Leveraging Bayesian Quadrature for Accurate and Fast CVA Calculations

N. Lehdili[*], P. Oswald[†], A. Azammam[‡] & O. Mirinioui[§]

June 26, 2024

Abstract

Counterparty risk, which combines market and credit risks, gained significant attention after the 2008 financial crisis due to its complexity and systemic implications. Traditional methods for managing this risk, such as netting agreements and collateralization, are computationally intensive under new regulatory frameworks like the Fundamental Review of the Trading Book (FRTB). This paper investigates the application of advanced machine learning techniques to enhance the efficiency and accuracy of counterparty risk metrics calculations, particularly Credit Valuation Adjustment (CVA). We focus on techniques interpolations and Gaussian processes. Gaussian process regression offers rapid and stable evaluations, reducing computational burdens significantly. Gaussian processes, through Bayesian quadrature and optimization, efficiently estimate risk metrics with minimal observations. These methods provide a promising balance between precision and computational efficiency, facilitating more effective counterparty risk management in the financial industry.

Keywords: Credit Valuation Adjustment, Expected Exposure, Basel III, FRTB, Potential Future Exposure, Gaussian processes regression, machine learning, Interest Rate Swaps.

Mathematics Subject Classification: 91G20, 91B05, 62G08, 60G15, 65D05.

1 Introduction

Counterparty risk is one of the most complex financial risk to measure and to manage, as it arises from the interaction of the market risk and credit risk and is also sensitive to systemic characteristics, such as the failure of large institutions ([10],[11],[9]). This especially concerns the credit risk associated with OTC derivatives counterparties and this risk has gained more importance since the global financial crisis ([17],[15], [18]). Before the financial crisis of 2008, many financial institutions managed their counterparty credit risk by dealing

[*] noureddine.lehdili@natixis.com. Expert Leader Market & Counterparty Risks Modelling.

[†] pascal.oswald@natixis.com. Expert Leader Market & Counterparty Risks Modelling.

[‡] othmane.mirinioui-ext@natixis.com

[§] amine.azammam-ext@natixis.com

only with the most robust counterparties, often relying on the perceived solvency of "too big to fail" institutions. However, the financial crisis highlighted that these same entities often represent the highest counterparty risk. Counterparty risk stems from the combination of market risk, which determines exposure, and credit risk, which assesses the credit quality of the counterparty. It is not always obvious whether a counterparty with a high probability of default and low exposure is preferable to one with higher exposure but a lower underlying default probability. Credit Valuation Adjustment (CVA) provides a precise value for counterparty risk and enables numerical distinction between these different cases. It evaluates the counterparty risk that an institution takes and potentially allows it to be traded or to be hedged ([20],[9],[12]). Various strategies are employed to mitigate counterparty risk in OTC derivatives.Netting agreements involve legally enforceable arrangements that allow offsetting of positive and negative exposures between counterparties. By consolidating multiple transactions into a single net position, netting reduces the overall credit exposure between counterparties. In the event of default, netting agreements help minimize the amount owed or payable between parties, thereby mitigating counterparty credit risk. Collateralization involves requiring counterparties to post collateral, providing a buffer against potential losses. Credit Support Annex (CSA, [21]) outlines terms for collateralization and margin requirements, further securing transactions. Additionally, hedging with credit derivatives, such as credit default swaps (CDS), transfers risk to third parties, safeguarding against defaults. These approaches collectively enhance risk management and financial stability.

Counterparty risk metrics must be assessed for economic, accounting, and regulatory purposes. Each of these purposes corresponds to a set of assumptions, mathematical, and numerical methods. The economic approach enables financial institutions to define exposure limits for different activities generating counterparty risk. Generally, this involves calculating Potential Future Exposure (PFE). The accounting approach involves directly incorporating adjustments of values (Cross Value Adjustments - XVA -), notably for the expected loss risk related to the deterioration of the counterparty's credit quality (Credit Value Adjustment - CVA -) into fair pricing. Finally, the regulatory approach allows evaluating the capital needed to cover unexpected losses related to i) counterparty defaults (for approved banks, this involves calculating Effective Expected Positive Exposure - EEPE -) ii) or CVA variability through VaR-CVA. For these three approaches, exposure is a common denominator. A true cornerstone of counterparty risk, its rapid and accurate estimation is of paramount importance. Banks use classical approaches that combine Monte Carlo simulations and pricing libraries to perform numerous repricings of all positions in the portfolio (from vanilla to exotic) on a multi-dimensional grid. A temporal dimension includes a time discretization between the exposure calculation date and the longest maturity of instruments in the portfolio. Spatial dimensions contain multivariate realizations of the (thousands of) risk factors disseminated by the bank, necessary for evaluating price functions at all times. This industrial context reflects the complexity that investment banks face in counterparty risk management, naturally leading them to seek avenues for improving

Under the purview of new regulations([17],[18],[15],[14]), the application of the Internal Method Approach (IMA) to evaluate risks such as Credit Counterparty Risk (CCR), Credit Valuation Adjustment (CVA), Value at Risk (VaR) CVA, Default Risk Charge (DRC), and Incremental Risk Charge (IRC) entails intricate and potentially time-consuming processes. For instance, the computation of capital under the Fundamental Review of the Trading Book regulation (FRTB, [14]) mandates the utilization of market risk metrics with varying liquidity horizons within the Internal Method Approach (IMA). This necessitates the pricing of portfolios across numerous scenarios, presenting a significant computational burden. The primary challenge for financial institutions lies in achieving accurate risk assessments while maintaining reasonable computation times. This challenge is equally pronounced in the computation of counterparty credit risk metrics, wherein multiple future pricing iterations are essential to determine exposures, fundamental components of many CCR metrics. While accuracy in calculations is imperative, it often comes at the cost of extended computation times, which can be exorbitant in some instances.

Traditionally, to address the computational complexity of financial risk models, financial institutions have opted for proxies. These approximations largely stem from the challenge of revaluing the derivatives portfolio for numerous different scenarios within a limited timeframe. In recent years, we have seen increasing adoption of machine learning methods in quantitative finance ([22],[6],[5],[13]). Indeed, many applications of machine learning have been implemented, and many are still being explored ([4], [7], [3],[8]). This includes derivatives pricing and hedgin, portfolio management, fraud detection, credit scoring, and more. The application of machine learning in financial risk management also represents a new avenue of research and could effectively implement financial risk models. Many studies are currently underway, and even more are yet to be discovered. One of the most promising research avenues explores the use of Gaussian Processes, Bayesian Optimization and Chebyshev interpolants technique in quantitative finance.

In this paper we will focus specifically on the application of these techniques for counterparty credit risk metrics calculations to adress the following problems : How can we efficiently implement these risk metrics, reducing computational time while ensuring satisfactory accuracy? In academic and professional literature, one finds the use of deep learning and Gaussian process regression to accurately and swiftly value derivatives. Another potential application involves leveraging Gaussian processes to estimate Value at Risk and Expected Shortfall for market risk measurement. Additionally, other machine learning applications, such as those based on neural networks and Chebyshev tensors, are utilized to compute risk metrics. In this study, we initially focus on a technique based on Chebyshev interpolations, which possess two unique mathematical properties. Firstly, given an analytical target function, the approximated interpolant converges exponentially towards the target. Secondly, the interpolation ensures a rapid and numerically stable evaluation. These characteristics render Chebyshev interpolation a distinctive pricing function approximator, significantly reducing the computational burden associated with risk calculations for portfolio evaluation across multiple scenarios. Subsequently, we explore the use of Gaussian processes via Bayesian quadrature (which also incorporates Bayesian optimization) to compute a numerical integral with very few observations. Both approaches will be applied to compute CCR metrics and Credit Valuation Adjustments optimally, striking an excellent balance between precision and computation time.

2 Counterparty Credit Risk Measures

In this section, we'll go over the basic concept and notations for counterparty credit risk (CCR), credit exposures, and Credit Valuation Adjustment (CVA). Counterparty Credit Risk (CCR) is the risk that the counterparty defaults or might not be able to pay what they owe before a derivative's payments are finalized. A financial loss occurs if the counterparty derivative has a positive economic value for the financial institutions at the time of default. Unlike a loan, where only the lender might lose out, Counterparty Credit Risk affects both counterparties: the market value could be positive or negative for either counterparty, and it might change over time due to market conditions. We define the counterparty exposure $E(t)$ of the financial institution at time t as the financial loss it could face on all the outstanding derivative transactions with the other counterparty if they can't pay at that time, considering netting and collateral but not taking into account any potential recoveries.

