, the number of interpolation points grows exponentially. As a matter of fact, if dimension of tensor is d and the number of points per dimension n , the number of points on tensor is n^d . Therefore, when dealing with functions presented in hundreds of dimensions, it becomes impossible to build an interpolator that can approximate the function in question. For instance, with a minimum of 2 points per dimension in a pricing function of dimension 100, one would need to build an interpolator of 1.27×10^{30} points.

To overchange this challenge, we investigate a well known dimensionality reduction technique with the aim of using it to side step the curse of dimensionality from which suffer Chebyshev interpolants. Principal component analysis (PCA) is a simple and fast technique that relies

on the correlation displayed by the input data to describe it in terms of variables that lay in a lower dimensional space, called latent space. Given that financial market risk factors often present high level of correlation (interest rates term structures, implied volatilities surfaces, credit default spreads..), we expect the PCA to work very well on them. Hence, we can describe the dynamics of market risk factors with a relatively low number of factors with good precision.

Compared to other dimensionality reduction techniques, the hyperparameter of the PCA is only the dimension of the domain of the latent space.

We use PCA in order to reduce the dimension of the risk factor space without losing much information. This enables the reduction of the building cost of Chebyshev interpolants.

In general, let m be the number of variables (risk factors) and n the number of numerical observations. Our data set can be seen as a matrix $X \in \mathbb{R}^{n \times m}$. Our aim is to find a representation of X in the subspace $F^k \subset \mathbb{R}^{n \times m}$ where ($k < m$) is the number of the new variables (principal components) that represent our data set with of information. Each of those variables will be a linear combination of the initial variables.

The algorithm of the PCA goes as follows

- first, we normalize the initial data so that contributions with different value ranges

$$\forall X_{ij}, X_{ij}^{norm} = \frac{X_{ij} - \mathbb{E}[X_j]}{\sigma(X_j)}$$

- Then we compute the covariance matrix C of the data set. Since the data is normalized,

$$C = \frac{1}{n} X^{norm} (X^{norm})^T$$

- C being a real valued symmetric matrix, it can be diagonalized. $C = P^T D P$, where D is the diagonal matrix of eigenvalues and P is the matrix of eigenvectors. The eigenvectors represent the directions of maximum variance and the eigenvalues represent the amount of variance explained by each eigenvector.
- Then we choose k , the number of principal components, depending on the amount of variance explained. They are then chosen by selecting the eigenvectors with the largest eigenvalues. These eigenvectors will be used to project the data onto the latent space, resulting in a lower-dimensional representation of the data.

When the domain of the function f , to be approximated, surpasses about 5 dimensions, we consider using the PCA technique before building the Chebyshev proxy. For a function f of dimension n , we apply the PCA function p to reduce the domain to k components ($k < n$), then we build a Chebyshev proxy C on the domain of dimension k to reconstruct the function f .

2.2.5 Sliding Technique

In this section we present the second alternative to bypass the curse of dimensionality : the sliding technique. A slide is an approximating object that can be a Chebyshev tensor. Slides are built in a way that several of these can be put together to form a bigger approximating object namely a slider. For our case, it is a Chebyshev slider.

In general, take a function f with domain of dimension n , and for some hyper-rectangle A in \mathbb{R}^n

$$f : A \rightarrow \mathbb{R}.$$

We pick a point $z = (z_1, \dots, z_n)$ in A . This is the pivot point for the slider. The function f can then be restricted to a lower dimensional space by letting some variables vary and fixing the remaining with the values given by the pivot point z .

If the studied function has a domain of high dimension, using sliders and choosing very low dimensions (3 or less) to build the slides can make building Chebyshev interpolants faster. In order to build a slider correctly, one must follow these conditions

- each variable in the domain of f is part of the domain of one and only one slide,
- all slides share the same pivot point z ,
- if we build k slides (s_1, \dots, s_k) for f , the sum of the dimensions of the slides has to be equal to the dimension of f : $\dim(s_1) + \dots + \dim(s_k) = n$.

Remark 2.2.2. If we consider k slides and each has l_i Chebyshev points on which f is evaluated, the total number of evaluations required to build the slider is $l_1 + \dots + l_k$. Usually we build slides of maximum dimension equals to 3, then the total number of points in the slider is considerably smaller than for an n -dimensional Chebyshev grid.

Before building a slider, one must make few choices:

- The number of slides that make the slider. We must take into account that increasing the number of slides reduces the number of evaluations required to build the slide and we can sidestep the exponential growth of the Chebyshev grids as the dimension increases.
- The dimension of each slide. The importance of this point is entirely related to the previous one. For instance, if we consider building n unidimensional slides, the Chebyshev grid becomes of linear growth. However, this might reduce the accuracy of the slider as we cancel all the correlation between the variables.
- The number of sampling points of each slide. The number of sampling points determines the number of evaluations to build the slider. One may not increase the number of points to reduce building effort. One may also choose to define more sampling points of a slide than the others if the slide corresponds to variables that are more significant than the others.
- The assignment of variables of f to slides. This is very important because it enables us to increase the accuracy of the slider without increasing the number of evaluations. In fact, one may put together variables that carry the highest non-linear relationships under f (Zeron and Ruiz, 2021).

To better understand how sliders are evaluated, we consider a function f defined on a domain of 6 dimensions. We first define the pivot point z which corresponds to a point of the domain of f , and we note v such that $v = f(z)$ the value of f at the point z . For instance, in order to approximate the function f using a slider, we assume that the slider's configuration is $(1, 2)$ meaning that the slider is composed of 2 slides (s_1, s_2) such that $\dim(s_1) = 1$ and $\dim(s_2) = 2$.

As mentioned before, slides are built by letting some variables vary and fixing the remaining ones with the values given by the pivot point. Therefore, the value of the slides on a given point $x = (x_1, x_2, x_3)$ from the domain of f are given by

$$\begin{aligned} s_1(x_1) &= f(x_1, z_2, z_3), \\ s_2(x_2, x_3) &= f(z_1, x_2, x_3). \end{aligned}$$

We note S_i the difference between the value of the slider s_i and v

$$\begin{aligned} S_1(x_1) &= s_1(x_1) - v, \\ S_2(x_2, x_3) &= s_2(x_2, x_3) - v. \end{aligned}$$

Then, we approximate the value of f at x by

$$f(x_1, x_2, x_3) \approx v + S_1(x_1) + S_2(x_2, x_3). \quad (2.2.8)$$

Sliders can be very useful to sidestep dimensionality problem for complex pricing functions that need to be evaluated thousands of times.

Figure 2.6 is an illustration of the grid points of sliders in comparison with the 2-dimensional Chebyshev grid points.

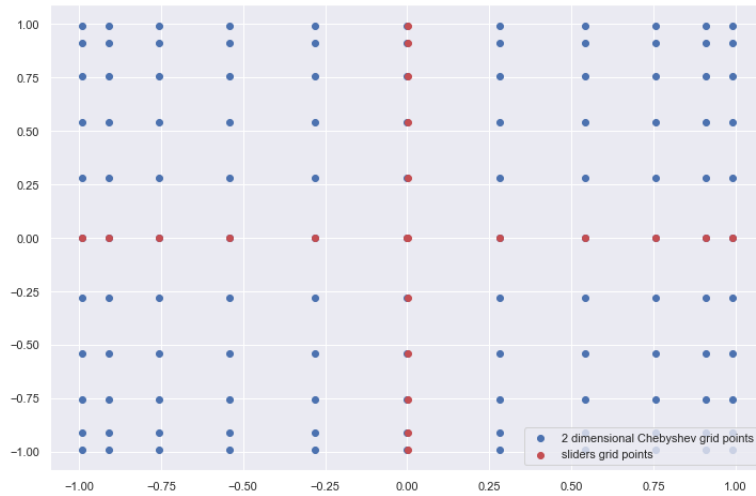
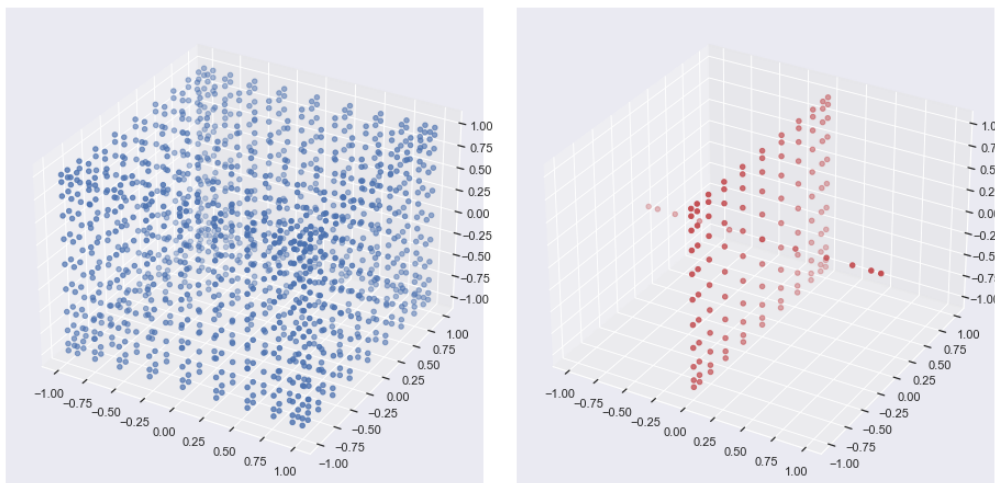


Figure 2.6: Two-dimensional sliders grid points (*red*) vs. full Chebyshev grid points (*blue*)

And for a three-dimensional grid, Figure 2.7 shows that the reduction of evaluation points is even more important.



