Leveraging Bayesian Quadrature for Accurate and Fast CVA Calculations

Noureddine Lehdili[*], Pascal Oswald[†] & Othman Mirinioui[‡]

July 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 Gaussian Process Regression (GPR) to enhance the efficiency and accuracy of counterparty risk metrics calculations, particularly Credit Valuation Adjustment (CVA). We focus on Bayesian quadrature to efficiently estimate risk metrics with minimal observations. These methods provide a promising balance between precision and computational efficiency. Our application focuses on a trading portfolio of fixed-income derivatives, such as interest rate swaps, swaptions, within the One Factor Gaussian Markov (LGM-1F) model framework. The LGM-1F model is widely accepted for valuing and managing risks of such derivatives, making it a suitable framework for credit risk estimation and CVA calculation.

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 ([9],[10],[8]). This especially concerns the credit risk associated with OTC derivatives counterparties and this risk has gained more importance since the global financial crisis ([15],[14], [16]). Before the financial crisis of 2008, many financial institutions managed their counterparty credit risk by dealing only with the most robust counterparties, often relying on the perceived solvency of "too big

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

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

[‡] othman.mirinioui-ext@natixis.com. Quantitative Analyst Intern Market & Counterparty Risks Modelling

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],[8],[11]). 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 evaluated for economic, accounting, and regulatory purposes. The economic approach calculates Potential Future Exposure (PFE) to set exposure limits. The accounting approach incorporates value adjustments (XVA), particularly CVA, into pricing. The regulatory approach assesses the capital needed to cover unexpected losses, such as EEPE and VaR-CVA. Exposure, essential in these three approaches, requires rapid and accurate estimates. Banks use Monte Carlo simulations and pricing libraries to evaluate portfolios, leading to complex and time-consuming processes, especially under new regulations. To overcome these challenges, institutions are increasingly adopting approximations ([22],[5],[4],[12]) and machine learning methods, such as deep learning algorithms ([3], [6], [2],[7]), to reduce computational complexity while maintaining accuracy.

In this article, we specifically focus on the application of these techniques for calculating counterparty credit risk metrics to address the following issues: How can we efficiently implement these risk metrics, reducing computation time while ensuring satisfactory accuracy? In academic and professional literature, deep learning and Gaussian process regression are used to accurately and quickly value derivatives. Another potential application involves using Gaussian processes to estimate Value at Risk (VaR) 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 explore the use of Gaussian processes via Bayesian quadrature (which also incorporates Bayesian optimization) to compute a numerical integral with very few observations. Specifically, we focus on Bayesian quadrature, a technique based on Gaussian process regression (GPR), which has two unique mathematical properties: given an analytical target function, the algorithm converges exponentially towards the target, and GPR ensures rapid and numerically stable evaluation. This characteristic makes GPR a distinctive pricing function approximator, significantly reducing the computational burden associated with risk calculations for portfolio evaluation across multiple scenarios. Both approaches will be applied to optimally calculate CCR metrics and Credit Valuation Adjustments, achieving an excellent balance between precision and computation time. For this purpose, we will focus our application on a trading portfolio composed of fixed-income derivatives such as interest rate swaps, swaptions, and cancellable swaps, adopting the One Factor Gaussian Markov (LGM-1F) model framework. Indeed, it is well known that the LGM-1F model is a consensus for the valuation and risk management of such derivatives. Therefore, it is natural to maintain the same modeling framework for credit risk estimation and CVA calculation.

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 discounted values of all trades with the counterparty $\{\mathbf{V}_i(t)\}_{i=1}^{N}$. The discounted 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 allowed, the exposure $\mathbf{E}(t)$ is expressed as follows:

$$E(t,T) = \max\{\mathbf{V}(t), 0\}. \tag{1}$$

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

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

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:

$$\mathbf{EE}(u,T) = \mathbf{E}^{Q}[E(u,T)]. \tag{3}$$

As we will explain in the following subsection, the Expected Exposure (EE) is the cornerstone in the calculation of Credit Valuation Adjustment.

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:

$$\mathbf{CVA}(t,T) = (1-\mathbf{R})\mathbf{E}^{Q}\left[\int_t^T \mathbf{EE}(t,u)dS(t,u)\right]. \tag{4}$$

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

$$\mathbf{CVA}(t,T) = (1-\mathbf{R})\sum_{i=1}^{m}\mathbf{EE}(t,t_i)\left[S(t,t_{i-1}) - S(t,t_i)\right]. \tag{5}$$

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

$$P(t,T) = 1 - S(t,u) = 1 - \exp\left[-\int_t^u \lambda(x)dx\right]. \tag{6}$$

- *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 $\mathcal{X}$,

$$\mathcal{I} = \int_{\mathcal{X}} f(x)dp(x). \tag{7}$$

When no closed-form solution to this integral is available, numerical methods are needed. Numerical integration schemes seek to approximate the integral (7) by weighted sum of function evaluations called a quadrature rule :

$$\mathcal{Q}(f, \mathbf{X}, \omega) = \sum_{i=1}^{N} \omega_i f(x_i) = \omega^{\perp} f_{\mathbf{X}} \tag{8}$$

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 (8).

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) ([19]). 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

$$f_{\mathbf{X}} \sim \mathcal{N}(m_{\mathbf{X}}, K_{\mathbf{XX}}) \tag{9}$$

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 $k(x, x') = \mathbb{E}\left[(f(x) - m(x))\,(f(x') - m(x'))\right]$.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:

$$k(x, x') = \sigma_f^2 exp\left(\frac{-(x - x')^2}{2l^2}\right) \tag{10}$$

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 [19], 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

$$y_n = f(x_n) + \epsilon_n \quad \text{where} \quad \epsilon_n \sim \mathcal{N}(0, \sigma^2) \tag{11}$$

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

$$\begin{pmatrix} y \\ f^* \end{pmatrix} \sim N\left(\begin{bmatrix} m \\ m(x^\star) \end{bmatrix}, \begin{bmatrix} K + \sigma^2 I & k(x^\star) \\ k(x^\star)^\perp & k(x^\star, x^\star) \end{bmatrix}\right) \tag{12}$$

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 equation (12) and the well known properties of Gaussian processes ([13] the law of $f^*|y$ follows a Gaussian distribution:

$$f^*|y \sim N(k(x^\star)^\perp (K + \sigma^2 I)^{-1} y, k(x^\star, x^\star) - (K + \sigma^2 I) k(x^\star)^\perp) \tag{13}$$