2.1 Expected Exposure

Consider a portfolio of N derivatives transactions of a financial institutions with a given counterparty. The counterparty defaults at a random time τ with a known risk-neutral distribution $P(t) = \mathbf{P}[\tau \leq t]$. Denote the value of the ith derivatives in the portfolio at time t by the financial insitution perspective by $V_i(t)$. The exposure corresponds to the value of all derivatives contracts of the financial institution that it has in common with a counterparty, which will lose if the latter defaults before maturity T (The maturity of the longest contract in the portfolio). At each time t, the exposure $\mathbf{E}(t)$ is determined by the values of all trades with the counterparty $\{\mathbf{V}_i(t)\}_{i=1}^N$. The value of the counterparty portfolio at time t is given by $\mathbf{V}(t) = \sum_{i=1}^N \mathbf{V}_i(t)$. If the netting is not allowed, the exposure $\mathbf{E}(t)$ is expressed as follows:

$$E(t,T) = \sum_{i=1}^{N} \max\{\mathbf{V_i(t)}, \mathbf{0}\}, \tag{1}$$

A netting agreement and a margin agreement are both important components in managing counterparty credit risk for derivatives. A netting agreement allows parties involved in multiple financial transactions, such as derivatives or securities trades, to offset the value of their obligations to each other. It enables the consolidation of all transactions between tow counterparties into one net amount, which reduces credit exposure. Netting agreements can take various forms, such a bilateral netting agreements which involves two parties, or multilateral netting agreements, which involve multiple parties. For a counterparty portfolio With a single netting agreement, the exposure is given by the following formula:

$$E(t,T) = \max\{\mathbf{V}(t), 0\}. \tag{2}$$

A margin agreement establishes the terms and conditions under which margin (collateral) is posted to cover potential losses arising from changes in the value of financial instruments, such as derivatives. Parties involved in derivatives transactions of exchange collateral to mitigate counterparty credit risk. Margin agreements specify the types of collateral that can be posted, the margin calculation methodology, margin call procedures, and dispute resolution mechanisms. The margin requirements may vary depending on factors such as the creditworthiness of the counterparties, the volatility of the underlying assets, and regulatory requirements.In the case where the netting agreement is further supported by a margin agreement, the counterparty must provide the bank with the collateral $\mathbf{C}(t)$ and we then have

$$E(t,T) = \max\{\mathbf{V}(t) - C(t), 0\}. \tag{3}$$

Credit exposure is pertinent only in the event of counterparty default. Therefore, the assessment of exposure should be contingent upon this occurrence. in the sequel, we will consider exposure regardless of default events, implicitly assuming no presence of "wrong-way risk". The

Expected Exposure (EE) defines the expected value knowing the mark-to-market (MtM) of the derivatives portfolio is positive so it represents the average of only the positive MtM values in the future. The expected exposure, calculated under the risk-neutral measure, is given as follows:

$$\mathbf{EE}(u,T) = \mathbf{E}^{Q}[E(u,T)]. \tag{4}$$

2.2 Credit Valuation Adjutment (CVA)

The credit valuation adjustment (CVA) is an important metric for financial institutions to accurately assess and manage counterparty credit risk, especially in the trading of derivatives where exposures can be sgnificant. It helps banks price derivatives transactions appropriately and allocate capital efficiently, ultimately contributing to the stability of financial markets. CVA is, by definition, the difference between the risk-free derivatives portfolio value and the true derivatives portfolio value that takes into account the counterparty's default. CVA is typically calculated as the present value of expected losses from counterparty default, discounted at the risk-free rate. It involves modeling the probability of default of the counterparty, the loss given default (LGD), and the exposure at default (EAD). Assuming no wrong-way risk and that the discount factors and survival probabilities are deterministics, the CVA can be expressed in the following way:

$$\mathbf{CVA}(t,T) = (1-\mathbf{R})\mathbf{E}^{Q}\left[\int_t^T D(t,u)\mathbf{EE}(t,u)dS(t,u)\right]. \tag{5}$$

This pricing formula can be computed using the following integration scheme:

$$\mathbf{CVA}(t,T) = (1-\mathbf{R})\sum_{i=1}^{m} D(t,t_i)\mathbf{EE}(t,t_i)\left[S(t,t_{i-1}) - S(t,t_i)\right]. \tag{6}$$

where we have m periods given by $t = t_0, t_1, t_2, ..., t_m = T$. We can observe that CVA depends on the following components:

- *Default Probability* (**PD**). For a non-constant intensity of default λ, the default probability and the survival probability are given by

$$P(t,T) = 1 - S(t,u) = 1 - \exp\left[-\int_t^u \lambda(x)dx\right]. \tag{7}$$

- *Loss Given Default* (**LGD**) : The portion of the exposure that is not recoverable in the default event. It represents the magnitute of losses in case of default and its value is unity less the recovery rate, $1-R$.

- *Exposure at Default* (**EAD**) : The expected value of the exposure to the counrerparty at the time of the default τ. This is given by $\mathbf{EE}(t,\tau)$.

- *Discount factors* : the expression $D(t,u)$ gives the risk-free discount factor at time t for a cash flow paid at time u.

CVA is influenced by various factors including the creditworthiness of the counterparty, market conditions, the volatility of the underlying assets, and the maturity of financial instruments.

3 Bayesian inference modeling

This section is meant to introduce to numerical integration from the view of Bayesian inference. We first give the toolset to consider basic the probabilistic integration algorithm. We consider then the problem of computing the integral over a function $f : \mathcal{X} \longrightarrow \mathbb{R}$ over a domain $\mathcal{X} \subset \mathbb{R}^d$ against a measure $p(x)$ on $\mathcal{X}$,

$$\mathbf{I} = \int_{\mathcal{X}} f(x) dp(x). \tag{8}$$

When no closed-form solution to this integral is available, numerical methods are needed. Numerical integration schemes seek to approximate the integral (22) by weighted sum of function evaluations called a quadrature rule :

$$\mathcal{Q}(f, \mathbf{X}, \omega) = \sum_{i=1}^{N} \omega_i f(x_i) = \omega^{\perp} f_{\mathbf{X}} \tag{9}$$

with real weights $\omega = [\omega_1, ..., \omega_N] \in \mathbb{R}^N$ and function evaluation $f_{\mathbf{X}} = [f(x_1), ..., f(x_N)]^{\perp} \in \mathbb{R}^N$ at design points $\mathbf{X} = \{x_1, ..., x_N\} \subset \mathcal{X}$. Roughly put, numerical integration is concerned with finding good weights, good design points, or both, in order to approximate (12).

3.1 Gaussian process regression

The marginalization property of Gaussian asserts that a subset of normal random variables is also governed by a normal distribution. Specifically, the marginalized Gaussian distribution remains unaffected by the parameters of the variables that have been marginalized over. It is thus possible to consider continuous index sets $\mathcal{X}$ such that the variables indexed at any finite subset thereof follow a multivariate normal distribution. The generalization of the normal distribution from a finite to an infinite collection of Gaussian random variables is called a Gaussian Process (GP) (??). A Gaussian process is therefore appropriate for representing a random function denoted as $f : \mathcal{X} \longrightarrow \mathbb{R}$. Realizations of the GP are real-valued functions $f : \mathcal{X} \longrightarrow \mathbb{R}$ and $x \in \mathcal{X}$ are considered as arguments of f rather than indices. More precisely, let $\mathcal{X} \subset \mathbb{R}^d$ be a non-empty set. A random process $f : \mathcal{X} \subset \mathbb{R}^d$ is said to be distributed according to a Gaussian process $f \sim \mathcal{GP}(m, k)$ with mean function $m : \mathcal{X} \longrightarrow \mathbb{R}$ and covariance function $k :: \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$, if for any finite set of arguments $\mathbf{X} = \{x_1, ..., x_N\} \subset \mathcal{X}$ and any $N \in \mathbb{N}$, the vector $f_{\mathbf{X}} = [f(x_1), ..., f(x_N)]^{\perp} \in \mathbb{R}^N$ is a mutltivariate normal random variable that follows the distribution

$$f_{\mathbf{X}} \sim \mathcal{N}(m_{\mathbf{X}}, K_{\mathbf{XX}}) \tag{10}$$

with mean vector $[m_{\mathbf{X}}]_i = m(x_i)$ and covariance matrix $[K_{\mathbf{XX}}]_{ij} = k(x_i, x_j)$ for $i, j = 1, ..., N$.

This means that the random process $\mathbf{f_X}$ plays the rôle of the surrogate for the latent function f. Assuming $f \sim \mathcal{GP}(m, k)$, our prior belief, before seeing the data, about the values that f may take at $\mathbf{X} = \{x_1, ..., x_N\}$ is

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(x_1) \\ \vdots \\ m(x_n) \end{bmatrix}, \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_N) \\ \vdots & \ddots & \vdots \\ k(x_N, x_1) & \cdots & k(x_N, x_N) \end{bmatrix} \right)$$

The main function is thus the expected value of f under the Gaussian process, $m(x) = \mathbb{E}\left[f(x)\right]$, and similarly, the covariance function denotes the covariance of function values at inputs $x, x' \in \mathcal{X}$ as $k(x,x') = \mathbb{E}\left[\left(f(x) - m(x)\right)\left(f(x') - m(x')\right)\right]$.The covariance function $k(.,.)$ is called kernel function and it is a central concept in the analysis of Gaussian processes. The popular choice, in machine learning, of the kernel function is the squared exponential:

$$k(x,x') = \sigma_f^2 exp\left(\frac{-(x-x')^2}{2l^2}\right) \tag{11}$$

The covariance matrix K has larger values, for points that are closer to each other, and smaller values for points further apart. This because the points are correlated by the difference in their means and their variances. If they are highly correlated, then their means are almost same, and their covariance is high.

Gaussian processes (GPs) (Rasmussen and Williams, 2006) are a rich class of models, which place probability distributions directly on classes of functions. Essentially, the performance of these models hinges on selecting appropriate kernels. They are often favored as a practical prior for modeling functions in Bayesian inference problems such as regression or classification. In the context of Bayesian regression, the goal is to estimate the underlying latent function from observations

$$y_n = f(x_n) + \epsilon_n \quad \text{where} \quad \epsilon_n \sim \mathcal{N}(0, \sigma^2) \tag{12}$$

that are corrupted by independent and identically distributed Gaussian noise. The likelihood for one datum is $p(y_n|x_n) = \mathcal{N}(y_n, f(x_n), \sigma^2)$. Let's assume we have N data points, $y = y_1, ..., y_n$ at input locations $\mathbf{X} = x_1, ..., x_n$ summarized as data $\mathcal{D} = \{\mathbf{X}, y\}$. The regression task is to compute the posterior $p(\mathbf{f}|\mathcal{D})$, and to make predictions about function values at a new input locations $x^\star$. More precisely, consider a matrix X^* consisting of n^* test inputs, each of dimension d, and denote the corresponding unknown vector of function values by f^*. The posterior can be obtained through the conditioning rules for Gaussians. This requires the joint distribution of function values $\mathbf{f}^\star := \mathbf{f}(x^\star)$ at a new location $x^\star \in \mathcal{X}$ and observations y

$$\begin{pmatrix} y \\ f^* \end{pmatrix} \sim N\left(\begin{bmatrix} m \\ m(x^\star) \end{bmatrix}, \begin{bmatrix} K + \sigma^2 I & k(x^\star) \\ k(x^\star)^\perp & k(x^\star, x^\star) \end{bmatrix}\right) \tag{13}$$

where we have introduced the shorthand notation $[k(x^\star)]_i = k(x_i, x^\star)$ for the column vector of kernel evaluations at $\mathbf{X}$.