(a) Full Chebyshev grid points

(b) Sliders grid points

Figure 2.7: Full Chebyshev grid points (*left*) VS sliders grid points (*right*).

Figure 2.6 and Figure 2.7 clearly illustrate an example of how we went from the need to evaluate on all the points in the full Cheybshev grid of 10 points per dimension (resulting in 10^3 points), to only evaluating on 110 ($= 10^2 + 10$) points if we consider a slider of configuration (2,1).

Chapter 3

Financial Trading Portfolios Case Study

In hereafter the composition of the portfolios that we considered for application interests.

Portfolio 1: for market risk calculation, we choose to work on a portfolio of 625 interest rate swaps and 425 european swaptions. This corresponds to taking into account the following risk factors

- ZC curves : each of 13 points.
- Implied volatilities surfaces : each of 100 points.

Portfolio 2: for counterparty risk calculation, we choose to work on a portfolio of 400 interest rate swaps. This corresponds to taking into account the following risk factors:

- ZC curves : each of 13 points.

3.1 Market Risk Calculation

3.1.1 Numerical Results of the Traditional Approach

We aim at computing the following market risk measures referring to the FRTB regulation:

- Value-at-risk at 99%
- Expected-shortfall at 97.5%.

We consider the portfolio 1 on a time horizon of 10 days. In order to compute the VaR and ES, we generate a large collection of 50,000 scenarios of our chosen risk factors which correspond to interest rate curve and implied volatility surfaces.

Note that the MtM of our portfolio at time 0 is equal to : 30 479.63€.

Running the evaluation of the portfolio on 50,000 Monte-Carlo simulations, we obtain the following results for VaR and ES calculation

	Full pricing
VaR _{99%}	950.49 €
ES _{97.75%}	952.22 €
Diffusion time	9min20s
Pricing time	14min02s

Table 3.1: Performace of the full pricing approach in VaR/ES calculation

These results were obtained for a not very complex portfolio that consists of swaps and swaptions. For banks having large portfolios that are usually composed of more complex instruments, such as Bermudan swaptions, such calculation can easily reach high level of complexity and MC approach becomes unsuitable for this task.

3.1.2 Application of the Alternative Approach

In order to compute the VaR and ES, we consider the same setting of the MC approach. Our aim is to predict the 50,000 scenarios of our selected risk factors which correspond to interest rate curves of 13 points each, and implied volatility surfaces of 100 points each.

3.1.2.1 Implementation of the Proxy

The main objective of this section is to compare the full pricing approach and the alternative approach using both PCA and Chebyshev interpolation (or the sliding technique) in terms of speed and accuracy.

The way this was achieved was by using 2 PCA transformations where different families of risk factor were transformed by different PCA models. More specifically, in the case of our defined portfolio, which depend on interest rates and implied volatilities both diffused in 10-day time horizon, a PCA function, denoted by p_1 , would be trained on the shocked interest rates, and a second one, independent of the first, denoted by p_2 , trained on the shocked implied volatilities. An Chebyshev proxy would then be built on the domain obtained by putting together the projections of these PCA models.

In practice, we made great use of the correlation presented in the domain of risk factors, and succeeded using PCA to reduce the domain of definition of the portfolio pricing function from 113 dimensions to only 2 dimensions, the first is the output of p_1 trained on interest rates, while the second is the output of p_2 trained on implied volatilities, each with high explained variance. Then, we plug these 2 PCA components to build a 2-dimensional Chebyshev proxy.

In practice, to build such a proxy, we create 2 Chebyshev grid points on the interval $[-1, 1]$. We then apply the transformation (2.2.1) to pass to the domain of the PCA components. Then, we get a generalized grid of Chebyshev points on which we will interpolate the pricing function. We take advantage of the inverse function of the PCA to get the inverses of each points of our grid which correspond to interest rate curves and implied volatility surfaces. We apply this inverse PCA transformation because we need to evaluate the pricing function on the points of Chebyshev, but this pricing function takes the interest rate curves and implied volatility surfaces as they are. It doesn't take the corresponding PCA component. Once we get the reduced number of curves and surfaces, we plug them to get the evaluation of the defined Cheybshev points. The images of these points serve then to build the interpolant. Afterwards we return to the Chebyshev domain $[-1, 1]$ because the calibbration of the coefficient c_j is only defined on this

interval. To do so, we inverse the transformation (2.2.1) to get the original Chebyshev points. The calibration of the coefficients c_j of the interpolant is done following the expression (2.2.7). Notice here that we only need the Chebyshev points and their corresponding images by the full pricing function. Once the coefficients are calibrated, it only takes to apply 2.2.6 to get the prediction of the portfolio prices on all the $50K$ scenarios.

3.1.2.2 Numerical Results

We illustrate the distribution of portfolio prices obtained using 2-dimensional Chebyshev interpolation below, plotted against the distribution obtained previously using MC method on the same input parameters and portfolio composition.

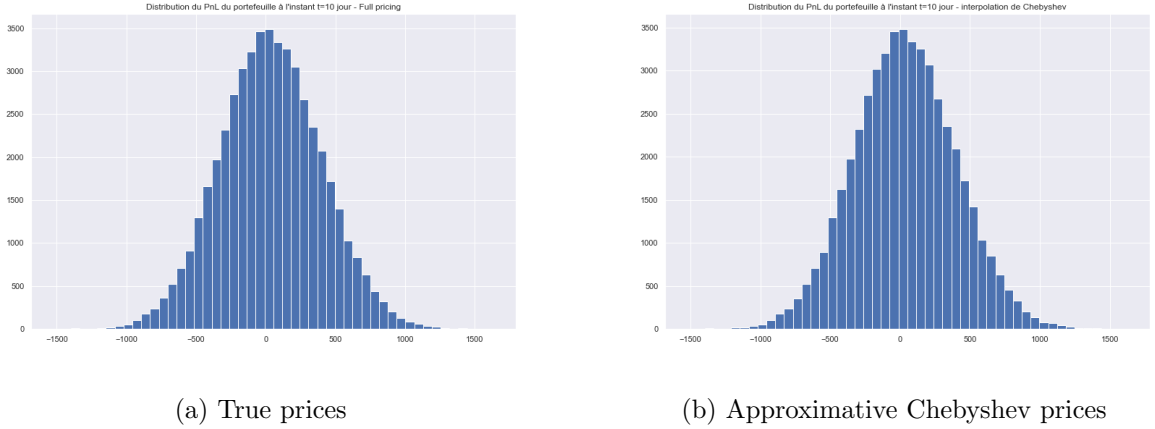


Figure 3.1: Comparaison of PnL distribution at time $t = 10$ days: The true prices distribution (*left*) is the result of the full pricing approach whereas the approximated prices (*right*) were obtained by applying the Chebyshev interpolation.

From Figure 3.1, it is clear that the distribution of prices obtained by applying the Chebyshev interpolation is almost identical to the the distribution of the full pricing approach. The adequacy of the distributions was also tested using Kolmogorov-Smirnov test on these two PnLs samples. The idea behind this test was to rule out the existence of any statistical evidence concerning the differences between the benchmark distribution (full pricing) and the one obtained with the Chebyshev interpolation. The test also confirms that the two distributions are very close.

We also compare the results by visualising the prices obtained with both approach against each others. Figure 3.2 shows the full pricing portfolio prices in the x -axis against the Chebyshev interpolation prices in the y -axis.

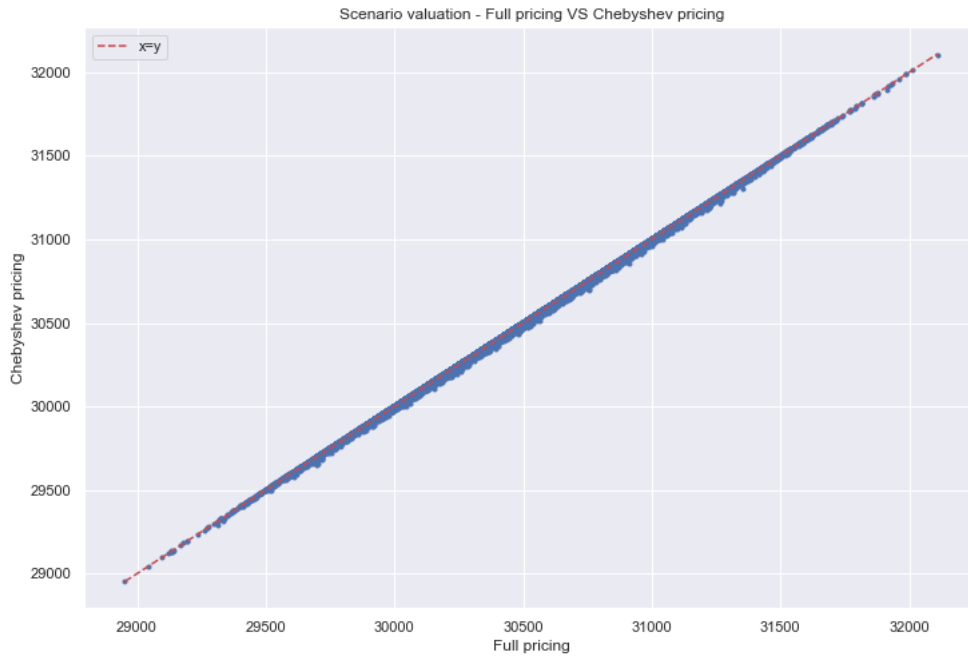


Figure 3.2: Q–Q plot for comparing prices distribution (full pricing approach vs. Chebyshev interpolation): a point (x, y) on the plot corresponds to one of the quantiles of the true prices' distribution (x -axis) plotted against the same quantile of the Chebyshev prices' distribution (y -axis).