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

$$\begin{aligned} \bar{f^*} &= E[f(x^*)|f(x)] \\ &= k(x^\star)^\perp (K + \sigma^2 I)^{-1} y \\ &= k(x^\star)^\perp (K + \sigma^2 I)^{-1} f(x) \end{aligned} \tag{14}$$

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

$$var(y^*) = k(x^\star, x^\star) - (K + \sigma^2 I) k(x^\star)^\perp \tag{15}$$

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:

$$E[f(x^*)|f(x)] = \sum_{i=1}^{n} \omega_i k(x_i, x^*) \quad \text{where} \quad \omega = (K + \sigma^2 I)^{-1} f(x) \tag{16}$$

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:

$$E[f(x^*)|f(x)] = \sum_{i=1}^{n} \phi_i f(x_i) \quad \text{where} \quad \phi = k(x^\star)^\perp (K + \sigma^2 I)^{-1} \tag{17}$$

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, σ_n, σ_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:

$$\begin{aligned} l(\theta) &= ln(p(y|\theta) \\ &= -\frac{1}{2} y^T (K + \sigma_n^2 I)^{-1} - \frac{1}{2} log(|K + \sigma_n^2 + I|) - -\frac{n}{2} log(2\pi) \end{aligned} \tag{18}$$

The first term in the equation 18 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:

$$\begin{aligned} \frac{\partial l(\theta)}{\partial \theta_j} &= \frac{1}{2} y^T K^{-1} \frac{\partial K}{\partial \theta_j} K^{-1} y - \frac{1}{2} tr \left(K^{-1} \frac{\partial K}{\partial \theta_j} \right) \\ &= \frac{1}{2} tr \left(\omega \omega^T - K^{-1} \frac{\partial K}{\partial \theta_j} \right) \end{aligned} \tag{19}$$

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 18 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 22 requires only time $0(n^2)$ per hyperparameter. Thus, the computational overhead of computing derivatives is small, so the optimization problem in the equation 18 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 ([1], [17]) 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 $\mathbf{I}$,

$$\mathbf{I} = \int_{\mathcal{X}} \mathbf{f}(x) dp(x). \tag{20}$$

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

$$m := \mathbb{E}_f[\mathbf{I}] = \int_{\mathcal{X}} m(x) dp(x) \quad \text{and} \quad \nu := \mathbb{V}_f[\mathbf{I}] = \int_{\mathcal{X}} k(x, x') dp(x) dp(x') \tag{21}$$

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:

$$\begin{pmatrix} y \\ \mathbf{I} \end{pmatrix} \sim N\left(\begin{bmatrix} \mathbf{m} \\ m \end{bmatrix}, \begin{bmatrix} K + \sigma^2 I & \int_{\mathcal{X}} k(x)dp(x) \\ \int_{\mathcal{X}} k(x)^{\perp}dp(x) & \iint_{\mathcal{X}} k(x,x')dp(x)dp(x') \end{bmatrix} \right) \tag{22}$$

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 $\nu_{\mathcal{D}}$:

$$\begin{aligned} \mathbf{m}_{\mathcal{D}} &= \mathbb{E}_{f|\mathcal{D}}[\mathbf{I}] \\ &= \int_{\mathcal{X}} \Big[m(x) + k(x)^{\perp}(K + \sigma^2 I_d)^{-1}(y - m) \Big] \, dp(x) \\ &= \int_{\mathcal{X}} m(x)dp(x) + \sum_{i=1}^{N} \int_{\mathcal{X}} k(x, x_i)dp(x) \left[(K + \sigma^2 I_d)^{-1}(y - m) \right]_i \end{aligned} \tag{23}$$

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

$$\begin{aligned} \nu_{\mathcal{D}} &= \mathbb{V}_{f|\mathcal{D}}[\mathbf{I}] \\ &= \iint_{\mathcal{X}} \Big[k(x,x') - k(x)^{\perp}(K + \sigma^2 I_d)^{-1}k(x) \Big] \, dp(x)dp(x') \\ &= \iint_{\mathcal{X}} k(x,x')dp(x)dp(x') - \sum_{i,j=1}^{N} \left[(K + \sigma^2 I_d)^{-1} \right]_{ij} \int_{\mathcal{X}} k(x, x_i)dp(x) \int_{\mathcal{X}} k(x_j, x')dp(x') \end{aligned} \tag{24}$$

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 (32),(24) can be rewritten in the following shorthand way:

$$\mathbf{m}_{\mathcal{D}} = \mathrm{m} + \kappa^{\perp}(K + \sigma^2 I_d)^{-1}(y - \mathbf{m}) \quad \text{and} \quad \nu_{\mathcal{D}} = \xi - \kappa^{\perp}(K + \sigma^2 I_d)^{-1}\kappa \tag{25}$$

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

$$\mathbf{m}_{\mathcal{D}} = \kappa^{\perp} K^{-1} f = \sum_{i=1}^{N} \omega_i f(x_i) \quad \text{with} \quad \omega = K^{-1} \times \kappa \tag{26}$$

This is the classic form of a quadrature rule (26) that represents the integral as a weighted sum of function evaluations. Bayesian Quadrature replaces the integral (8) 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. The implementation of the Bayesian quadrature described above to numerically calculate integral (8) can be summarized in the steps outlined in Algorithm 1 below.

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

$$\rho^2_{\mathcal{D}}(X^{\star}) = \nu_D^{-1} \kappa^{\perp}_{\star|\mathcal{D}} C^{-1}_{\star|\mathcal{D}} \kappa_{\star|\mathcal{D}} \tag{27}$$

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*

$$\kappa_{\star|\mathcal{D}} = \kappa(X^{\star}) - \kappa(X) \left(K + \sigma^2 \mathbf{I} \right)^{-1} k(X, X^{\star}) \tag{28}$$

Algorithm 1. Bayesian quadrature

```
Input  : $f(.), \mathcal{GP}(m,k), \nu(.), \mathcal{D}\{\mathbf{X}, y\}, \sigma^2$
Output: Bayesian quadrature estimation of the integral $\mathbb{I}$
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
```

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

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

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*

$$H[\mathbf{x}] = \int_{\mathcal{X}} p(x)\log p(x)dx \quad \textit{and} \quad H[\mathbf{x}|\mathbf{y}] = \int_{\mathcal{X}} p(x,y)\log p(x|y)dxdy \tag{30}$$

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

$$H[\mathbf{x}] = \frac{D}{2}\log(2\pi \exp) + \frac{1}{2}\log\det\Sigma. \tag{31}$$

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 the entropy criteria 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:

$$\alpha^{\mathbf{MI}}_{\mathcal{D}} = -\frac{1}{2}\log\left(1 - \rho^2_{\mathcal{D}}(X^{\star})\right) \tag{32}$$

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

$$\begin{aligned} \alpha_{\mathcal{D}}^{\mathbf{IVR}} &= \frac{\mathbb{V}[\mathbf{I}|\mathcal{D} - \mathbb{V}[\mathbf{I}|\mathcal{D} \cup \{\mathbf{X}^\star, \mathbf{y}^\star\}]}{\mathbb{V}[\mathbf{I}|\mathcal{D}]} \\ &= \rho_{\mathcal{D}}^2 \left(X^\star\right) \end{aligned} \tag{33}$$

which is independent of upcoming functions evaluations $\mathbf{y}^\star$. Algorithm 2 follows the steps of Algorithm 1 but incorporates the described approach of selecting discretization points based on Bayesian optimization and entropy.