Note that $y = f(x)$ when the there is no noise ($\sigma^2 = 0$). This happens when we have clean observations of the underlying process f. We are of course interested in the conditional probability $p(f^*|X, f, y)$: given the data, how likely is a certain prediction for y^*. The mean is assumed to be $m(x) = m(x^{`\star} = 0$.According to equations **??** to **??**, the probability follows a Gaussian distribution:

$$f^*|y \sim N(k(x^\star)^\perp (K + \sigma^2 I)^{-1} y, k(x^\star, x^\star) - (K + \sigma^2 I) k(x^\star)^\perp) \tag{14}$$

Our best estimate for f^* is the posterior mean $E[f(x^*)|f(x)]$ of this distribution is given as follows:

$$\begin{aligned} \bar{f^*} &= E[f(x^*)|f(x)] \\ &= k(x^\star)^\perp (K + \sigma^2 I)^{-1} y \\ &= k(x^\star)^\perp (K + \sigma^2 I)^{-1} f(x) \end{aligned} \tag{15}$$

And the uncertainty in the Gaussian process regression is captured in its variance:

$$var(y^{*}) = k(x^{\star}, x^{\star}) - (K + \sigma^{2}I)k(x^{\star})^{\perp}) \tag{16}$$

The posterior mean $E[f(x^{*})|f(x)]$ can be interpreted in two ways. If the last the two terms of $(K + \sigma^{2}I)^{-1}f(x)$ are grouped, the posterior mean be expressed a linear combination of the kernel functions values, namely:

$$E[f(x^{*})|f(x)] = \sum_{i=1}^{n} \omega_{i} k(x_{i}, x^{*}) \quad \text{where} \quad \omega = (K + \sigma^{2}I)^{-1}f(x) \tag{17}$$

This means that the posterior mean can be computed without explicitly inverting the matrix K, by solving the equation $(K + \sigma^{2}I)\omega = f(x)$ instead. Similarly, by grouping the first two terms $k(x^{\star})^{\perp}(K + \sigma^{2}I)^{-1}$, the posterior mean can be written as a linear combination of the observed function values:

$$E[f(x^{*})|f(x)] = \sum_{i=1}^{n} \phi_{i} f(x_{i}) \quad \text{where} \quad \phi_{i} = k(x^{\star})^{\perp}(K + \sigma^{2}I)^{-1} \tag{18}$$

Now, if the prior mean function is no-zero, we can still use the previous derivation by noting that if $f \sim N(\mu, K(X, X))$, then function $h = f - \mu$ is a zero-mean Gaussian process if $h \sim N(0, K(X, X))$. Hence, if the observations are deducted from the values of f, we can subtract the prior mean function values to get observations of h, do the inference on h, and finally once we obtain the posterior on $h(x^{*})$, we can simply add back the prior mean $\mu(x^{*})$ to the posterior mean, to obtain the posterior on f.

3.2 Training Gaussian process algorithms

The practical implementation of the Gaussian process regression is dependent how well we choose the covariance function. For this, the parameters $l, \sigma_{n}, \sigma_{f}$ must appropriately selected. Set $\theta = \{l, \sigma_{n}, \sigma_{f}\}$ the parameters of the model. The usual way of selecting parameters is to maximize the log marginal likelihood function $l(\theta) = ln(p(y|\theta))$ given by:

$$\begin{aligned} l(\theta) &= ln(p(y|\theta) \\ &= -\frac{1}{2}y^{T}(K + \sigma_{n}^{2}I)^{-1} - \frac{1}{2}log(|K + \sigma_{n}^{2} + I|) - -\frac{n}{2}log(2\pi) \end{aligned} \tag{19}$$

The first term in the equation 19 can be interpreted as a data-fit term, the second term is a complexity penalty and the last term is a normalizing constant. The derivatives of the log marginal likelihood with respect to the hyperparameters are given by:

$$\begin{aligned} \frac{\partial l(\theta)}{\partial \theta_{j}} &= \frac{1}{2}y^{T}K^{-1}\frac{\partial K}{\partial \theta_{j}}K^{-1}y - \frac{1}{2}tr\left(K^{-1}\frac{\partial K}{\partial \theta_{j}}\right) \\ &= \frac{1}{2}tr\left(\omega\omega^{T} - K^{-1}\frac{\partial K}{\partial \theta_{j}}\right) \end{aligned} \tag{20}$$

where $\omega = K^{-1}y$. One drawback of Gaussian processes is that it scales very badly with the number of observations n. Indeed, the complexity of computing the marginal likelihood in the equation 19 is dominated by the need to invert the K matrix. The log determinant of

K is easily computed as by-product of the inverse. Standard methods for matrix inversion of positive definite symmetric matrices require time $0(n^3)$ for inversion of an n by n matrix. Once the matrix K^{-1} is determined, the computation of the derivatives in the equation 23 requires only time $0(n^2)$ per hyperparameter. Thus, the computational overhead of computing derivatives is small, so the optimization problem in the equation 19 can be easily solved using gradient-descent, conjugate gradient or quasi-Newton algorithms.

3.3 Bayesian Quadrature

Now that we have the Gaussian process regression as a modeling tool available, let's revisit the intractable integral (8). Bayesian quadrature is a model-based approach to approximately evaluating such integral by modeling f as a Gaussian process. Consequently, the deterministic function f is represented by a stochastic process $\mathbf{f}$ and the integral (8) is replaced by the random variable $\mathbf{I}$,

$$\mathbf{I} = \int_{\mathcal{X}} \mathbf{f}(x) dp(x). \tag{21}$$

As the integration is a linear functional, the integral of a Gaussian process takes a Gaussian form as well. It's then natural to consider a prior distribution for the integrand rather than for the integral itself, since quadrature rules (9) depend on pointwise evaluations $f(x_1), ..., f(x_N)$. For suitable kernel-prior combinations the quadrature weights can be evaluated analytically.

Despite the computational effort required to maintain the Gaussian process model, it enables the generalization of sample information across the integration domain. Consequently, samples can be chosen in a targeted fashion, and Bayesian quadrature emerges as a competitive and more efficient alternative in terms of evaluation compared to Monte Carlo methods. The choice of a Gaussian process prior $f \sim \mathcal{GP}(m, k)$ induces an univariate Gaussian prior on $\mathcal{I}$: $\mathcal{I} \sim \mathcal{N}(m, \nu)$ where

$$m := \mathbb{E}_f[\mathcal{I}] = \int_{\mathcal{X}} m(x) dp(x) \quad \text{and} \quad \nu := \mathbb{V}_f[\mathcal{I}] = \int_{\mathcal{X}} k(x, x') dp(x) dp(x') \tag{22}$$

Once the nodes and function evaluations $\mathcal{D} = \{\mathbf{X}, y\}$ are available, the inference step is nothing else but the Gaussian process regression (GPR), with the integral operator applied to the posterior$f|\mathcal{D}$. Bayesian quadrature hence provides a straightforward method for integrating noisy observations $y = f(x) + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma^2)$. The linearity property of GPs allows to write down the joint distribution of (y, I) as follows:

$$\begin{pmatrix} y \\ \mathbf{I} \end{pmatrix} \sim N \left(\begin{bmatrix} \mathbf{m} \\ m \end{bmatrix}, \begin{bmatrix} K + \sigma^2 I & \int_{\mathcal{X}} k(x) dp(x) \\ \int_{\mathcal{X}} k(x)^{\perp} dp(x) & \iint_{\mathcal{X}} k(x, x') dp(x) dp(x') \end{bmatrix} \right) \tag{23}$$

Conditioning on the observations $\mathbf{y}$ results in the posterior distribution $\mathbf{I}|\mathbf{y} \sim \mathcal{N}(\mathbf{m}_{\mathcal{D}}, \nu_{\mathcal{D}})$, characterized by its posterior mean $\mathbf{m}_{\mathcal{D}}$ and variance $\nu_{\mathcal{D}}$:

$$\begin{aligned} \mathbf{m}_{\mathcal{D}} &= \mathbb{E}_{f|\mathcal{D}}[\mathbf{I}] \\ &= \int_{\mathcal{X}} m(x) + k(x)^{\perp} (K + \sigma^2 I)^{-1} (y - m) dp(x) \\ &= \int_{\mathcal{X}} m(x) dp(x) + \sum_{i=1}^{N} \int_{\mathcal{X}} k(x, x_i) dp(x) \left[(K + \sigma^2 I)^{-1} (y - m) \right]_i \end{aligned} \tag{24}$$

and its variance $\nu_{\mathcal{D}}$

$$\begin{aligned}\nu_{\mathcal{D}} &= \mathbb{V}_{f|\mathcal{D}}[\mathbf{I}] \\ &= \iint_{\mathcal{X}} k(x,x') - k(x)^{\perp}(K+\sigma^2 I)^{-1}k(x)dp(x)dp(x') \\ &= \iint_{\mathcal{X}} k(x,x')dp(x)dp(x') - \sum_{i,j=1}^{N}\left[(K+\sigma^2 I)^{-1}\right]_{ij}\int_{\mathcal{X}} k(x,x_i)dp(x)\int_{\mathcal{X}} k(x_j,x')dp(x')\end{aligned} \tag{25}$$

