End-of-Studies Internship Report

Modeling Forward Initial Margin and Counterparty Credit Risk in Uncleared Derivatives

Université Paris Cité
Master M2MO

Ernst & Young
Quantitative Advisory Services

Author:
SANGLIER Nathan
nathan.sanglier@etu.u-paris.fr

EY Supervisors:
ARCHIMBAUD Antoine, COLIN Arthur

M2MO Jury:
CRÉPEY Stéphane

Date: September 5, 2025
Academic Year: 2024-2025

Acknowledgements

I would like to express my sincere gratitude to EY QAS team for providing me the invaluable opportunity to complete my internship with them. This experience has been greatly rewarding, and I am grateful for the trust and resources the company extended to me during my time there. I would like to thank my supervisors, Mr. Archimbaud Antoine, and Mr. Colin Arthur for their guidance and support. Their insights were of most importance in helping me navigate the challenges of this internship and in shaping my learning experience.

Finally, I wish to extend my appreciation to my university and professors for equipping me with the knowledge and skills necessary to make the most of this opportunity.

Abstract

This report focuses on the modeling of forward initial margin (IM) in the context of counterparty credit risk. In particular, we may answer the following questions:

1. Where does this forward IM appear and why is it important to model and estimate it? Is there clear regulatory guidance on this topic, as for the spot IM (ie. IM exchanged in the "real world")?
2. The common practice is to make a Gaussian distribution assumption. Is it relevant to make a Johnson distribution assumption instead? What does it implies ?
3. Are there alternatives that don't need distributional hypothesis? Can neural networks be used to estimate forward IM?

In Section 1, we highlight the need to model such quantity as part of the prudential framework, ie. the bank capital requirements. We follow a top-down approach starting from the computation of RWA, up to the modeling of collateralized exposure. **In Section 2**, we investigate where spot and forward IM appear beyond the prudential framework and we provide the main methods for calculating the spot IM. We also give the mathematical definition of forward IM and the general procedure for its estimation in the context of Monte-Carlo simulations, which are necessary for computing the expected exposure. **In Section 3**, we focus on the forward IM models studied in this report: nested Monte-Carlo, Monte-Carlo with Gaussian distribution, Monte-Carlo with Johnson distribution, and neural networks quantile regression. We also provide a exhaustive state-of-the-art of the other techniques. **In Section 4**, we present a detailed analysis of the performance of the forward IM models for two case studies: a European put option where the stock price follows a Black-Scholes dynamics, and a swaption where the riskfree rate follows a one factor Hull-White dynamics. **Finally, Section 5** summarizes the findings of this report and discuss potential further developments in the field of forward IM modeling. Several technical appendices are also available for the reader unfamiliar with the mathematical topics discussed. A Github repository is available at Nathan-Sanglier/EY-Initial-Margin.

Throughout this report, we consider a frictionless and arbitrage-free continuous-time financial market with a finite time horizon T. In addition and unless stated otherwise, let $(\Omega, \mathcal{F}, \mathbb{Q})$ be the probability space with the market filtration $\mathcal{F} = (\mathcal{F}_t)_{0 \leq t \leq T}$ and the risk-neutral probability measure $\mathbb{Q}$, with $\mathbb{E}[\cdot]$ being the expectation under $\mathbb{Q}$. We also denote the physical probability measure $\mathbb{P}$.

Contents

1 Initial Margin in the Regulatory Framework

In this section, we introduce the necessity to model forward IM in the context of the prudential Basel frameworks. Thus, it is important to understand and define the metrics used for a bank capital requirements.

1.1 The Basel frameworks

The Basel Committee of Banking Supervision (BCBS) is a committee of the Bank for International Settlements (BIS) responsible for defining global standards for banking regulation worldwide. The Basel frameworks focus only on the banking sector and have no legal power. Indeed, each jurisdictional regulatory body is responsible for adopting and adapting the BCBS guidance into local regulation. For instance, in the European Union (EU), the European Commission proposes legislative texts with the help of the European Parliament, the European Banking Authority (EBA), and the European Central Bank (ECB). Some of them are directly applicable in all EU member States (regulations) while others (directives) must be transposed into the national law by national competent authorities, like Autorité de Contrôle Prudentiel et de Résolution (ACPR) in France.

The first Basel Accord was published in the 1980's and has evolved through subsequent Accords into the cumulative Basel framework, which is based on three pillars (in addition to separate regulatory tools for leverage and liquidity aspects which are not studied here):

1. **Capital requirements**. The bank has to meet a minimum level of solvency at all times by having enough capital to cover the risk of expected and unexpected losses, based on capital ratio (CR):

$$\text{CR} := \frac{\text{Capital}}{\text{RWA}}, \tag{1}$$

where RWA are the risk-weighted assets, which will be detailed later. Capital is highly important: if the value of bank assets falls, the losses are subtracted from its capital and not from customers deposits or debt holders. We can distinguish 3 types of capital. The first one is Common Equity Tier 1 (CET1) which includes common shares, retained earnings and is considered the most stable. The second one is Additional Tier 1 whose purpose is to absorb losses prior to or at the point of insolvency[1] and constitutes the

[1]Notice that insolvency is different from default and bankruptcy. Default occurs when a bank fails to miss a specific debt obligation (eg. missed coupon payment) and may lead to bankruptcy if the default can't be cured. Bankruptcy refers to the legal process through which the bank seeks relief from some of its debts by liquidating assets. On the other hand, insolvency means that the bank has not enough assets or cash flows to cover its liabilities. Insolvency does not always imply default as the bank could still meet its debt obligations by borrowing money or acknowledging incapacity to meet future payments. However, it may lead to debt restructuring (modify terms with creditors to survive) or bankruptcy directly if no other solution is conceivable.

Tier 1 capital with CET1. The last one is Tier 2 capital, used to absorb losses in the event of liquidation and forms the total capital with the tier 1 capital. From the Basel III framework published by the BCBS in 2010, and implemented legally in the EU as Capital Requirement Regulation (CRR) and Capital Requirement Directive (CRD) IV as of 1st January 2014, the minimum capital requirements (MCR) are:

$$\text{MCR} \coloneqq \begin{cases} 4.5\% & \text{if Capital} = \text{CET1}, \\ 6.0\% & \text{if Capital} = \text{Tier 1}, \\ 8.0\% & \text{if Capital} = \text{Total Capital}. \end{cases} \tag{2}$$

There also exists capital buffers (conservation, countercyclical and systemic) to protect banks against the variability of their capital requirements. For more information, consult RBC20 [oBS20a].

2. **Supervisory review process.** This pillar focuses on forward-looking capital adequacy of the bank, assessed internally for each material risk and is required annually through the Internal Capital Adequacy Assessment Process (ICAAP) as part of the Supervisory Review and Evaluation Process (SREP). Moreover, it provides sound guidelines about risk management and reporting, as well as stress tests (bank performance during simulated macroeconomic and market adverse scenarios). which are also controlled by supervisors during the SREP, who could require higher capital ratios or impose qualitative measures for risk management.

3. **Market disclosure.** In order to ensure transparency and accountability, this last pillar requires banks to publicly disclose key information on their risk exposure, capital adequacy, and governance and risk management practices.

Since its publication, the Basel III framework has been reformed and the latest modifications have been proposed in December 2017. They have been legally implemented in the EU since 1st January 2025 with CRR III and CRD VI.

In this report, we consider the regulations related to Counterparty Credit Risk (CCR), defined as the risk that a counterparty to a transaction could default before the final settlement of the transaction's cash flows in cases where there is a bilateral risk of loss (see CRE50 [oBS24a]). CCR mainly relates to the trading book, where financial instruments are recorded in the balance sheet as mark-to-market if they are exchange-traded (eg. futures), or mark-to-model if they are traded over-the-counter (OTC). Notice that some OTC derivatives can be traded through clearing houses (cleared OTC derivatives). Moreover, even if a credit valuation adjustment (CVA) has been charged to the client in addition to the derivative price, the bank still needs to put aside capital reserves as the credit situation of the counterparty can change during the instrument life. CCR is opposed to credit risk associated with the banking book, where financial instruments are recorded by their amortized cost (eg. loans, mortgages). Fore more information,

see CRE51 [oBS20b].

1.2 CCR for uncleared OTC derivatives

When computing its RWA, the bank has the choice to either use internal methods or standard approaches. While internal methods are tailored to the portfolio of the bank, they must also be authorized by the competent authorities and require more investment. According to CRE51 [oBS20b], the CCR RWA for uncleared OTC derivatives are calculated either by the standardized or the IRB[2] approaches to credit risk. In all cases (see CRE20 [oBS24b] and CRE31 [oBS20f]):

$$\text{RWA} := \frac{1}{0.08} \cdot \sum_j w_j \text{EAD}_j, \tag{3}$$

where EAD_j is the Exposure at Default of counterparty j, and w_j is the weight attributed to this counterparty. Notice that $1/0.08 = 12.5$ enables to express total required capital directly as RWA. The difference between the standardized and IRB approaches lies in the calculation of w_j. While it follows simple predetermined grids for standardized approach (see CRE20 [oBS24b]), it is more complex for the IRB approach. According to CRE31 [oBS20f], it features the one year probability of default (PD) calculated based on the rating of the counterparty, and the loss given default (LGD) calculated based on the EAD of the counterparty:

$$\begin{aligned} w_j := & \left(\Phi \left(\sqrt{\frac{1}{1-R_j}} \Phi^{-1}\left(\text{PD}_j\right) + \sqrt{\frac{R_j}{1-R_j}} \Phi^{-1}\left(0.999\right) \right) - \text{PD}_j \right) \\ & \times \text{LGD}_j \cdot \frac{1 + (M_j - 2.5) b_j}{1 - 1.5 b_j}, \end{aligned} \tag{4}$$

where $b_j = (0.11852 - 0.05478 \ln(\text{PD}_j))^2$, $\Phi(\cdot)$ is the cdf.cumulative distribution function (cdf.) of standard Gaussian distribution, R_j is the correlation parameter (see CRE31 [oBS20f]), and M_j the maturity factor (see CRE53 [oBS20c]), such that in the general case:

$$\begin{cases} R_j & := 0.12 \frac{1-e^{-50\text{PD}_j}}{1-e^{-50}} + 0.24 \left(1 - \frac{1-e^{-50\text{PD}_j}}{1-e^{-50}}\right), \\ M_j & := \min\left(1 + \frac{\int_1^T \text{EE}(t) B_0(t) dt}{\int_0^1 \text{EEE}(t) B_0(t) dt}, 5\right), \end{cases} \tag{5}$$

where $B_0(t)$ is the discount factor (ie. the market price of the zero-coupon of maturity t), T is the last maturity of contracts related to counterparty j (considering a run-off portfolio), and $\text{EE}(t)$ is the expected exposure at time t, which will be detailed later. For the IRB approach presented here, the PD is always estimated internally (see CRE36 [oBS20d]), and the LGD can be estimated using either the F-IRB approach or a fully internal one (A-IRB), see CRE32 [oBS20e]. Herafter, we will only focus on the estimation of EAD.

While the calculation of EAD obeys to simple accounting rules in the context of credit risk

[2]IRB: Internal Rating Based.

related to on-balance sheet items (see article 166 of CRR III), it is a much tougher task in the context of CCR for uncleared OTC derivatives as they are marked-to-model and require posting/receiving collateral in order to mitigate excessive credit exposure. Indeed, the derivatives contracts related to a counterparty are grouped into a unique legal framework called the ISDA Master Agreement (IMA), which defines the general terms of the transactions. In this structure, derivatives can be grouped (not a requirement) into several Master Netting Agreements (MNA), which are in turn made of different Credit Support Annexes (CSA).

When computing EAD for uncleared OTC derivatives, the bank has the choice to use either the standardized approach (SA-CCR) based on CRE52 [oBS20h], or the internal model method IMM). We will only focus on the IMM. For a counterparty j, denoting $\text{EAD} \coloneqq \text{EAD}_j$ for simplicity, we have based on CRE53 [oBS20c]:

$$\text{EAD} \coloneqq \alpha \cdot \max\left(\text{EEPE}, \text{EEPE}_{\text{stressed}}\right), \tag{6}$$

where $\alpha \geq 1.4$ is the regulatory multiplier, and EEPE and $\text{EEPE}_{\text{stressed}}$ are the Effective Expected Positive Exposures calculated respectively under unstressed and stressed market conditions (calibration of model parameters will change), and such that:

$$\text{EEPE} \coloneqq \int_0^{T\wedge 1} \text{EEE}(t)dt, \tag{7}$$

where

$$\text{EEE}(t) \coloneqq \sup_{0\leq s\leq t} \text{EE}(s) \tag{8}$$

is the Effective Expected Exposure at time t. This formula enables to have a conservative estimation on the calculation of EEE to reflect the fact we consider in theory a run-off portfolio (the portfolio values amortizes through time), whereas it is not true in practice. Moreover,

$$\text{EE}(t) \coloneqq \mathbb{E}\left[E_t \mid \mathcal{F}_0\right] \tag{9}$$

is the Expected Exposure at time t (assuming the counterparty defaults at time t), where (E_t) is the collateralized exposure which needs to be modeled taking into account the IMA legal framework (see Section 1.3). Notice that the expectation is computed under the risk-neutral one $\mathbb{Q}$ (ie. calibrating with market implied data), but it can also be computed under the historical probability measure $\mathbb{P}$ (ie. calibrating the modeling of (E_t) with at least three years historical market data). For more information, see article 292 of CRR III.

In the definition (9) of EE, we have assumed that there is no specific wrong-wray risk. According to [oBS24a], such risk arises when the exposure to the counterparty is positively correlated with its probability of default, due to the nature of the transaction. In other words, we have

assumed that:

$$\mathbb{E}\left[E_t \mid \tau^\star = t, \mathcal{F}_0\right] = \mathbb{E}\left[E_t \mid \mathcal{F}_0\right], \tag{10}$$

where $\tau^\star$ is the materialized default time of the counterparty[3]. From articles 284 and 291 of CRR III, expected exposure for instruments subject to specific wrong-way risk should be calculated separately. For instance, [Pyk24] and [AK24] investigate the modeling of EE subject to a modified "leveraged" wrong-way risk by applying a Gaussian copula to link the counterparty default event with the increment of the portfolio value over the MPOR defined in Section 1.3.

1.3 Modeling collateralized exposure

The expected exposure defined in (9) relies on the stochastic process $(E_t)_{0\leq t\leq T}$, which is the collateralized Exposure. It represents the estimated "net" exposure of the bank if the counterparty defaults and the amount of losses becomes known at a future time t. Let us denote M a MNA inside the ISDA agreement[4] I, and C a CSA inside a specific MNA. We then have:

$$E_t = \sum_{\mathrm{M}\in\mathrm{I}} E_t^{(\mathrm{M})} = \sum_{\mathrm{M}\in\mathrm{I}} \left(\sum_{\mathrm{C}\in\mathrm{M}} \left(V_t^{(C)} + \mathrm{UTF}_t^{(C)} - K_t^{(C)} \right) \right)_+, \tag{11}$$

where $V_t^{(C)}$ is the mark-to-market process (to the bank) associated to CSA C, so that MtMs are aggregated (netted) at the CSA level. Notice that we use the term "mark-to-market", but it is in fact a mark-to-model process as we focus on the pricing of OTC products at future dates. Moreover, $\mathrm{UTF}_t^{(C)}$ represents the unpaid trade flows to the bank (flows related to contractual cash flows) at the CSA level during the margin period of risk (MPOR). $K_t^{(C)}$ is the effective collateral available to the bank. For simplicity, assume we have only one CSA (equivalently a "portfolio") in our ISDA agreement, ie.

$$E_t \coloneqq \left(V_t + \mathrm{UTF}_t - K_t\right)_+, \tag{12}$$

where (V_t) is the portfolio value. In order to model the quantities in (12), we need to make some hypotheses about collateral exchanges (margin flows) and trade flows. We will use the model of [Pyk09], described in [ASP16] as the "Classical+" model, but enhanced with an IM component.

Assumptions on margin flows. First, we define the MPOR represented in Figure 1 as the period of time between the last successful margin call and the time when the amount of loss becomes known and transaction is closed out (ETD[5]), assuming the counterparty has defaulted

[3] This means the time at which the counterparty default is known. In practice the true default time can happen any time during the MPOR defined in Section 1.3.

[4] One may find the term "netting set" in the Basel documentation. Under the IMM, it represents in general a CSA, see Article 272 of CRR III.

[5] Early Termination Date.

between these two dates.

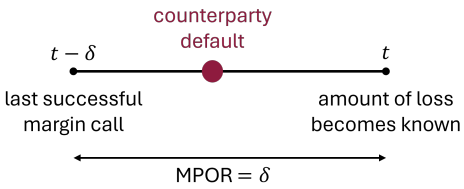

Figure 1: Stylized representation of the MPOR.

Indeed, a firm under stress may stop fully honoring margin calls by attempting a dispute over the collateral requirement calculation, before an Event of Default would be officially issued, and a portfolio termination date decided. Here, we set a fixed MPOR $\delta = 10$ business days, which is coherent with the standard case in the regulation (see article 304 of CRR III). In practice, the timeline of events prior to ETD is more complex, so that the model may be enhanced as in [ASP16]. We also consider that both the bank and its counterparty stop paying margins at $t-\delta$ (t being the ETD) and that there is no minimal amount threshold for receiving or posting collateral (ie. no margin threshold). Thus, for all $t \in [\delta, T]$:

$$\begin{cases} K_t &= \mathrm{VM}_{t-\delta} + \mathrm{IM}_{t-\delta} \\ \mathrm{VM}_t &= V_t, \end{cases} \tag{13}$$

where VM and IM are respectively the Variation Margin and *received* IM (ie. IM posted by the counterparty to the bank), calculated at the CSA level[6]. VM captures the day-to-day movement of the portfolio, and we have assumed perfect continuous variation margining as it is not the focus of our study. IM covers portfolio gap risk (also called close-out risk) which is the risk of adverse market movement during the MPOR, at some high percentage. Although the term "initial" may be misleading, notice that both these quantities are dynamically refreshed. However, the IM amount must be held in a default-remote way (eg. by a custodian), so that IM posted by a counterparty should be immediately available to it should the other counterparty default. Since 2015, posting/receiving VM and IM is mandatory on a regular basis (eg. daily) and follows specific uncleared margin rules (UMR) under the BCBS and the International Organization of Securities Commissions (IOSCO) guidelines [oBS20g]. The MPOR implies that the counterparty cannot default before δ (otherwise the bank would not enter the trade), hence the integral in equation (7) starts at $t = \delta$.

Assumptions on trade flows. We consider that both the bank and the counterparty continue paying all trade flows during the MPOR:

$$\mathrm{UTF}_t = 0. \tag{14}$$

This is not a too restrictive hypothesis (and greatly simplifies implementations) as most of the

[6]Notice one could accrue collateral from $t-\delta$ to t at the riskfree rate, such that $K_t = e^{\int_{t-\delta}^t r_s ds}(\mathrm{VM}_{t-\delta} + \mathrm{IM}_{t-\delta})$.

time, there is no trade flow between $t-\delta$ and t ($\delta = 10$ days). Moreover, the final quantity EAD is based on an integral of the collateralized exposure during the portfolio life, making the consideration of trade flows less significant. Nonetheless, any possibility of the bank making a trade payment to the counterparty in the future results in upward spikes in the expected exposure profile. These spikes appear because the portfolio value (from the bank point of view) will jump following the payment by the bank, but the counterparty does not update the VM in consequence, ie. there is no offsetting. In some situations, it could yield to an undesirable instability in expected exposure[7] (or CVA, defined later). As mentioned before, this exposure model can be refined, to better account for the timeline complexity of IMA/CSA events, eg. see [ASP16].