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

It is well known that calculating counterparty risk indicators and the CVA of a trading portfolio is generally done through Monte Carlo simulations, as this method is typically easy to implement. The main drawback of this approach is its performance, as accuracy increases with the number of simulations. However, when adopting a specific modeling framework such as One Factor Linear Gaussian Markov (LGM-1F), other alternative methods can serve as a serious option for calculating these risk indicators, offering performance that compares very favorably to the Monte Carlo method. As mentioned earlier, the choice of the LGM-1F model is motivated by its status as the market consensus for valuing the majority of interest rate derivatives. Therefore, it seems appropriate to use the same model to estimate the valuation adjustment for counterparty risk (CVA). In the sequel, the derivative products comprising our trading portfolio are interest rate swaps and swaptions. Within this framework and alongside the Monte Carlo method, we also present the numerical integration method for calculating the CVA of a portfolio of swaps and/or swaptions, as well as an analytical formula, used as a proxy, as reference methods for calculating the CVA of a portfolio of interest rate swaps.

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:

$$dX_t = \left[\phi(t) - \lambda X_t\right] dt + \sigma(t) dW_t \tag{34}$$

where

$$X_0 = 0 \quad \text{and} \quad \phi(t) = \int_0^t \sigma(s)^2 e^{-2\lambda(t-s)} ds \tag{35}$$

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:

$$B(t,T) = \frac{B(0,T)}{B(0,t)} \exp\left(-\frac{1}{2}\beta(t,T)^2\phi(t) - \beta(t,T)X_t\right) \quad \text{and} \quad \beta(t,T) = \frac{1-\exp(-\lambda(T-t))}{\lambda} \tag{36}$$

Consider an interest rate swap and denote by $T_0 < T_1 < ... < T_n$ the settlement dates for the fixed leg (no flows exchanged in T_0). The fixed swap rate defined at date t ($t \leq T_0$) equals

$$S(t,T_0,T_n) = \frac{B(t,T_0) - B(t,T_n)}{LVL(t,T_0,T_n)} \quad \text{where} \quad LVL(t,T_0,T_n) = \sum_{i=1}^{n} \delta_i B(t,T_i). \tag{37}$$

The parameter δ_i is the year fraction between T_{i-1} and T_i calculated in the adequate basis. In this setup, the value of a fixed coupon interest rate swap with rate K at a future date t can be expressed as follows:

$$V(t,X_t) = B(t,T_0) - B(t,T_n) - K \times \sum_{i=1}^{n} \delta_i B(t,T_i). \tag{38}$$

The calculation of the CVA for this swap can be expressed as follows using its value $V(t,X_t)$ and the density $\phi(.)$ of the Gaussian variable X_t:

$$\mathbf{CVA} = (1-R) \times \mathbb{E}^{\mathbb{Q}}\left[\left(\int_{-\infty}^{+\infty} V(\tau,x)^+ \phi(x) dx\right) \times 1_{\tau \leq T_e}\right] \tag{39}$$

4.1 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

$$S(T_e,T_0,T_n) = \frac{B(T_e,T_0) - B(T_e,T_n)}{LVL(T_e,T_0,T_n)} \quad \text{where} \quad LVL(T_e,T_0,T_n) = \sum_{i=1}^{n} \delta_i B(T_e,T_i). \tag{40}$$

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:

$$\begin{aligned} LVL(T_e, T_0, T_n)(S(T_e, T_0, T_n) - K)^+ = &\left(B(T_e, T_0) - B(T_e, T_n) - \sum_{i=1}^{n} \delta_i K B(T_e, T_i) \right)^+ \\ = &\sum_{i=1}^{n} c_i \left(\frac{B(T_e, T_i)}{B(T_e, T_0)} - K_i \right)^+ \end{aligned} \tag{41}$$

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:

$$\begin{aligned} V(0, T_e, T_0, T_n) = &B(0, T_e) \times \mathbb{E}^{\mathbb{Q}^{T_e}} \left[\left(B(T_e, T_0) - B(T_e, T_n) - \sum_{i=1}^{n} \delta_i K B(T_e, T_i) \right)^+ \right] \\ = &B(0, T_0) \times \sum_{i=1}^{n} c_i \times P_i^{bs}(F_i, 0, \sigma_i^{bs}, K_i, T) \end{aligned} \tag{42}$$

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

$$F_i = \frac{B(0, T_i)}{B(0, T_e)} \quad \text{and} \quad \sigma_i^{bs} = \frac{1}{\sqrt{T_e}} \beta(T_e, T_i) \sqrt{\phi(T_e)}. \tag{43}$$

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 a trading portfolio including swaptions 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 the differential $dS(t)$ using formula (38) by applying Itô's lemma and neglecting the terms present in the drift:

$$dS(t) = \left[\frac{dB(t, T_0) - dB(t, T_n)}{\sum_{i=1}^{n} \delta_i B(t, T_i)} - \frac{B(t, T_0) - B(t, T_n)}{\left(\sum_{i=1}^{n} \delta_i B(t, T_i)\right)^2} \sum_{i=1}^{n} \delta_i dB(t, T_i) \right] + (...)dt \tag{44}$$

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

$$dS(t) = \frac{\sigma(t)}{\lambda} e^{\lambda t} S(t) \left(\frac{B(t, T_0) e^{-\lambda T_0} - B(t, T_n) e^{-\lambda T_n}}{B(t, T_0) - B(t, T_n)} - \frac{\sum_{i=1}^{n} \delta_i B(t, T_i) e^{-\lambda T_i}}{\sum_{i=1}^{n} \delta_i B(t, T_i)} \right) dW^{\mathbb{Q}} + (...)dt \tag{45}$$

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

$$dS(t) = \sigma(t) e^{\lambda t} g(t) S(t) dW^{\mathbb{Q}} + (...)dt \tag{46}$$

[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.