It thus becomes clear that the effective numerical implementation of Bayesian quadrature essentially involves computing two integrals, namely the kernel mean $\kappa(x) = \int_{\mathcal{X}} k(x,x')dp(x')$ and the initial variance $\xi = \iint_{\mathcal{X}} k(x,x')dp(x)dp(x')$. Denoting the column vector $\kappa := \kappa(x)$ as the kernel mean evaluated at nodes $\mathbf{X}$, equations (33),(25) can be rewritten in the following shorthand way:

$$\mathbf{m}_{\mathcal{D}} = \mathrm{m} + \kappa^{\perp}(K+\sigma^2 I)^{-1}(y-\mathbf{m}) \quad \text{and} \quad \nu_{\mathcal{D}} = \xi - \kappa^{\perp}(K+\sigma^2 I)^{-1}\kappa \tag{26}$$

For a zero prior mean and noise-free observations, the expected value of $\mathbf{I}$ is

$$\mathbf{m}_{\mathcal{D}} = \kappa^{\perp}K^{-1}f = \sum_{i=1}^{N}\omega_i f(x_i) \quad \text{with} \quad \omega = K^{-1}\times\kappa \tag{27}$$

This is the classic form of a quadrature rule (??) that represents the integral as a weighted sum of function evaluations. Bayesian Quadrature replaces the integral (??) by integrals over the kernel. To gain a computational advantage, the kernel used for Bayesian Quadrature is typically chosen such that the kernel mean and initial variance are available in closed form.

Figure ?? shows the Gaussian prior and the resulting Gaussian distribution over the integral, as well as the updated posteriors on **f** and given three and eight exact evaluations of the integrand, respectively. Pseudocode for this procedure is given in Algorithm 2 above.

Algorithm 1. Bayesian quadrature

```
Input  : $f(.), \mathcal{GP}(m,k), \nu(.), \mathcal{D}\{\mathbf{X},y\}, \sigma^2$
Output: Bayesian quadrature estimation of the expected exposure EE and CVA
procedure BQ($f(.), \mathcal{GP}(m,k), \nu(.), \mathcal{D}\{\mathbf{X},y\}, \sigma^2$)
$\kappa \leftarrow \int_{\mathcal{X}} k(X,x)dp(x)$                    //Compute kernel mean
$\nu \leftarrow \iint_{\mathcal{X}} k(x,x')dp(x)dp(x')$               //Compute kernel variance
$m \leftarrow \iint_{\mathcal{X}} m(x)dp(x)$                          //Integrate prior mean mean
$\mathbf{m} \leftarrow m(\mathbf{X})$
$\omega \leftarrow (\mathbf{K}+\sigma^2\mathbf{I})^{-1}\kappa$         //Compute quadrature weights
$m_{\mathcal{D}} \leftarrow m + \omega^{\perp}(y-m)$                  //Compute Bayesian quadrature mean
$\nu_{\mathcal{D}} \leftarrow \nu - \omega^{\perp}\kappa$             //Compute kernel variance
$p(\mathbf{I}|y) \leftarrow \mathcal{N}(m_{\mathcal{D}}, \nu_{\mathcal{D}})$
return $p(\mathbf{I}|y)$
end procedure
```

Remark 1. *We define the canonical squared correlation for Bayesian quadrature as*

$$\rho^2_{\mathcal{D}}(X^{\star}) = \nu_D^{-1}\kappa^{\perp}_{\star|\mathcal{D}}C^{-1}_{\star|\mathcal{D}}\kappa_{\star|\mathcal{D}} \tag{28}$$

with the posterior variance of the integral $\mathbb{V}(I|\mathcal{D}) = \nu_{\mathcal{D}}$ and the posterior kernel mean evaluated at the new inputs

$$\kappa_{\star|\mathcal{D}} = \kappa(X^\star) - \kappa(X)\left(K + \sigma^2\mathbf{I}\right)^{-1} k(X, X^\star) \tag{29}$$

as well as the noise-corrected posterior covariance matrix of the new data points

$$C_{\star|\mathcal{D}} = K(X^\star, X^\star) - k(X^\star, X)\left(K + \sigma^2\mathbf{I}_N\right)^{-1} k(X, X^\star) + \sigma^2\mathbf{I}_N^\star. \tag{30}$$

with the $N \times N$ identity matrix $\mathbf{I}_N$ and the $N^\star \times N^\star$ version $\mathbf{I}_N^\star$. The coefficient $\rho^2_{\mathcal{D}}(X^\star)$ is a scalar quantity that measures the correlation between the integral $\mathbf{I}$ and new observation $\mathbf{y}^\star$.

Remark 2. *The entropy $H[x]$ and conditional entropy $H[x|y]$ refers to the expected information content of the random variable $\mathbf{x}$ and they are expressed as follows*

$$H[\mathbf{x}] = \int_{\mathcal{X}} p(x)\log p(x)dx \quad \textit{and} \quad H[\mathbf{x}|\mathbf{y}] = \int_{\mathcal{X}} p(x, y)\log p(x|y)dxdy \tag{31}$$

In the case of a Gaussian random variable $\mathbf{x} \sim \mathcal{N}(\mu, \Sigma)$ where $\mu \in \mathbb{R}^D$ and $\Sigma \in \mathbb{R}^{D\times\mathcal{D}}$ the entropy is

$$H[\mathbf{x}] = \frac{D}{2}\log(2\pi\exp) + \frac{1}{2}\log\det\Sigma. \tag{32}$$

3.4 Optimal design for Bayesian Quadrature

So far, we have assumed the nodes $\mathbf{X} := \{x_1, ..., x_n\}$ to be given for Bayesian quadrature. In the following, we will employ an approach based on Bayesian optimization and entropy to optimally select the nodes X for computing the integral $\mathbf{I}$. To this end, consider the quantity of interest is the integral $\mathbf{I}$, data $\mathcal{D} = \{X, y\}$ given and $\mathcal{D}^\star = \{X^\star, y^\star\}$ to be observed. The *mutual information* between the Gaussian integral $\mathbf{I}$ and new observations $y^\star$ at locations $X^\star$ is founded of the entropy of Gaussian Random variables and it is defined as follows:

$$\alpha_{\mathcal{D}}^{\mathbf{MI}} = -\frac{1}{2}\log\left(1 - \rho^2_{\mathcal{D}}(X^\star)\right) \tag{33}$$

It measures the expected amount of information gained about the integral $\mathbf{I}$ by observing the yet unseen $y^\star$, and vice versa.

It is widely known that in Gaussian models, information criteria yield a policy that maximally reduces the variance on the quantity of interest.There are multiple equivalent ways to phrase the reduction of integral variance that results in different functional forms of the acquisition function.The change of variance by evaluating at $\mathbf{X}^\star$ and observing $\mathbf{y}^\star$ relative to the variance at the current step is

$$\begin{aligned} \alpha_{\mathcal{D}}^{\mathbf{IVR}} &= \frac{\mathbb{V}[\mathbf{I}|\mathcal{D} - \mathbb{V}[\mathbf{I}|\mathcal{D} \cup \{\mathbf{X}^\star, \mathbf{y}^\star\}]}{\mathbb{V}[\mathbf{I}|\mathcal{D}]} \\ &= \rho^2_{\mathcal{D}}\left(X^\star\right) \end{aligned} \tag{34}$$

which is independent of upcoming functions evaluations $\mathbf{y}^\star$.

Algorithm 2. Sequential Bayesian quadrature

```
Input  : $f(.), \mathcal{GP}(m,k), \nu(.), \mathcal{D}\{\mathbf{X}, y\}, \sigma^2$
procedure BQ($f(.), \mathcal{GP}(m,k), \nu(.), \mathcal{D}\{\mathbf{X}, y\}, \sigma^2$)
  $\mathcal{D} = \{\}$                                              //initialize data
  for $n = 1 : N_{\max}$ do
    $x \leftarrow \text{argmax}_{x \in \mathcal{X}} \alpha(x)$                  //Optimize acquisition function
    $y \leftarrow f(x) + \epsilon$                                   //Evaluate integrand
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{x, y\}$                              //Evaluate integrand
    $\mathbf{f}_\theta \leftarrow \mathbf{f}_\theta | \mathcal{D}$                                    //condition GP on data
    $\theta \leftarrow \text{argmax}_{\theta \in \times} \log p(y|\theta)$           //Optimize acquisition function
    $\alpha \leftarrow \alpha_{\mathcal{D}}$                                     ////Optimize acquisition function
    $p(\mathbf{I}|y) \leftarrow BQ(f, \mathbf{f}_\theta, \nu, \mathcal{D})$                       //call BQ Algorithm 2
  end for
  return $p(\mathbf{I}|y)$
end procedure
```