In summary, one needs to estimate $\forall \delta \leq t \leq T \wedge 1$:

$$\mathrm{EE}(t) = \mathbb{E}\left[(V_t + \mathrm{UTF}_t - \mathrm{VM}_{t-\delta} - \mathrm{IM}_{t-\delta})_+ \mid \mathcal{F}_0\right]. \tag{15}$$

In particular, we need to define the quantity IM_t. Typically, the expectation under $\mathbb{Q}$ will be approximated by its empirical version, and the expected exposure will be computed on a time grid[8] $G \coloneqq [\delta =: t_0, \cdots, t_K \coloneqq T \wedge 1]$, such that for all $k \in 0:K$,

$$\mathrm{EE}(t_k) \approx \frac{1}{M} \sum_{m=1}^{M} \left(V_{t_k}^{(m)} + \mathrm{UTF}_{t_k}^{(m)} - \mathrm{VM}_{t_k-\delta}^{(m)} - \mathrm{IM}_{t_k-\delta}^{(m)}\right)_+. \tag{16}$$

It implies the necessity to:

1. Model and compute the mark-to-market process on each path (m) and *at least*[9] at each timestep t_k.
2. Model and compute the IM on each path (m) and at each date $t_k - \delta$. We call this quantity **forward IM**.

[7] For a comprehensive study, see [ASP17].

[8] A finer mesh is usually used for short-term dates in the time grid G, with increasing timestep towards maturity.

[9] We will see that computing IM also requires computing portfolio value at given dates that may not be in our time grid G.

2 Computing Initial Margin

2.1 Initial margin in the XVA framework

As explained in Section 1, computing spot IM is needed for "real-world" transactions of OTC derivatives under the BCBS-IOSCO guidelines [oBS20g], and forward IM is mandatory from a prudential point of view. Moreover, it is also required in the context of fair value pricing of a derivative product. In incomplete markets, a model price assuming a completeness hypothesis has to be adjusted by the valuation of market imperfections. Derivative dealers charge to their clients so-called XVAs, standing for the different value adjustments, meant to account for counterparty risk and its capital and funding implications. In particular, posting IM dynamically during the life of the contract induces a funding cost which is known as Margin Value Adjustment (MVA) [VL24]:

$$\mathrm{MVA}_0 = \int_0^T f(t)\mathbb{E}\left[e^{-\int_0^t r_s ds}\mathrm{PIM}_t \mid \mathcal{F}_0\right] dt, \tag{17}$$

where f is the risky funding spread of the bank, (r_t) is the risk-free rate, and PIM_t is the *posted* IM by the bank. In addition, the Credit Value Adjustment (CVA) represents the difference between the price of the product with and without default risk the counterparty. Let us assume default time of the bank is beyond T and denote $\tau := \tau^\star - \delta$ where $\tau^\star$ is the materialized default time (as explained in Section 1.2). According to [BCG+24] and [CBB14], we have:

$$\begin{aligned}\mathrm{CVA}_t &= \mathbb{E}\left[e^{-\int_t^{\tau+\delta} r_s ds}\mathbf{1}_{t<\tau+\delta<T}(1-R)E_{\tau+\delta} \mid \mathcal{F}_t\right] &(18)\\ &= (1-R)\mathbf{1}_{t<\tau+\delta}\mathbb{E}\left[e^{-\int_t^{\tau+\delta} r_s ds}\mathbf{1}_{\tau+\delta<T}E_{\tau+\delta} \mid \mathcal{F}_t\right], &(19)\end{aligned}$$

where R is the recovery rate of the counterparty upon default, and E_τ is the collateralized exposure defined in (12) which features the IM received by the bank. We then have:

$$\begin{aligned}\mathrm{CVA}_t =&(1-R)\mathbf{1}_{t<\tau}\mathbb{E}\left[\mathbb{E}\left[e^{-\int_t^{\tau+\delta} r_s ds}\mathbf{1}_{\tau+\delta<T}E_{\tau+\delta} \mid \mathcal{F}_\tau\right] \mid \mathcal{F}_t\right]\\ &+ (1-R)\mathbf{1}_{\tau<t<\tau+\delta}\mathbb{E}\left[e^{-\int_t^{\tau+\delta} r_s ds}\mathbf{1}_{\tau+\delta<T}E_{\tau+\delta} \mid \mathcal{F}_t\right]. \end{aligned}\tag{20}$$

Assuming a reduced-form credit default model for the counterparty and denoting $\gamma(t)$ its default intensity (ie. the hazard rate), we have:

$$\begin{aligned}\mathrm{CVA}_t =&(1-R)\mathbf{1}_{t<\tau}\mathbb{E}\left[\int_t^{T-\delta}\mathbb{E}\left[e^{-\int_t^{s+\delta} r_u du}E_{s+\delta} \mid \mathcal{F}_s\right]\gamma(s)e^{-\int_t^s \gamma(u)du}ds \mid \mathcal{F}_t\right]\\ &+ (1-R)\mathbf{1}_{\tau<t<\tau+\delta}\mathbb{E}\left[e^{-\int_t^{\tau+\delta} r_s ds}\mathbf{1}_{\tau+\delta<T}E_{\tau+\delta} \mid \mathcal{F}_t\right]. \end{aligned}\tag{21}$$

For more information on CVA, MVA, other XVA and the implications of collateralization, refer to [CBB14], [Gre15]. In conclusion, as for the regulatory point of view, we face the challenge of

computing a forward (and spot) IM, re-affirming the importance of such problem.

2.2 Spot initial margin

As explained above, calculating spot IM (ie. IM at the date $t = 0$) is done on a daily basis by the banks, in order to announce margin calls to their counterparties. It falls under clear regulations and requirements described in [oBS20g], where financial institutions can either use a standardized margin schedule or an IM model.

2.2.1 Standardized margin schedule

This standardized method is provided by the BCBS and the IOSCO, such that:

$$\mathrm{IM}_0 = (0.4 + 0.6 \cdot \mathrm{NGR}) \cdot \sum_j \mathrm{GIM}_j, \tag{22}$$

where $\mathrm{GIM}_j = w_j \cdot \mathrm{Notional}_j$, is the gross IM amount expressed based on the notional size of the derivative contract j inside the portfolio (ie. inside a CSA), and on a margin rate w_j pre-determined for specific asset classes given in Appendix A of [oBS20g]. Notice that a limited degree of netting may be performed to compute the notional amount that is applied to the margin rate. Moreover, according to [Aut19], the NGR ratio[10] is:

$$\mathrm{NGR} := \frac{\left(V_0^{(C)}\right)_+}{\sum_{j \in C} \left(V_0^{(j)}\right)_+}, \tag{23}$$

where $V_0^{(C)} := \sum_{j \in C} V_0^{(j)}$ is the (netted) mark-to-market value of the portfolio at CSA level, and $V_0^{(j)}$ is the mark-to-market value of instrument j inside the CSA considered.

2.2.2 ISDA standard initial margin model (SIMM)

If the bank does not use the standardized approach, then it uses its own IM model, which regroups either the ISDA[11] SIMM or an internal model. In accordance with regulatory standards [oBS20g], the ISDA suggested a method in [Ass23] (first version in 2016) to prevent disputes between counterparties, that is straightforward to implement while being more complex than the standardized approach. This method is now widely used by the industry.

In a portfolio made of several instruments, the methodology considers four product classes: Interest Rates and Foreign Exchange (RatesFX), Credit, Equity, and Commodity. Moreover,

[10] NGR: ratio of the net current replacement cost to gross current replacement cost.
[11] ISDA: Interest Swap Derivatives Association.

there are six risk classes: Interest, Credit (Qualifying), Credit (Non-Qualifying), Equity, Commodity, FX. The margin that will be calculated for each risk class is defined as the sum of (potentially) four margin classes: Delta, Vega, Curvature[12], and Base Correlation margins[13]. These margins are defined based on sensitivities with respect to risk factors. Notice that the method to compute these sensitivities is left up to the bank. Moreover, they are assigned to risk buckets according to the characteristics of the risk factor associated. The first step is to classify our instruments and related sensitivities with Algorithm 1. We can then calculate the margins with Algorithm 2 (see [Ass23] for more details).

Algorithm 1: ISDA SIMM - Sensitivities Classification

for each instrument I_j **do**
- assign it to a product class PC_p and risk class RC_r
- **for** each risk factor RF_k related to I_j **do**
  - assign it to a risk bucket RB_b
  - **for** each margin class MC_m **do**
    - compute the sensitivity $s_{j,k,m}$ of I_j wrt. RF_k, for MC_m
    - compute the weighted sensitivity $ws_{j,k,m}$ based on RB_b

Algorithm 2: ISDA SIMM - Initial Margin Computation

Input: $\forall r \neq s$, risk classes correlation $\psi_{r,s}$

for each product class PC_p **do**
- **for** each risk class RC_r **do**
  - **for** each margin class MC_m **do**
    - **for** each risk bucket RB_b **do**
      - $K_b \leftarrow$ aggregation of $\{ws_{j,k,m}$ st. $I_j \in \text{PC}_p$ and RC_r, $\text{RF}_k \in \text{RB}_b$, $\forall (j,k)\}$
    - $\text{margin}_m \leftarrow$ aggregation of $\{K_{m,b}, \forall b\}$
  - gross initial margin $\text{IM}_{p,r} \leftarrow \sum_m \text{margin}_m$
- initial margin $\text{SIMM}_p \leftarrow \sqrt{\sum_r \text{IM}_{p,r}^2 + \sum_{s \neq r} \psi_{r,s} \text{IM}_{p,r} \text{IM}_{p,s}}$

total initial margin $\text{IM}_0 \leftarrow \sum_p \text{SIMM}_p$

2.2.3 Internal model

If the financial institution does not use the ISDA SIMM for its own IM model, then it can use an internal one, which should be approved by the EBA (European Banking Authority) [Par24], and should respect the BCBS-IOSCO guidelines (see [oBS20g]). We do not provide examples of such model as our goal in this report is to calculate a forward IM, and not a spot IM that

[12] It represents the Gamma sensitivity.
[13] Only for credit qualifying, sensitivity to correlation between defaults of different credits within a basket.

would be exchanged in real life.

The spot IM must reflect an extreme but plausible estimate of an increase in the value of the instrument that is consistent with a one-tailed 99 per cent confidence interval over a 10-day horizon (in case variation margin is exchanged daily). The model calibration should be done based on an historical data period that do not exceed 5 years and that incorporates a period of significant financial stress. This period of financial stress should be identified and applied separately for each broad asset class in the portfolio. Notice that this calibration step is different from the one implied by the Basel capital requirements detailed in Section 1.2.

Moreover, a bank should not be allowed to switch between model and schedule-based margin calculations in an effort to cherry-pick the most favorable IM terms. However, a model-based IM can be used for one class of derivatives commonly dealt, and a schedule-based one for derivatives that are less routinely employed in its trading activity.

2.3 Forward initial margin

Contrary to the computation of spot IM analyzed in Section 2.2, there are no regulatory requirements nor standardized approaches for calculating the forward IM. Nonetheless, in compliance with guidelines for the spot IM provided in Section 2.2.3 and in [Aut19], we define the forward IM for our portfolio as[14] $\forall t \in [0, T]$:

$$\mathrm{IM}_t \coloneqq \left(\mathrm{VaR}^{\alpha}\left(\Delta V_{t+\delta} \mid \mathcal{F}_t\right)\right)_+ , \tag{24}$$

where $\Delta V_{t+\delta} \coloneqq V_{(t+\delta)\wedge T} - V_t$, and $\mathrm{VaR}^{\alpha}(Y) = \inf\{y \in \mathbb{R} \text{ st. } F_Y(y) \geq \alpha\}$ is the quantile of the random variable Y at level α, with $F_Y(\cdot)$ being the cdf. of Y under probability $\mathbb{Q}$. Thus, estimating the forward IM amounts to estimating a conditional quantile under $\mathbb{Q}$. In practice, we will choose $\alpha = 99\%$, as described in [oBS20g]. Notice that if the cdf. of Y is continuous, then $\mathbb{Q}\left(Y \leq \mathrm{VaR}^{\alpha}(Y)\right) = \alpha$. For rigorous mathematical definitions of conditional expectation and conditional value-at-risk (or conditional quantile), we refer the reader to Appendix A.1. of [MKN+18], Section 2. of [BCG+24], as well as [Kal02]. We assume here that all the assumptions are satisfied so that both these operators are defined and applied accordingly.

The construction of a forward IM model is typically done in five main steps. As described in Section 1.3 and given the expressions in Section 2.1, we will model and compute the initial margin $\mathrm{IM}_{t_i}^{(m)}$ on each path $(m) \in [|1, M|]$ at each evenly spaced date t_i in the grid:

$$\pi \coloneqq \{0 =: t_0, \cdots, t_N \coloneqq T \wedge 1\} , \tag{25}$$

[14] Notice that one could add the term $\mathrm{UTF}_{t+\delta}$ inside the value-at-risk. One could also discount the portfolio value at $t+\delta$ by $e^{-\int_t^{t+\delta} r_s ds}$. In order to be consistent with our collateralized exposure model of Section 1.3, we did not take them into account.

with timestep h. For simplicity, we assume $\delta \in \pi$ so that $(t_i + \delta) \wedge T \in \pi$, since we need to simulate our portfolio value both at $t_i + \delta$ and t_i given the formula (24). As in Section 1.3, we choose $\delta = 10$ days and we will take $h = 1$ day. Notice that the time grid G from Section 1.3 is not necessarily the same as the time grid π, even if in general we have $G \subseteq \pi$.

1. **Diffusing the risk factors.** In order to calculate our portfolio value at each $t_i \in \pi$ in a model-based approach, we first need to identify the risk factors $(X_t)_t$ associated to the portfolio (taking values in $\mathbb{R}^d$, $d \in \mathbb{N}^*$), and model their dynamics by stochastic differential equations (SDEs), eg. Black-Scholes model. Then, one should calibrate the model parameters with market-implied or historical data. Finally, simulate the (m)-th path $\{X_{t_0}^{(m)}, \cdots, X_{t_N}^{(m)}\}$ of these risk factors with a given discretization scheme. Notice that for complex models, the scheme used could produce numerical errors, that should be quantified.

2. **Pricing the portfolio.** Based on the risk factors paths previously computed, calculate the portfolio value paths $\{V_{t_0}^{(m)}, \cdots, V_{t_N}^{(m)}\}$ through the usual techniques such as closed-form formulas, Monte-Carlo methods, partial differential equations (PDEs) resolution, etc.

3. **Estimating the value-at-risk.** Assuming the risk factors to be a Markov process, we now need to calculate for each $i \in [|0, N|]$ and for each $m \in [|1, M|]$:

$$f_i\left(X_{t_i}^{(m)}\right) \coloneqq \mathrm{VaR}^{\alpha}\left(\Delta V_{t_i+\delta} \mid X_{t_i}^{(m)}\right), \tag{26}$$

where $(\Delta V_{t+\delta} \mid x) \coloneqq (V_{(t+\delta)\wedge T} - V_t \mid X_t = x)$, for $x \in \mathbb{R}^d$. For each $i \in [|0, N|]$, the function $f_i(\cdot)$ (or directly the values $f_i\left(V_{t_i}^{(m)}\right)$ in the case of nested Monte-Carlo) will be estimated from our "training" dataset:

$$\mathcal{D} \coloneqq \left\{\left(X_{t_i}^{(1)}, \Delta V_{t_i+\delta}^{(1)}\right), \cdots, \left(X_{t_i}^{(M)}, \Delta V_{t_i+\delta}^{(M)}\right)\right\}, \tag{27}$$

and/or from new simulations of risk factors[15]. We will see that it is often simpler to calculate for each $i \in [|0, N|]$ and for each $m \in [|1, M|]$:

$$g_i\left(V_{t_i}^{(m)}\right) \coloneqq \mathrm{VaR}^{\alpha}\left(\Delta V_{t_i+\delta} \mid V_{t_i}^{(m)}\right) \approx f_i\left(X_{t_i}^{(m)}\right), \tag{28}$$

where $(\Delta V_{t+\delta} \mid v) \coloneqq (V_{(t+\delta)\wedge T} - V_t \mid V_t = v)$. Again, for each $i \in [|0, N|]$, one can estimate the function $g_i(\cdot)$ from the following training dataset:

$$\mathcal{D}' \coloneqq \left\{\left(V_{t_i}^{(1)}, \Delta V_{t_i+\delta}^{(1)}\right), \cdots, \left(V_{t_i}^{(M)}, \Delta V_{t_i+\delta}^{(M)}\right)\right\}. \tag{29}$$

The forward IM is then simply estimated by taking the positive part of the estimate of $f_i(\cdot)$ or $g_i(\cdot)$.

[15] It will be the case for techniques involving nested Monte-Carlo simulations.

4. **Adjusting to external methodology**. At this point, there is little chance that our IM estimation at t_0 is consistent with the realized IM value for this portfolio (eg. given by SIMM), as the associated external IM methodology usually differs from our model, and our risk factors model may not be accurate enough. In order to adjust the two quantities at $t = t_0$ and trying to reduce the gap at $t > t_0$ to limit the number of exceptions in the backtesting (and to have a conservative value), it is possible to scale the IM estimations by a deterministic coefficient $a(t)$, as in [AAGL17]. However, this raises a number of questions about stability and backtesting of this scaling function, investigated in [CA24]. If the external methodology is based on datasets of historical information, [THM16] proposes to scale the IM estimation at t_0 and to re-weight $\{V_t^{(1)}, \cdots, V_t^{(M)}\}$ for $t > t_0$. This re-weighting step can be seen as a kind of measure change to the subjective measure $\mathbb{P}^*$ (perception of the probability $\mathbb{P}$) that is implied by the historical data available.

5. **Backtesting the method.** According to article 294 of CRR III, the last *mandatory* step is to backtest our estimation of value-at-risk. The standard approach in the industry (see [CA24] for an exhaustive review) is to define a backtesting that examines jointly the model's risk factors generation and accuracy of IM forecast. More precisely, [AAGL17] defines a procedure which involves diffusing B risk factors paths from their observed value at a given past time t during a backtesting horizon H (ie. up to time $t + H$). Then, one should estimate IM for these B risk factor values[16] and compare them to the realized IM value. Finally, one should aggregate the results for different past times and horizons. It is also possible to backtest the top-level expected exposure metric as in [AKN13].

In this report, we do not focus on the calibration of risk factors model parameters, nor on a sound backtesting although these are crucial tasks. The emphasis is put on the method used to estimate the value-at-risk functions $f_i(\cdot)$ (or at least the function g_i).

[16] It implies to diffuse the risk factors up to $t + h + \delta$.

3 Forward Initial Margin Models

3.1 A first approach: nested Monte-Carlo

A first naive approach to directly estimate the values $f_i\left(X_{t_i}^{(m)}\right)$ is to calculate the quantile in equation (26) empirically, giving rise to a nested Monte-Carlo. Indeed, for a given $m \in [|1, M|]$ we first need to simulate K iid. realizations of $X_{t_i+\delta}$ conditional on $X_{t_i} = X_{t_i}^{(m)}$, denoted $X_{t_i+\delta}^{(m,k)}$ for each $k \in [|1, K|]$. Then, we can compute the portfolio values $V_{t_i+\delta}^{(m,k)}$ associated, and the sample quantile at level α based on the K iid. samples $\Delta V_{t_i+\delta}^{(m,k)} := V_{t_i+\delta}^{(m,k)} - V_{t_i}^{(m)}$ of $\left(\Delta V_{t_i+\delta} \mid X_{t_i}^{(m)}\right)$.

In Figure 2, we show a stylized methodology of such approach, for a specific time $t_i \in \pi$ and a specific path $m \in [|1, M|]$. On the left graph, we generate the nested paths of the risk factor ($d = 1$). They are used on the right graph to compute the nested paths of portfolio value (in red), from which we display (in gray) the empirical distribution of $\left(\Delta V_{t_i+\delta} \mid X_{t_i}^{(m)}\right)$ and the associated empirical quantile $\hat{q}_i^{(m)}$. This value is the estimation of $f_i\left(X_{t_i}^{(m)}\right)$ defined in equation (26).