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 $S(t)$:

$$dS(t) = \sigma(t)e^{\lambda t}g(0)S(0)dW^{\mathbb{Q}}. \tag{47}$$

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 $S(t)$ are governed by the normal process (45), can be obtained using the Bachelier formula as follows :

$$V^\star(0) = LVL(0,T_e)\left[S(0)\sigma^\star\sqrt{T}\phi\left(\frac{K-S(0)}{S(0)\sigma^\star\sqrt{T}}\right) + (S(0)-K)\left(1 - N\left(\frac{K-S(0)}{S(0)\sigma^\star\sqrt{T-t}}\right)\right)\right] \tag{48}$$

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 Figure 1, 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. Indeed, the expected exposure $\mathbb{EE}(t)$ associated with the interest rate swap at a future date t in the context of LGM-1F is simply the value of the swaption at maturity t:

$$\mathbf{EE}(t) = \int_{-\infty}^{+\infty} V(\tau,x)^+\phi(x)dx \tag{49}$$

where $V(t, X_t)$ is the interest rate swap value at future time t and the function $\phi(.)$ is the density of the Gaussian variable X_t. In fact, the Expected Exposure can be calculated exactly using the closed-form formula (43) or through the approximation (49) ($\mathbb{EE}(t) = V^\star(t)$). So, the CVA of the interest rate swap is expressed as follows:

$$\mathbf{CVA} = (1-\mathbf{R})\sum_{i=1}^{m}[S(0,t_{i-1}) - S(0,t_i)] \times V^\star(t_i). \tag{50}$$

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.

Now, denote the value of the swaption at future time t as $V^\star(t, X_t)$ and the default time

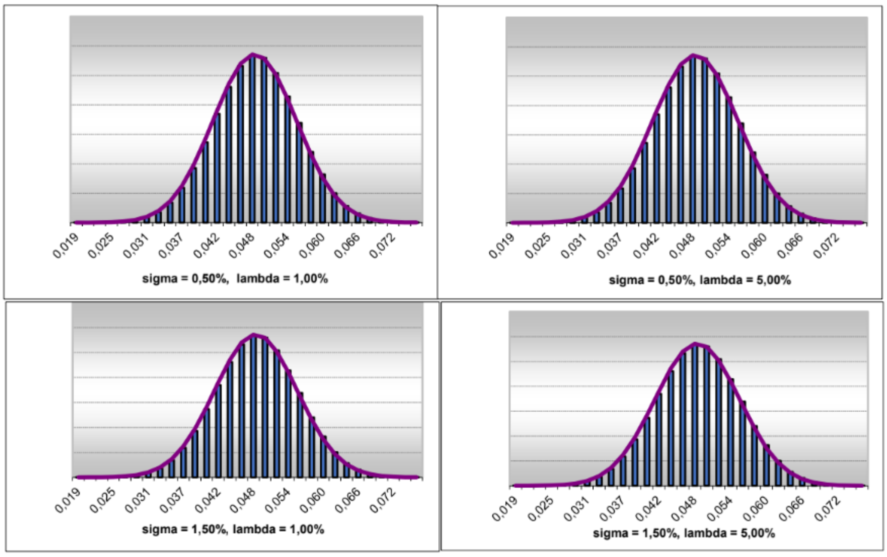

Figure 1: Comparison of the empirical distribution of a forward swap rate obtained with the LGM-1F model via Monte Carlo simulations and the theoretical distribution of the Gaussian approximation. This comparison is performed for different values of σ and λ.

of the counterparty as τ. In this case, the credit valuation adjustment (CVA) can be rewritten as follows:

$$\mathbf{CVA} = (1-R) \times \mathbb{E}^{\mathbb{Q}}\left[\left(\int_{-\infty}^{+\infty} V^{\star}(\tau,x)^{+}\phi(x)dx\right) \times 1_{\tau \leq T_e}\right]. \tag{51}$$

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

$$\begin{aligned} \mathbf{CVA} =& (1-R) \times \mathbb{E}^{\mathbb{Q}}\left[V^{\star}(\tau, X_\tau)\right] \times \mathbb{E}^{\mathbb{Q}}\left[1_{\tau \leq T_e}\right] \\ =& (1-R) \times P(t,T) \times V^{\star}(0). \end{aligned} \tag{52}$$

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.

4.2 Analytical CVA Approximation under LGM-1F Model

It is also practical to quickly estimate the expected exposures and CVA of an interest rate swaps portfolio outside the simulation system. To achieve this, we will demonstrate that the value distribution of such a portfolio under LGM-1F can be approximated by a Gaussian distribution. We consider a portfolio of n interest rate swaps, where the value of each swap at a future date t is denoted by $V_i(t)$ for $i = 1, 2, \ldots, n$. Without loss of generality, if we consider the swaps with values V_i as payers, we can express $V_i(t)$ as follows:

$$V_i(t) = \sum_{k=1}^{m} c_k^i B\left(t, T_k^i\right) \tag{53}$$

where

$$c_0^i = 1, \quad c_m = 1 - (T_m^i - T_{m-1}^i)K_i, \quad c_k^i = -(T_k^i - T_{k-1}^i)K_i \quad k = 1, ..., m-1 \tag{54}$$

By performing a first-order Taylor expansion of $B(t, T_k^i)$ from formula (37) with respect to X_t, we obtain the following approximation by performing a first-order Taylor expansion of $B(t, T)$ from formula (37) with respect to X_t, we obtain the following approximation:

$$B\left(t, T_k^i\right) \approx A(t, T_k^i) - A(t, T_k^i)\beta(t, T_k^i)X_t \tag{55}$$

where

$$A\left(t, T_k^i\right) = \frac{B\left(0, T_k^i\right)}{B\left(0, t\right)} \times \exp\left(-\frac{1}{2}\beta(t, T_k^i)^2\phi(t)\right) \tag{56}$$

Taking this approximation into account, the future value $V_i(t)$ of swap i for $i = 1, 2, ..., n$ in the portfolio is normally distributed and can be estimated as an affine function of the Gaussian variable X_t, which characterizes the dynamics of LGM-1F :

$$V_i(t) \approx \left[\sum_{k=1}^{m} c_k^i A\left(t, T_k^i\right)\right] - \left[\sum_{k=1}^{m} c_k^i A\left(t, T_k^i\right)\beta(t, T_k^i)\right] X_t \tag{57}$$