4 CVA Calculation under One Factor Linear Gauss Markov

The derivatives used to carry out numerical tests, results of which we present in this section, are interest rate swaps, cross-currency swaps, and swaptions. Credit Counterpary Risk and CVA calculations are done using Monte Carlo simulations and market risk factors are diffused using One Factor Linear Gaussian Markov (LGM-1F) as diffusion model. In the LGM-1F framework, the dynamics of the zero-coupon bond $B(t, T)$ of a maturity T is a lognormal process under the risk neutral probability $\mathbb{Q}$. In this case the volatility of the zero-coupon bond is deterministic and it is given by $\Gamma(t, T) = \frac{\sigma}{\lambda}\left(\exp(-\lambda(T-t)) - 1\right)$ where σ is piecewise constant and λ is constant. For the LGM-1F assumption for interest rate modeling, derived from the HGM framework, has the adantage of simplicity and tractability. The dynamics of the yield curve is specified by the piecwise constant short term volatility σ and the mean reversion parameter λ and it is given as follows:

$$dX_t = \left[\phi(t) - \lambda X_t\right] dt + \sigma(t) dW_t \tag{35}$$

where

$$X_0 = 0 \quad \text{and} \quad \phi(t) = \int_0^t \sigma(s)^2 e^{-2\lambda(t-s)} ds \tag{36}$$

Notice that at any future date t, all the discount factors are obtained as deterministic functions of the the state variable X_t. This deterministic relation is referred to as the construction formula which is given as follows:

$$B(t, T) = \frac{B(0, T)}{B(0, t)} \exp\left(-\frac{1}{2}\beta(t, T)^2 \phi(t) - \beta(t, T) X_t\right) \quad \text{and} \quad \beta(t, T) = \frac{1 - \exp(-\lambda(T-t))}{\lambda} \tag{37}$$

Another attractive feature of the LGM-1F model is the straightforward evaluation of vanilla interest rate options: caplets and swaptions prices are obtained *via* quasi-analytical formula.

4.1 Swap and Swaption valuation with LGM-1F model

Consider a swap with expiry date T_e. Denote by $T_0 < T_1 < ... < T_n$ (with $T_e < T_0$) the settlement dates or the fixed leg (no flows exchanged in T_0). The fixed swap rate defined at date T_e equals

$$S(T_e, T_0, T_n) = \frac{B(T_e, T_0) - B(T_e, T_n)}{LVL(T_e, T_0, T_n)} \quad \text{where} \quad LVL(T_e, T_0, T_n) = \sum_{i=1}^{n} \delta_i B(T_e, T_i). \tag{38}$$

The parameter δ_i is the year fraction between T_{i-1} and T_i calculated in the adequate basis. A payer swaption of strike K written on the above swap is an option maturing in T_e with payoff:

$$\begin{aligned} LVL(T_e, T_0, T_n)(S(T_e, T_0, T_n) - K)^+ &= \left(B(T_e, T_0) - B(T_e, T_n) - \sum_{i=1}^{n} \delta_i K B(T_e, T_i) \right)^+ \\ &= \sum_{i=1}^{n} c_i \left(\frac{B(T_e, T_i)}{B(T_e, T_0)} - K_i \right)^+ \end{aligned} \tag{39}$$

where $c_i = \delta_i K$ for $i = 1, ..., n-1$, $c_n = 1 + \delta_n K$ and $K_i = \frac{B_{T_e,T_i}(x_0)}{B_{T_e,T_0}(x_0)}$1. Hence, its price in date $t = 0$ is given as follows:

$$\begin{aligned} V(T_e, T_0, T_n, K) =& B(0, T_e) \times \mathbb{E}^{\mathbb{Q}^{T_e}} \left[\left(B(T_e, T_0) - B(T_e, T_n) - \sum_{i=1}^{n} \delta_i K B(T_e, T_i) \right)^+ \right] \\ =& B(0, T_0) \times \sum_{i=1}^{n} c_i \times P_i^{bs}(F_i, 0, \sigma_i^{bs}, K_i, T) \end{aligned} \tag{40}$$

where P_i^{bs} denotes the Black & Scholes price of a call option, and F_i and σ_i^{bs} are given as follows

$$F_i = \frac{B(0, T_i)}{B(0, T_e)} \quad \text{and} \quad \sigma_i^{bs} = \frac{1}{\sqrt{T_e}} \beta(T_e, T_i) \sqrt{\phi(T_e)}. \tag{41}$$

4.2 Approximation of the Dynamics of a Swap Rate under LGM-1F

The pricing formula for European swaptions is quasi-closed because the calculation of the strikes K_i for $i = 1, \ldots, n$ requires numerically solving the zeros of a nonlinear function. This can be time-consuming if this formula is used repeatedly, such as in the case of calculating the CVA of an interest rate swap through Monte Carlo simulations. In the following, we will propose an alternative method for pricing a swaption based on an approximation of the dynamics of a swap rate within the framework of LGM-1F. To achieve this, we seek to calculate

[1] If we write $B(T_e, T_i) = B_{T_e,T_i}(X_{T_e})$, we can see from equation (37), for all i, $x \to \sum_{i=1}^{n} c_i \frac{B_{T_e,T_i}(x)}{B_{T_e,T_0}(x)}$ is strictly decreasing, so that it is a bijection between $\mathbb{R}$ and $]0, +\infty[$. As a consequence, there exists a unique $x_0 \in \mathbb{R}$ such that $\sum_{i=1}^{n} c_i \frac{B_{T_e,T_i}(x_0)}{B_{T_e,T_0}(x_0)} = 1$. By construction, the n call options in (39) all have the same exercise domain -defined by $X_{T_e} < x_0$ - as the payer swaption.

the differential $dS(t)$ using formula (38) by applying Itô's lemma and neglecting the terms present in the drift. [10][1] [2] [19] [16] [15] [18] [22]

$$dS(t) = \left[\frac{dB(t,T_0) - dB(t,T_n)}{\sum_{i=1}^{n} \delta_i B(t,T_i)} - \frac{B(t,T_0) - B(t,T_n)}{\left(\sum_{i=1}^{n} \delta_i B(t,T_i)\right)^2} \sum_{i=1}^{n} \delta_i dB(t,T_i)\right] + (...)dt \tag{42}$$

Using the assumptions made about the dynamics of zero-coupon rates within the framework of LGM-1F, we obtain the following expression for $\lambda \neq 0$:

$$dS(t) = \frac{\sigma(t)}{\lambda} e^{\lambda t} S(t) \left(\frac{B(t,T_0)e^{-\lambda T_0} - B(t,T_n)e^{-\lambda T_n}}{B(t,T_0) - B(t,T_n)} - \frac{\sum_{i=1}^{n} \delta_i B(t,T_i)e^{-\lambda T_i}}{\sum_{i=1}^{n} \delta_i B(t,T_i)}\right) dW^{\mathbb{Q}} + (...)dt \tag{43}$$

In order to simplify the notation, we introduce a function $g(.)$ such that:

$$dS(t) = \sigma(t) e^{\lambda t} g(t) S(t) dW^{\mathbb{Q}} + (...)dt \tag{44}$$

To obtain the desired approximations, we set aside the drift term and then freeze the expression of the function $g(t)S(t)$ at its initial value, i.e. $\forall t > 0, g(t)S(t) \approx g(0)S(0)$. We thus obtain the approximation of the normal dynamics of the interest rate $S(t)$:

$$dS(t) = \sigma(t) e^{\lambda t} g(0) S(0) dW^{\mathbb{Q}}. \tag{45}$$

In the following, we will only consider the case where $\lambda \neq 0$, but the equations and the function g can be adapted for $\lambda = 0$. The price of a swaption, in the case where the dynamics of the interest rate S(t)S(t) are governed by the normal process (45), can be obtained using the Bachelier formula as follows :

$$V(0,T_e) = LVL(0,T_e) \left[S(0)\sigma^{\star}\sqrt{T}\phi\left(\frac{K - S(0)}{S(0)\sigma^{\star}\sqrt{T}}\right) + (S(0) - K)\left(1 - N\left(\frac{K - S(0)}{S(0)\sigma^{\star}\sqrt{T - t}}\right)\right)\right] \tag{46}$$

where $\sigma^{\star} \triangleq \frac{\sigma g(0)}{\sqrt{(T)}}\sqrt{\frac{e^{2\lambda T} - 1}{2\lambda}}$, ϕ is the probability density function of the standard normal distribution, and N corresponds to its cumulative distribution function. One way to test the validity of the normality assumption of swap rates under the LGM-1F model used in the approximation is to compare the empirical distribution of a swap rate with the Gaussian distribution of swap rates at a given date t. To do this, we focus on the distribution of the swap rate at the date $T = 3Y$. This swap starts at $T_0 = 3Y$ and matures at $T_n = 10Y$. The payment frequency is semi-annual. We use 30,000 simulations. The empirical distribution is constructed based on 30 homogeneous classes. In the graphs above, the histogram corresponds to the empirical distribution obtained through Monte Carlo simulations. The curve represents the theoretical Gaussian distribution of the approximation. The distribution tests show that the Gaussian approximation of the swap rate dynamics is very close to the dynamics obtained within the LGM-1F framework. The quality of the approximation is stable for different levels of instantaneous volatility σ and mean-reversion λ. Therefore, the approximation seems more suitable for our modeling choice. Before presenting the method for calculating the expected exposure and the CVA for a portfolio of swaps and swaptions, we will do so for the case of a single swap and a single swaption since this can still be done using closed-form formula. Denote the risky value of the swaption with maturity T_e as $\tilde{V}(t,T_e)$ and the default time of