Figure 3.2 also confirms the similarity of the two distributions since the points in the Q–Q plot approximately lie on the identity line $y = x$.

We also wanted to test the performance of sliders. We took the same portfolio and risk factors on which we apply PCA transformations the same way done previously. We choose to build a slider of 2 uni-dimensional slides, one for each PCA component obtained previously. Once a dimension has been fixed, two parameters need to be specified to define a slider: the dimension of each of the slide, and the number of interpolation points for each slide. For simplicity, the number of points for each dimension in each slide was fixed to 10.

The evaluation of the slider is done with use of the formula (2.2.8), and the construction of Chebyshev tensors for each slide is done the same way as described in the previous test.

The resulting distribution of portfolio prices is displayed in Figure 3.3.

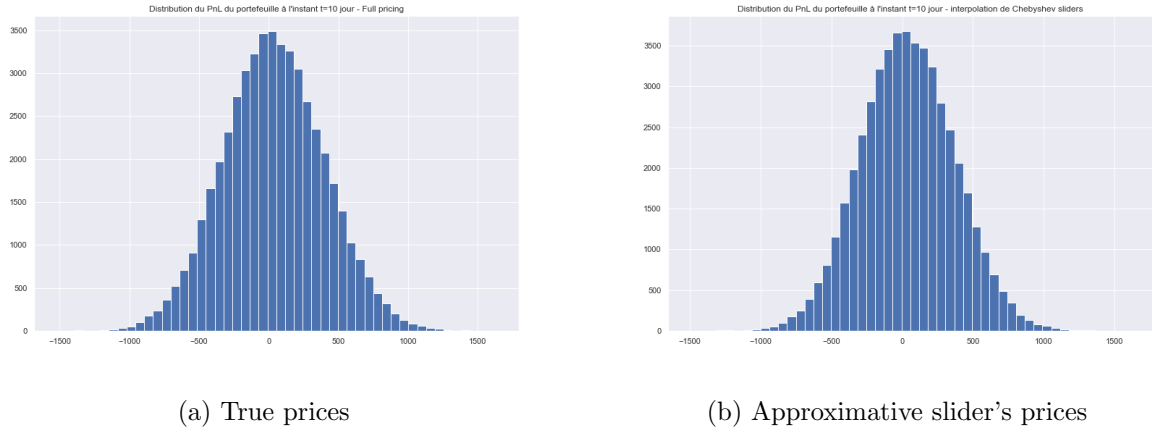


Figure 3.3: Comparaison of PnL distribution at time $t = 10$ days: The true prices's distribution (*left*) is the result of the full pricing approach whereas the approximative prices (*right*) were obtained by applying the sliding technique.

Although visually the distributions in Figure 3.3 seem to be close, but it is not at the same level of similarity observed previously in Figure 3.1. Therefore, we get a comparison of the PnLs obtained by plotting the prices distribution obtained with MC approach (x -axis) against the prices obtained by the Chebyshev Sliding Technique (y -axis) in Figure 3.4.

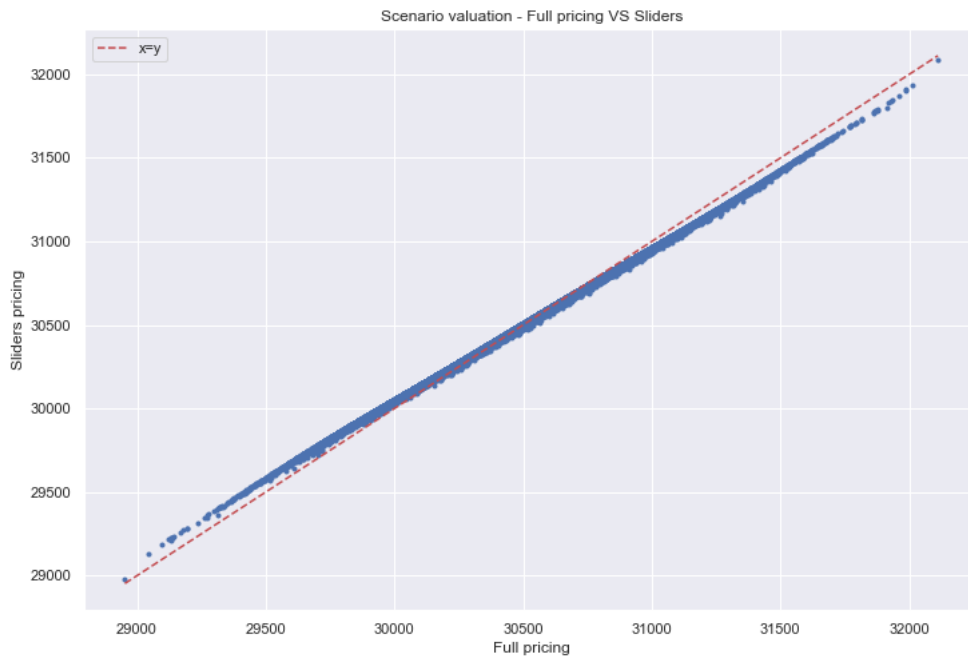


Figure 3.4: Q-Q plot for comparing prices distribution (full pricing approach vs. sliding technique): a point (x, y) on the plot corresponds to one of the quantiles of the true prices' distribution (x -axis) plotted against the same quantile of the slider prices' distribution (y -axis).

This gives visual evidence of the error incurred by the Chebyshev Slider alone.

The comparison of these three approaches is summarized in Table ??.

	Full pricing	2-d Chebyshev	Sliders
VaR _{99%}	950.49 €	951.36 €	883.18 €
ES _{97.75%}	952.22 €	952.31 €	885.90 €
Diffusion time	9min20s	9min20s	9min20s
Pricing time	14min02s	11s	2s

Table 3.2: Comparaison of performace in VaR/ES calculation

The accuracy of both alternative techniques was ultimately measured on the ES values obtained. Starting with the study of multivariate Chebyshev interpolation's performance, the interpolants reduced the reevaluation number of the pricing function from 50,000 to only 100 evaluation resulting in a computational gain of 99.98% while the relative error on EE is less than 0.01%. The combination of PCA and Chebyshev interpolation is then proven to be very beneficial for computing risk metrics of large portfolios involving many risk factors. This can be generalized to any portfolio and size and we expect an equally interesting reduction in computational effort while maintaining high level of accuracy.

As for sliders, we observe that indeed, we obtained excellent results in term of pricing time reduction, however, this was at the expense of accuracy. The relative error of ES is about 6.9%, and it is very high compared with the error of multivariate Chebyshev interpolation.

The solution that we judge to be the most suitable is the combination of PCA and multivariate Chebyshev interpolation.

3.2 Counterparty Risk

3.2.1 Numerical Results of the Traditional Approach

We aim at computing the Expected Exposure and Credit Valuation Adjustment risk measures considering portfolio 2 for numerical application. We consider a time horizon of 8 years which corresponds to the maximum maturity of the swaps. We aim at computing a daily exposure which means 2880 time steps throughout the defined time horizon.

In order to compute the EE and CVA, we generate a large collection of 10,000 scenarios of our chosen risk factors at all dates of exposure calculation (2,880 time steps).

We also set the following parameters for the CVA calculation $R = 0.4$ and $\lambda = 0.5\%$.

Remark 3.2.1. The MC simulation that we are about to apply generates 50,000 paths of market risk factors in the future at 2,880 time points. Under these assumptions, these are 28,800,000 nodes generated in the calculation of the desired counterparty risk metrics.

Before proceeding with the determination of the EE profile and the CVA calculation for the portfolio defined in the previous section, we give the numerical results of this calculation applied to a single swap. We also introduce an approximation to the valuation of a swap, which will enable us to calculate the EE and CVA more simply, without the need for Monte-Carlo simulations. Secondly, we present the results obtained for the whole portfolio.

We apply the MC algorithm to obtain the EE profile and the CVA of a portfolio consisting of 400 swaps.