Since the sum of normal variables is also normal, the portfolio value $V(t) = \sum_{i=1}^{n} V_i(t)$ is normally distributed and it is expressed as an affine function of the Gaussian Variable X_t

$$V(t) \approx \left[\sum_{i=1}^{n}\sum_{k=1}^{m} c_k^i A\left(t, T_k^i\right)\right] - \left[\sum_{i=1}^{n}\sum_{k=1}^{m} c_k^i A\left(t, T_k^i\right)\beta(t, T_k^i)\right] X_t \tag{58}$$

The mean and the standard deviation of the portfolio value $V(t)$ at each future time t are given by

$$\mu_P(t) = \left[\sum_{i=1}^{n}\sum_{k=1}^{m} c_k^i A\left(t, T_k^i\right)\right] - \left[\sum_{i=1}^{n}\sum_{k=1}^{m} c_k^i A\left(t, T_k^i\right)\beta(t, T_k^i)\right] \mathbf{E}(X_t) \tag{59}$$

and

$$\sigma_P^2(t) = \left[\sum_{i=1}^{n}\sum_{k=1}^{m} c_k^i A\left(t, T_k^i\right)\beta(t, T_k^i)\right]^2 \times \mathbb{VAR}(X_t) \tag{60}$$

The expected Exposure (EE) of the interest rate swaps portfolio is given by:

$$\begin{aligned}\mathbf{EE}(t) =& \mathbf{E}\left(\max(\mu_P(t) + \sigma_P(t)Z, 0)\right) \\ =& \int_{-\mu_P/\sigma_P}^{+\infty} \left(\mu_p(t) + \sigma_P(t)x\right)\phi(x)dx \\ =& \mu_P\Phi(\mu_P/\sigma_P) + \sigma_P\phi(\mu_P/\sigma_P)\end{aligned} \tag{61}$$

The actual calculation of the mean μ_P and standard deviation σ_P of the interest rate swap portfolio is based on the mean and standard deviation of the variable X_t. In the context of the LGM-1F model, the integrated form of the dynamics of X_t is given as follows:

$$X_t = \int_0^t \phi(s)e^{-\lambda(t-s)}ds + \int_0^t \sigma(s)e^{-\lambda(t-s)}dW_s \tag{62}$$

So, the mean and standard deviation can be obtained analytically as follows :

$$\mathbf{E}\left(X_t\right) = \int_0^t \phi(s) e^{-\lambda(t-s)} ds \qquad \mathbb{VAR}(X_t) = \int_0^t \sigma(s)^2 e^{-2\lambda(t-s)} ds \tag{63}$$

CVA is assessed at the counterparty level, but there are instances where it's beneficial to identify the contributions of individual trades to the counterparty-level CVA. This can be straightforwardly achieved using an analytical formula when the exposure distribution is Gaussian, as it is in this context. For further details, refer to ([18]).

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.

5.1 Convergence tests for a unitary Interest Rate Swap

Before presenting the method for calculating the expected exposure (EE) and CVA for a portfolio of swaps, we will first do so for a single swap. This can be done precisely using a closed-form formula for the EE and a quasi-closed formula (numerical integration) for the CVA. In this numerical test, we ensure the convergence conditions of the Bayesian quadrature method for calculating the EE and CVA for a single swap.

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 60,000 simulations for the right graph). The results of this test demonstrate that the Gaussian approximation is valid and that 60,000 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.

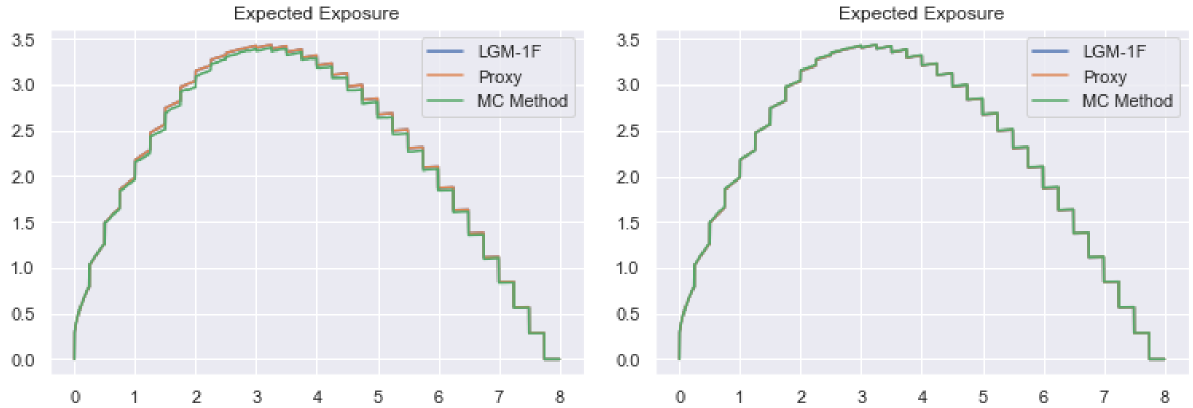

Figure 2: The expected exposure of an interest rate swap with a 5-year tenor and semi-annual payments on both legs, calculated in three different ways (quasi-closed formula LGM-1F (40), closed formula based on the Gaussian approximation (46), Monte Carlo simulation pricing). In the left graph, the Monte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

Figure 3 represents the Expected Exposure (EE) 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 Gaussian Process Regression (GPR). The left figure uses 10 LGM-1F price observations, and the right figure uses 20 LGM-2F price observations. This numerical test highlights that the Gaussian Process Regression fits the Expected Exposure profile in a very satisfactory manner with a reasonable number of observations.

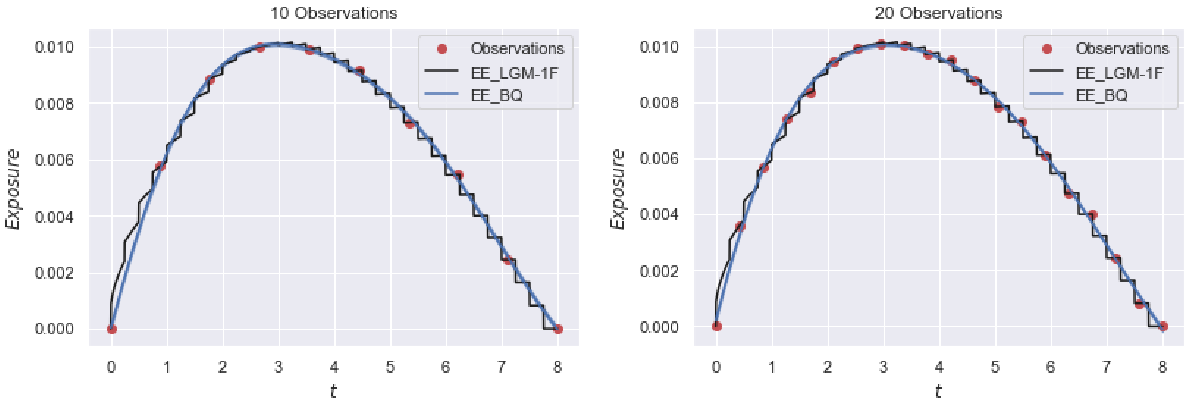

Figure 3: The Expected Exposure profile of an 8-year tenor interest rate swap, calculated directly using the quasi-closed formula (40) under the LGM-1F model with daily discretization and with the Gaussian Process Regression (GPR) (the left figure uses 10 LGM-1F price observations, and the right figure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