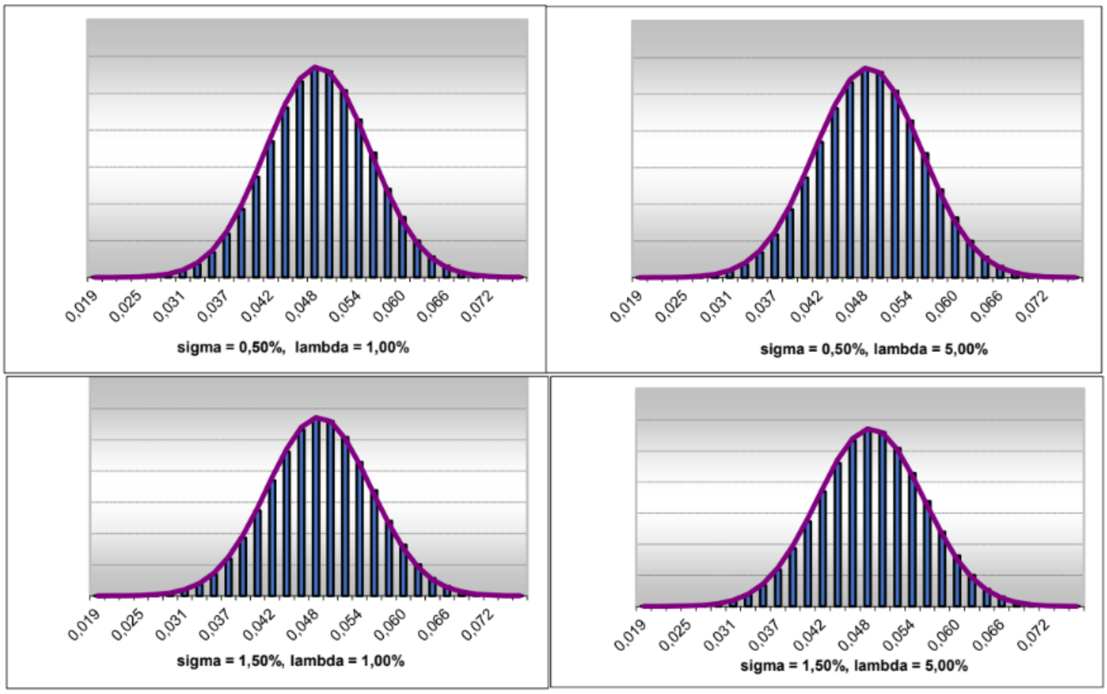

Figure 1: Comparison of the empirical distribution of a forward swap rate obtained with the LGM-1F model via Monte Carlo simulations and the theoretical distribution of the Gaussian approximation. This comparison is performed for different values of σ and λ.

the counterparty as τ. In this case, the risky value of the swaption is obtained by adjusting the risk-free value $V(t, T_e)$ in the following way:

$$\tilde{V}(t, T_e) = V(t, T_e) - \mathbf{CVA}(t, T_e) \tag{47}$$

where

$$\mathbf{CVA}(t, T_e) = (1 - R) \times \mathbb{E}^{\mathbb{Q}} \left[V(\tau, T_e)^+ \times 1_{\tau \leq T_e} \right] \tag{48}$$

As the exposure of the long swaption position can never be negative, the credit valuation adjustment (CVA) can be rewritten as follows:

$$\begin{aligned} \mathbf{CVA}(t, T_e) =& (1 - R) \times \mathbb{E}^{\mathbb{Q}} [V(\tau, T_e)] \times \mathbb{E}^{\mathbb{Q}} [1_{\tau \leq T_e}] \\ =& (1 - R) \times P(t, T) \times V(t, T_e) \end{aligned} \tag{49}$$

In the case where the exposure to the conterparty consists of a portfolio of long and short positions in swaptions, this can be negative and consequently, a closed-form formula for calculating the CVA is no longer available and its computation become challenging.

Now, consider the exposure to the conterparty is a payer interest rate swap with the tenor T. Notice that the expected exposure $\mathbf{EE}(t, t_i)$ of this interest rate swap at a future date t_i is nothing else but the value of the swaption with maturity t_i of a swap with tenor $T - t_i$. In this case, the CVA formula interest rate swap (6) can be expressed as follows:

$$\mathbf{CVA}(t, T) \approx (1 - \mathbf{R}) \sum_{i=1}^{m} [S(t, t_{i-1}) - S(t, t_i)] \times V(t, t_i, T). \tag{50}$$

As with the case of a swaption, the calculation of the CVA for an interest rate swap is done using a closed-form formula but based on the valuation of m swaptions (which corresponds approximately to $m = 2500$ swaption valuations for the calculation of the CVA of a 10-year swap, assuming a daily discretization step!),which can quickly become very time-consuming.

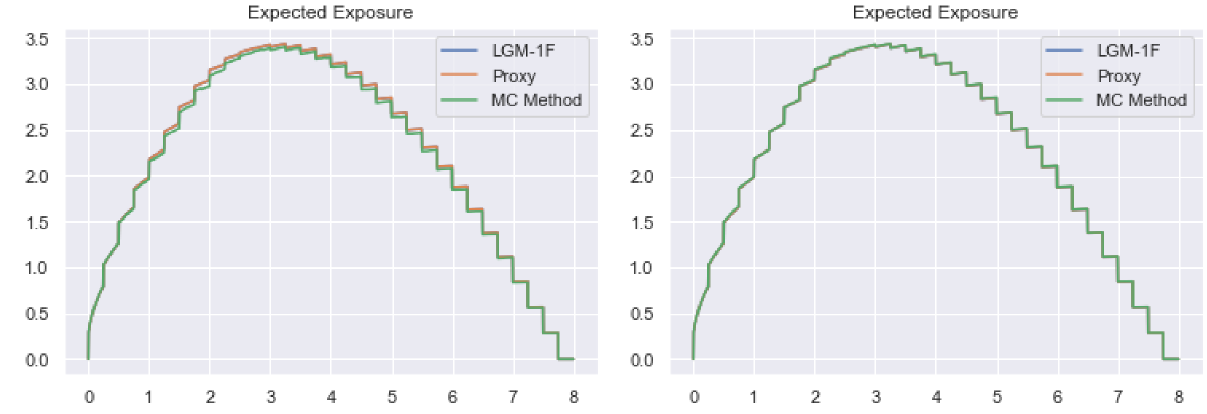

Figure 2: The expected exposure of an interest rate swap with a 5-year tenor and semi-annual payments on both legs, calculated in three different ways (quasi-closed formula LGM-1F (40), closed formula based on the Gaussian approximation (46), Monte Carlo simulation pricing). In the left graph, the Monte Carlo method uses 10,000 simulations, and in the right graph, it uses 50,000 simulations.

Figure 2 represents the Expected Exposure profile of an 8-year tenor interest rate swap, calculated directly using the quasi-closed formula (40) under the LGM-1F model with daily discretization. Alternatively, the same Expected Exposure profile is calculated using the Gaussian approximation (46) and Monte Carlo simulations (with 10,000 simulations for the left graph and 50,000 simulations for the right graph). The results of this test demonstrate that the Gaussian approximation is valid and that 10,000 simulations are sufficient to ensure convergence to the reference swaption prices (LGM-1F price). In the sequel, we will use the Gaussian approximation of swap rates and the Bachelier formula as benchmarks for pricing swaptions.

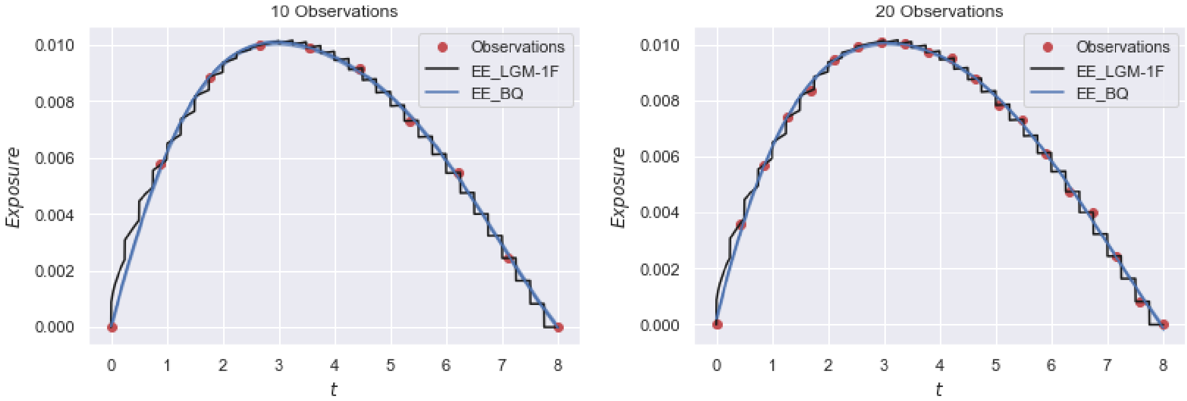

Figure 3: The Expected Exposure profile of an 8-year tenor interest rate swap, calculated directly using the quasi-closed formula (40) under the LGM-1F model with daily discretization and with the Gaussian Process Regression (GPR) (the left figure uses 10 LGM-1F price observations, and the right figure uses 20 LGM-2F price observations).

5 Numerical Applications for CVA Calculation

This section aims to numerically evaluate the precision and speed of Bayesian quadrature for computing CVA in a derivatives portfolio. We consider two case studies: In the first case study, we analyze a portfolio of 400 interest rate swaps in a single currency. The Gaussian Process Regression (GPR) is trained on the portfolio's residual maturity and a single risk factor, namely the short-term interest rate of the Hull-White one factor model. In the second case study, we examine a portfolio of 600 interest rate swaps across 10 different currencies. Here, GPR is trained on the portfolio's residual maturity and on each of the 11 risk factors representing the yield curve of 11 economies (10 foreign economies plus one domestic economy). In this context, Monte Carlo simulations serve as the benchmark method for our comparative study of Expected Exposure (EE) and CVA calculations. Indeed, most counrerparty credit risk (CCR) calculations are carried out at the level of the netting set using Monte Carlo simulations. Some CCR metrics are done at portfolio level, but in the sequel we consider, without loss of generality, CRR measurements at netting set level. For Monte Carlo simulations purpose, 10,000 paths of market risk factors into the future at 500 time points will be generate. In this context, 5,000,000 simulations will be generated and used as reference for Monte Carlo simulations.