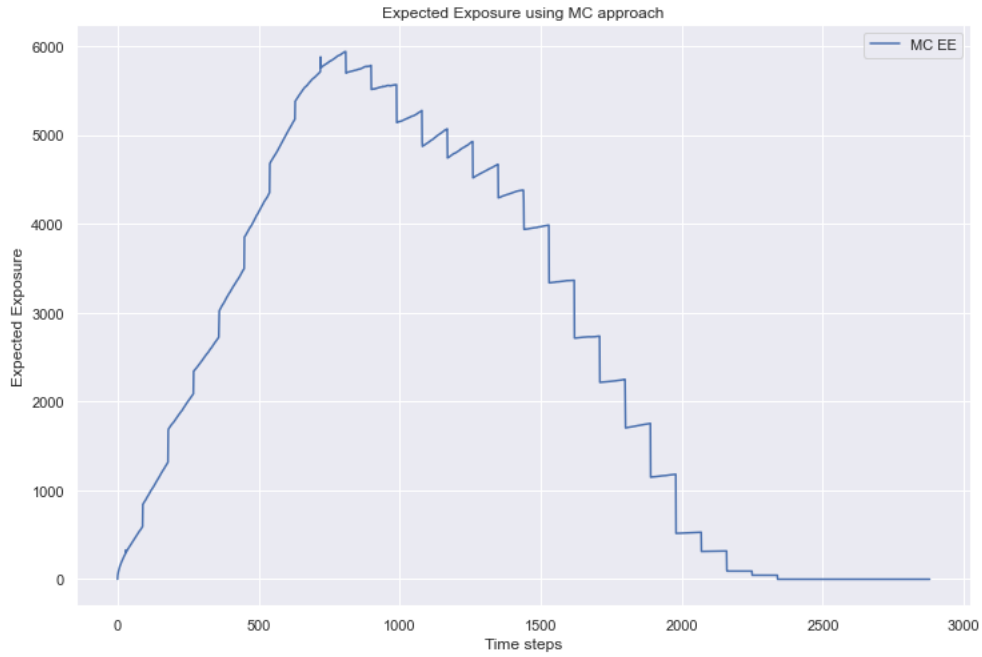


Figure 3.5: Expected Exposure profile of portfolio 2 on 8-year time horizon calculated with MC approach.

The profile presented in Figure 3.5 corresponds well to what is expected. In fact, the exposure increases since we are facing several cash flows if swaps payments in the future. At time $T/3$, we observe a maximum of exposure. Then, the curve decreases as the swaps' cash-flows are less and some of the swap contracts figuring in the portfolio begin conclude.

We also obtain the following analysis for the CVA calculation at time 0

	Full pricing
CVA_0	64.27 €
Diffusion time	2min
Pricing time	2h29min

Table 3.3: Performance of the full pricing approach in CVA calculation

These results are obtained for a standard portfolio of interest rate swaps. The task can become drastically expensive once we consider complex portfolio with thousands of trades. The calculation of EE and CVA require reevaluation of the portfolio price considering long time horizons. The MC approach can thus become unsuitable for this calculation.

3.2.2 Approximation to the Valuation of a Swap

In this section, we introduce an approximation for the dynamics of the swap rate in the framework of LGM-1F. We recall the expression of the swap rate

$$S(t, T_0, T_n) = \frac{B(t, T_0) - B(t, T_n)}{\sum_{i=1}^n \delta_i B(t, T_i)}.$$

We calculate dS_t with itô formula and we neglect the drift term

$$dS(t) = \left[\frac{dB(t, T_0) - dB(t, T_n)}{\sum_{i=1}^n \delta_i B(t, T_i)} - \frac{(B(t, T_0) - B(t, T_n))}{(\sum_{i=1}^n \delta_i B(t, T_i))^2} \sum_{i=1}^n \delta_i dB(t, T_i) \right] + (\dots)dt$$

With LGM1F dynamics, we get

$$dS(t) = \frac{\sigma(t)}{\lambda} e^{\lambda t} S(t) \left(\frac{B(t, T_0) e^{-\lambda T_0} - B(t, T_n) e^{-\lambda T_n}}{B(t, T_0) - B(t, T_n)} - \frac{\sum_{i=1}^n \delta_i B(t, T_i) e^{-\lambda T_i}}{\sum_{i=1}^n \delta_i B(t, T_i)} \right) dW_t^Q + (\dots)dt$$

We introduce the function $g(t)$ in order to simplify the expression above

$$dS(t) = \sigma(t) e^{\lambda t} g(t) S(t) dW_t^Q + (\dots)dt$$

where

$$g(t) = \frac{B(t, T_0) e^{-\lambda T_0} - B(t, T_n) e^{-\lambda T_n}}{B(t, T_0) - B(t, T_n)} - \frac{\sum_{i=1}^n \delta_i B(t, T_i) e^{-\lambda T_i}}{\sum_{i=1}^n \delta_i B(t, T_i)}$$

We freeze $g(t) = g(0)$ to get the log normal dynamics

$$dS(t) = \sigma(t) e^{\lambda t} g(0) S(t) dW_t^Q,$$

And the normal dynamics

$$dS(t) = \sigma(t) e^{\lambda t} g(0) S(0) dW_t^Q. \quad (3.2.1)$$

Roux (2008) test the normal dynamics for the swap rate in order to validate its adaptability to the problem. Numerical tests for pricing european swaptions have been done. For instance, for constant value σ of the instantaneous volatility obtained from the calibration of LGM-1F, and for constant mean reversion λ , we calculate the price of the same swaption using both the expression (1.3.1.1) and the Black-Scholes formula as shown in equation (1.2.2) taking into account the normal dynamics of the swap rate (3.2.2).

The price of the swaption in the case of normal dynamics is

$$P_{swaption}^{approx}(\sigma, \lambda, T_e, T_0, T_n, K) = A(0) \left[S(0) \sigma^* \sqrt{T_e} \varphi \left(\frac{K - S(0)}{S(0) \sigma^* \sqrt{T_e}} \right) + (S(0) - K) \left(1 - \Phi \left(\frac{K - S(0)}{S(0) \sigma^* \sqrt{T_e}} \right) \right) \right], \quad (3.2.2)$$

where

- $A(0) = \sum_{i=1}^n \delta_i B(0, T_i)$,
- $S(0) = \frac{B(0, T_0) - B(0, T_n)}{\sum_{i=1}^n \delta_i B(0, T_i)}$,
- $\sigma^* = \frac{\sigma g(0)}{\sqrt{T_e}} \sqrt{\frac{e^{2\lambda T_e} - 1}{2\lambda}}$,
- φ is the probability density function of the standard Gaussian variable.

For more details on the approximation and the validity tests, we refer to the article Roux (2008).

The tests performed by Roux (2008) show that the gaussian approximation of the dynamics of the swap rate as detailed previously, is really close to the dynamics obtained in the LGM-1F framework. Therefore, this approximation is suitable and we are going to use it in order to calculate the exposure of portfolio of a single swap and compare the result to the exposure obtained with MC approach.

In fact, in the case of normal dynamics of the swap rate, we can approximate the value of a receiver swap. First, from the dynamics (3.2.2), we deduce the expression of the swap rate through the volatility defined with

$$\begin{aligned} (\sigma^*)^2 t &= \int_0^t (\sigma e^{\lambda t} g(0) S(0))^2 dt, \\ \sigma^* &= \frac{\sigma g(0)}{\sqrt{t}} \sqrt{\frac{e^{2\lambda t} - 1}{2\lambda}}. \end{aligned} \quad (3.2.3)$$

Then

$$S(t, T_0, T_n) = S(0) + \sigma^* \sqrt{t} \epsilon, \quad (3.2.4)$$

where $\epsilon \sim \mathcal{N}(0, 1)$.

Note that the value of a payer swap is

$$\begin{aligned} P_{swap}(t, T_0, T_n) &= N \left[B(t, T_0) - B(t, T_n) - \sum_{i=1}^n \delta_i K B(t, T_i) \right] \\ &= N A(t) (S(t, T_0, T_n) - S(0)), \end{aligned}$$

where $K = S(0)$ is the fixed rate.

$A(t)$, the annuity of the swap, can be approximated as follows

$$A(t) = \sum_{i=1}^n \delta_i B(t, T_i) \approx \frac{1 - e^{-S_t(T_n - t)}}{S_t} \approx (T_n - t). \quad (3.2.5)$$

Under the normal dynamics of the swap rate (3.2.4) and considering the approximation 3.2.2, the value of the swap contract becomes

$$P_{swap}(t, T_0, T_n) = N(S(t, T_0, T_n) - S(0))(T_n - t) \sim \mathcal{N}(0, (\sigma^*)^2 (T_n - t)^2 t).$$

In this case of normal model, the expected exposure is given by

$$EE(t) = \frac{2}{3\sqrt{2\pi}} \sigma^* \sqrt{t} (T - t). \quad (3.2.6)$$

We consider a single interest rate swap of maturity 8 years and calculate the daily expected exposure through an 8-year time horizon using MC approach. Figure 3.6 gives an illustration of the EE profile calculated with MC approach and the profile calculated with the approximation (3.2.6):

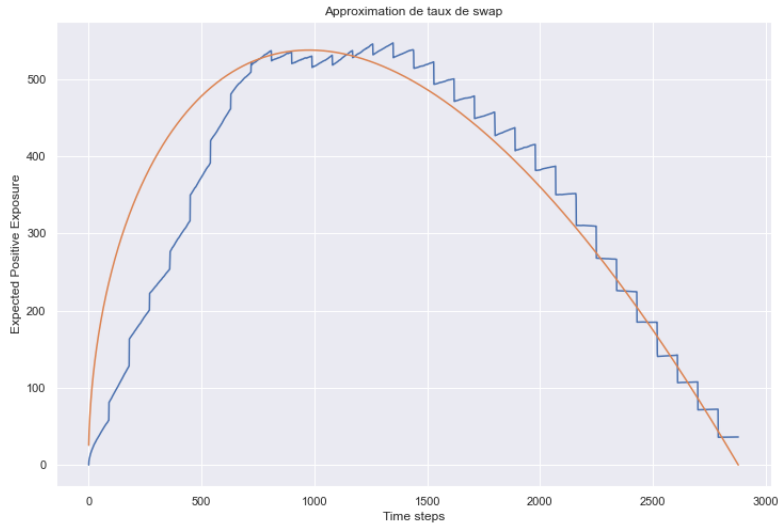


Figure 3.6: EE profile of a single interest rate swap by the normal dynamics approximation of swap rate vs. EE profile of the same swap by MC approach.