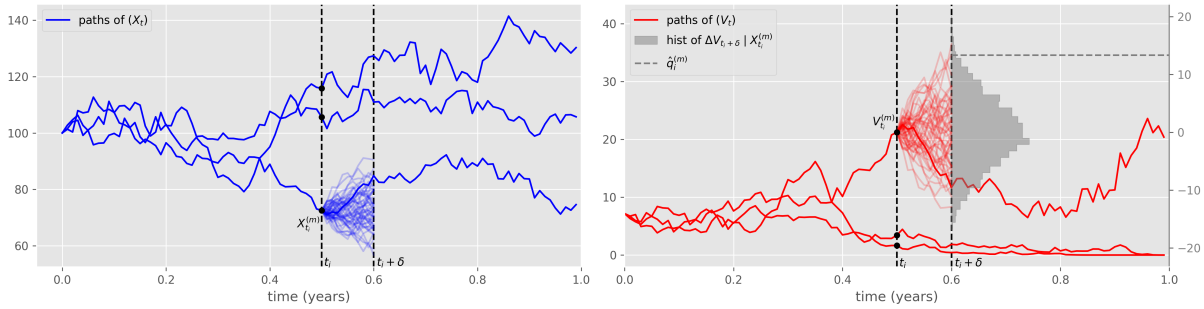

Figure 2: Nested Monte-Carlo, stylized methodology at a specific (t_i, m).

At this point, we need to specify what we mean by a sample/empirical quantile $\hat{q}_i^{(m)}$. Indeed, according to [HY96], there exist a large number of different definitions used for sample quantiles in statistical packages. Here, we consider the oldest and most studied version.

Definition 1 (Sample quantile)**.** *Let $\{Y_1, \cdots, Y_K\}$ be a set of iid. observations of a random variable Y with probability density function (pdf.) h, and denote $\{Y_{(1)}, \cdots, Y_{(K)}\}$ the associated order statistics. Then, the sample quantile at level α is:*

$$\hat{q} := \begin{cases} Y_{(\lfloor \alpha K \rfloor + 1)} & \text{if } \lfloor \alpha K \rfloor < \alpha K \leq \lfloor \alpha K \rfloor + 1 \\ Y_{(\lfloor \alpha K \rfloor)} & \text{if } \alpha K = \lfloor \alpha K \rfloor \,, \end{cases}$$

where $\lfloor \cdot \rfloor$ denotes the integer part. A visualization is given in Figure 3.

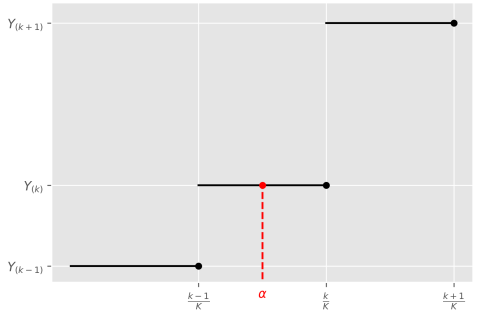

Figure 3: Sample quantile of Definition 1.

[OR01] shows that this estimator is biased[17] (ie. $\mathbb{E}[\hat{q}] \neq q$, where q is the true quantile), and all the more for extreme quantiles compared to central ones. The authors also provide a "quantile Central Limit Theorem".

Theorem 1 (Quantile CLT)**.** *Let $\hat{q}$ be the sample quantile of Definition 1 and q the true quantile, both at level α. Then we have:*

$$\sqrt{K}\,(\hat{q} - q) \xrightarrow[K\to+\infty]{\mathcal{D}} \mathcal{N}\left(0, \frac{\alpha(1-\alpha)}{h(q)^2}\right). \tag{30}$$

Further detail on sample and asymptotic bounds are investigated in [OR01] and [Leh98]. We see that the sample quantile is an asymptotically unbiased estimator, which implies that we need to simulate a high number K of nested paths, but simulating risk factors and pricing portfolio $K \times M \times N$ times is too costly. [GJ10] investigates in details the computational budget to be allocated based on a mean-square error criterion. [ATCD18] provides advanced nested Monte-Carlo based on GPU optimization in the more general context of XVA computations.

3.2 State-of-the-art

In order to alleviate the computational burden implied by nested Monte Carlo methods, it is common practice in the industry to rely on more advanced techniques that we group into different broad categories.

1. **Distribution-based methods.** The first step is to make an assumption on the distribution of $(\Delta V_{t+\delta} \mid v)$. Indeed, we will estimate the function $g_i(\cdot)$ instead of $f_i(\cdot)$ (see equations (26)-(28)) to avoid specifying a distribution in high dimension which implies estimating mixed moments that would not be feasible in practice. In particular, we will study the method GLSMC [AAGL17] where we make the hypothesis that $(\Delta V_{t+\delta} \mid v)$ follows a Gaussian distribution (see Section 3.3), as well as the method JLSMC [MKN+18] which makes the assumption that $(\Delta V_{t+\delta} \mid v)$ follows a Johnson distribution (see Section 3.4.1).

[17] This is not specific to the estimator studied here.

Once we have determined the adequate distribution, the next step is to estimate some metrics of the distribution (eg. moments, quantiles, etc.) and fit the distribution parameters based on these estimations. Usually, one estimates the first conditional raw moments of $(\Delta V_{t+\delta} \mid v)$ up to a given order and performs a moment-matching procedure. In this case, the conditional raw moments functions are estimated by regression using the dataset D' defined in (29). At this point, we can either use non-parametric methods like kernel regression, k-NN, Random Forest, etc. or parametric methods. If one chooses the last option, the regression is performed by specifying the conditional raw moment function as a linear combination of functions from a given truncated basis. It amounts to a least-squares Monte-Carlo approach as in the standard American option pricing (see eg. the Longstaff-Schwartz method [LS01]). Although ordinary least squares (OLS) approach is the most natural for parametric regression, other approaches can be explored such as Huber regression [HR09], generalized linear models (GLM, [NW72]), Ridge regression etc. Finally, value-at-risk can be estimated as the quantile at level α of the fitted distribution.

2. **Parametric distribution-free methods.** While assuming a given distribution for $(\Delta V_{t+\delta} \mid v)$ yields a more tractable problem, it is also prone to high bias (eg. invalid distribution) and high variance (eg. estimation of high-order moments for moment matching). Fortunately, some techniques do not require such hypothesis and can be coined as parametric quantile regression methods. The function representing the conditional value-at-risk minimizes the expectation of the "pinball loss" (see equation (105) of Appendix A). The task of estimating the value-at-risk function by minimizing this loss is called *quantile regression*. We will explore the approach of [BCG+24] (see Section 3.5) who propose a two-steps methodology for estimating value-at-risk and expected shortfall by neural networks (parametric functions) in a potentially multi-quantile setup (ie. estimating value-at-risk at different levels α). Regulators are not especially eager to the adoption of such approach as they consider Deep Learning techniques to be black boxes with little control. However, as we will recall, there are many theoretical results on neural networks that enable to provide so-called *a priori* and *a posteriori* bounds on the error between neural network estimation of value-at-risk (or expected shortfall) and the corresponding groundtruth.

3. **Non-parametric methods.** If one does not want to specify a parametric form for the function $f_i(\cdot)$ of equation (26), it is possible to use nearest-neighbor quantile regression [BG90], or kernel quantile regression of [LLZ07] which includes the entire family of smoothing splines and additive and interaction spline models. [CPS15] also investigates an estimator through optimal quantization. Moreover, one can use quantile regression tree [CL02] or quantile regression forest [Mei06] that are build upon regression trees and random forests. These methods are non-supervised, and one may instead try supervised learning methods such as Gaussian process regression (see [MZ21] and [BBC12]), which assumes a Gaussian process prior over the function to estimate. However, it requires labels (ie. true value-at-risk function valued at given points) that we don't have at first hand. Hence, we need

to produce noisy labels using nested Monte-Carlo simulations, which adds computational burden over the Gaussian process regression. Notice that these techniques may be costly to run for a large dataset (especially if hyperparameters are data-driven selected) and some of them may suffer from the curse of dimensionality. In view of these issues, one can select a subset of the training set and estimate $g_i(\cdot)$ instead of $f_i(\cdot)$.

4. **Sensitivity-based methods.** Another approach is to use sensitivity-based methods, which focus on the sensitivity of the portfolio value to changes in risk factors. The Delta-Gamma method (see [GH21], [BJS99]) is based on a second-order Taylor expansion of the portfolio pricing function $u(t,x)$ around $(t+\delta, X_{t+\delta})$, and further assumes that the risk factors return during the MPOR follows a normal distribution (ie. it is coherent when we have a multivariate Black-Scholes diffusion). Then, value-at-risk can be obtained as a function of Delta and Gamma (and potentially Theta) sensitivities either through Cornish-Fisher expansion or a Gaussian distribution hypothesis for $(\Delta V_{t+\delta} \mid v)$, matching the moments with the ones implied by the Taylor expansion (featuring the sensitivities). However, this Delta-Gamma approach is not very useful in our Monte-Carlo framework since we can directly estimate any moment of $(\Delta V_{t+\delta} \mid v)$ by regression, as we do for GLSMC/JLSMC. One can also adapt the SIMM method of Section 2.2.2 by setting the sensitivities to the corresponding derivatives of the portfolio pricing function $u(t,x)$, see eg. [FKLV18]. However, if there are no analytical expressions (or already computed estimates) of sensitivities available, then these sensitivity-based techniques imply a high computational burden. For instance, they may be computed with automatic adjoint differentiation (AAD) in the context of differential Machine Learning as in [HS20].

3.3 Monte-Carlo with Gaussian distribution

The Gaussian Least-Squares Monte Carlo (GLSMC) method of [AAGL17] falls into the category of distribution-based method described in Section 3.2 and assumes that[18]:

$$(\Delta V_{t_i+\delta} \mid v) \sim \mathcal{N}\left(0, \sigma_i^2(v)\right). \tag{31}$$

As explained in Appendix A, we can approximate $\sigma_i^2(\cdot)$ as a linear combination of some basis functions:

$$\sigma_i^2(v) \approx \hat{\sigma}_i^2(v) \coloneqq \sum_{k=0}^{K} \hat{a}_k L_k(v), \tag{32}$$

where $\hat{a} \coloneqq (\hat{a}_0, \cdots, \hat{a}_K)^T$ is estimated using the dataset $\mathcal{D}'$ defined in (29). In practice, [AAGL17] takes $L_k(x) = x^k$ as basis functions. Although they are simple, they are not valid theoretically. Indeed, because of multicolinearity, the Gram matrix $\mathbf{L}^T\mathbf{L}$ in equation (102) tends

[18] For the reasons mentioned in Section 3.2, we do not model directly the law of $(\Delta V_{t+\delta} \mid x)$.

to be ill-conditioned, which can make our coefficient estimates highly sensitive to noise. As explained in Appendix A, a solution would be to take an orthogonal basis in a well-chosen weighted L^2 space, such as the Laguerre polynomials used in [MKN+18]. However, we have found that empirically, the choice of the basis functions (among the usual ones) has little importance on the quality of the estimation. The function $g_i(\cdot)$ of equation (28) is then estimated by $\hat{g}_i(\cdot)$ such that:

$$\hat{g}_i(v) = \hat{\sigma}_i(v)\,\Phi^{-1}(\alpha), \tag{33}$$

where $\Phi(\cdot)$ is the cdf of the standard normal distribution.

Nonetheless, there is an important issue with this method. Depending on the value taken by $v = V_{t_i}^{(m)}$, it is possible that our estimated function $\hat{\sigma}_i^2$ may take negative values, which happened in our numerical experiments. Unfortunately, the authors in [AAGL17] do not investigate this problem and we have tried several methods. First, one can enforce positivity of $\hat{\sigma}_i(v)$ by using a GLM with $\ln(\cdot)$ as the link function (log-linear model):

$$\ln\left(\mathbb{E}\left[(\Delta V_{t_i+\delta})^2 \mid v\right]\right) = \sum_{k=0}^{K} \hat{a}_k L_k(v) \tag{34}$$

$$\Leftrightarrow \mathbb{E}\left[(\Delta V_{t_i+\delta})^2 \mid v\right] = \exp\left(\sum_{k=0}^{K} \hat{a}_k L_k(v)\right) =: \hat{\sigma}_i^2(v). \tag{35}$$

Although it does not yield an OLS type minimization problem (because we don't have a linear combination of basis functions), it is possible to find the optimal $\hat{a}_k$ by maximum likelihood estimation (MLE), through an iteratively reweighted least-squares procedure (IRLS), see [NW72][19]. An other method is simply to discard the points for which it produces a negative variance. Indeed, as the final goal is to take an empirical mean, we can afford to discard some points as long as they are not too numerous. More precisely, assuming that the timegrid π defined in Section 2.3 is the same as the timegrid G and recalling equation (16), the estimated expected exposure at a time t_i will be calculated as:

$$\widehat{\mathrm{EE}}(t_i) \approx \frac{1}{\mathrm{Card}(A)} \sum_{m \in A} \left(V_{t_i}^{(m)} + \mathrm{UTF}_{t_i}^{(m)} - \mathrm{VM}_{t_i-\delta}^{(m)} - \hat{g}_{i^\star}\left(V_{t_i-\delta}^{(m)}\right)_+ \right)_+, \tag{36}$$

where $t_{i^\star} \coloneqq t_i - \delta$, A is the set of points $m \in [|1, M|]$ for which $\hat{\sigma}_{i^\star}\left(V_{t_i-\delta}^{(m)}\right) > 0$. Notice that this reasoning is also applicable for any metric estimated by an empirical expectation, eg. DIM (assuming constant riskfree rate for simplicity):

$$\widehat{\mathrm{DIM}}(t_i) \approx \frac{1}{\mathrm{Card}(A)} \sum_{m \in A} \hat{g}_i\left(V_{t_i}^{(m)}\right)_+, \tag{37}$$

[19] As a warning, notice that our application of the GLM framework is not fully satisfying from a theoretical point of view.

where A is the set of points $m \in [|1, M|]$ for which $\hat{\sigma}_i\left(V_{t_i}^{(m)}\right) > 0$. We have tested to set the mean of $(\Delta V_{t+\delta} \mid v)$ different from zero by assuming that:

$$(\Delta V_{t_i+\delta} \mid v) \sim \mathcal{N}\left(\eta_{i,1}(v), \sigma_i^2(v)\right), \tag{38}$$

where $\eta_{i,j}(v)$ is the j-th conditional raw (ie. not centred) moment of $(\Delta V_{t_i+\delta} \mid v)$. In order to estimate $\eta_{i,1}(\cdot)$ and $\sigma_i^2(\cdot)$, we need to estimate both the conditional raw moments of order one and two by regression:

$$\hat{\sigma}_i^2(v) = \hat{\eta}_{2,i}(v) - \hat{\eta}_{1,i}(v)^2. \tag{39}$$

However, even when enforcing positivity of $\hat{\eta}_{2,i}(\cdot)$ by regressing using GLM with log-link, it does not mean that $\hat{\sigma}_i^2(\cdot) > 0$. This issue is extended in details in Section 3.4, and investigated empirically in Section 4.

3.4 Monte-Carlo with Johnson distribution

However, the Gaussian hypothesis may not be valid. For instance, consider the put example of Section 4.1 at a given time $t_i = 1/12$ and for a given portfolio value $V_{t_i} = 6.875$ (corresponding to $X_{t_i} = 100$). On the left graph of Figure 4, we plot the histogram of $(\Delta V_{t+\delta} \mid X_{t_i} = 100)$ and the estimated Gaussian density[20], based on Monte-Carlo simulations. We also plot the empirical quantile and the estimated quantile under the Gaussian hypothesis at level $\alpha = 99\%$. On the right graph, we display the QQ-plot.

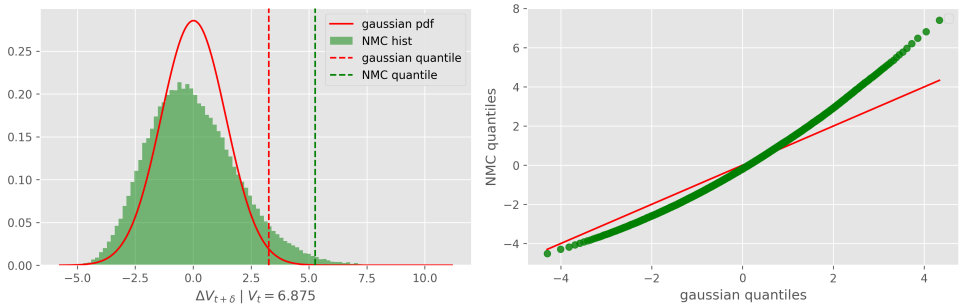

Figure 4: Distribution of $(\Delta V_{t_i+\delta} \mid X_{t_i} = 100)$ for $t_i = 1/12$.

We clearly see that even in the simple case of a put in a Black-Scholes model, the Gaussian assumption is not valid. This has been confirmed statistically with Andersen-Darling and Kolmogorov-Smirnov tests reporting p-values well below the 5% threshold. It implies the need

[20] It is computed here based on Monte-Carlo simulations, and not based on the GLSMC method that assumes a centred Gaussian distribution with variance estimated as described in Section 3.3. Indeed, we don't want to take into account the bias in our estimation of the distribution parameters to justify that a Gaussian distribution is not adapted.

to model the distribution of $(\Delta V_{t+\delta} \mid v)$ in a more flexible way. We will assume that:

$$(\Delta V_{t_i+\delta} \mid v) \sim \text{Johnson}\left(J_{i,v}(\cdot), \gamma_{i,v}, \delta_{i,v}, \xi_{i,v}, \lambda_{i,v}\right). \tag{40}$$

We recall the main properties of this family of distributions as well as parameters estimation methods in Appendix C.

3.4.1 Johnson least-squares Monte-Carlo

In the Johnson Least-Squares Monte-Carlo (JLSMC) method of [MKN+18], the estimation of the Johnson distribution parameters is done using a moment matching method, described in Appendix C.2.1. It requires computing the mean, variance, skewness and kurtosis of $(\Delta V_{t_i+\delta} \mid v)$. Although conditional variance, skewness and kurtosis functions can't be estimated directly by regression, they are based on conditional raw moments of order one to four, which can be estimated by regression. Indeed, from equation (153), the estimated skewness and kurtosis of $(\Delta V_{t_i+\delta} \mid v)$ are respectively given by:

$$\begin{cases} \hat{\gamma}_i(v) & \coloneqq \frac{\hat{\mu}_{i,3}(v)}{\hat{\mu}_{i,2}(v)^{3/2}}, \\ \\ \hat{\kappa}_i(v) & \coloneqq \frac{\hat{\mu}_{i,4}(v)}{\hat{\mu}_{i,2}(v)^2}, \end{cases} \tag{41}$$

where $\hat{\mu}_{i,j}(v)$ is the estimation of $\mu_{i,j}(v)$, the j-th conditional centred moment[21] of $(\Delta V_{t_i+\delta} \mid v)$, such that:

$$\hat{\mu}_{i,j}(v) \coloneqq \sum_{k=0}^{j} \binom{j}{k} (-1)^{j-k} \hat{\eta}_{i,k}(v) \hat{\eta}_{i,1}(v)^{j-k}, \tag{42}$$

where $\hat{\eta}_{i,j}(v)$ is the estimated $\eta_{i,j}(v)$ by regression, defined as in Section 3.3.

Although [MKN+18] do not mention this problem, a first remark we can make is that $\hat{\eta}_{i,2}(\cdot)$ and $\hat{\eta}_{i,4}(\cdot)$ may be negative, whereas it is not possible theoretically. It implies that $\hat{\mu}_{i,j}(v)$ for $j \in [|2,4|]$ may be ill-defined for some values of v. It impacts in turn the regions where estimated conditional skewness $\hat{\gamma}_i(\cdot)$ and kurtosis $\hat{\kappa}_i(\cdot)$ functions are well-defined. There are not many approaches to deal with this issue, and one can only partially solve the problem. As explained in Section 3.3, we can discard the values $v = V_{t_i}^{(m)}$ for which the estimated conditional raw moments of order two and four are negative, and discard in addition the values for which the estimated conditional centred moments of order two and four are negative. Notice that depending on the quality of our regression, it could yield to discard a too large number of points, which is not desirable. As explained in Section 3.3, an alternative is to use a GLM with log-link function for the regression of $\eta_{i,2}(\cdot)$ and $\eta_{i,4}(\cdot)$ to ensure their estimate is positivite. However,

[21] Notice that the variance of $(\Delta V_{t_i+\delta} \mid v)$ is then denoted $\mu_{i,2}(v)$.

it does not mean that the estimates of $\mu_{i,2}(\cdot)$ and of $\mu_{i,4}(\cdot)$ are positive. It only reduces the number of points to discard but at the cost of a potentially high bias in the regression.