Algorithm 3. Calculation of **EE** and **CVA** by Monte Carlo simulations

```
input  : Loss given default $1 - R$, survival probability $S(0, t), ...$
output: estimation of the expected exposure EE and CVA
Generate $m$ risk factors scenarios on $n$ points of the time grid $0 = t_0, t_1, ..., t_n = T$
For each scenario $j = 1, 2, ..., m$ and for each future date $t_k$, calculate the value of the portfolio.
Compute the exposure of the bank for the scenario $j$ at time $t_k$: $E^{(j)}(t_k) = \max(V^{(j)}(t_k), 0)$
Compute the expected exposure of the bank at $t_k$ : $\mathbf{EE}(t_k) = \frac{1}{m}\sum_{i=1}^{m} D^{(j)}(0, t_k)E^{(j)}(t_k)$
Compute the CVA: $\mathbf{CVA}(0, T) = (1 - \mathbf{R})\sum_{i=1}^{m} D(0, t_i)\mathbf{EE}(0, t_i)\left[S(0, t_{i-1}) - S(0, t_i)\right].$
```

5.1 Case Study 1: Portfolio of Interest Rate Swaps in the Same Currency

We will present a numerical example of an interest rate swap portfolio. The number of swaps is arbitrarily set at 400, with notional amounts uniformly set at 10,000 euros. Other swap characteristics, such as payment frequency and maturity, are randomly assigned. Without loss of generality, the valuation framework used is the single-curve approach. However, this does not affect our quantitative analysis, as our approach can be easily adapted to a multi-curve framework. As a benchmark, we consider the full pricing by the Monte Carlo method using 10,000 simulations on 700 time steps to compute the Expected Exposure profile over the maximum maturity of the portfolio.

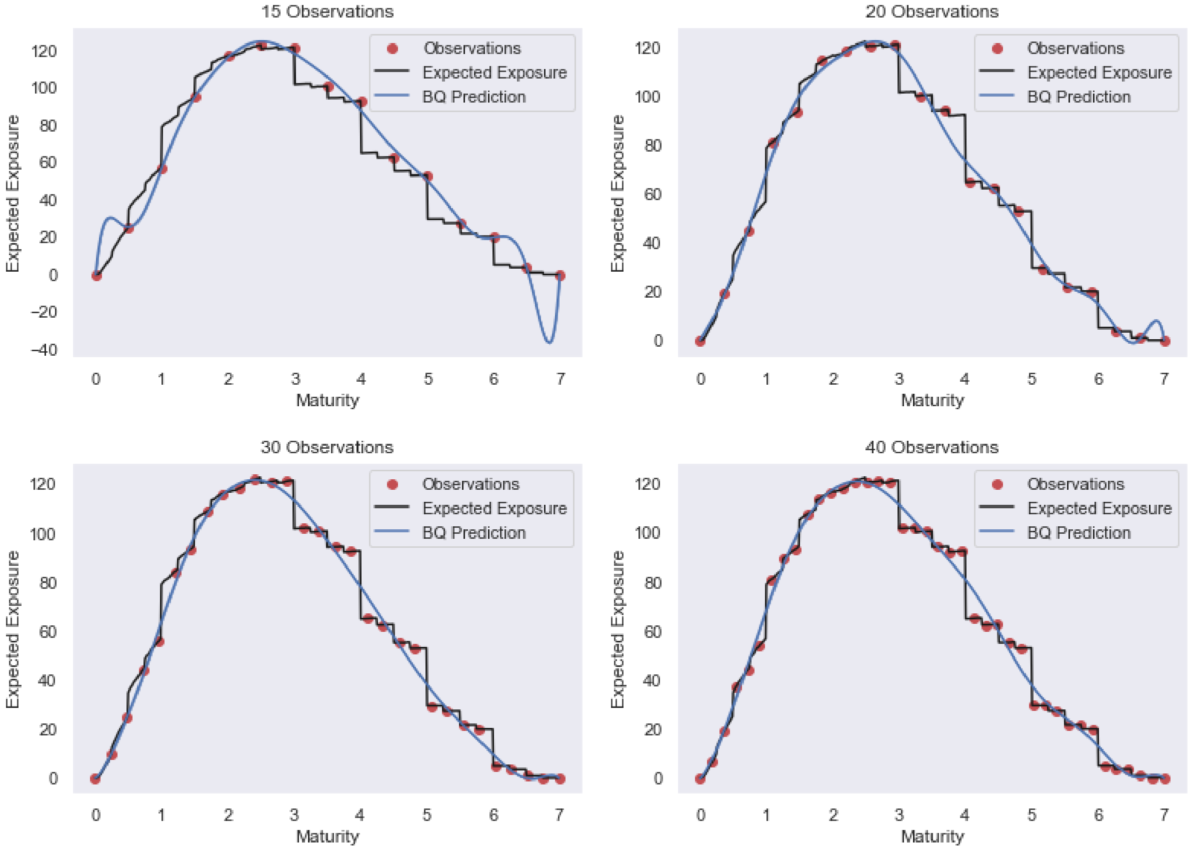

Figure 4: The expected exposure of the portfolio of 400 interest rate swaps for 4 level of number of observations (15, 20, 30, and 40) used for training the Gaussian process.

The value of the interest rate swaps portfolio is estimated at -54,507 €, with a corresponding CVA value of 432 €. The execution time for the CVA calculation is approximately 41 minutes.

Training points Number	**Bayesian Quadrature Prediction**	**Relative Error**	**Computation Time**	**Time Savings**
15	446 €	0,26	26	93
25	433 €	0,03	40	62
30	423 €	0,17	50	50
35	435 €	0,05	62	40
40	432 €	0,00	76	32

Table 1: Calculation of CVA using the Bayesian Quadrature method based on the number of training points and the training of the GPR is done based on time maturity.

The numerical test demonstrates that using 40 training points for the GPR, with training currently performed only on the time axis, the Bayesian Quadrature method achieves a high level of accuracy while significantly reducing computation time to 76 seconds, compared to 41 minutes for Monte Carlo simulations. This indicates that under the conditions of this numerical test, Bayesian Quadrature is 32 times faster than the Monte Carlo method.

As previously mentioned, the calculation of Expected Exposure for this numerical test involves 10,000 simulations of the swap rate portfolio value at each discretization date. This process is time-consuming since each of the 400 swaps is individually valued for all 10,000 simulations. In the following numerical test, instead of calculating the portfolio value 10,000 times, we will do it for a much smaller number of times (e.g., 5 to 15 times). These valuations will serve as training points for the GPR. The two following graphs show that the GPR provides an excellent approximation with only 5 training points.

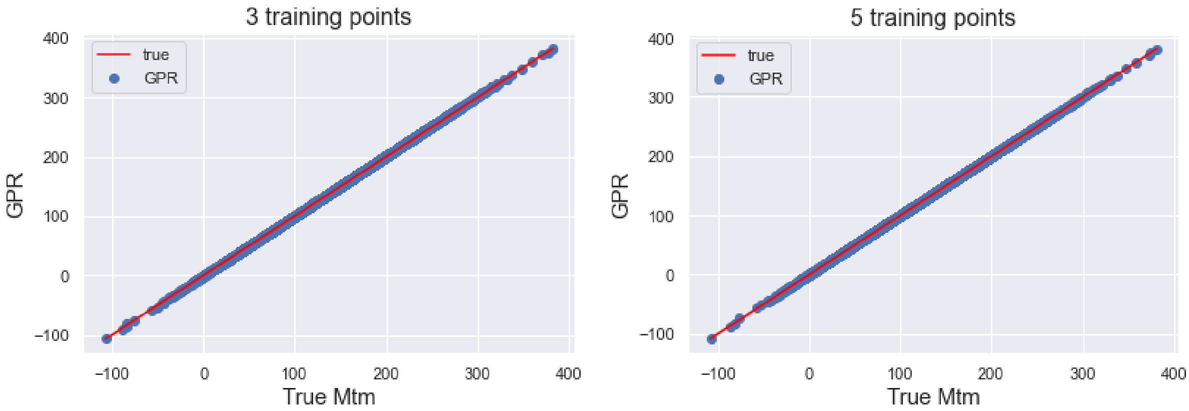

Figure 5: Comparison of Monte Carlo prices and GPR prices for 10,000 simulations on a given discretization date.

We conducted the same numerical test as above to calculate the CVA for the portfolio of 400 interest rate swaps, but this time the portfolio value was calculated using Gaussian Process Regression (trained with only 5 points) for the 10,000 simulations. The results of this numerical test are presented in the following table.

Training points Number	**Bayesian Quadrature Prediction**	**Relative Error**	**Computation Time**	**Time Savings**
15	446 €	0,26	4	615
25	434 €	0,01	6	400
30	423 €	0,17	7	369
35	435 €	0,05	8	317
40	432 €	0,00	9	276

Table 2: Calculation of CVA using the Bayesian Quadrature method based on the number of training points and the training of the GPR is done based on time to maturity and X_t

In terms of precision, we achieved the same results as the previous numerical test, but with significantly reduced computation time. This time, the Bayesian Quadrature method ensured a high level of accuracy while reducing the computation time to 9 seconds, compared to 41 minutes for Monte Carlo simulations. This means that under the conditions of this numerical test, Bayesian Quadrature is 276 times faster than the Monte Carlo method.