This approximation was made in order to avoid the use of the MC approach, being very expensive. It can serve to reduce computational burden while getting an approximation for CVA.

3.2.3 Numerical Results of the Alternative Approach

In order to compute the EE and CVA, we consider the same setting of the MC approach and we work with the same portfolio setting with 400 interest rate swaps. Our aim is to predict the 10,000 scenarios of our chosen risk factors which correspond to interest rate curves of 13 points each. We also make use of the PCA to reduce the domain of the swap pricing function from 13 dimensions to only one.

In addition to interpolation on the number of scenarios to evaluate the portfolio, the objective is to apply Chebyshev interpolation on the time domain in order to optimise the determination of the exposure distribution and subsequently calculate the CVA

$$CVA_0 = \sum_{k=1}^n \sum_{j=1}^M D^{(j)}(t_k) E^{(j)}(t_k) [P(t_k) - P(t_k - 1)]. \quad (3.2.7)$$

This makes of the time to maturity a risk factor in the calculation of CVA.

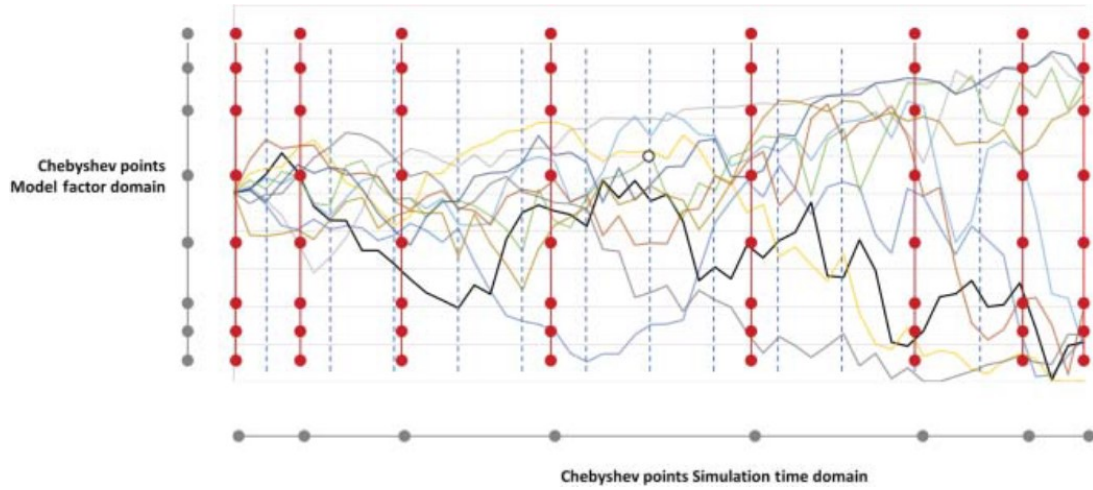


Figure 3.7: Chebyshev interpolation on time and risk factors domain.

The configuration of the proxy for this application will be the following: two uni-dimensional Chebyshev interpolants for each dimension (one for the PCA component that explains interest rate curves, and another for the time domain) evaluated with the barycentric formula (2.2.5). The objective is then to use our proxy to attempt reconstructing the EE profile obtained with MC approach in order to calculate the CVA with high degree of accuracy. We note M' the number of grid points on the scenarios domain and n' the number of grid points on the time domain. We also note $n' \times M'$ the configuration of the Chebyshev interpolant. In order to get the following illustration of EE profile we use 20×10 configuration.

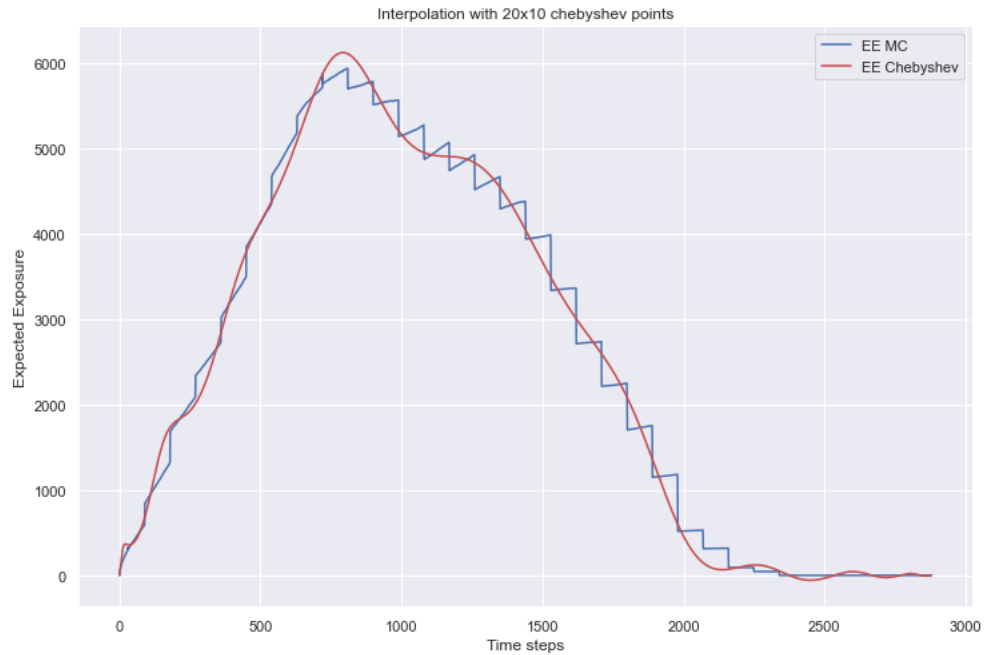


Figure 3.8: EE profile for portfolio 2 by MC approach (blue) vs. Chebyshev interpolation (red).

We wanted to see the influence of the number grid points attribution on the precision of reconstructing the EE profile. We ran the interpolation algorithm on different configurations of grid points:

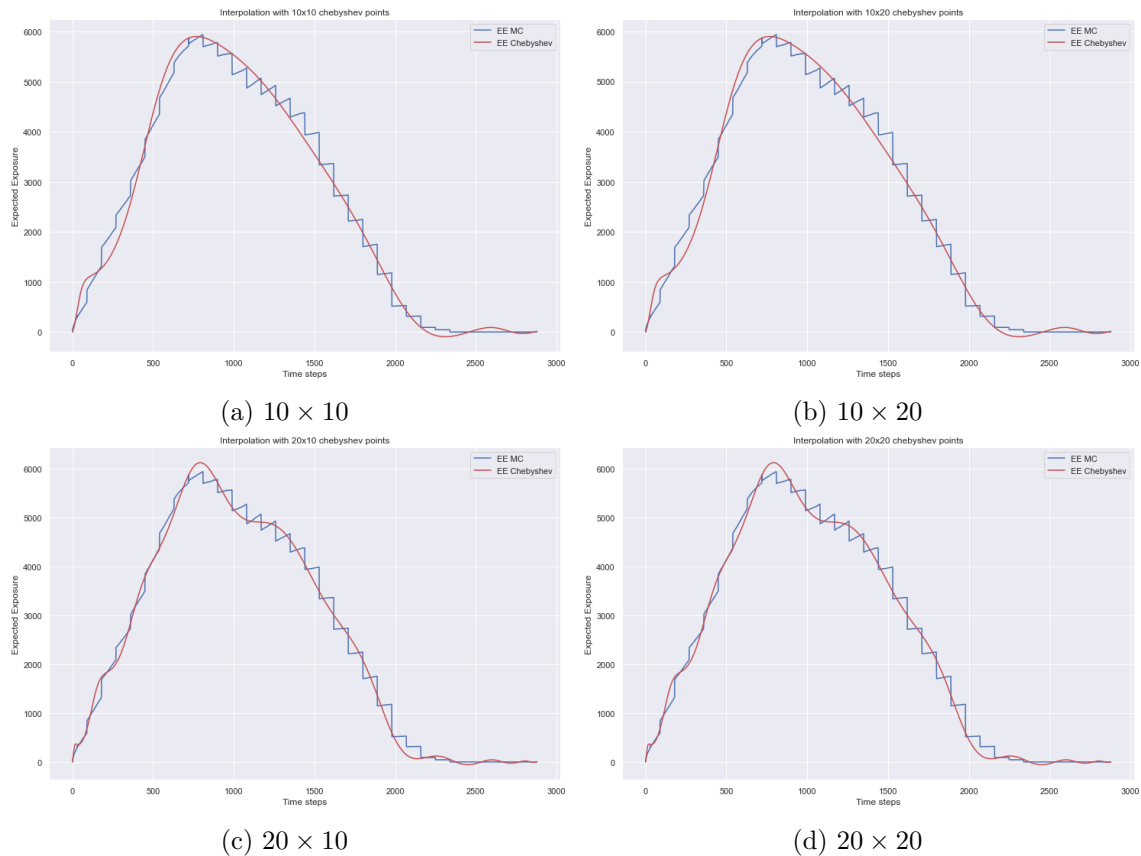


Figure 3.9: Influence of the number of interpolation nodes on the EE profile. For each subplot, the number of grid points is indicated.