A second observation is that the biases in estimations of the conditional raw moments accumulate when computing variance, skewness and kurtosis. It may lead to Johnson distribution parameters totally misspecified while $(\Delta V_{t_i+\delta} \mid v)$ may indeed follow a Johnson distribution but with very different parameters values. The only way to limit this issue is to try several regression methods beyond the natural OLS formulation from Appendix A, as mentionned in Section 3.2.

Moreover, as explained in Appendix C.2.1, the moment matching procedure is costly in time and represents too much computational burden if we want to fit a Johnson distribution (ie. finding its parameters) to $(\Delta V_{t_i+\delta} \mid v)$ for each value $v = V_{t_i}^{(m)}$ with $m \in [|1, M|]$. Thus, we will fit a Johnson distribution at time t_i only for some given *support values* denoted:

$$\mathcal{V}_i = \left\{ V_{t_i}^{(m_{i,1})}, \cdots, V_{t_i}^{(m_{i,R})} \right\}, \tag{43}$$

where R is the number of support values chosen. Then, the function $g_i(\cdot)$ of equation (28) is estimated by minimizing a given empirical loss function on a given set of functions or by non-parametric techniques[22], based on the theoretical quantiles $\{q_{i,1}, \cdots, q_{i,R}\}$ at level α of the Johnson distributions fitted to $(\Delta V_{t_i+\delta} \mid v)$ for each $v \in \mathcal{V}_i$. Indeed, it is possible to compute the quantile of a Johnson distribution in closed-form, as explained in Appendix C. In [MKN+18], the authors simply minimize a mean square error criterion over the set $\mathcal{G}$ of linear combination of Laguerre polynomials truncated at fourth order:

$$\hat{g}_i(v) = \arg\min_{g \in \mathcal{G}} \frac{1}{R} \sum_{r=1}^{R} \left(g\left(V_{t_i}^{(m_{i,r})} \right) - q_{i,r} \right)^2, \tag{44}$$

where $g(v) \coloneqq \sum_{k=0}^{4} a_k L_k(v)$. This approach is useful as we will have an estimate of forward IM value for any value v, which was not the case with the GLSMC method.

A fundamental question lies in how to choose the support values $\mathcal{V}_i$ at a given time t_i. In [MKN+18], they are chosen as some empirical quantiles[23] of V_{t_i}, based on the dataset of iid. observations $\left\{ V_{t_i}^{(1)}, \cdots, V_{t_i}^{(M)} \right\}$. More precisely, we use $M_q = 100$ empirical quantiles at evenly-spaced levels, and we add more evenly-spaced levels in the tails below 1% and above 99%, such that the added number of empirical quantiles in each tail is $\left\lfloor \frac{0.1 \cdot M_q}{2} \right\rfloor$ (if no duplicate with the previous levels related to M_q). These supplementary quantiles enable us to capture effectively the behaviour in the distribution tail, which is crucial for risk management applications. However, there is no clear justification on the extreme quantile levels taken (1% and above 99%),

[22] Any conventional technique may be worth it to consider, as long as the computational cost remains low. Thus, OLS formulations are usually preferred.

[23] We have estimated conditional mean, variance, skewness and kurtosis of $(\Delta V_{t_i+\delta} \mid v)$ as *functions* of v, thus we can choose any definition of empirical quantile. For simplicity, we take the Definition 1 already seen, so that the support values are existing portfolio values among $\{V_{t_i}^{(1)}, \cdots, V_{t_i}^{(M)}\}$.

which should ideally depend on the empirical distribution V_{t_i}. Thus, an alternative idea to build the set of support values could be to use a weak discrepancy sequence over the interval $\left[\min_m V_{t_i}^{(m)},\ \max_m V_{t_i}^{(m)}\right]$. For instance, in the spirit of [MZ21], we may generate the support values from R points of a Sobol sequence [Sob67] on $[0,1]$, such that $\mathcal{V}_i = \left\{V_i^{(1)}, \cdots, V_i^{(R)}\right\}$ where:

$$V_i^{(r)} = \min_m V_{t_i}^{(m)} + \text{sob}_r \cdot \left(\max_m V_{t_i}^{(m)} - \min_m V_{t_i}^{(m)},\right) \tag{45}$$

with sob_r being the r-th element of a Sobol sequence over $[0,1]$. We have not tried this idea in practice, but it may be good to analyze it in the future.

Notice that we have tested other methods for estimating $g_i(\cdot)$ based on the theoretical quantiles $\{q_{i,1}, \cdots, q_{i,R}\}$, which are investigated in details in Section 4. Moreover, one may argue that we could use nested Monte-Carlo simulations in order to compute the raw moments of $(\Delta V_{t_i+\delta} \mid v)$ for $v \in \mathcal{V}_i$ instead of doing a regression. Nonetheless, it would be too costly in practice, even for a reasonable number of support values as the algorithm AS99 already requires a computation time which is unknown ex-ante and may be high.

3.4.2 Johnson percentile matching Monte-Carlo

In order to alleviate the drawbacks of the JLSMC method, we propose a new method called Johnson Percentile Matching Monte-Carlo (JPMMC), which has also been investigated in [GH21]. The main idea is to use an other technique to perform the fitting of Johnson distributions to $(\Delta V_{t_i+\delta} \mid v)$ for each support value $v \in \mathcal{V}_i$, which does not have the flaws of the moment matching procedure explained in Section 3.4.1. We have chosen to use the percentile matching method detailed in Appendix C.2.2 since it does not require the estimation of high-order metrics like skewness and kurtosis and one has more control over the computational cost. As mentionned in Appendix C.2.2, the cost only comes from estimating some quantiles of $(\Delta V_{t_i+\delta} \mid v)$ for each support value $v \in \mathcal{V}_i$ defined in Section 3.4.1, through nested Monte-Carlo. It is controlled by the number of inner paths M_{IN}, where we have found $M_{\text{IN}} = 1000$ to be a good compromise between accuracy and computational cost. Notice we don't estimate quantiles, ie. perform a nested Monte-Carlo, for all the values $v = V_{t_i}^{(m)}$ with $m \in [|1, M|]$ but only for the support values $v \in \mathcal{V}_i$. Indeed, it would be too costly to simulate $N \times M \times K$ paths of the risk factors and to price our portfolio on these paths. In this case, we would be better of doing a quantile regression, explained in Appendix A and investigated in Section 3.5.

Finally, the same procedure as in Section 3.4.1 can be employed to estimate the function $g_i(\cdot)$, but notice that the support values cannot be chosen as freely as for the JLSMC method. Indeed, for each support value $v \in \mathcal{V}_i$, the nested Monte-Carlo simulations imply to generate K iid. samples of $(\Delta V_{t_i+\delta} \mid v)$. Recalling the procedure of Section 3.1, we first need to generate risk factors paths starting from the risk factors value $X_{t_i}^\star$ that would produce the protfolio value

v. However, it is not clear how to obtain this value $X^\star_{t_i}$ if there is not a bijection between the risk factors and portfolio values, or if the function that relates them is not easily invertible. Thus, our only choice is to take the support values as some of the portfolio values $V^{(m)}_{t_i}$ for $m \in [|1, M|]$ already generated, because we know the risk factors value $X^{(m)}_{t_i}$ associated to each portfolio value $V^{(m)}_{t_i}$. For simplicity, we choose the support values as in Section 3.4.1 where the empirical quantile is given in Definition 1.

The only drawback of this method is that the nested Monte-Carlo simulations may be too costly in two cases. First, the risk factors simulation could require costly numerical methods other than the classic schemes like Euler-Maruyama. Second, there may not be analytical (or fast enough semi-analytical) expressions for the portfolio price and one may be forced to use Monte-Carlo simulations or other costly techniques to estimate the portfolio value at each time t_i and for each path m. In this case, it is not a good option to rely on the JPMMC method and one may use a simpler technique like GLSMC explained in Section 3.3. Nonetheless, there may be ways to reduce the variance in the estimation of the four quantiles, hence reducing the number of inner paths while keeping the same degree of accuracy. For instance, one may use techniques like antithetic variables, control variates [HN98] or importance sampling [EL10] [Gly96], but their implementation depends on the risk factors diffusion and portfolio specified. One may also try to estimate the four quantiles by doing quantile regression in a multi-level setup with a very small neural network, but it is not clear how much accuracy we can achieve with such a small model and we may be better off using a larger model and follow the distribution-free approach of [BCG+24] described in Section 3.5.

3.5 Neural networks quantile regression

In this section, we investigate the work of [BCG+24] who propose a two-steps methodology for estimating value-at-risk and expected shortfall in a potentially multi-quantile setup, leveraging neural networks and transfer learning. We recommend the reader unfamiliar with this topic to go through Appendix A.2, Appendix A.3, and Appendix B.

3.5.1 A larger scope of study

The estimation of both value-at-risk (or quantile) and expected shortfall (or superquantile) don't arise directly in forward IM, and we exceptionnally take a larger perspective in this section. Indeed, expected shortfall appears from a risk management point of view or when calculating the CVA (see [BCG+24], [AAC20]). Let us provide a short explanation and assume the riskfree rate is deterministic for the sake of simplicity. From equation (21), we need to calculate for all

$t \in [0, T-\delta]$ and for all $s \in [t, T-\delta]$:

$$\mathbb{E}\left[E_{s+\delta} \mid \mathcal{F}_s\right] = \mathbb{E}\left[\left(\Delta V_{s+\delta} - \mathrm{IM}_s\right)_+ \mid \mathcal{F}_s\right] \tag{46}$$

$$= \mathbb{E}\left[\left(\Delta V_{s+\delta} - \mathrm{VaR}^\alpha\left(\Delta V_{s+\delta} \mid \mathcal{F}_s\right)_+\right)_+ \mid \mathcal{F}_s\right]. \tag{47}$$

Assuming that taking $\mathrm{VaR}^\alpha\left(\Delta V_{s+\delta} \mid \mathcal{F}_s\right)_+ \approx \mathrm{VaR}^\alpha\left(\Delta V_{s+\delta} \mid \mathcal{F}_s\right)$ has little impact on the CVA value, it yields:

$$\begin{aligned}\mathbb{E}\left[E_{s+\delta} \mid \mathcal{F}_s\right] =& \mathbb{E}\left[\Delta V_{s+\delta} \cdot \mathbf{1}_{\Delta V_{s+\delta} > \mathrm{VaR}^\alpha(\Delta V_{s+\delta} \mid \mathcal{F}_s)} \mid \mathcal{F}_s\right] \\ & - \mathrm{VaR}^\alpha\left(\Delta V_{s+\delta} \mid \mathcal{F}_s\right) \mathbb{Q}\left(\Delta V_{s+\delta} > \mathrm{VaR}^\alpha\left(\Delta V_{s+\delta} \mid \mathcal{F}_s\right)\right).\end{aligned} \tag{48}$$

Using the identity in Proposition 2, we have:

$$\mathbb{E}\left[E_{s+\delta} \mid \mathcal{F}_s\right] = (1-\alpha)\left(\mathrm{ES}^\alpha\left(\Delta V_{s+\delta} \mid \mathcal{F}_s\right) - \mathrm{VaR}^\alpha\left(\Delta V_{s+\delta} \mid \mathcal{F}_s\right)\right), \tag{49}$$

where $\mathrm{ES}^\alpha(Y) = \mathbb{E}\left[Y \mid Y > \mathrm{VaR}^\alpha(Y)\right]$. For rigorous mathematical definition of conditional expected shortfall (or conditional superquantile), we refer the reader to Section 2. of [BCG+24].

3.5.2 Learning conditional value-at-risk and expected shortfall

It implies that for each $i \in [|0, N|]$ and for each $m \in [|1, M|]$, we need to estimate the function $f_i(\cdot)$ defined in equation (26), as well as the function $\mathrm{es}_i(\cdot)$ such that:

$$\mathrm{es}_i\left(X_{t_i}^{(m)}\right) \coloneqq \mathrm{ES}^\alpha\left(\Delta V_{t_i+\delta} \mid X_{t_i}^{(m)}\right). \tag{50}$$

As mentionned in the procedure for estimating expected shortfall in Appendix A.3, the first step is to estimate the value-at-risk function $f_i(\cdot)$. In [BCG+24], the authors propose to learn this function using neural networks, in order to lower the bias compared to a basis functions approach investigated in Appendix A.2. Building on the results of Appendix A.2 and using the approach from Appendix B, we solve the following stochastic optimization problem for each $i \in [|0, N|]$:

$$h_{\theta^\star} \coloneqq \arg\min_{h_\theta \in \mathcal{H}_\theta} \mathbb{E}\left[p\left(\Delta V_{t_i+\delta}, h_\theta\left(X_{t_i}\right)\right)\right], \tag{51}$$

where $p(\cdot,\cdot)$ is the pinball loss of equation (106), and $\mathcal{H}_\theta$ defined in equation (142) is the set of (feed-forward) neural network functions parametrized by θ. Here, we use mini-batch SGD given by Algorithm 4 on the dataset $\mathcal{D}$ defined in equation (27), and Ridge regularisation with parameter $\kappa > 0$, so that the loss used in the SGD procedure is:

$$\mathcal{L}\left(x, y; \theta\right) = p\left(y, h_\theta(x)\right) + \kappa\|\theta\|_2^2, \tag{52}$$

where $\|\theta\|_2$ is the euclidian norm of the flattened vector θ and . We then have:

$$\hat{f}_i(\cdot) = h_{\hat{\theta}}(\cdot), \tag{53}$$

where $\hat{\theta}$ is the output of the mini-batch SGD algorithm. Then, one can estimate the expected shortfall function as we have $\mathrm{es}_i(\cdot) = f_i(\cdot) + r_i(\cdot)$, and in view of equation (133) from Appendix A.3:

$$r_i = \arg\min_{h \in \mathcal{B}} \mathbb{E}\left[\left(p\left(\Delta V_{t_i+\delta}, f_i\left(X_{t_i}\right)\right) - f_i\left(X_{t_i}\right) - h\left(X_{t_i}\right)\right)^2\right]. \tag{54}$$

Replacing $f_i(\cdot)$ by $\hat{f}_i(\cdot)$ and using a new neural network to estimate $r_i(\cdot)$ in order to have minimal bias, we obtain the following stochastic optimization problem for each $i \in [|0, N|]$:

$$\zeta_{\theta^\star} \coloneqq \arg\min_{h_\theta \in \mathcal{H}_\theta} \mathbb{E}\left[\left(p\left(\Delta V_{t_i+\delta}, \hat{f}_i\left(X_{t_i}\right)\right) - \hat{f}_i\left(X_{t_i}\right) - h_\theta\left(X_{t_i}\right)\right)^2\right]. \tag{55}$$

Performing the mini-batch SGD algorithm on this problem, we would obtain $\hat{r}_i(\cdot) = \zeta_{\hat{\theta}}(\cdot)$ as an estimation of $r_i(\cdot)$, where $\hat{\theta}$ is the output of the mini-batch SGD algorithm. However, [BCG+24] propose a transfer learning approach to free from the computational burden of training a new neural network from scratch. Indeed, we can use the trained hidden layers of the value-at-risk neural network function $\hat{f}_i(\cdot)$ to estimate the residual function $r_i(\cdot)$. More precisely, they use the following procedure:

1. Define a neural network $\zeta_{\tilde{\theta}}$ for learning the residual function $r_i(\cdot)$, with the same architecture as the value-at-risk neural network $\hat{f}_i(\cdot) = h_{\hat{\theta}}(\cdot)$.

2. Freeze the weight matrices of the hidden layers of the expected shortfall neural network $\zeta_{\tilde{\theta}}$ to be equal to the weight matrices of the hidden layers of the value-at-risk neural network $h_{\hat{\theta}}$. Similarly to equation (143) of Appendix B, let us denote $\hat{\theta} = \{\hat{W}_1, \cdots, \hat{W}_{l+1}\}$ and $\tilde{\theta} = \{\tilde{W}_1, \cdots, \tilde{W}_{l+1}\}$. It means we set:

$$\tilde{W}_k \coloneqq \hat{W}_k, \quad \forall k \in [|1, l|]. \tag{56}$$

3. Thus, the only parameter to optimize is the output weight matrix, such that:

$$\tilde{W}_{l+1} = \arg\min_{W \in \mathbb{R}^{1 \times n_l}} \mathbb{E}\left[\left(p\left(\Delta V_{t_i+\delta}, \hat{f}_i\left(X_{t_i}\right)\right) - \hat{f}_i\left(X_{t_i}\right) - W\zeta_l\left(X_{t_i}\right)\right)^2\right], \tag{57}$$

where $\zeta_l(\cdot)$ is the output (taking values in $\mathbb{R}^{n_l}$) of the last hidden layer of the expected shortfall neural network $\zeta_{\tilde{\theta}}(\cdot) = \tilde{W}_{l+1}\zeta_l(\cdot)$. Taking the empirical version of this loss function over the whole dataset, we then have a classic linear regression problem (ie. OLS

formulation), which can be solved in closed-form:

$$\tilde{W}_{l+1} \approx \arg \min_{W \in \mathbb{R}^{1 \times n_l}} \frac{1}{M} \sum_{m=1}^{M} \left(p\left(\Delta V_{t_i+\delta}^{(m)}, \hat{f}_i\left(X_{t_i}^{(m)}\right)\right) - \hat{f}_i\left(X_{t_i}^{(m)}\right) - W\zeta_l\left(X_{t_i}^{(m)}\right)\right)^2. \quad (58)$$

4. Finally, we set $\hat{\text{es}}_i(\cdot) = \hat{f}_i(\cdot) + \hat{r}_i(\cdot)$ where $\hat{r}_i(\cdot) = \tilde{W}_{l+1}\zeta_l(\cdot) = \zeta_{\tilde{\theta}}(\cdot)$.

Notice that this transfer learning approach is only possible if we estimate the value-at-risk function $f_i(\cdot)$ by a neural network. Moreover, we have one neural network training (to estimate the value-at-risk function) per timestep $i \in [|0, N|]$. If N is large, one may wonder if we can leverage the knowledge gained from $\hat{f}_i(\cdot)$ learned at previous timesteps to improve the training process for later timesteps. In [BCG+24], the authors build upon the work of [ATCS22] by doing first the training of $\hat{f}_N(\cdot)$ and then using the learned parameters to initialize the training of $\hat{f}_{N-1}(\cdot)$, and so on, until $\hat{f}_0(\cdot)$. Hence, if we denote the learned parameters $\hat{\theta}_i$ at step i, then we initialize $\hat{\theta}_{i-1} = \hat{\theta}_i$ for the training of $\hat{f}_{i-1}(\cdot)$. For the last timestep, the parameters are initialized using the default `pytorch` procedure ie. each value is drawn from $\mathcal{U}\left(-\frac{1}{\sqrt{n_l}}, \frac{1}{\sqrt{n_l}}\right)$.

3.5.3 A priori error, a posteriori error, and multi-quantile setup

Although using neural networks may be seen as a black-box approach, there are actually many theoretical results on the SGD and value-at-risk error bound. Assuming that the optimization procedure converges to the global minimum of the empirical version of the loss function from equation (51) (which is a very strong assumption though), then the upper bound of the statistical error between $\hat{f}_i(\cdot)$ (which the theoretical estimate of equation (51) here and not the actual one) and $f_i(\cdot)$ exhibits a rate of order[24] $\mathcal{O}\left(M^{-1/4}\right)$. We call this error the *a priori error* as it does not consider the error arising in practice from the optimization procedure. Analyzing *a priori* the error resulting from the optimization procedure is a much tougher task.