5.2 Case Study 2 : IRS Portfolio in different currencies

For this case study, we consider a portfolio of 600 interest rate swaps in 10 different currencies (one domestic currency and nine foreign currencies). The characteristics such as notional amount, frequency, fixed coupon, etc., for each interest rate swap are randomly chosen. In this context, we consider 10 yield curves, each governed by the HW-1F diffusion, and 9 cross-currency rates, each assumed to follow a lognormal process. The objective of this setup is to test the Bayesian quadrature approach in a multidimensional framework. Similar to the previous test dedicated to the portfolio of interest rate swaps in a single currency, we applied Bayesian quadrature for CVA calculation in the multi-currency framework in two different ways. First, the integral calculation is performed over a given number of discretization points (ranging from 5 to 40), which are then used to train the GPR algorithm along the time axis (time to maturity). In this case, the Expected Exposure is calculated based on 10,000 simulations of the portfolio's value, with each swap being valued individually. Second, we train the GPR along the time axis and on the risk factors. This is achieved by applying the GPR algorithm to each component of the Mark-to-Market (MtM) that depends on each risk factor. As a reminder, the direct calculation of the CVA for the portfolio of 600 interest rate swaps is based on 10,000 simulations with 500 discretization steps. In this case, the portfolio value is -75,806 €and the corresponding CVA is estimated at 210 €. The execution time for the CVA calculation is approximately 31 minutes.

The following graph displays the Expected Exposure profile of the interest rate swap portfolio in a multidimensional framework, calculated using two different methods: (i) Monte Carlo simulations and (ii) GPR with three levels of training points (15, 20, 30, and 40). The numerical results highlight that the Expected Exposure profile is an irregular function of time. In contrast, the Expected Exposure profile estimated by Gaussian Process Regression (GPR) is a smooth function that approximates the Monte Carlo-calculated Expected Exposure profile very satisfactorily. Moreover, using 40 discretization points appears to be an excellent compromise for training the GPR and subsequently calculating the CVA. It should be noted that for each discretization point, we also use 5 portfolio valuations via Monte Carlo simulations to train the GPR and approximate the Mark-to-Market calculation for the 10,000 simulations

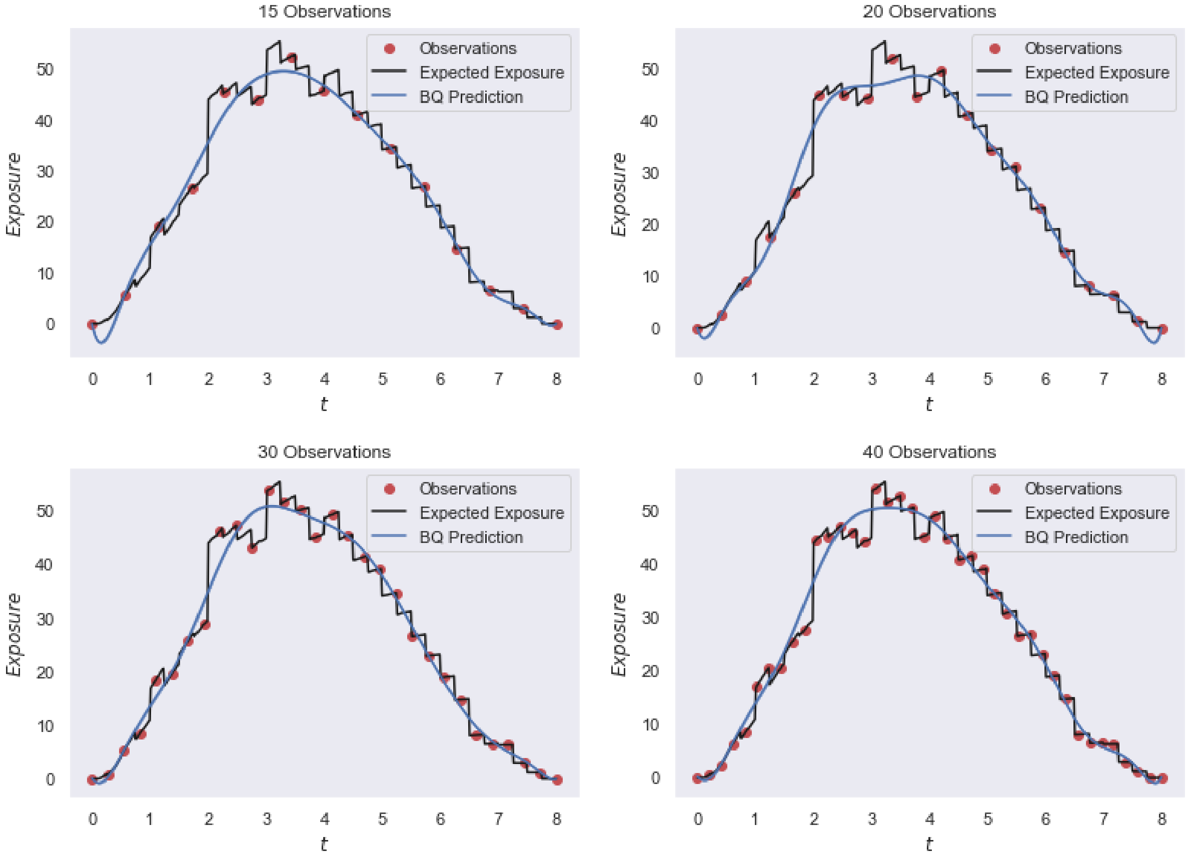

Figure 6: The expected exposure of the portfolio of 600 interest rate swaps in 10 different currencies for 4 level of number of observations (15, 20, 30, and 40) used for training the Gaussian process.

Training points Number	**Bayesian Quadrature Prediction**	**Relative Error**	**Computation Time**	**Time Savings**
15	208 €	0,04	26	53
25	206 €	0,06	62	22
30	210 €	0,00	63	22
35	210 €	0,01	72	19
40	212€	0,01	84	16

Table 3: Calculation of CVA using the Bayesian Quadrature method based on the number of training points and the training of the GPR is done based on time to maturity and X_t

The Bayesian Quadrature method ensured a good level of accuracy while reducing the computation time to 63 seconds, compared to 31 minutes for Monte Carlo simulations. This means that under the conditions of this numerical test, Bayesian Quadrature is 22 times faster than the Monte Carlo method.

References

[1] Damiano Brigo. *Interest Rate Models with Credit Risk, Collateral, Funding Liquidity Risk and Multiple Curve.* Imperial College London, Spring Term, 2021.

[2] Fabio Mercurio & Damiano Brigo. *Interest Rates Models Theory and Practice. II, Continuous Time Models.* Wiley Finance, 2008.

[3] Robert Culkin Sanjiv R. Das. Machine learning in finance: The case of deep learning for option pricing. 2017.

[4] C. DeBrusk and E. Du. Why wall street needs to make investing in machine learning a higher priority. *Oliver Wyman Report*, 2018.

[5] Stéphane Crépey & Mattew F. Dixon. Gaussian process regression for derivative portfolio modeling and application to cva computations. 2019.

[6] Qian Fen. *Advanced Estimation of Credit Valuation Adjustment.* PhD thesis - Delft University of Technology), 2017.

[7] Financial Stability Board (FSB). Artificial intelligence and machine learning in financial services - market developments and financial stability implications. 2017.

[8] Antonio Maffia & Yong Chao Sun Gang Mu, Teodor Godina. Supervised machine learning with control variates for american option pricing. *Foundations of computing and decision science*, 2018.

[9] Andrew Green. *XVA: Credit, Funding and Capital Valuation Adjustments.* XVA: Credit, Funding and Capital Valuation Adjustments, 2015.

[10] Jon Gregory. *Couterparty Credit Risk and Credit Value Adjustment.* Wiley Finance, 2012.

[11] Fred Hoffman. *Credit Valuation Adjustment (CVA): Theory, Mathematics and Risk Management.* LAP LAMBERT Academic Publishing, 2012.

[12] John C. Hull. *Risk Management and Financial Institutions.* Wiley Finance), 2014.

[13] Thierry Roncalli Joan Gonzalvez, Edmond Lezmi and Jiali Xu. Financial applications of gaussian processes and bayesian optimization. 2019.

[14] Basel Committee on Banking Supervision. Fundamental review of the trading book: A revised market risk framework. *https://www.bis.org/publ/bcbs265.pdf*, 2013.

[15] Basel Committee on Banking Supervision. Minimum capital requirements for market risk. *Available at https://www.bis.org/bcbs/publ/d352.pdf*, 2016.

[16] Basel Committee on Banking Supervision. Basel iii: Finalising post-crisis reforms. *Available at https://www.bis.org/bcbs/publ/d424.htm*, 2017.

[17] Basel Committee on Banking Supervision. Credit valuation adjustment risk: targeted final revisions. *Bank of International Settlements*, Available at https://www.bis.org/bcbs/publ/d488.pdf, 2020.

[18] Basel Committee on Banking Supervision. Minimum capital requirements for credit risk. $https://www.bis.org/basel_{f}ramework/chapter/CRE/20.htm$, 2023.

[19] Steven E. Shreve. *Stochastic Calculus for Finance.* Springer.

[20] Alexander Sokol. *Long-Term Portfolio Simulation - For XVA, Limits, Liquidity and Regulatory Capital.* Risk Books, 2014.

[21] INTERNATIONAL SWAPS and INC DERIVATIVES ASSOCIATION. *User's Guide to the ISDA Credit Support Documents under English Law.* ISDA), 1999.

[22] Ignacio Ruiz & Mariano Zeron. *Machine Learning for Risk Calculations: A practionner's View.* John Wiley & Sons, 2021.