Naturally, we observe, in Figure 3.9, that the calculation of EE is more accurate when we increase the grid points on the time domain. Thus, the ideal setup is to allocate more number of points on the time domain than the scenarios domain. With the 20×10 configuration, we reduce the number of evaluation of the portfolio pricing function from 28,800,000 nodes to only 200.

Table 3.4 illustrates the analysis for CVA calculation

	Monte-Carlo	Chebyshev
CVA_0	64.27 €	64.31 €
Diffusion time	2min	2min
Pricing time	2h29m	13s

Table 3.4: Performace of the Chebyshev interpolation in CVA calculation

In terms of accuracy, the relative error of the CVA obtained with Chebyshev interpolation compared to MC approach is of 0.0622% giving a total computational gain of 99.98%.

It is worth noting that the interest rate swap pricer is usually fast, so the incentive to run the calculation with Chebyshev interpolants is limited. However, banks tend to have a very large portfolios consisting of more complex products, so the computational savings obtained when we consider such complex portfolios will not be negligible.

Conclusion

In order to accurately quantify our risk measures (VaR and ES), we need to re-evaluate our entire portfolio multiple times. This step is extremely expensive, therefore the bank is forced to use simpler models that are less representative of the reality and less time-consuming. To bypass this issue, we implemented a proxy using Chebyshev interpolation and PCA to be able to speed-up the evaluation of the portfolio without losing much precision. The numerical experiments that were presented showed that the approximation approaches greatly improve a portfolio evaluation capacity, while reducing the computational burden and maintaining a solid level of accuracy.

Similar question was addressed for the Counterparty risk calculations which rely on thousands of scenarios and time steps, easily creating a grid of millions of simulation nodes inducing massive computations. In addition, portfolios of financial institutions tend to be large and complex, making the computational load heavier. And again, building a pricer proxy based on Chebyshev interpolants which require only a small number of calls of the pricing functions to be calibrated, was numerically proven to be efficient.

Analysis of the previous results naturally lead us to question the use of the classical re-pricing method in the context of risk measurement. In fact, while the classical approach suggests to perform M pricings to obtain a distribution of prices allowing to infer all sort of risk measures, the alternative proxy approach in turn, only needs to evaluate the pricer on m ($m \ll M$) trajectories, to be built. The resulting interpolator, would then be evaluated on M trajectories to obtain the desired distribution at a lower computational cost. In fact, as our numerical results show, this approach represent a non-negligible time saving technique once implemented inside the systems of banks to approximate Front Office pricing functions with the aim of reducing the substantial computational burden associated within the FRTB IMA capital calculation. In fact, we have shown through the numerical results that the computational gain of using the multidimensional Chebyshev interpolation could exceed 99% which is outstanding compared to MC method.

We also implemented the sliders as a dimensionality reduction technique on Chebyshev interpolants for the calculation of the VaR and ES. Tests and numerical results confirmed the huge gain in terms of computational time, however, this improvement comes at the expense of the accuracy giving a relative error on ES higher than 5%. Thus, we tend to favor the combination of PCA and Chebyshev interpolation over the sliding technique for this problem.

The study case portfolios considered for the tests are composed of swaps and swaptions. These products represent a large percentage of the Interest Rate derivative products traded globally, and they account for a significant part of the computational burden in any risk calculation. Also, interest rate swaps represent 87% of the gross market value of Interest Rate contracts worldwide. Therefore, accuracy at the level of these products is crucial to obtain accuracy at the level of the whole portfolio. In view of the promising performance of Chebyshev interpolation, we propose to extend this work to more complex applications considering large portfolios consisting

of complex financial products such as Bermuda swaptions.

Another point to examine is the stability of the technique. In this respect, the central questions are whether the Chebyshev interpolation will work under stressed market conditions. Results of this work show that the stability of the combination of PCA with Chebyshev interpolants is remarkable. However, further tests on more stressed conditions need to be studied.

Furthermore, this methodology can be adapted to calibration problems, in particular the correct construction of implied volatility surfaces. This would represent an intriguing and practical application that can significantly benefit financial markets and risk calculation.

Appendix A

LGM-1F

The instantaneous forward rate

In the HJM framework, the dynamics of the zero-coupon bond of maturity T is given by

$$\frac{dB(t, T)}{B(t, T)} = r_t dt + \Gamma(t, T) dW_t^{\mathbb{Q}}.$$

The instantaneous forward rate of maturity T is defined by

$$f(t, T) = -\partial_T \ln(B(t, T)), \text{ or } B(t, T) = e^{-\int_t^T f(t, s) ds}.$$

The corresponding dynamics of $f(t, T)$ under \mathbb{Q} is:

$$df(t, T) = -\gamma(t, T)\Gamma(t, T)dt + \gamma(t, T)dW_t^{\mathbb{Q}},$$

where

$$\gamma(t, T) = -\partial_T \Gamma(t, T),$$

with the LGM form of volatility, the instantaneous forward rate is:

$$f(t, T) = f(0, T) - \int_0^t \frac{\sigma(s)^2}{\lambda} \left(e^{-2\lambda(T-s)} - e^{-\lambda(T-s)} \right) ds + \int_0^t \sigma(s) e^{-\lambda(T-s)} dW_s.$$

The short rate process

$$\begin{aligned} r_t &= f(t, t) \\ &= f(0, t) - \int_0^t \frac{\sigma(s)^2}{\lambda} \left(e^{-2\lambda(t-s)} - e^{-\lambda(t-s)} \right) ds + \int_0^t \sigma(s) e^{-\lambda(t-s)} dW_s \\ &= f(0, t) - \frac{e^{-2\lambda t}}{\lambda} \int_0^t \sigma(s)^2 e^{2\lambda s} ds + \frac{e^{-\lambda t}}{\lambda} \int_0^t \sigma(s)^2 e^{\lambda s} ds + e^{-\lambda t} \int_0^t \sigma(s) e^{\lambda s} dW_s. \end{aligned}$$

On the one hand we have

$$d(e^{\lambda t} r_t) = e^{\lambda t} dr_t + \lambda e^{\lambda t} r_t dt.$$

Putting the two previous equations together, we obtain

$$dr_t = \left[\lambda (f(0, t) - r_t) + \partial_2 f(0, t) + e^{-2\lambda t} \int_0^t \sigma(s)^2 e^{2\lambda s} ds \right] dt + \sigma(t) dW_t.$$

Reconstruction formula for zero-coupon bonds

This equation can be written as

$$dX_t = [\phi(t) - \lambda X_t] dt + \sigma(t) dW_t,$$

where

$$X_t = r_t - f(0, t) \quad \text{and} \quad \phi(t) = \int_0^t \sigma(s)^2 e^{-2\lambda(t-s)} ds.$$

Note that we have $X_0 = 0$. The aim of this subsection is to write the zero-coupon bonds as deterministic functions of the state variable X_t , namely:

$$\forall(t, T), \quad B(t, T) = B_{t,T}(X_t),$$

where $x \mapsto B_{t,T}(x)$ is a deterministic function.

First, we find the expression of the instantaneous forward rate $f(t, T)$ as a function of X_t . Idea: note that $f(t, T)$ and $e^{-\lambda(T-t)}r_t$ have the same stochastic parts, and calculate the difference $f(t, T) - e^{-\lambda(T-t)}r_t$ to get the result. The expression of the short rate r_t provides

$$\begin{aligned} e^{-\lambda(T-t)}r_t &= e^{-\lambda(T-t)}f(0, t) \\ &= -e^{-\lambda(T-t)} \int_0^t \frac{\sigma(s)^2}{\lambda} \left(e^{-2\lambda(t-s)} - e^{-\lambda(t-s)} \right) ds + \int_0^t \sigma(s) e^{-\lambda(T-s)} dW_s. \end{aligned}$$

Using the expression of the forward rate given in the previous section, we get

$$\begin{aligned} f(t, T) - e^{-\lambda(T-t)}r_t &= f(0, T) - \int_0^t \frac{\sigma(s)^2}{\lambda} \left(e^{-2\lambda(T-s)} - e^{-\lambda(T-s)} \right) ds \\ &\quad + e^{-\lambda(T-t)} \int_0^t \frac{\sigma(s)^2}{\lambda} \left(e^{-2\lambda(t-s)} - e^{-\lambda(t-s)} \right) ds - e^{-\lambda(T-t)}f(0, t), \end{aligned}$$

rearranging, and introducing X_t we obtain:

$$f(t, T) = f(0, T) + e^{-\lambda(T-t)}X_t + \int_0^t \frac{\sigma(s)^2}{\lambda} \left(e^{-\lambda(T+t-2s)} - e^{-2\lambda(T-s)} \right) ds,$$

we now have to compute the integral

$$\begin{aligned} \int_t^T f(t, u) du &= \int_t^T f(0, u) du + X_t \int_t^T e^{-\lambda(u-t)} du \\ &\quad + \int_t^T \int_0^t \frac{\sigma(s)^2}{\lambda} \left(e^{-\lambda(u+t-2s)} - e^{-2\lambda(u-s)} \right) ds du. \end{aligned}$$