Thus, [BCG+24] also provide a way to estimate the distance of $\hat{f}_i(\cdot)$ and $\hat{\text{es}}_i(\cdot)$ (which are the actual estimates we get with our data and the SGD procedure) to the groundtruth $f_i(\cdot)$ and $\text{es}_i(\cdot)$ respectively. These *a posteriori errors* involve a companion out-of-sample "twin" Monte-Carlo procedure, and are not restricted to neural network estimators. Notice that the twin Monte-Carlo is only appropriate to ensure that the a posteriori error is small enough for a given type of estimator, but not for comparing the performance accross different types of estimators. For instance, if we find an a posteriori error too large for our neural network estimator, we can try to modify its architecture or the optimization procedure and recompute the a posteriori error. Comparing different types of estimators is simply done computing a MSE based on the groundtruth or nested Monte-Carlo.

[24] [BCG+24] provide explicitly the upper bound of this error.

Finally, [BCG+24] provide an extension of this approach to the problem of learning value-at-risk and expected shortfall at different levels α. In particular, they tackle the crossing quantile issue, which refers to the violation of value-at-risk monotonicity with respect to α.

4 Case Study and Numerical Results

We consider now numerical applications of our forward IM models for two case studies.

4.1 European put option in Black-Scholes model

4.1.1 Model setup and analytical expressions

In this section, we consider that our portfolio is a European put option on an underlying no dividend-paying stock price X, with strike K and maturity T. The payoff at maturity is given by:

$$V_T = (K - X_T)_+. \tag{59}$$

We assume the stock price follows a Black-Scholes model [BS73] under $\mathbb{Q}$ such that:

$$dX_t = rX_t dt + \sigma X_t dW_t, \tag{60}$$

where r is the constant riskfree rate, σ is the constant volatility, and W is a $\mathbb{Q}$-Brownian motion. From Itô lemma applied to the function $f(x) = \ln(x)$, we know that on the timegrid π defined in Section 2.3, we have:

$$\begin{cases} X_{t_i} &= X_{t_{i-1}} e^{(r-\frac{\sigma^2}{2})h+\sigma(W_{t_i}-W_{t_{i-1}})}, \\ X_{t_0} &= X_0, \end{cases} \tag{61}$$

We construct an exact approximation[25] of X using independent Gaussian random variables $Z_i \sim \mathcal{N}(0,1)$ such that:

$$\begin{cases} X_{t_i} &= X_{t_{i-1}} e^{(r-\frac{\sigma^2}{2})h+\sigma\sqrt{h}Z_i}, \\ X_{t_0} &= X_0. \end{cases} \tag{62}$$

Thus, we simulate our risk factor ($d = 1$) paths such that:

$$\begin{cases} X_{t_i}^{(m)} &= X_{t_{i-1}}^{(m)} e^{(r-\frac{\sigma^2}{2})h+\sigma\sqrt{h}Z_{i,m}}, \\ X_{t_0}^{(m)} &= X_0, \end{cases} \tag{63}$$

[25] The joint distribution of our approximation given by equation (62) on π coincides with the joint distribution of the true process given by equation (61) on π.

where $Z_{i,m} \overset{\text{iid.}}{\sim} \mathcal{N}(0,1)$. Moreover, we know that we have an analytical expression for the price of our portfolio for all $t < T$, given by:

$$V_t = u(t, X_t) \coloneqq Ke^{-r(T-t)}\Phi(-d_2(t, X_t)) - X_t\Phi(-d_1(t, X_t)), \tag{64}$$

where::

$$\begin{cases} d_1(t,x) & \coloneqq \frac{\ln\left(\frac{x}{K}\right)+\left(r+\frac{\sigma^2}{2}\right)(T-t)}{\sigma\sqrt{T-t}}, \\ d_2(t,x) & \coloneqq d_1(t,x) - \sigma\sqrt{T-t}. \end{cases} \tag{65}$$

Thus, we simulate with no additional cost our portfolio price paths such that $V_{t_i}^{(m)} = u\left(t_i, X_{t_i}^{(m)}\right)$.

Although it is well known that the Black-Scholes model fails to capture important empirical features such as the volatility smile, it is still a good example to illustrate and benchmark our methods. In particular, we have an analytical expression for the true conditional value-at-risk function $f_i(\cdot)$ (and so for forward IM) defined in equation (26) of Section 2.3, based on the monotonicity property of value-at-risk from Proposition 1 of Appendix A. We have:

$$\begin{aligned} f_i(x) &= \mathrm{VaR}^{\alpha}\left(\Delta V_{t_i+\delta} \mid x\right) && (66)\\ &= \mathrm{VaR}^{\alpha}\left(u\left(t_i+\delta', xe^{\left(r-\frac{\sigma^2}{2}\right)\delta'+\sigma(W_{t_i+\delta'}-W_{t_i})}\right) - u(t_i,x)\right) && (67)\\ &= \mathrm{VaR}^{\alpha}\left(u\left(t_i+\delta', xe^{\left(r-\frac{\sigma^2}{2}\right)\delta'+\sigma\sqrt{\delta'}Z_i}\right) - u(t_i,x)\right) && (68)\\ &= \mathrm{VaR}^{\alpha}\left(w_i(Z_i)\right), && (69) \end{aligned}$$

where $Z_i \sim \mathcal{N}(0,1)$, $w_i(z) \coloneqq u\left(t_i+\delta', xe^{\left(r-\frac{\sigma^2}{2}\right)\delta'+\sigma\sqrt{\delta'}z}\right) - u(t_i,x)$ and $\delta' \coloneqq \delta \wedge (T-t_i)$. The derivative of w_i is given by:

$$w_i'(z) = x\sigma\sqrt{\delta'}e^{\left(r-\frac{\sigma^2}{2}\right)\delta'+\sigma\sqrt{\delta'}z} \cdot \partial_x u\left(t_i+\delta', xe^{\left(r-\frac{\sigma^2}{2}\right)\delta'+\sigma\sqrt{\delta'}z}\right), \tag{70}$$

where $\Delta \coloneqq \partial_x u(t,x)$ is the partial derivative of u with respect to its second variable. It is well-known that the Greek Δ of a European put option is strictly negative, so that $w_i' < 0$, ie. w_i is strictly decreasing. From Proposition 1, we have:

$$\begin{aligned} f_i(x) &= w_i\left(\mathrm{VaR}^{1-\alpha}(Z_i)\right) && (71)\\ &= u\left(t_i+\delta', xe^{\left(r-\frac{\sigma^2}{2}\right)\delta'+\sigma\sqrt{\delta'}\Phi^{-1}(1-\alpha)}\right) - u(t_i,x). && (72) \end{aligned}$$

4.1.2 Numerical results

In this section, we present detailed numerical results of our forward IM estimation methods for the European put option in the Black-Scholes model. For consistency with [MKN+18], we will take the following parameters values in Table 1 assuming one year is 240 days.

riskfree rate	r	5%	MPOR	δ	1/24
spot price	X_0	100\$	timestep	h	1/240
volatility	σ	30%	nb paths train set	M_{train}	10^4
maturity	T	1 year	nb paths test set	M_{test}	10^3
strike	K	95\$			
value-at-risk level	α	99%			

Table 1: Parameters values European put Black-Scholes.

If one uses the JLSMC method of Section 3.4.1, the first step is to estimate by regression the first four raw moments of $(\Delta V_{t_i+\delta} \mid v)$. Although the natural way is to do a linear regression (ie. OLS-type), we have tested other techniques including Hubert regression and GLM, with/without Ridge penalization. We have also tested kernel regression with a Silvermann bandwidth and Gaussian kernel. Moreover, we tried Laguerre and polynomial basis functions with trunctature degree of 2, 3, and 7. Finally, we considered assuming the mean of $(\Delta V_{t_i+\delta} \mid v)$ equal or different from zero. As an example, we plot in Figure 5 the regressed mean, variance, skewness, and kurtosis of $(\Delta V_{t_i+\delta} \mid v)$ for $t_i = 1/12$, using linear regression with Laguerre polynomials truncated at a degree of 2 and without Ridge penalty (JLSMC). The raw moments obtained by nested Monte-Carlo and the GLSMC setting of [AAGL17] are displayed for comparison purposes. We notice there are 7 invalid support values, ie. the values $v \in \mathcal{V}_i$ for which the estimated metrics values do not belong to the corresponding definition domains. Moreover, we see that the estimations of skewness and kurtosis are not accurate for small values v, due to the accumulation of errors and invalid support values in regression of the conditional raw moments.

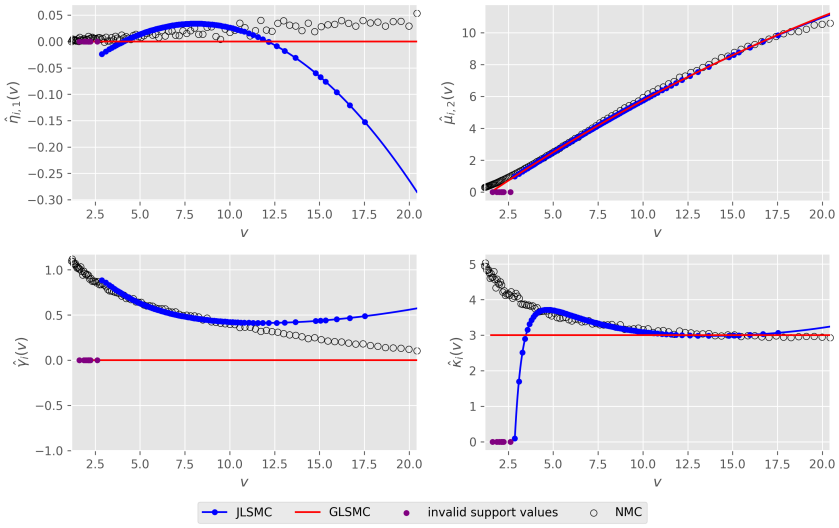

Figure 5: Regressed mean, variance, skewness, and kurtosis for $t_i = 1/12$.

The second step of the JLSMC method is to estimate the Johnson distribution parameters (based on the estimated metrics described above) and calculate the associated quantile at level α, for each support value $v \in \mathcal{V}_i$. In Figure 6, we display the quantiles of fitted Johnson distributions for each support value using the same setup as in Figure 5. We also display the estimated $f_i(\cdot)$ obtained from GLSMC and the analytical expression given in equation (72) of Section 4.1.1.

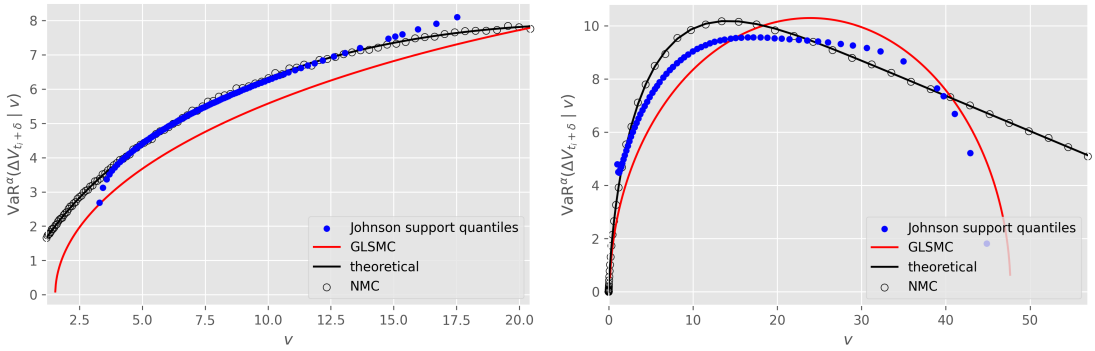

Figure 6: Estimated Johnson quantiles with moment matching. Left: $t = 1/12$. Right: $t = 0.8$.

Notice that we have respectively 9 and 44 invalid support values for $t_i = 1/12$ and $t_i = 0.8$. Moreover, we see that the fitting of Johnson distribution to $(\Delta V_{t_i+\delta} \mid v)$ for $v \in \mathcal{V}_i$ is bad for $t_i = 0.8$. It is mainly due to the errors that accumulate in the estimation of skewness and kurtosis, leading to inaccurate Johnson parameters values. As an example, we display in Figure 7 the empirical distribution for the largest support value $v \approx 44.864$ of Figure 6 for $t_i = 0.8$, computed using Monte-Carlo. Comparing with the pdf. of the fitted Johnson distribution, we see that although a Johnson distribution is adapted to the distribution of $(\Delta V_{t_i+\delta} \mid v)$, the fitted

parameters are not accurate enough to capture the distribution shape.

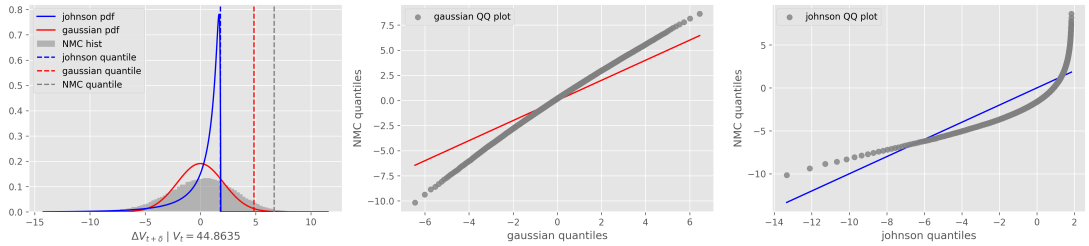

Figure 7: Fitted distributions for largest support value when $t_i = 1/12$.

One may wonder if we have the same problem with the JPMMC method of Section 3.4.2, ie. when we estimate the Johnson distribution parameters using the percentile matching method. In Figure 8, we display the same setting as in Figure 6 but using the percentile matching method instead of the moment-matching method, with $M_{\text{IN}} = 1000$ (see Section 3.4.2) and $z = 1$ (see Appendix C.2.2). We see that we do not have the estimation issues of the moment-matching method, as we already mentionned before. In particular, we do not have any invalid support values.

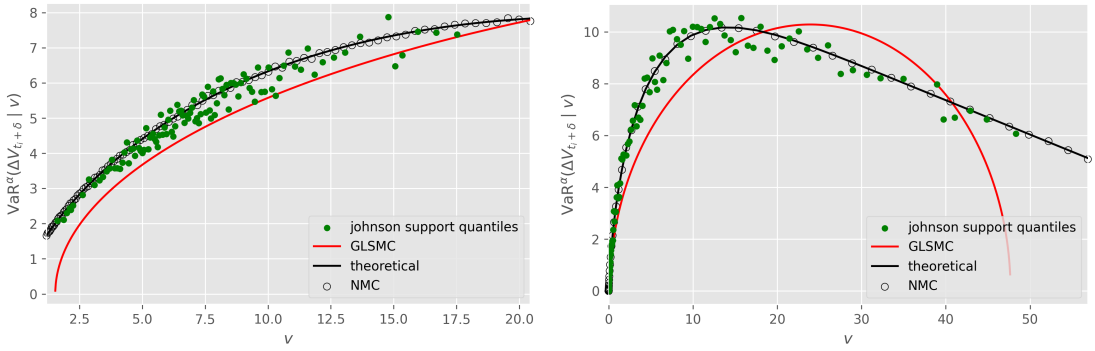

Figure 8: Estimated Johnson quantiles with percentile matching. Left: $t_i = 1/12$. Right: $t_i = 0.8$.

We may check if the hypothesis of Johnson distribution is well adapted to the distribution of $(\Delta V_{t_i+\delta} \mid v)$ for each $v \in \mathcal{V}_i$ we face in this case study. In Figure 9, we show the empirical distribution for the 60-th smallest support value $v \approx 0.496$ (corresponding to $X_{t_i} \approx 97.051$) for $t_i = 0.9875$ and the estimated Johnson density[26], based on Monte-Carlo simulations.

[26]It is computed here based on Monte-Carlo simulations, and not from on the JLSMC/JPMMC method. Indeed, we don't want to take into account the bias in our estimation of the distribution parameters to justify that a Johnson distribution would not adapted in some cases.

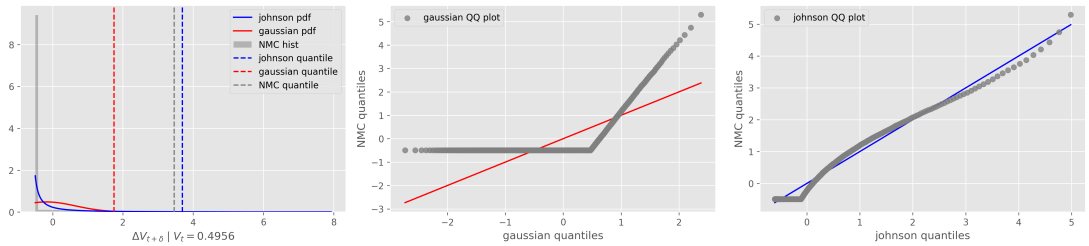

Figure 9: Distribution of $(\Delta V_{t_i+\delta} \mid v)$ for $t_i = 0.9875$.

We see that the distribution of $(\Delta V_{t_i+\delta} \mid v)$ has a spike on the left extremity of the support, which is not well captured by the Johnson distribution. Such case happens when t_i is near maturity and the put is at-the-money, ie. $X_{t_i}^{(m)} \approx K$. If we denote the considered support value by $v = V_{t_i}^{(m)}$, then some values $X_{t_i+\delta}^{(m,k)}$ from the nested Monte-Carlo simulation (recall notations of Section 3.1) will be above the strike K, yielding some values $V_{t_i+\delta}^{(k)} = 0$ and so a negative spike. The empirical and Johnson quantiles are roughly the same, but we have observed in practice that it is especially difficult to estimate the Johnson distribution parameters in the context of JLSMC/JPMMC methods in this case.

The last step is to estimate the function $g_i(\cdot)$ of equation (28) using an error based on the fitted quantiles $\{q_{i,1}, \cdots, q_{i,R}\}$. As mentionned in Section 3.4, we have tested parametric techniques like linear regression with Laguerre basis functions and truncature degree of 2, 3, and 4. Moreover, we investigated non-parametric techniques like kernel regression with cross-validated bandwidth, or k-nearest neighbors (k-NN) with $k = 1$, 3, 5. In Figure 10 and Figure 11, we display the function $\hat{g}_i(\cdot)$ computed using respectively linear regression with order 2 and k-NN with $k = 3$, for $t_i = 1/12$ and $t_i = 0.8$. In order to isolate the Johnson fittings quality from this final step, we take the Johnson quantiles given by the fitted Johnson distributions through percentile-matching procedure with $z = 1$ and $M_{\text{IN}} = 10^4$.

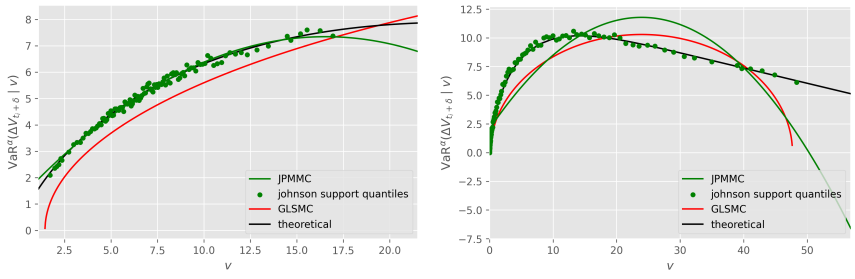

Figure 10: $\hat{g}_i(\cdot)$ using linear regression. Left: $t_i = 1/12$. Right: $t_i = 0.80$.

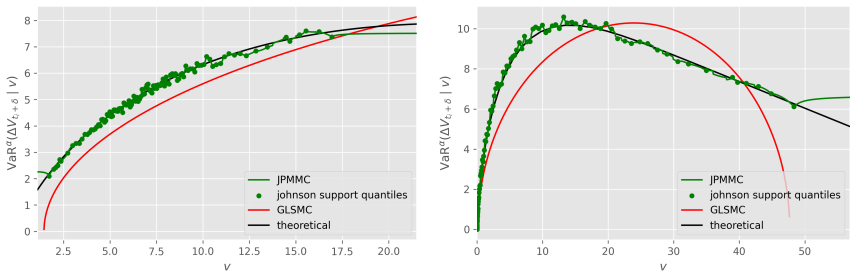

Figure 11: $\hat{g}_i(\cdot)$ using k-NN. Left: $t_i = 1/12$. Right: $t_i = 0.80$.