5.2 CVA calculation by classical approaches of a IRS portfolio

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 two approaches : (i) the full pricing by the Monte Carlo method combined with a numerical integration, and (ii) two successive numerical integrations. More specifically, the Expected Exposure profile is computed either through the Monte Carlo method or via numerical integration. Subsequently, the CVA is determined through numerical integration.

Simulations Number	**CVA Value**	**Computation Time**
10000	434 €	1822
40000	434 €	1053
60000	431 €	2006
80000	431 €	3165
100000	431 €	9866

Table 1: Calculation of CVA using The Monte Carlo simulations

Discretization Steps	**CVA Value**	**Computation Time**
50	436 €	6
100	430€	13
300	431 €	41
500	432 €	72
700	432€	96

Table 2: Calculation of CVA using The numerical integration method

Table 1 presents the CVA value, computed using The Monte Carlo Method, as a function of the number of simulations along with the corresponding computation times. Meanwhile, Table 2 displays the same results based on the number of discretization. The numerical test highlights that the Monte Carlo method converges at 60,000 simulations, yielding a CVA value of 431 euros with a computation time of 33 minutes. In contrast, numerical integration achieves nearly the same value with a discretization step of 300, but with a computation time of only 40 seconds. It is important to note that, in the example presented here, the implementation of numerical integration is straightforward. However, in the general case where the portfolio is driven by multiple risk factors, this approach becomes challenging. Consequently, only the Monte Carlo method can provide an effective numerical implementation, albeit with a high computation time. For this numerical example it is possible to use also the Gaussian proxy method presented in in subsection 4.2. and based on a quasi-analytical formula for CVA calculation. In this numerical example, the Gaussian proxy method, presented in subsection 4.2 and based on a quasi-analytical formula for CVA calculation, can also be used. Figure 4 shows the Expected Exposure profile of the interest rate swaps portfolio estimated by the three methods. Notably, the Expected Exposure profile calculated with the Gaussian proxy overlaps with the two other profiles obtained using the alternative methods.

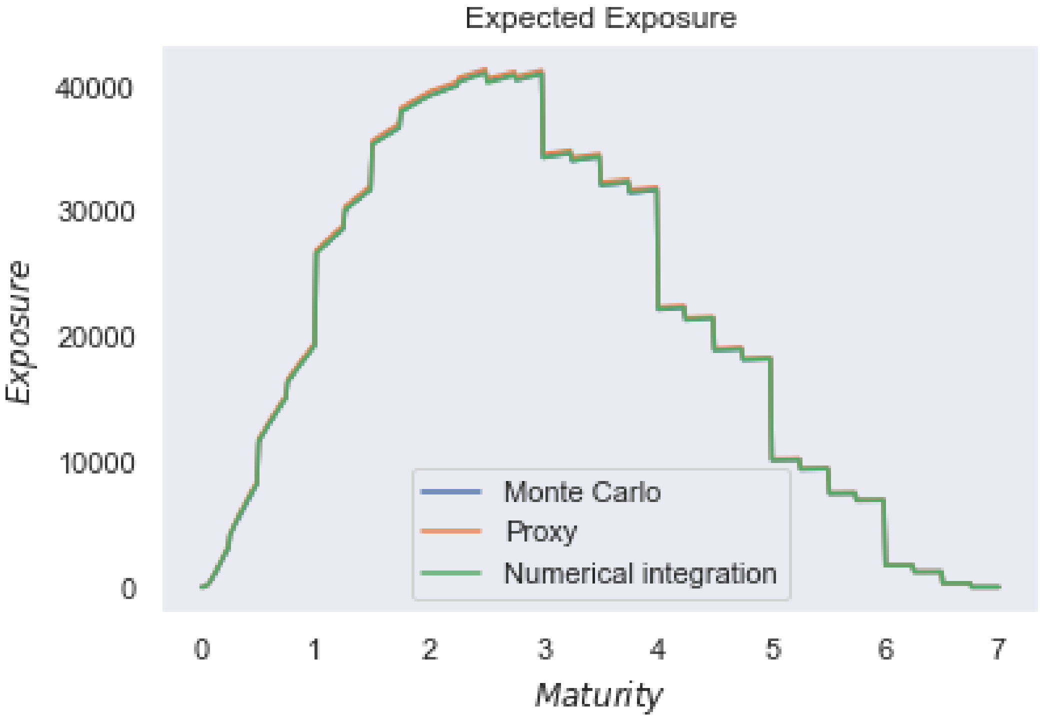

Figure 4: The expected exposure of the portfolio of 400 interest rate swaps computed with the Gaussian proxy, Monte Carlo and NI methods.

Table 3 presents the numerical results of a comparative study between the Monte Carlo method, numerical integration (NI), and the Gaussian proxy. The proxy yields good results, particularly for relatively low volatility levels.

LGM1F Volatility	**MC Value (€)**	**NI Value (€)**	**Proxy Value (€)**
0,5%	432	431	435
1%	629	628	642
1,5%	845	841	874
2%	1066	1062	1116

Table 3: Calculation of CVA using The numerical integration method

To ensure the robustness of the Gaussian approximation, we compared the density of the simulated value of an interest rate swaps portfolio at a future date with the densities obtained using two other alternative methods, for LGM1F volatility levels ranging between 50 and 200 basis points. The densities of the three methods are shown in Figure 5. This test demonstrates that the Gaussian proxy offers a good compromise between accuracy and computational time (13 seconds for the Gaussian proxy method, 72 seconds for the numerical integration method, and 33 minutes for the Monte Carlo method).