The first integral in the right member above is:

$$\begin{aligned} \int_t^T f(0, u) du &= \int_0^T f(0, u) du - \int_0^t f(0, u) du \\ &= -\ln B(0, T) + \ln B(0, t). \end{aligned}$$

After straightforward calculations

$$\begin{aligned} \int_t^T \int_0^t \frac{\sigma(s)^2}{\lambda} \left(e^{-\lambda(u+t-2s)} - e^{-2\lambda(u-s)} \right) ds du \\ = -\frac{1}{2} \left(1 - e^{-\lambda(T-t)} \right)^2 \int_0^t \left(\frac{\sigma(s)}{\lambda} \right)^2 e^{-2\lambda(t-s)} ds \\ = -\frac{1}{2} \left(\frac{1 - e^{-\lambda(T-t)}}{\lambda} \right)^2 \phi(t). \end{aligned}$$

Defining $\beta(t, T)$ by

$$\beta(t, T) = \frac{1 - e^{-\lambda(T-t)}}{\lambda} = \int_t^T e^{-\lambda(u-t)} du,$$

then

$$\begin{aligned} B(t, T) &= \exp \left(- \int_t^T f(t, u) du \right) \\ &= \frac{B(0, T)}{B(0, t)} \exp \left\{ -\frac{1}{2} \beta(t, T)^2 \phi(t) - \beta(t, T) X_t \right\}. \end{aligned}$$

In summary, if we define

$$X_t = r_t - f(0, t),$$

then the process (X_t) satisfies the SDE

$$\begin{cases} dX_t = [\phi(t) - \lambda X_t] dt + \sigma(t) dW_t, \\ X_0 = 0. \end{cases}$$

At date t , any zero-coupon bond can be written as a deterministic function of X_t :

$$\begin{aligned} B(t, T) &= B_{t,T}(X_t) \\ &= \frac{B(0, T)}{B(0, t)} \exp \left\{ -\frac{1}{2} \beta(t, T)^2 \phi(t) - \beta(t, T) X_t \right\}, \end{aligned}$$

with

$$\phi(t) = \int_0^t \sigma(s)^2 e^{-2\lambda(t-s)} ds \quad \beta(t, T) = \frac{1 - e^{-\lambda(T-t)}}{\lambda} = \int_t^T e^{-\lambda(u-t)} du.$$

Appendix B

Change of Numeraire

Definition B.0.1. A numeraire is any strictly positive $(\mathcal{F}_t)_{t \in \mathbb{R}}$ -adapted stochastic process $(N_t)_{t \in \mathbb{R}}$ that can be taken as a unit of reference when pricing an asset or a claim. In general, the price A_t of an asset when quoted in terms of numeraire N_t is given by

$$\hat{S}_t := \frac{S_t}{N_t}.$$

In order to perform a fair pricing, one has to determine a probability measure under which the transformed process \hat{S}_t will be a martingale (Privault, 2022)

Assumption B.0.1. for a numeraire N_t , the discounted value

$$t \mapsto M_t := e^{-\int_0^t r_s ds} N_t$$

is an \mathcal{F}_t -martingale under the risk-neutral probability \mathbb{Q}

Definition B.0.2. Given $(N_t)_{t \in [0, T]}$ a numeraire process, the associated forward measure \mathbb{P}^T is defined by its Radon-Nikodym density

$$\frac{d\mathbb{Q}^T}{d\mathbb{Q}} := \frac{M_T}{M_0} = e^{-\int_0^T r_s ds} \frac{N_T}{N_0}.$$

Equivalently, \forall integrable \mathcal{F}_T -measurable random variable F

$$\begin{aligned} \mathbb{E}^{\mathbb{Q}^T}[F] &= \int_{\Omega} F(w) d\mathbb{Q}^T(w) \\ &= \int_{\Omega} F(w) e^{-\int_0^T r_s ds} \frac{N_T}{N_0} d\mathbb{Q}(w) \\ &= \mathbb{E} \left[F e^{-\int_0^T r_s ds} \frac{N_T}{N_0} \right]. \end{aligned}$$

Since the process $(M_t)_t$ is an \mathcal{F}_t -martingale under \mathbb{Q} , we find that

$$\begin{aligned} \mathbb{E} \left[\frac{d\mathbb{Q}^T}{d\mathbb{Q}} \middle| \mathcal{F}_t \right] &= \mathbb{E} \left[\frac{N_T}{N_0} e^{-\int_0^T r_s ds} \middle| \mathcal{F}_t \right] \\ &= \mathbb{E} \left[\frac{M_T}{M_0} \middle| \mathcal{F}_t \right] \\ &= \frac{M_t}{M_0} = \frac{N_t}{N_0} e^{-\int_0^t r_s ds}. \end{aligned}$$

Similarity, the radon-Nikody density $\frac{d\mathbb{Q}_{\mathcal{F}_t}^T}{d\mathbb{Q}_{/\mathcal{F}_t}}$ satisfies

$$\begin{aligned}\mathbb{E}^{\mathbb{Q}^T}[F/\mathcal{F}_t] &= \int_{\Omega} F(\omega) d\mathbb{Q}_{/\mathcal{F}_t}^T(\omega) \\ &= \int_{\Omega} F(\omega) e^{-\int_0^T r_s ds} \frac{d\mathbb{Q}_{/\mathcal{F}_t}^T}{d\mathbb{Q}_{/\mathcal{F}_t}} d\mathbb{Q}_{/\mathcal{F}_t}(\omega) \\ &= \mathbb{E} \left[F \frac{d\mathbb{Q}_{/\mathcal{F}_t}^T}{d\mathbb{Q}_{/\mathcal{F}_t}} \mid \mathcal{F}_t \right].\end{aligned}$$

for all \mathcal{F}_T measurable random variable.

Lemma

$$\frac{d\mathbb{Q}_{/\mathcal{F}_t}^T}{d\mathbb{Q}_{/\mathcal{F}_t}} = \frac{M_T}{M_t} = e^{-\int_t^T r_s ds} \frac{N_T}{N_t}.$$

Proposition

let $(X_t)_t$ denote a continuous $(\mathcal{F}_t)_t$ adapted asset price process such that: $t \mapsto \int e^{-\int_0^t r_s ds} X_t$ is a martingale under \mathbb{Q} .

Then, under the change of numeraire, the deflated process $(\hat{X}_t)_t = \left(\frac{X_t}{N_t}\right)_t$ of forward prices is an \mathcal{F}_t -martingale under \mathbb{Q}^T .

$$\mathbb{E} \left[\frac{X_t}{N_t} / \mathcal{F}_s \right] = \frac{X_s}{N_s}.$$

Each application of the change of numeraire will require to:

1. pick a suitable numeraire $(N_t)_t$ satisfying assumption B.0.1,
2. make sure that the ratio $\frac{C}{N_T}$ takes a sufficiently simple form,
3. use the Girsanov theorem in order to determine the dynamics of asset prices under the new probability measure \mathbb{Q}^T in order to compute the expectation under \mathbb{Q} .

$$\mathbb{E} \left[e^{-\int_t^T r_s ds} C / \mathcal{F}_t \right] = N_t \mathbb{E}^{\mathbb{Q}^T} \left[\frac{C}{N_T} / \mathcal{F}_t \right].$$

Appendix C

Bond Options Pricing

The calibration of an LGM-1F model is done on diagonal swaptions. One has to fit the prices of set of diagonal swaptions under LGM 1F with the market prices. Therefore, we need to express the swaptions' price according to LGM framework. To do so, we make use of the bond options pricing.