We observe that extrapolation (ie. estimating $g_i(\cdot)$ outside the two extreme support values) is especially challenging as the function has many degrees of freedom. Nonetheless, bad extrapolation is not critical (unless it is very bad, like rapidly decreasing function). Indeed, the support values cover around 99.66% of values taken by V_{t_i} in our Monte-Carlo experiment, so that extrapolation is relevant only for a very small number of paths at time t_i (around 30 points for $M = 10^4$). We have tried to add $\max_m V_{t_i}^{(m)}$ and $\min_m V_{t_i}^{(m)}$ to $\mathcal{V}_i$, or to have an even more finer resolution in the tails to improve extrapolation, but it did not change the final results of Table 2. There are many twists one can provide to the JLSMC/JPMMC methods, but it is more of adding ingredients in a recipe than a solid theoretical framework.

For the neural network quantile regression, we have taken softplus (smoothed ReLu) activation functions to be consistent with [BCG+24], tried Adam and SGD (with momentum to speed-up convergence) optimizers, and tested several values for hyperparameters like batch size, number of epochs, number of hidden layers, number of neurons per layer, learning rate, Ridge penalty strength, and patience (number of epochs without improvement before stopping).

Finally, one can compare all the methods investigated using a mean-square error between the estimated forward IM $\hat{g}_i(\cdot)_+$, or $\hat{f}_i(\cdot)_+$ in the neural network case, and the true forward IM $f_i(\cdot)_+$ given by[27] equation (72) aggregated over all paths and timesteps:

$$\text{MSE} = \frac{1}{M \cdot (N+1)} \sum_{i=0}^{N} \sum_{m=1}^{M} \left[\hat{g}_i \left(V_{t_i}^{(m)} \right)_+ - f_i \left(X_{t_i}^{(m)} \right)_+ \right]^2, \tag{73}$$

computed on both training set and test set to check for potential underfitting/overfitting issues. In Table 2, we show the MSE and running time for the JPMMC, JLSMC from [MKN+18], and GLSMC from [AAGL17], as well as for variants in moments regressions and in the estimation of $g_i(\cdot)$. Here, "MM" and "PM" stand for moment-matching and percentile-matching procedures to fit the Johnson distribution. Then, "xk" represents the number of nested paths $M_{\text{IN}} = \mathtt{x} \times 1000$ used in percentile-matching. "LRx" is the linear regression method (ie. OLS-type expression) with truncature degree x of Laguerre polynomials, for the regression of the conditional raw

[27] If no analytical formula was available, we could simply have used the nested Monte-Carlo estimations as ground truth, although it takes a long time to run.

moments in the moment-matching method. In the same way, "GLM`x`" is the GLM method but only for regressing conditional raw moments of order two and four, the other ones being regressed with linear regression. Finally, "NN`x`" and "LR`x`" are respectively `x`-NN and linear regression with Laguerre polynomials of degree `x`, for the estimation of the function $g_i(\cdot)$. Notice that in this Table, we do not display all the 70 variants we have tested for the sake of brevity.

Id	Fit. type	Fit. params	Reg. type	MSE train	MSE test	Runtime
GLSMC*	MM	LR4	-	0.78	**0.76**	1s
JPMMC*	PM	10k	LR4	0.76	**0.77**	3mn9s
JPMMC	PM	1k	LR4	0.90	**0.92**	29s
-	PM	1k	NN3	0.98	0.98	28s
JLSMC*	MM	GLM4	NN3	1.19	**1.23**	5mn33s
GLSMC	MM	LR2	-	1.30	**1.30**	0s
-	MM	GLM4	LR4	1.40	1.49	6mn28s
-	MM	LR4	LR4	2.00	2.10	3mn36s
-	MM	LR4	NN3	2.16	2.23	3mn53s
JLSMC	MM	LR2	LR4	2.23	**2.34**	1mn53s
-	PM	10k	LR2	2.48	2.50	3mn51s
-	PM	1k	LR2	2.55	2.57	29s
-	MM	LR2	LR2	3.63	3.78	1mn55s
-	MM	GLM2	LR2	3.58	3.79	1mn03s
-	MM	GLM2	LR4	4.19	4.66	59s

Table 2: Forward IM MSE European put Black-Scholes model for GLSMC, JLSMC, JPMMC.

We observe that the methods using percentile matching for the fitting of the distributions perform better overall compared to the ones using moment-matching. We have checked that the cases for which the performance is worse is essentially due to a bad behavior from $T-\delta$ to T, where the distribution of $\left(\Delta V_{t_i+\delta} \mid V_{t_i}^{(m)}\right)$ is higly spiked as the put option maturity is T. It means that the estimation of conditional skewness and kurtosis in the moment-matching method is too unstable. Thus, our JPMMC method is preferable compared to the JLSMC method of [MKN$^+$18], although the computational cost is higher because of the need to generate some nested paths. In particular, the JPMMC* method may not be adapted in practice because of its high runtime, although it provides the best MSE. Moreover, we managed to improve the MSE of the JLSMC method by a factor two, compared to [MKN$^+$18] by using other regression techniques (JLSMC* vs. JLSMC). Finally, the JLSMC method performs poorer than the GLSMC one, which is better than half of the settings considered. It means the Gaussian assumption is robust enough although not realistic. The only difference between GLSMC* and GLSMC is that we set the mean different from zero and regress the moments by linear regression with order four (instead of two) Laguerre polynomials.

In Table 3, we show the MSE and running time for the neural network quantile regression

(denoted ML thereafter) for 10 hyperparameters/optimizer configurations. Here "Architecture" is given by `x(yy)` where `x` is the number of layers and `yy` is the number of neurons per layer. Then "Optimizer" is given by "`opt(lr)`" where `opt` is the optimizer name and `lr` is the learning rate. Finally, "Iterations" is given by `xx(yyy)` where `xx` is the number of epochs and `yy` is the batch size. We include the best methods GLSMC*, JLSMC*, and JPMMC* from Table 2 for comparison purposes.

Id	Architecture	Optimizer	Iterations	MSE train	MSE test	Runtime
ML*	2(32)	Adam(5e-4)	50(128)	0.08	**0.08**	26mn6s
-	2(64)	Adam(3e-4)	25(64)	0.18	0.18	22mn27s
ML	2(10)	Adam(1e-3)	40(512)	0.23	**0.24**	11mn37s
-	1(10)	Adam(1e-2)	20(1024)	0.24	0.24	5mn4s
-	2(4)	SGD(5e-2)	40(256)	0.51	0.53	8mn41s
-	1(4)	SGD(1e-2)	15(1024)	0.64	0.66	3mn59s
GLSMC*	-	-	-	0.78	**0.76**	1s
-	1(16)	Adam(1e-3)	30(256)	0.74	0.77	12mn60s
JPMMC*	-	-	-	0.76	**0.77**	3mn9s
JLSMC*	-	-	-	1.19	**1.23**	5mn33s
-	1(4)	Adam(1e-3)	30(2048)	1.31	1.42	12mn54s
-	1(8)	Adam(5e-4)	20(512)	1.62	1.81	9mn58s
-	1(128)	SGD(1e-2)	20(256)	3.08	3.61	7mn52s

Table 3: Forward IM MSE European put Black-Scholes model for ML.

We observe that the ML method gives the best results overall, but often require more computational ressources. Notice that the neural network architectures have been chosen such that their total running time[28] is approximately of same magnitude order than the distribution-based methods. Moreover, neural networks with more layers and neurons perform better, provided that the number of epochs (and hence training time) is large enough.

In Figure 12, we display the forward IM profiles of some methods from Table 2 and 3 by representing the empirical mean (DIM) and quantiles at 5% and 95% based on the training and test sets.

[28]In this case study, if the total running time is around 10-15mn, then a training at a given timestep lasts 5s in average.

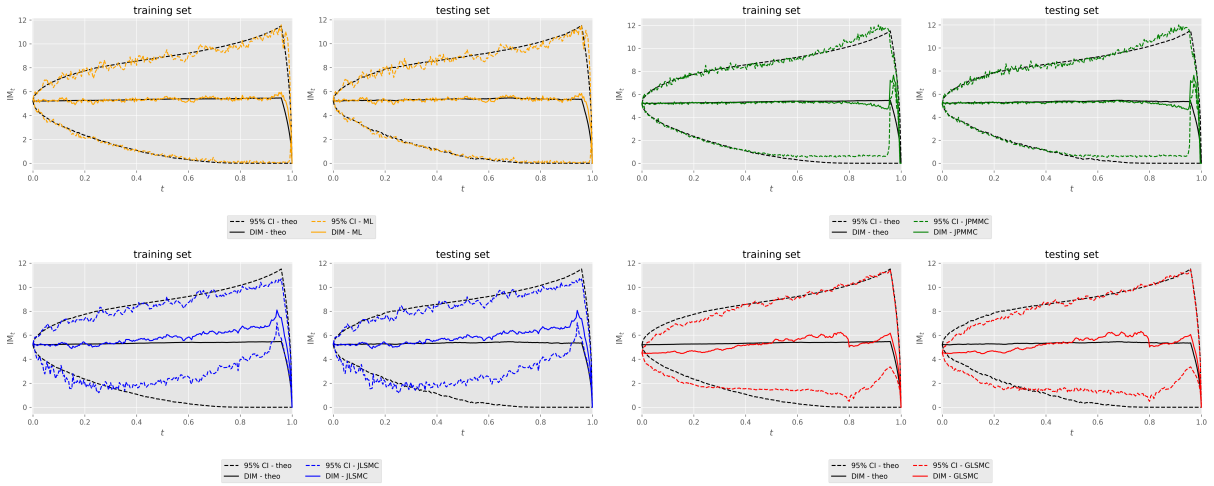

Figure 12: IM profiles. Top left, right: ML, JPMMC. Bottom left, right: JLSMC, GLSMC.

4.2 Swaption in Hull-White model

4.2.1 Model setup and analytical expressions

In this section, we consider that our portfolio is a European swaption (see Appendix D.2) with payoff at time $T = T_0$ given by:

$$V_T = \left(V_{T_0}^{\text{swap}}\right)_+, \tag{74}$$

where $V_{T_0}^{\text{swap}}$ is the price at time T_0 of the payer swap exchanging a fixed rate R against the floating rate $L(T_{i-1}, T_i)$ for the period $[T_{i-1}, T_i]$ and for a notional FV, where $\pi^* \coloneqq \{T_1, \cdots, T_n\}$ is the timegrid of the fixing dates (ie. swap payment dates) assumed to be equally spaced such that $\Delta T \coloneqq T_i - T_{i-1}$. Moreover, we assume the riskfree rate under $\mathbb{Q}$ follows a HW1F model, described in Appendix D.1 such that:

$$dr_t = (\theta(t) - kr_t)dt + \sigma dW_t, \tag{75}$$

From Appendix D.1, we know that on the timegrid π defined in Section 2.3, we have:

$$\begin{cases} r_{t_i} \mid \mathcal{F}_{t_{i-1}} & \sim \mathcal{N}\left((r_{t_{i-1}} - \beta(t_{i-1}))e^{-kh} + \beta(t_i), \frac{\sigma^2}{2k}\left(1 - e^{-2kh}\right)\right), \\ r_{t_0} & = \hat{r}_0, \end{cases} \tag{76}$$

where $\hat{r}_0$ is the observed spot short rate. As in [MKN+18], we specify here that the observed continuously compounded zero-coupon (ZC) yield curve $T \mapsto \hat{y}(T)$ is given by:

$$\hat{y}(T) \coloneqq C_1 + C_2 e^{C_3 T}, \tag{77}$$

with $\hat{B}_0(T) = e^{-\hat{y}(T)\cdot T}$ being the observed price of the ZC with maturity T. It yields that $\hat{r}_0 \coloneqq \lim_{T\to 0} \hat{y}(T) = C_1 + C_2$ and the observed instantaneous forward rate is given by:

$$\hat{f}(0,T) = -\partial_T \ln \hat{B}_0(T) \tag{78}$$

$$= \hat{y}'(T)\cdot T + \hat{y}(T), \tag{79}$$

with $\hat{y}'(T) = C_2C_3e^{C_3T}$. We could have used a real ZC prices curve taken from a market data provider, but for replication purposes we have chosen to use the same (possibly unrealistic) setup as [MKN+18] since it does not change the analysis. We construct an exact approximation of r using independent Gaussian random variables $Z_i \sim \mathcal{N}(0,1)$ such that:

$$\begin{cases} r_{t_i} &= (r_{t_{i-1}} - \beta(t_{i-1}))e^{-kh} + \beta(t_i) + \sqrt{\frac{\sigma^2}{2k}(1-e^{-2kh})}Z_i, \\ r_{t_0} &= \hat{r}_0. \end{cases} \tag{80}$$

Thus, we simulate our risk factor $X_t = r_t$ $(d=1)$ paths such that:

$$\begin{cases} X_{t_i}^{(m)} &= (X_{t_{i-1}}^{(m)} - \beta(t_{i-1}))e^{-kh} + \beta(t_i) + \sqrt{\frac{\sigma^2}{2k}(1-e^{-2kh})}Z_{i,m}, \\ X_{t_0}^{(m)} &= \hat{r}_0, \end{cases} \tag{81}$$

where $Z_{i,m} \overset{\text{iid.}}{\sim} \mathcal{N}(0,1)$. Notice that equation (80) can be seen as a shifted AR(1) filter, so that we can leverage the function `scipy.signal.lfilter` in `Python` for simulating our risk factor. Moreover, we know from the results in Appendix D.2 that we have an analytical expression for the price of our portfolio (made of one swaption) for all $t < T$, given by:

$$V_t = u(t, X_t) = \text{FV}\cdot \sum_{k=1}^{n} c_k u_k^{\text{put}}(t, X_t), \tag{82}$$

where the price function of a put with maturity $T = T_0$ and strike K_k on the ZC with maturity T_k is given by:

$$u_k^{\text{put}}(t,x) \coloneqq K_k u_0^{\text{zc}}(t,x)\Phi(-d_{2,k}(t,x)) - u_k^{\text{zc}}(t,x)\Phi(-d_{1,k}(t,x)), \tag{83}$$

with:

$$\begin{cases} d_{1,k}(t,x) &\coloneqq \frac{\ln\left(\frac{u_k^{\text{zc}}(t,x)}{K_k u_0^{\text{zc}}(t,x)}\right) + \frac{1}{2}\bar{v}(t,T,T_k)}{\sqrt{\bar{v}(t,T,T_k)}}, \\ d_{2,k}(t,x) &\coloneqq d_{1,k}(t,x) - \sqrt{\bar{v}(t,T,T_k)}. \end{cases} \tag{84}$$

Moreover, we recall the price function of the ZC with maturity T_k is:

$$u_k^{\text{zc}}(t,x) \coloneqq e^{m(t,T_k) - n(t,T_k)x}. \tag{85}$$

Thus, we simulate with almost no additional cost our portfolio price paths such that $V_{t_i}^{(m)} = u\left(t_i, X_{t_i}^{(m)}\right)$. Notice that calculating the cdf of the standard normal distribution $2n$ times for computing one portfolio value $v = V_{t_i}^{(m)}$ is not negligible when the number of fixing dates is large. Now, one may wonder if we have an analytical expression for conditional value-at-risk function $f_i(\cdot)$ defined in equation (26) of Section 2.3, as in Section 4.1.1. We have:

$$f_i(x) = \mathrm{VaR}^{\alpha}\left(\Delta V_{t_i+\delta} \mid x\right) \tag{86}$$

$$= \mathrm{VaR}^{\alpha}\left(u\left(t_i+\delta', (x-\beta(t_i-\delta'))e^{-k\delta'} + \beta(t_i) + \sqrt{\frac{\sigma^2}{2k}\left(1-e^{-2k\delta'}\right)}Z_i\right) - u(t_i,x)\right) \tag{87}$$

$$= \mathrm{VaR}^{\alpha}\left(w_i(Z_i)\right), \tag{88}$$

where $w_i(z) \coloneqq u\left(t_i+\delta', (x-\beta(t_i-\delta'))e^{-k\delta'} + \beta(t_i) + \sqrt{\frac{\sigma^2}{2k}\left(1-e^{-2k\delta'}\right)}z\right) - u(t_i,x)$. We need to check if the derivative of this function w_i is strictly positive/negative. The derivative of w_i is given by:

$$w_i'(z) = \sqrt{\frac{\sigma^2}{2k}\left(1-e^{-2k\delta'}\right)} \cdot \partial_x u\left(t_i+\delta', (x-\beta(t_i-\delta'))e^{-k\delta'} + \beta(t_i) + \sqrt{\frac{\sigma^2}{2k}\left(1-e^{-2k\delta'}\right)}z\right), \tag{89}$$

where $\Delta \coloneqq \partial_x u(t,x)$ is the partial derivative of u with respect to its second variable. Checking the sign of w_i' is not straightforward as $\partial_x u(t,\cdot)$ is made of a sum of complex functions that are not all positive or all negative. Relying on numerical simulations, we observed that for some $i \in [|1,N|]$, $u(t,\cdot)$ is not strictly monotonic contrary to what is mentionned in [MKN+18], who gives this result without any proof. We have found that with our parameters values, w_i is strictly increasing on an interval containing most of the values of z (which would be drawn from $\mathcal{N}(0,1)$), for most values of x (which would be drawn from a Gaussian distribution with a given mean and variance), and for most of $i \in [|1,N|]$. Hence, we have the following approximated analytical formula using the Proposition 1 of Appendix A:

$$f_i(x) \approx w_i\left(\mathrm{VaR}^{\alpha}(Z_i)\right) \tag{90}$$

$$= u\left(t_i+\delta', (x-\beta(t_{i-1}))e^{-k\delta'} + \beta(t_i) + \sqrt{\frac{\sigma^2}{2k}\left(1-e^{-2k\delta'}\right)}\Phi^{-1}(\alpha)\right) - u(t_i,x). \tag{91}$$

It is difficult to quantify the error we make by assuming that w_i is strictly increasing, and to quantify the interval of parameters values for which this assumption is violated. For our numerical experiments, we observed that the nested Monte-Carlo provides the same results as equation (91).

4.2.2 Numerical results

In this section, we present detailed numerical results of our forward IM estimation methods for the swaption in the HW1F model. For consistency with [MKN+18], we will take the following parameters values in Table 4 assuming one year is 240 days. We recall that the initial value of risk factors $X_0 = \hat{r}_0$ is entirely determined by the ZC yield curve.

long-term mean	k	1.5%	MPOR	δ	1/24
param ZC yield curve	C_1	0.05	timestep	h	1/240
-	C_2	-0.03	nb paths train set	M_{train}	10^4
-	C_3	-0.18	nb paths test set	M_{test}	10^3
volatility	σ	1%			
swaption maturity	T	1 year			
swap length	T^*	5 years			
swap strike	R	4%			
swap notional	FV	10000$			
swap accrual period	ΔT	0.25 years			
value-at-risk level	α	99%			

Table 4: Parameters values swaption HW1F.

As we follow the same methodology as for the put in the Black-Scholes model of Section 4.1.2, we do not repeat the details mentionned and provide a brief analysis of the results. In Figure 13, we plot the regressed mean, variance, skewness, and kurtosis of $(\Delta V_{t_i+\delta} \mid v)$ for $t_i = 1/12$ using the same setting as in Figure 5. We notice there are 0 invalid support values.