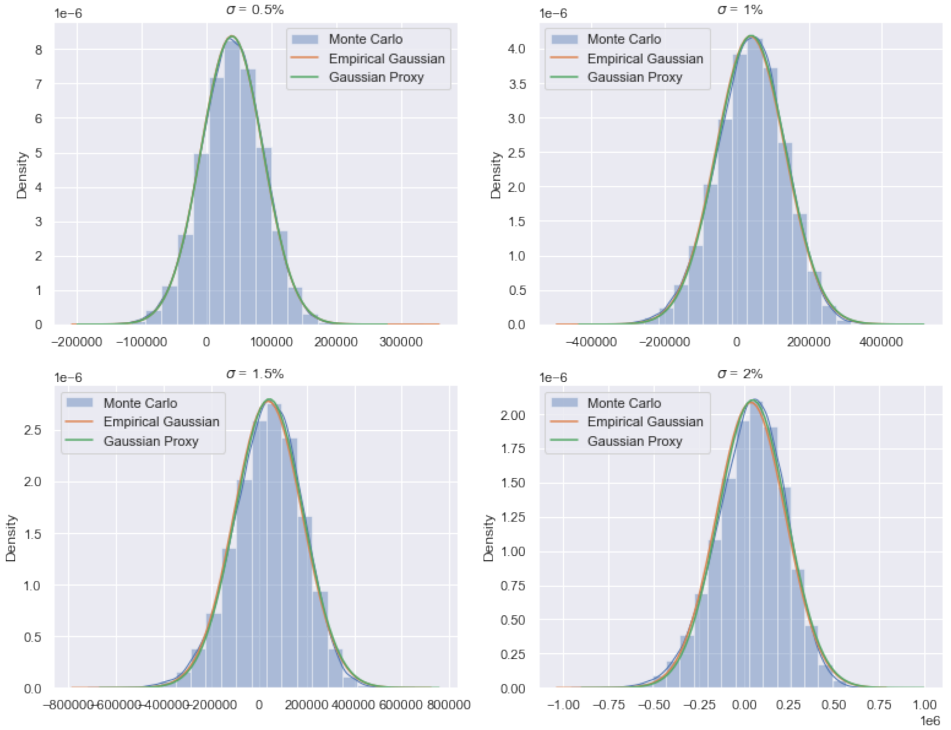

Figure 5: Comparison of Density Functions for Interest Rate Swap Portfolio Simulation under LGM1F Volatility

5.3 Numerical tests for CVA Calucation by GPR and Bayesian Quandarure

In this subsection, we will revisit the previous numerical example of an interest rate swap portfolio to evaluate the performance of the Bayesian quadrature in terms of accuracy and computational efficiency. As a benchmark, we compare it to the CVA calculation using the Monte Carlo method with 60,000 simulations and 500 time steps to compute the Expected Exposure profile across the portfolio's maximum maturity. As explained earlier, Bayesian quadrature (BQ) and Gaussian process regression (GPR) training occurs along the time axis for numerical integration, as well as on the risk factors driving the portfolio, represented here by the Gaussian variable X_t. Therefore, we conducted tests by varying the number of points used for training GPR and BQ.

Figure 6 displays values of the portfolio consisting of 400 interest rate swaps at a future date, calculated simultaneously using a traditional swap pricer and the GPR algorithm for 10,000 simulated values of the Gaussian variable X_t.

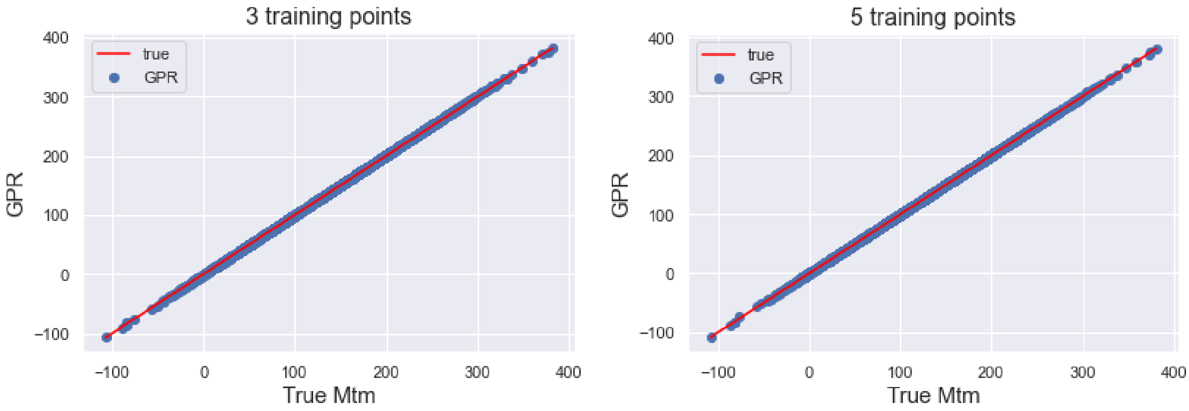

Figure 6: Comparison of Monte Carlo prices and GPR prices for 10,000 simulations on a given discretization date.

The calculation of the CVA and the Expected Exposure involves 60,000 simulations of the swap rate portfolio value at for 500 discretization dates. This process is time-consuming since each of the 400 swaps is individually valued for all 60,000 simulations. In the following numerical test, instead of calculating the portfolio value 60,000 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 above graphs show that the GPR provides an excellent approximation with only 5 training points.Therefore, we will select five prices to train the GPR algorithm at each future date for the valuation of the portfolio's mark-to-market. Subsequently, these valuations will be used to compute the expected exposures necessary for calculating the portfolio's CVA. Concurrently, we will employ Bayesian quadrature, built upon Gaussian process regression, to compute the numerical integral underlying the CVA calculation. To achieve this, a very limited number of Expected Exposure computations will be used to derive the entire Expected Exposure profile, which serves as the cornerstone for CVA computation. Figure 7 displays the Expected Exposure profiles of the portfolio of 400 interest rate swaps for 15, 20, 30, and 40 observation points used in training the Bayesian quadrature algorithm