C.1 Bond Option

We define the call bond option as the option whose payoff in T is:

$$\left(\sum_{i=1}^n c_i B(T, T_i) - K \right)^+,$$

with

$$K > 0, \quad T < T_1 < \dots < T_n \quad \text{and} \quad \forall i \in \{1, \dots, n\}, c_i > 0.$$

Denote by \mathbb{Q}^T the T -forward measure. The price in $t = 0$ of the call bond option can be written as

$$P_{\text{CBO}}(T, K, (T_i)_{1 \dots n}, (c_i)_{1 \dots n}) = B(0, T) \mathbb{E}^{\mathbb{Q}^T} \left[\left(\sum_{i=1}^n c_i B(T, T_i) - K \right)^+ \right], \quad (\text{C.1.1})$$

where $B(T, T_i) = B_{T, T_i}(X_T)$. As we can see from equation 1.3.3, for all i , B_{T, T_i} is strictly decreasing, so that it is a bijection between \mathbb{R} and $]0, +\infty[$. Considering that all c_i 's are positive, we have the same property for the function $x \mapsto \sum_{i=1}^n c_i B_{T, T_i}(x)$. Chaix (2003) As a consequence, there exists a unique $x_0 \in \mathbb{R}$ such that

$$\sum_{i=1}^n c_i B_{T, T_i}(x_0) = K,$$

x_0 corresponds to the exercise frontier of our bond option. Now define

$$K_i = B_{T, T_i}(x_0),$$

and consider the product paying in T :

$$\sum_{i=1}^n c_i (B(T, T_i) - K_i)^+.$$

By construction, the n call options in C.1.1 all have the same exercise domain - defined by $\{X_T < x_0\}$ - as the call bond option. When exercise occurs we see that the payoff of the call bond option and the payoff C.1.1 coincide. In one word, we just proved the equality:

$$\left(\sum_{i=1}^n c_i B(T, T_i) - K \right)^+ = \sum_{i=1}^n c_i (B(T, T_i) - K_i)^+.$$

As soon as x_0 is determined, our call bond option can be evaluated as a sum of BS calls since zero-coupon bonds are lognormal. More precisely:

$$\begin{aligned} P_{\text{CBO}}(T, K, (T_i)_{1\dots n}, (c_i)_{1\dots n}) &= B(0, T) \mathbb{E}^{\mathbb{Q}^T} \left[\sum_{i=1}^n c_i (B(T, T_i) - K_i)^+ \right] \\ &= B(0, T) \sum_{i=1}^n c_i \mathbb{E}^{\mathbb{Q}^T} \left[\left(\frac{B(T, T_i)}{B(T, T)} - K_i \right)^+ \right]. \end{aligned}$$

For all $i \in \{1, \dots, n\}$, $B(\cdot, T_i) / B(\cdot, T)$ is a \mathbb{Q}^T -martingale. In the one factor LGM framework, its instantaneous volatility equals

$$\Gamma(t, T_i) - \Gamma(t, T) = \frac{\sigma(t)}{\lambda} e^{-\lambda(T-t)} \left[e^{-\lambda(T_i-T)} - 1 \right],$$

so that the integrated volatility σ_i^{bs} to take as input of BS formula is defined by

$$\begin{aligned} T \left(\sigma_i^{bs} \right)^2 &= \int_0^T |\Gamma(t, T_i) - \Gamma(t, T)|^2 dt \\ &= \left(\frac{e^{-\lambda(T_i-T)} - 1}{\lambda} \right)^2 \int_0^T \sigma(t)^2 e^{-2\lambda(T-t)} dt \\ &= \beta(T, T_i)^2 \phi(T). \end{aligned}$$

The price of the call bond option is then obtained as:

$$P_{\text{CBO}}(T, K, (T_i)_{1\dots n}, (c_i)_{1\dots n}) = B(0, T) \sum_{i=1}^n c_i \text{BScall} \left(F_i, 0, \sigma_i^{bs}, K_i, T \right),$$

where

$$F_i = \frac{B(0, T_i)}{B(0, T)} \quad \text{and} \quad \sigma_i^{bs} = \frac{1}{\sqrt{T}} \beta(T, T_i) \sqrt{\phi(T)},$$

The case of the put bond option is treated similarly:

$$P_{\text{PBO}}(T, K, (T_i)_{1\dots n}, (c_i)_{1\dots n}) = B(0, T) \sum_{i=1}^n c_i \text{BSput} \left(F_i, 0, \sigma_i^{bs}, K_i, T \right).$$

C.2 Modified Bond Option

The aim of this section is to introduce price expression of a modified bond option since a swaption can be considered as particular modified options.

A slight modification of the bond option has to be introduced to take properly into account the time lag between the expiry dates and the start dates of calibration instruments. (Chaix, 2003)

We define the modified call bond option as the option whose payoff in T is

$$\left(\sum_{i=1}^n c_i B(T, T_i) - K B(T, T_0) \right)^+,$$

where

$$K > 0, \quad T \leq T_0 < T_1 < \dots < T_n \quad \text{and} \quad \forall i \in \{1, \dots, n\}, c_i > 0.$$

Note that this option coincides with a "classical" bond option if $T_0 = T$. As before:

$$\tilde{P}_{\text{CBO}}(T, T_0, K, (T_i)_{1\dots n}, (c_i)_{1\dots n}) = B(0, T) \mathbb{E}^{\mathbb{Q}^T} \left[\left(\sum_{i=1}^n c_i B(T, T_i) - K B(T, T_0) \right)^+ \right].$$

Using, the T_0 -forward measure, the price can be rewritten as

$$\tilde{P}_{\text{CBO}}(T, T_0, K, (T_i)_{1\dots n}, (c_i)_{1\dots n}) = B(0, T_0) \mathbb{E}^{\mathbb{Q}^{T_0}} \left[\left(\sum_{i=1}^n c_i \frac{B(T, T_i)}{B(T, T_0)} - K \right)^+ \right].$$

As for all $i \geq 1$, $x \mapsto B_{T, T_i}(x)/B_{T, T_0}(x)$ is monotonic, we can proceed exactly as for the standard bond option. x_0 and K_i are now defined by the relations

$$\sum_{i=1}^n c_i \frac{B_{T, T_i}(x_0)}{B_{T, T_0}(x_0)} = K \quad \text{and} \quad K_i = \frac{B_{T, T_i}(x_0)}{B_{T, T_0}(x_0)}.$$

The volatility parameter that enters in BS formulas is changed, since the martingale underlying of option i is now $B(t, T_i)/B(t, T_0)$. Consequently, σ_i^{bs} is now such that:

$$\begin{aligned} T \left(\sigma_i^{bs} \right)^2 &= \int_0^T |\Gamma(t, T_i) - \Gamma(t, T_0)|^2 dt \\ &= \left(\frac{e^{-\lambda(T_0-T)} - e^{-\lambda(T_i-T)}}{\lambda} \right)^2 \phi(T). \end{aligned}$$

Finally:

$$\tilde{P}_{\text{CBO}}(T, T_0, K, (T_i)_{1 \dots n}, (c_i)_{1 \dots n}) = B(0, T_0) \sum_{i=1}^n c_i \text{BScall}\left(F_i, 0, \sigma_i^{bs}, K_i, T\right).$$

where

$$F_i = \frac{B(0, T_i)}{B(0, T_0)} \quad \text{and} \quad \sigma_i^{bs} = \frac{e^{-\lambda(T_0-T)} - e^{-\lambda(T_i-T)}}{\lambda\sqrt{T}} \sqrt{\phi(T)}.$$

The case of the put bond option is treated similarly:

$$\tilde{P}_{\text{PBO}}(T, T_0, K, (T_i)_{1 \dots n}, (c_i)_{1 \dots n}) = B(0, T_0) \sum_{i=1}^n c_i \text{BSput}\left(F_i, 0, \sigma_i^{bs}, K_i, T\right).$$


Note that for Black-Scholes dynamics

$$\sigma = \begin{cases} \frac{dX_t}{X_t} = \mu dt + \sigma dW_t \\ X_0 = x, \end{cases} \quad (\text{C.2.1})$$

we define

$$\begin{aligned} \text{BScall}(x, \mu, \sigma, K, T) &= \mathbb{E}[(X_T - K)^+] , \\ \text{BSput}(x, \mu, \sigma, K, T) &= \mathbb{E}[(K - X_T)^+] . \end{aligned}$$

Bibliography

- Allan Clark (2019). Calculation and drivers of the credit valuation adjustment for corporate treasurers. Available at <https://www.gbm.hsbc.com/insights/markets/calculations-and-drivers-of-the-credit-valuation-adjustment>.
- Basel Committee on Banking Supervision (2017). Basel iii: international regulatory framework for banks. Available at <https://www.bis.org/bcbs/basel3.htm?m=2572>.
- Basel Committee on Banking Supervision (2020). Calculation of rwa for market risk. Available at https://www.bis.org/basel_framework/chapter/MAR/10.htm.
- Basel Committee on Banking Supervision (2023). Calculation of RWA for credit risk.
- Brigo, D. and F. Mercurio (2001). *Interest-Rate Models: Theory and Practice*. Springer Finance.
- Chaix, A. (2003). One factor linear Gauss Markov. *Natixis's internal document*.
- European Banking Authority, A. . Capital requirements regulation (crr). Available at <https://www.eba.europa.eu/regulation-and-policy/single-rulebook/interactive-single-rulebook/101153>. 
- Lehdili, N., P. Oswald, and H. Gueneau (2019). Market risk assessment of a trading book using statistical and machine learning.
- M. Dupuy. (1948). Le calcul numérique des fonctions par l'interpolation barycentrique. *Comptes Rendus de l'Académie des Sciences. Série I, Mathématique*.
- on Banking Supervision, B. C. (2015). Review of the credit valuation adjustment risk framework.
- Privault, N. (2022). *Introduction to Stochastic Finance with Market Examples*. Chapman and Hall.
- Roux, A. (2008). Hull&White-1F : Pricing des swaptions Bermuda à nominal variable. *Natixis's internal document*.
- Ruiz, I. and M. Zeron (2021). *Machine Learning for Risk Calculations: A Practitioner's View*. John Wiley & Sons.
- Zeron, M. and I. Ruiz (2021). Denting the FRTB-IMA computational challenge via orthogonal chebyshev sliding technique. *Wilmott 2021*(111), 74–93.