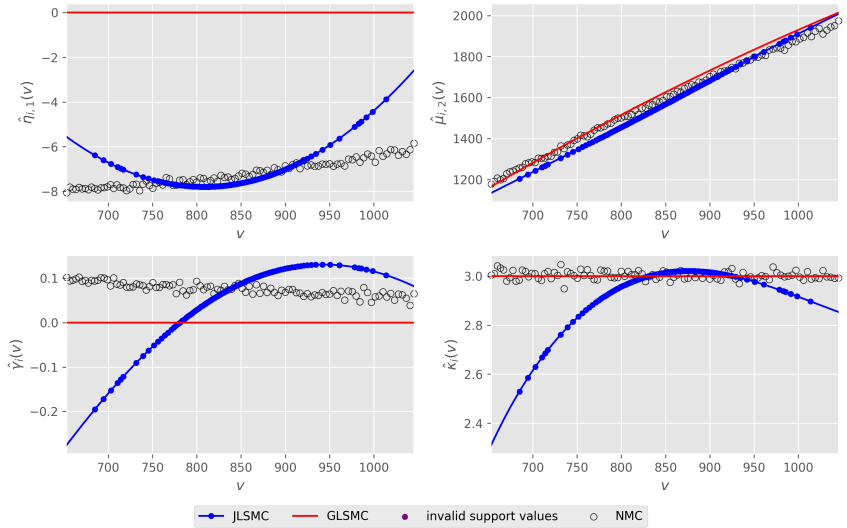

Figure 13: Regressed mean, variance, skewness, and kurtosis for $t_i = 1/12$.

In Figure 14 and Figure 15, we display the quantiles of fitted Johnson distributions for each support value, respectively in the same way as in Figure 6 and Figure 8. In these graphs, we add the nested Monte-Carlo estimations of value-at-risk since the analytical formula for $f_i(\cdot)$ in equation (91) is only valid as an approximation, recall the discussion of Section 4.2.1.

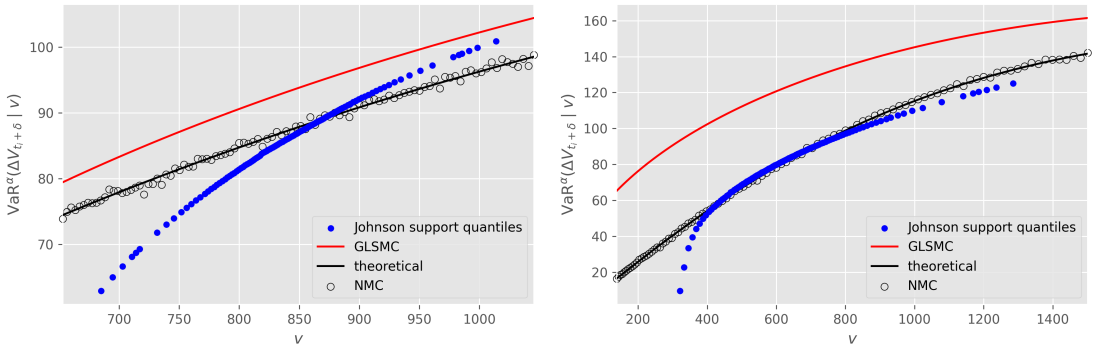

Figure 14: Estimated Johnson quantiles with moment matching. Left: $t = 1/12$. Right: $t = 0.8$.

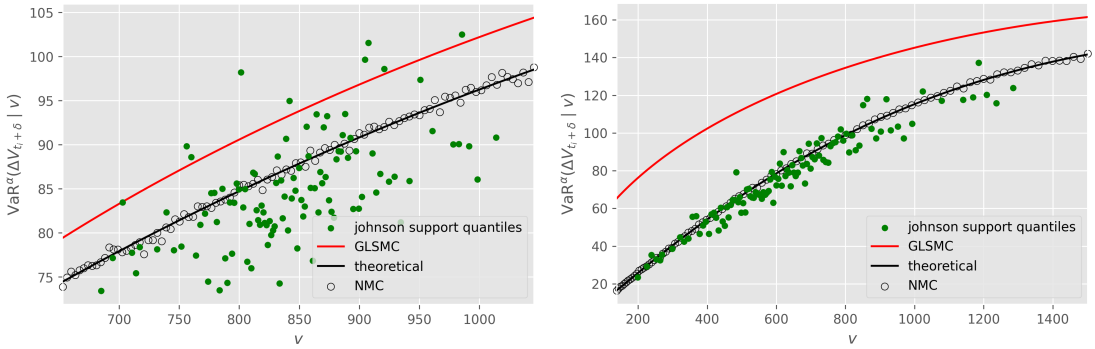

Figure 15: Estimated Johnson quantiles with percentile matching. Left: $t = 1/12$. Right: $t = 0.8$.

We observe that in both cases, we do not have any invalid support values, and the fittings of the Johnson distributions using moment-matching are good (at least at these two timesteps). However, we have much more variance in the quantiles of the fitted Johnson distributions using the percentile matching procedure, although taking the same number of nested paths $M_{\text{IN}} = 1000$ as in the Black-Scholes case. We have checked that by increasing the number of nested paths to $M_{\text{IN}} = 10^4$, we obtain a much better fitting. In Figure 16, we display the function $\hat{g}_i(\cdot)$ computed using respectively linear regression with order 2 and k-NN with $k = 3$. In order to investigate the estimation of $g_i(\cdot)$ in a more delicate situation, we take the Johnson quantiles given by the fitted Johnson distributions through the percentile matching procedure with $z = 1$ and $M_{\text{IN}} = 10^3$ in the same way as in Figure 10 and Figure 11, but only for $t_i = 1/12$ which is the interesting case here.

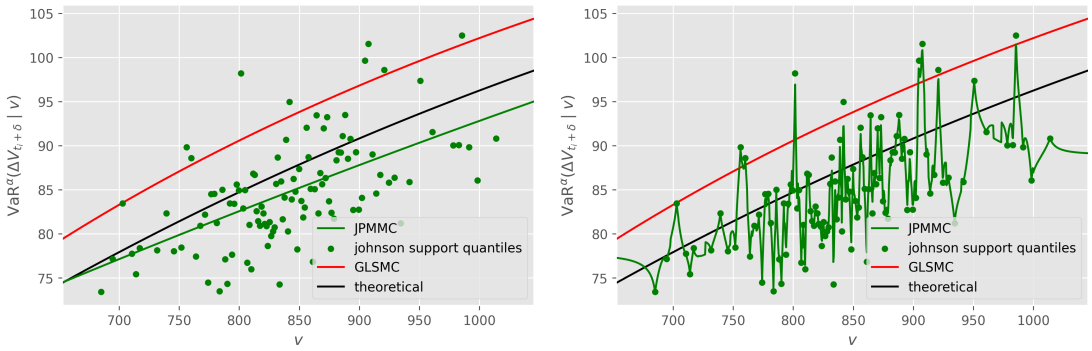

Figure 16: $\hat{g}_i(\cdot)$ when $t_i = 1/12$. Left: using linear regression. Right: using k-NN.

We remark that using k-NN is not adapted here, because of the large variance in the quantiles of fitted Johnson distributions. On the other hand, MSE-based techniques provide smoother estimates that are more appreciated by regulators. Notice that we have tried the same settings for the neural network quantile regression as in Section 4.1.2.

Finally, one can compare the methods investigated using a mean-square error as in Section 4.1.2, by displaying the Table 5 similarly to Table 2.

Id	Fit. type	Fit. params	Reg. type	MSE train	MSE test	Runtime
JPMMC*	PM	10k	LR4	3.10	**2.66**	12mn44s
JPMMC	PM	1k	LR4	5.59	**5.00**	4mn20s
JLSMC*	MM	LR4	NN3	18.53	**17.12**	1mn55s
-	MM	LR2	NN3	19.68	18.75	2mn46s
-	PM	1k	NN3	22.66	22.63	5mn19s
GLSMC*	MM	LR2	-	25.63	**25.89**	1s
-	MM	LR4	LR2	32.21	31.28	1mn45s
-	MM	LR4	LR4	50.30	40.63	1mn52s
-	PM	1k	NN3	46.43	45.85	4mn27s
JLSMC	MM	LR2	LR4	60.56	**51.31**	2mn40s
-	MM	LR2	LR2	143.03	133.25	2mn42s
JLSMC†	MM	GLM4	NN3	1942.54	**1944.58**	22mn33s
GLSMC	MM	LR2	-	11777.55	**11810.43**	1s
-	MM	GLM2	LR4	27008.60	25066.21	16mn31s

Table 5: Forward IM MSE swaption HW1F model for GLSMC, JLSMC, JPMMC.

As for the put in the Black-Scholes model, we observe that the JPMMC performs better than the JLSMC method. The best JLSMC setting for the put in the Black-Scholes model (denoted JLSMC in Table 2) is one of the worst for the swaption in the HW1F model (denoted JLSMC† in Table 5). It means that relying on GLM technique to guarantee positivity of conditional raw moments of order two and four may not be a good idea in practice. Finally, the GLSMC setting

of [AAGL17] is very bad, but the performance improves drastically if we consider the mean different from zero and estimate the first conditional raw moment using eg. linear regression with truncature degree 2 (GLSMC*). Notice the MSE is calculated based on the approximated analytical formula of $f_i(\cdot)$ given by equation (91), as we have checked that the forward IM profile obtained by nested Monte-Carlo matches the one obtained with the analytical expression. In Table 6, we also display the MSE of the IM profiles for neural quantile regression, similarly to Table 3.

Id	Architecture	Optimizer	Iterations	MSE train	MSE test	Runtime
JPMMC*	-	-	-	3.10	**2.66**	12mn44s
ML*	2(32)	Adam(5e-4)	50(128)	16.89	**16.34**	20mn1s
JLSMC*	-	-	-	18.53	**17.12**	1mn55s
-	1(4)	Adam(1e-3)	30(2048)	17.44	17.13	8mn56s
ML	2(10)	Adam(1e-3)	40(512)	18.83	**17.37**	9mn43s
-	1(8)	Adam(5e-4)	20(512)	22.21	19.55	10mn19s
-	2(64)	Adam(3e-4)	25(64)	23.92	22.55	17mn52s
GLSMC*	-	-	-	25.63	**25.89**	1s
-	1(10)	Adam(1e-2)	20(1024)	45.38	41.87	5mn30s
-	1(16)	Adam(1e-3)	30(256)	56.47	50.52	10mn55s
-	1(4)	SGD(1e-2)	15(1024)	69.99	52.75	4mn18s
-	1(128)	SGD(1e-2)	20(256)	70.86	67.17	6mn56s
ML$^\times$	2(4)	SGD(5e-2)	40(256)	460.12	1499.97	8mn44s

Table 6: Forward IM MSE swaption HW1F model for ML.

We observe that the ML method is not as competitive as in the previous case study. Indeed, the downgrade in performance comes from a convergence slowdown for timesteps between $T-\delta$ and T, as we have checked that most of MSE is due to large spikes in the IM profile at the last timesteps. The ML$^\times$ configuration shows that a misspecification in the hyperparameters can lead to poor results. In particular, we will prefer to use Adam optimizer instead of SGD in practice, as it provides faster convergence and better performance. In Figure 17 and Figure 18, we display the forward IM profiles of some methods from Table 5 and Table 6.

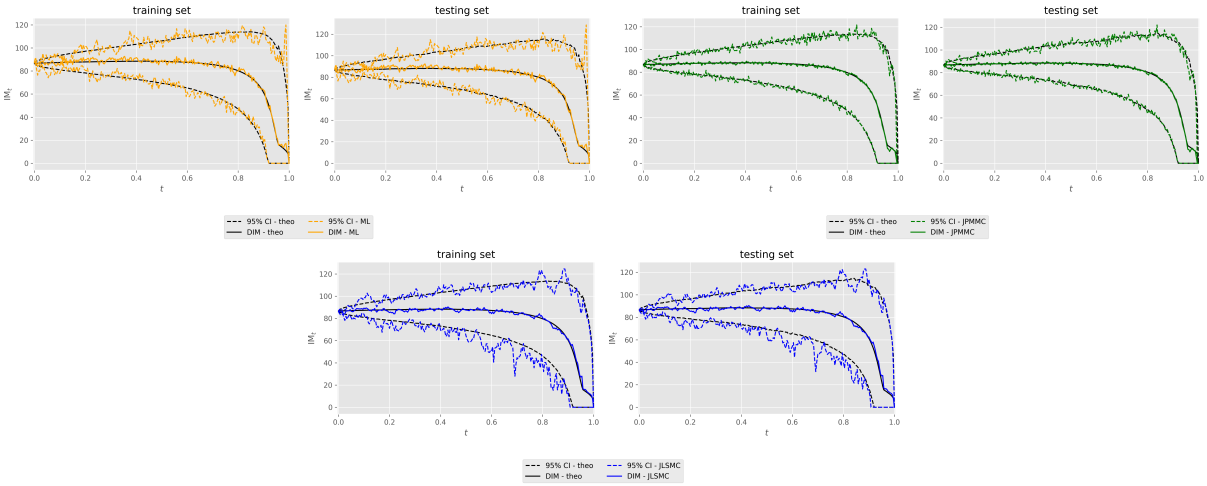

Figure 17: IM profiles. Top left, right: ML, JPMMC. Bottom: JLSMC.

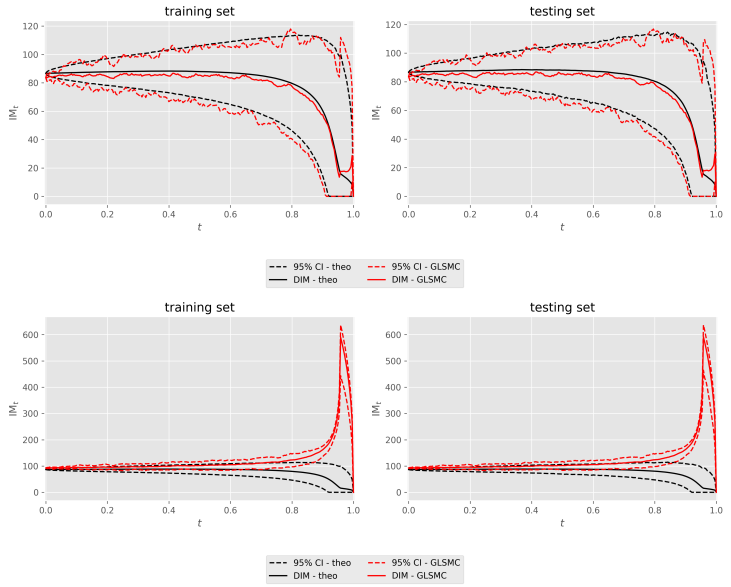

Figure 18: IM profiles. Top: GLSMC*. Bottom: GLSMC.

4.3 Conclusion

Throughout these numerical applications, we have seen that Monte-Carlo with Johnson distribution hypothesis is not recommended if one uses the JLSMC method of [MKN+18]. Indeed, the moment-matching procedure to fit the Johnson distributions requires conditional skewness and kurtosis, whose estimations are unstable and cannot guarantee robust results. Moreover, we can't control the runtime of AS99 algorithm used for moment-matching as it is an iterative procedure, and may even fail to converge. Still, Monte-Carlo with Johnson distribution is a good alternative to the Gaussian assumption if (and only if) one uses the JPMMC method we provide (and briefly mentionned in [GH21]), with the percentile-matching procedure to fit the Johnson distributions and minimizing a MSE to estimate $g_i(\cdot)$.

However, if simulating risk factors and pricing portfolio is too costly, then the JPMMC method is not adapted as it involves some nested Monte-Carlo simulations. In this case, we recommend to use the GLSMC method of [AAGL17]. Although the Gaussian hypothesis is not realistic, it provides stable performance since it does not require to estimate high-order conditional metrics. Contrary to [AAGL17], we recommend to assume the mean of the Gaussian distribution to be different from zero, so that one should estimate the first two conditional raw moments. Nonetheless, we observed in practice that the GLSMC method frequently produces `NaN` values for IM estimates from $T-\delta$ to T, due to negative estimated variances. Setting the mean to zero and using GLM to guarantee positivity solves this problem but consistently yields bad estimates.

A point of attention is that we have dealt with cases involving bijection between risk factors and portfolio value, which is often not the case in practice. Under this bijection property, it is equivalent to estimate the function $f_i(\cdot)$ or the function $g_i(\cdot)$. Otherwise (eg. simply consider a European put option in a Heston model), there is no equivalence and we would observe further discrepancies between the true forward IM profile based on $f_i(\cdot)$ and the one based on $\hat{g}_i(\cdot)$. The Machine Learning approach, ie. neural network quantile regression of [BCG$^+$24], does not have this drawbacks and accomodates easily for multiple risk factors. Moreover, it does not need any distribution hypothesis and has lower bias, compared to the distribution-based techniques. It does not require nested Monte-Carlo simulations and can't produce `NaN` values, contrary to JPMMC and GLSMC. Finally, one can easily control the runtime by increasing batch size or reducing the number of epochs, contrary to JLSMC. Still, training is costly and misspecifying the hyperparameters may lead to slow convergence of the neural network and hence poor results.

5 Further Developments and Conclusion

In this report, we have investigated some models used to estimate forward IM in the context of counterparty credit risk and uncleared OTC derivatives. This quantity arises first the calculation of capital requirements, and more precisely in the modeling of available collateral to the bank for collateralized exposure. It is also required for the computation of XVA such as MVA and CVA, highlighting the importance of its accurate estimation.

Both these applications underpin a Monte-Carlo framework, where one needs to simulate several risk factors value paths, as well as the corresponding portfolio price paths. The forward IM is defined as a conditional quantile at level α of the increment in portfolio price during the MPOR, and there is currently no consensus on the best way to estimate it.

We have tried several estimation techniques (ie. forward IM models), including nested Monte-Carlo, distribution-based methods assuming a Gaussian or Johnson distribution, and neural networks quantile regression. Evaluating different variants of these methods on two case studies, we have found that the JLSMC method of [MKN+18] is not recommended and we proposed the JPMMC method that solves the issues of JLSMC, but at the cost of some nested Monte-Carlo simulations. The GLSMC method of [AAGL17] may serve as a first baseline, although a Gaussian distribution hypothesis was not appropriate in our case studies. Finally, the neural network quantile regression provided good results, reducing bias compared to other methods. Although its training may be more costly, it is expected to achieve higher accuracy for more complex risk factors and portfolios.

There are further developments that could be considered. First one may test some non-parametric or sensitivity-based methods described in our state-of-the-art. One may also try to evaluate the forward IM models on more realistic case studies, featuring several risk factors and financial instruments, while calibrating the parameters to real market data. In addition, it would be the opportunity to assess the impact of learning the value-at-risk conditionally on the portfolio value instead of the risk factors value, ie. it may be enough to estimate $g_i(\cdot)$ instead of $f_i(\cdot)$, with the methods described in our state-of-the-art. Finally, one can investigate adjustment to external CCP/bank methodology as well as a sound backtesting of the forward IM models.

References

[AAC20] Claudio Albanese, Yannick Armenti, and Stephane Crepey. Xva metrics for ccp optimisation. Statistics and Risk Modeling with Applications in Finance and Insurance, 2020.

[AAGL17] Fabrizio Anfuso, Daniel Aziz, Paul Giltinan, and Klearchos Loukopoulos. A sound modelling and backtesting framework for forecasting initial margin requirements, 2017.

[AK24] Fabrizio Anfuso and Dimitris Karyampas. The wwr in the tail: a monte carlo framework for ccr stress testing, 2024.

[AKN13] Fabrizio Anfuso, Dimitris Karyampas, and Andreas Nawroth. A sound basel iii compliant framework for backtesting credit exposure models, 2013.

[ASP16] Leif Andersen, Alexander Sokol, and Michael Pykhtin. Rethinking the margin period of risk. Journal of Credit Risk, 2016.

[ASP17] Leif Andersen, Alexander Sokol, and Michael Pykhtin. Credit exposure in the presence of initial margin, 2017.

[Ass23] Interest Swap Derivatives Association. Isda simm methodology, 2023.

[ATCD18] Lokman Abbas-Turki, Stéphane Crépey, and Babacar Diallo. Xva principles, nested monte carlo strategies, and gpu optimizations, 2018.

[ATCS22] Lokman Abbas-Turki, Stéphane Crépey, and Bouazza Saadeddine. Pathwise cva regressions with oversimulated defaults, 2022.