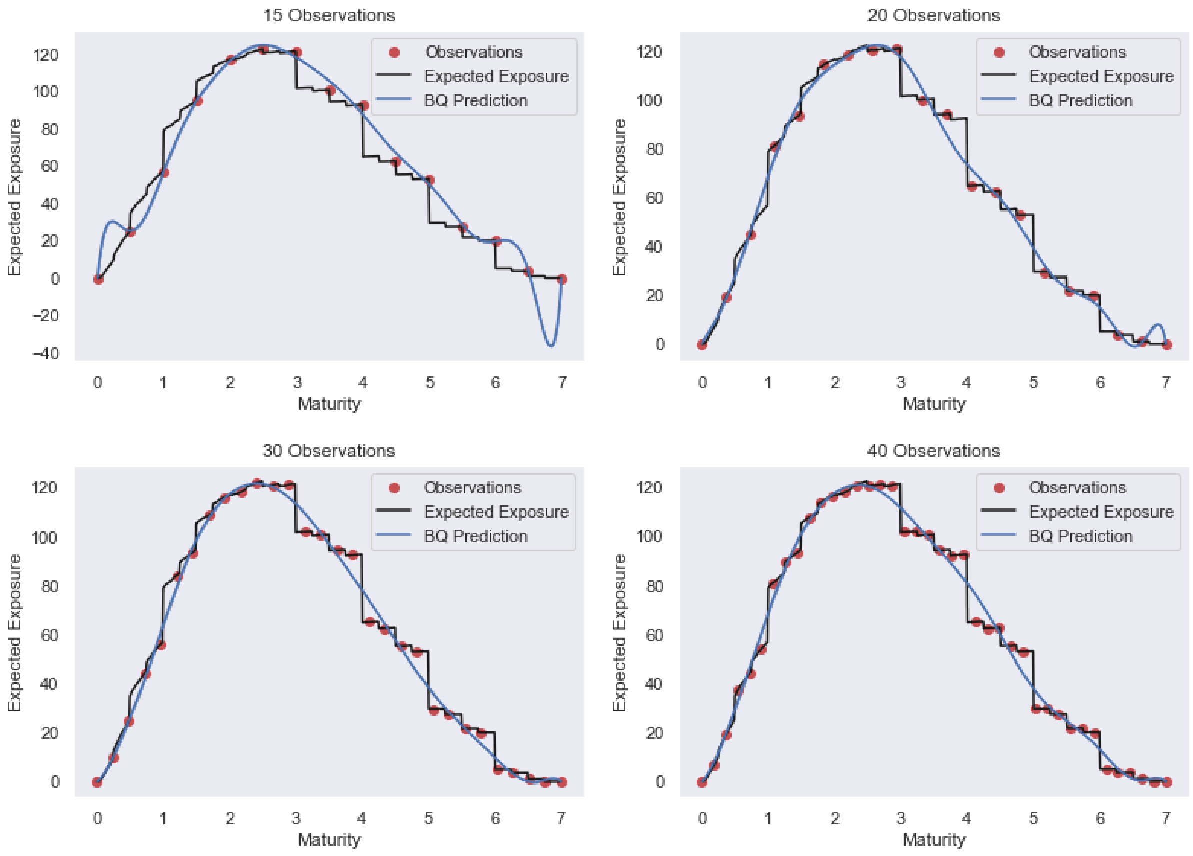

Figure 7: The expected exposure of the portfolio of 400 interest rate swaps for 4 level of number of observations (15, 20, 30, and 40) used for training the Gaussian process.

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 31 minutes. We conducted a numerical test using 40 training points for the GPR, with training currently performed on the time axis, and where the portfolio value was calculated using Gaussian Process Regression (trained with only 5 points) for the 60,000 simulations. The Bayesian Quadrature method achieves a high level of accuracy while significantly reducing computation time to 9 seconds, compared to 31 minutes for Monte Carlo simulations. This indicates that

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 4: 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

under the conditions of this numerical test, Bayesian Quadrature is 276 times faster than the Monte Carlo method.

Remark 3. *We conducted similar numerical tests, this time for a comparative study between traditional CVA calculation methods (numerical integration and Monte Carlo simulations) and the Bayesian quadrature method for a portfolio of approximately one hundred swaptions. To achieve this, we used a proxy to value the swaptions with the Bachelier formula (48), assuming that the forward swap rate follows a Gaussian distribution (47) within the LGM1F model. This approach significantly improves computation time compared to valuing swaptions using the LGM1F formula (42). The numerical results confirm that the Bayesian quadrature method outperforms traditional approaches (numerical integration method and Monte Carlo Simulations) in terms of computation time while maintaining comparable accuracy.*

6 Conclusion

In this study, we have demonstrated the significant advantages of using Bayesian quadrature for CVA calculations within the One Factor Linear Gaussian Markov (LGM1F) model framework. Traditional methods for CVA calculation, while robust, are often computationally intensive and time-consuming. Our research focused on applying Bayesian quadrature to CVA calculation and comparing it to traditional approaches. Bayesian quadrature, a method based on Gaussian process regression (GPR), is known for its exponential convergence and numerical stability. By employing Bayesian quadrature, we have achieved substantial reductions in computation time without compromising accuracy, making it a compelling alternative for real-time risk management. To optimally apply the Bayesian quadrature method, we highlighted a relevant approximation for valuing a swaption within the LGM1F framework using an analytical formula. Specifically, we showed that the value of an interest rate swap portfolio approximately follows a Gaussian distribution, and in this case, the CVA calculation can be performed using a quasi-analytical formula.

The numerical results confirm that Bayesian quadrature not only outperforms traditional methods in terms of efficiency but also maintains comparable precision. This balance of speed and accuracy is critical for the practical implementation of CVA calculations, particularly under stringent regulatory requirements. Furthermore, our application to a portfolio of fixed-income derivatives, including interest rate swaps and swaptions, within the LGM1F model framework, underscores the versatility and robustness of Bayesian quadrature. This model is widely accepted for the valuation and risk management of such derivatives, making our findings highly relevant for financial institutions seeking to enhance their counterparty risk assessment processes. In conclusion, Bayesian quadrature presents a powerful tool for the efficient and accurate computation of CVA, offering a promising direction for future research and practical applications in financial risk management.

References

[1] Naslidnyk M. Gretton A. Briol F. X. Chen, Z. Conditional bayesian quadrature. *arXiv preprint arXiv:2406.16530*, 2024.

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

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

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

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

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

[7] 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.

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

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

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

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

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

[13] Guéneau H. Oswald P. Lehdili, N. Market risk assessment of a trading book using statistical and machine learning. *Preprint*, 2029.

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

[15] 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.

[16] Basel Committee on Banking Supervision. Minimum capital requirements for credit risk. *https://www.bis.org/basel*$_f$*ramework/chapter/CRE/*20*.htm*, 2023.

[17] Garnett R. Ghahramani Z. Duvenaud D. K. Roberts S. J. Rasmussen C. Osborne, M. Active learning of model evidence using bayesian quadrature. *Advances in neural information processing systems, 25*, 2012.

[18] Michael Pykhtin and Dan Rosen. Pricing counterparty risk at the trade level and cva allocations. *Finance and Economics Discussion Series (FEDS)*, 2010.

[19] C. E. Rasmussen and C. K. I. Williams. Gaussian processes for machine learning. *MIT Press. Cambridge, Massachusetts*, 2006.

[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.