[Aut19] Australian Prudential Regulation Authority. Margining and risk mitigation for non-centrally cleared derivatives, 2019.

[BBC12] Alexis Boukouvalas, Remi Barillec, and Dan Cornford. Gaussian process quantile regression using expectation propagation, 2012.

[BCG+24] D Barrera, S Crépey, E Gobet, Hoang-Dung Nguyen, and B Saadeddine. Statistical learning of value-at-risk and expected shortfall, 2024.

[BG90] P. K. Bhattacharya and Ashis K. Gangopadhyay. Kernel and nearest-neighbor estimation of a conditional quantile. The Annals of Statistics, 18, 1990.

[BJS99] Mark Britten-Jones and Stephen M. Schaefer. Non-linear value-at-risk. Review of Finance, 2, 1999.

[BM13] D. Brigo and F. Mercurio. Interest Rate Models Theory and Practice. Springer, 2013.

[BS73] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. Journal of Political Economy, 81, 1973.

[CA24] Vladimir Chorniy and Sergii Arkhypov. Dissecting initial margin forecasts: Models, limitations and backtesting, 2024.

[CBB14] Stéphane Crépey, Tomasz Bielecki, and S. Brigo. Counterparty Risk and Funding: A Tale of Two Puzzles. Chapman and Hall, 2014.

[CL02] Probal Chaudhuri and Wei-Yin Loh. Nonparametric estimation of conditional quantiles using quantile regression trees. Bernoulli, 8, 2002.

[CPS15] Isabelle Charlier, Davy Paindaveine, and Jérôme Saracco. Conditional quantile estimation based on optimal quantization: From theory to practice. Computational Statistics and Data Analysis, 91, 2015.

[EL10] Daniel Egloff and Markus Leippold. Quantile estimation with adaptative importance sampling. The Annals of Statistics, 38, 2010.

[FKLV18] Christian Fries, Peter Kohl-Landgraf, and Mario Viehmann. Melting sensitivities - exact and approximate margin valuation adjustments, 2018.

[GH21] Narayan Ganesan and Bernhard Hientzsch. Estimating future var from value samples and applications to future initial margin, 2021.

[GJ10] Michael B. Gordy and Sandeep Juneja. Nested simulation in portfolio risk measurement. Management Science, 56, 2010.

[Gly96] Peter W. Glynn. Importance sampling for monte carlo estimation of quantiles, 1996.

[GR11] Florence George and Kandethody Ramachandran. Estimation of parameters of johnson's system of distributions. Journal of Modern Applied Statistical Methods, 10, 2011.

[Gre15] Andrew Green. XVA: Credit, Funding and Capital Valuation Adjustments. Wiley, 2015.

[HHH76] I. D. Hill, R. Hill, and R. L. Holder. Algorithm as 99: Fitting johnson curves by moments. Journal of the Royal Statistical Society. Series C (Applied Statistics), 25, 1976.

[HJM92] David Heath, Robert Jarrow, and Andrew Morton. Bond pricing and the term structure of interest rates: A new methodology for contingent claims valuation. Econometrica, 60, 1992.

[HN98] Timothy C. Hesterberg and Barry L. Nelson. Control variates for probability and quantile estimation. Management Science, 44, 1998.

[HR09] Peter Huber and Elvezio Ronchetti. Robust Statistics. John Wiley and Sons, 2009.

[HS20] Brian Huge and Antoine Savine. Differential machine learning, 2020.

[HW08] Patrick Hagan and Graeme West. Methods for constructing a yield curve, 2008.

[HY96] Rob J. Hyndman and Fan Yanan. Sample quantiles in statistical packages. The American Statistician, 1996.

[Jam89] Farshid Jamshidian. An exact bond option formula. The Journal of Finance, 44, 1989.

[Kal02] O. Kallenberg. Foundations of Modern Probability. Springer New York, 2002.

[KB17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[Leh98] Erich Leo Lehmann. Elements of Large-Sample Theory. Springer New York, 1998.

[LLZ07] Youjuan Li, Yufeng Liu, and Ji Zhu. Quantile regression in reproducing kernel hilbert spaces. Journal of the American Statistical Association, 102, 2007.

[LS01] Francis Longstaff and Eduardo S Schwartz. Valuing american options by simulation: A simple least-squares approach. The Review of Financial Studies, 14, 2001.

[Mei06] Nicolai Meinshausen. Quantile regression forests. Journal of Machine Learning Research, 7, 2006.

[MKN+18] Thomas A. McWalter, JÃrg Kienitz, Nikolai Nowaczyk, Ralph Rudd, and Sarp K. Acar. Dynamic initial margin estimation based on quantiles of johnson distributions. Journal of Credit Risk, 2018.

[MZ21] Adrien Misko and Amine Zarbi. Metamodeling de l'initial margin/margin value adjustment, 2021.

[NW72] J. A. Nelder and R. W. M. Wedderburn. Generalized linear models. Journal of the Royal Statistical Society. Series A (General), 135, 1972.

[oBS20a] Basel Committee on Banking Supervision. Calculation of minimum risk-based capital requirements, 2020.

[oBS20b] Basel Committee on Banking Supervision. Counterparty credit risk overview, 2020.

[oBS20c] Basel Committee on Banking Supervision. Internal models method for counterparty credit risk, 2020.

[oBS20d] Basel Committee on Banking Supervision. Irb approach: minimum requirements to use irb approach, 2020.

[oBS20e] Basel Committee on Banking Supervision. Irb approach: risk components, 2020.

[oBS20f] Basel Committee on Banking Supervision. Irb approach: risk weight functions, 2020.

[oBS20g] Basel Committee on Banking Supervision. Margin requirements for non-centrally cleared derivatives, 2020.

[oBS20h] Basel Committee on Banking Supervision. Standardised approach to counterparty credit risk, 2020.

[oBS24a] Basel Committee on Banking Supervision. Counterparty credit risk definitions and terminology, 2024.

[oBS24b] Basel Committee on Banking Supervision. Standardised approach: individual exposures, 2024.

[OR01] Andrzej Okolewski and Tomasz Rychlik. Sharp distribution-free bounds on the bias in estimating quantiles via order statistics. Statistics and Probability Letters, 52, 2001.

[Par24] European Parliament. Regulation (eu) 2024/2987, 2024.

[Pyk09] Michael Pykhtin. Modeling credit exposure for collateralized counterparties. Journal of Credit risk, 2009.

[Pyk24] Michael Pykhtin. Credit exposure to leveraged counterparties, 2024.

[Sav18] Antoine Savine. Modern Computational Finance: AAD and Parallel Simulations. Wiley, 2018.

[Sob67] I.M Sobol'. On the distribution of points in a cube and the approximate evaluation of integrals. USSR Computational Mathematics and Mathematical Physics, 7, 1967.

[SS80] James F. Slifker and Samuel S. Shapiro. The johnson system: Selection and parameter estimation. Technometrics, 22, 1980.

[THM16] Camilo A. Garcia Trillos, Marc P. A. Henrard, and Andrea Macrina. Estimation of future initial margins in a multi-curve interest rate framework, 2016.

[VL24] Joel P. Villarino and Alvaro Leitao. On deep learning for computing the dynamic initial margin and margin value adjustment, 2024.

[WH90] Alan White and John Hull. Pricing interest-rate-derivative securities. Review of Financial Studies, 3, 1990.

A On Conditional Expectation, Quantile, and Superquantile

We highlight here the property of conditional expectation as the minimizer of a least-squares criterion. Let $(\Omega, \mathcal{F}, \mathbb{P})$ a probability space with $\mathbb{E}[\cdot]$ being the expectation under $\mathbb{P}$. Let X and Z be two random variables with values in $\mathbb{R}$, and let $\mathcal{G}$ be a sub σ-field of $\mathcal{F}$. Assume that X is $\mathcal{G}$-measurable and that Z is independent of $\mathcal{G}$.

A.1 Conditional expectation

We know that for any nonnegative (or bounded) Borel function Φ, the function ϕ defined for all $x \in \mathbb{R}$ by $\phi(x) \coloneqq \mathbb{E}[\Phi(x, Z)]$ is a measurable function and we have with probability one:

$$\phi(X) = \mathbb{E}[\Phi(X, Z) \mid \mathcal{G}]. \tag{92}$$

Moreover, assume $Y \coloneqq \Phi(X, Z) \in L^2(\Omega, \mathbb{P})$:

$$\mathbb{E}\left[Y^2\right] = \int_\Omega Y(\omega)^2 \mathbb{P}(d\omega) \tag{93}$$

$$= \int_{\mathbb{R}} y^2 \mu_Y(dy) < +\infty, \tag{94}$$

where $\mu_Y(\cdot)$ is the law of Y under $\mathbb{P}$ and $Y(\Omega) \subset \mathbb{R}$. Then $\phi(X)$ is the (unique) orthogonal projection of Y in Hilbert space $L^2(\Omega, \mathbb{P})$ onto the set of functions $\{g(X) : g \in L^2(\mathbb{R}, \mu_X)\} \subset L^2(\Omega, \mathbb{P})$. Moreover, it is easy to show that:

$$\phi = \arg\min_{h \in \mathcal{B}} \mathbb{E}\left[(Y - h(X))^2\right], \tag{95}$$

where $\mathcal{B}$ is the set of all functions in $L^2(\mathbb{R}, \mu_X)$. Now, from the properties of this Hilbert space, we know there exists an orthonormal basis of functions $\{L_k(\cdot)\}_k$ (possibly uncountable) such that:

$$\phi = \sum_{k=0}^{+\infty} \tilde{a}_k L_k, \tag{96}$$

where $\tilde{a}_k \coloneqq \langle \phi, L_k \rangle$, with $\langle \cdot, \cdot \rangle$ being the natural inner product in $L^2(\mathbb{R}, \mu_X)$. If X is a continuous random variable with probability density function (pdf.) $f_X(\cdot)$, then:

$$\langle \phi, L_k \rangle = \int_{\mathbb{R}} \phi(x) L_k(x) f_X(x) dx, \tag{97}$$

and $L^2(\mathbb{R}, \mu_X)$ can be seen as the $L^2(\mathbb{R})$[29] space weighted by f_X. Since equation (95) yields an infinite-dimensional optimization, we truncate the series in equation (96):

$$\phi(X) \approx \sum_{k=0}^{K} \tilde{a}_k L_k(X), \tag{98}$$

with $\tilde{a} \coloneqq (\tilde{a}_0, \cdots, \tilde{a}_K)^T$ being such that:

$$\tilde{a} = \arg \min_{a \in \mathbb{R}^{K+1}} \mathbb{E}\left[\left(Y - L(X)a \right)^2 \right], \tag{99}$$

where $L(X) = (L_0(X), \cdots, L_K(X))$. Assuming we are not able to express easily the probability density functions involved, it may be difficult to find an orthonormal basis $\{L_k\}_k$ of $L^2(\mathbb{R}, \mu_X)$. Thus, we usually rely on a simpler weighted $L^2(\mathbb{R})$ space. For instance, the weight $x \mapsto e^{-x}$ would be the pdf of $X \sim \mathcal{E}(1)$, and we would look for a basis $\{L_k(\cdot)\}_k$ such that:

$$\langle L_i, L_j \rangle = \int_{\mathbb{R}_+} L_i(x) L_j(x) e^{-x} dx = \delta_{ij}. \tag{100}$$

Such basis can be given by the Laguerre polynomials, which are obtained thanks to the Gram-Schmidt algorithm that orthogonalizes the monomials $\{x \mapsto x^k\}_k$ in this weighted $L^2(\mathbb{R})$ space. Notice that the Laguerre polynomials can be normalized so that they form an orthonormal basis.

Most of the time, we are not able to calculate explicitly the expectation in equation (99). However, if we have access to iid. samples $\{(X_1, Y_1), \cdots, (X_M, Y_M)\}$, we can approximate it by its empirical counterpart, building on the Ordinary Least Squares (OLS) method:

$$\hat{a} = \arg \min_{a \in \mathbb{R}^{K+1}} \frac{1}{M} \sum_{m=1}^{M} \left(Y_m - \mathbf{L}(X_m)a \right)^2 \tag{101}$$

$$= \left(\mathbf{L}^T \mathbf{L} \right)^{-1} \mathbf{L}^T \mathbf{Y}, \tag{102}$$

where $\mathbf{L} \coloneqq [L(X_1), \cdots, L(X_M)]^T \in \mathbb{R}^{M \times K+1}$ and $\mathbf{Y} \coloneqq (Y_1, \cdots, Y_M) \in \mathbb{R}^M$. Then, we have:

$$\phi(X) \approx \hat{\phi}(X) = \sum_{k=0}^{K} \hat{a}_k L_k(X). \tag{103}$$

A.2 Conditional quantile

Assume now we want to estimate a conditional quantile (ie. a conditional value-at-risk) instead of a conditional expectation. We know there exist a function ψ such that:

$$\psi(X) = \mathrm{VaR}^{\alpha} \left(\Phi(X, Z) \mid \mathcal{G} \right). \tag{104}$$

[29] The measure being the Lebesgue measure.

It is easy to show that such function is given by:

$$\psi = \arg\min_{h \in \mathcal{B}} \mathbb{E}\left[p(Y, h(X))\right], \tag{105}$$

where $Y \coloneqq \Phi(X, Z)$ and $p(\cdot, \cdot)$ is the pinball loss displayed in Figure 19, such that:

$$p(y, \hat{y}) \coloneqq \frac{1}{1-\alpha}(y - \hat{y})^+ + \hat{y}. \tag{106}$$

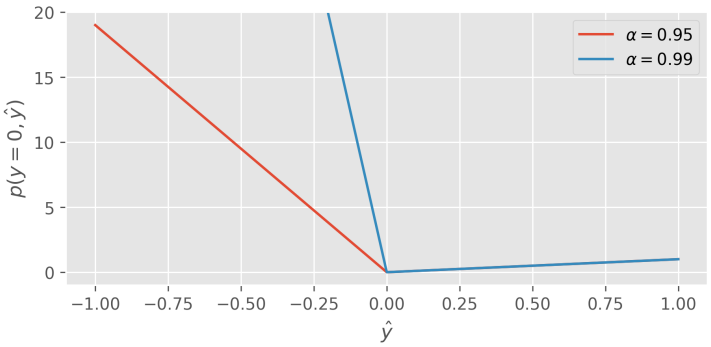

Figure 19: Pinball loss for $y = 0$ and $\alpha = 0.95,\ 0.99$.

If we have access to iid. samples $\{(X_1, Y_1), \cdots, (X_M, Y_M)\}$, we can approximate the expectation in equation (105) by its empirical counterpart:

$$\psi \approx \arg\min_{h \in \mathcal{B}} \frac{1}{M} \sum_{m=1}^{M} p(Y_m, h(X_m)). \tag{107}$$

As in Appendix A.1, one could be tempted to use Hilbert space properties by performing the mnimization over functions $h(\cdot)$ defined as in equation (98):

$$\hat{a} = \arg\min_{a \in \mathbb{R}^{K+1}} \frac{1}{M} \sum_{m=1}^{M} p\left(Y_m, L(X_m)a\right). \tag{108}$$

Notice that this empirical loss is convex wrt. a, but does not feature a quadratic expression as for the conditional expectation. Hence, we don't have an analytical expression for $\hat{a}$ and the minimization problem can be solved by the simplex method (eg. `scipy.optimize.linprog` in `Python`) applied to the following linear program:

$$\begin{aligned} &\min_{a \in \mathbb{R}^{K+1},\, b \in \mathbb{R}^M} \frac{1}{M} \sum_{m=1}^{M} \frac{1}{1-\alpha} b_m + L(X_m)a, \\ &\text{st.} \begin{cases} b_m \geq Y_m - L(X_m)a, & \forall m \in [|1, M|], \\ b_m \geq 0, & \forall m \in [|1, M|]. \end{cases} \end{aligned} \tag{109}$$

Then, we would have:

$$\hat{\psi}(X) = \sum_{k=0}^{K} \hat{a}_k L_k(X), \tag{110}$$

where $\hat{a}$ is the solution of the linear program (109) corresponding to a.

If one wants to derive an analytical expression for the value-at-risk of a given random variable, the following property is of high interest.

Proposition 1. *Assume in addition that $\Phi(X, \cdot)$ is strictly increasing. Then, we have:*

$$\begin{aligned} \psi(X) &= VaR^{\alpha}\left(\Phi(X, Z) \mid \mathcal{G}\right) \\ &= \Phi\left(X, VaR^{\alpha}(Z)\right). \end{aligned} \tag{111}$$

If $\Phi(X, \cdot)$ is strictly decreasing, then $\psi(X) = \Phi\left(X, VaR^{1-\alpha}(Z)\right)$.

Proof. First, let us denote $U \coloneqq \Phi(x, Z)$ and $g(z) \coloneqq \Phi(x, z)$ for a given $x \in \mathbb{R}$. Notice that if g is strictly increasing, then g^{-1} is also strictly increasing. Indeed,

$$g \circ g^{-1}(z) = z \Rightarrow \left(g^{-1}\right)'(z) \cdot g' \circ g^{-1}(z) = 1 \tag{112}$$

$$\Leftrightarrow \left(g^{-1}\right)'(z) = \frac{1}{g' \circ g^{-1}(z)}. \tag{113}$$

Since $g' > 0$, we have $\left(g^{-1}\right)' > 0$ and so g^{-1} is strictly increasing. Thus, we have:

$$F_U(u) \coloneqq \mathbb{P}(U \leq u) = \mathbb{P}(g(Z) \leq u) \tag{114}$$

$$= \mathbb{P}(Z \leq g^{-1}(u)) \tag{115}$$

$$= F_Z(g^{-1}(u)), \tag{116}$$

where F_U and F_Z are the cumulative density functions (cdf.) of U and Z, respectively. Then, we get:

$$u \geq \mathrm{VaR}^{\alpha}(\Phi(x, Z)) \Leftrightarrow F_U(u) \geq \alpha \tag{117}$$

$$\Leftrightarrow F_Z(g^{-1}(u)) \geq \alpha \tag{118}$$

$$\Leftrightarrow g^{-1}(u) \geq \mathrm{VaR}^{\alpha}(Z) \tag{119}$$

$$\Leftrightarrow u \geq g\left(\mathrm{VaR}^{\alpha}(Z)\right) \tag{120}$$

$$\Leftrightarrow u \geq \Phi\left(x, \mathrm{VaR}^{\alpha}(Z)\right). \tag{121}$$

Thus, $\mathrm{VaR}^{\alpha}\left(\Phi(x, Z)\right) = \Phi\left(x, \mathrm{VaR}^{\alpha}\left(Z\right)\right)$ for all $x \in \mathbb{R}$. Now if g is strictly decreasing, it is easy

to show that g^{-1} is also strictly decreasing and that $F_U(u) = 1 - F_Z(g^{-1}(u))$. Thus, we have:

$$\begin{aligned} u \geq \mathrm{VaR}^{\alpha}(\Phi(x, Z)) &\Leftrightarrow F_U(u) \geq \alpha && (122)\\ &\Leftrightarrow 1 - F_Z(g^{-1}(u)) \geq \alpha && (123)\\ &\Leftrightarrow F_Z(g^{-1}(u)) \leq 1 - \alpha && (124)\\ &\Leftrightarrow g^{-1}(u) \leq \mathrm{VaR}^{1-\alpha}(Z) && (125)\\ &\Leftrightarrow u \geq g\left(\mathrm{VaR}^{1-\alpha}(Z)\right) && (126)\\ &\Leftrightarrow u \geq \Phi\left(x, \mathrm{VaR}^{1-\alpha}(Z)\right), && (127) \end{aligned}$$

and we find that $\mathrm{VaR}^{\alpha}\left(\Phi(x, Z)\right) = \Phi\left(x, \mathrm{VaR}^{1-\alpha}\left(Z\right)\right)$ for all $x \in \mathbb{R}$. □

A.3 Conditional superquantile

Assume we now want to estimate a conditional superquantile (ie. a conditional expected shortfall) in addition to a conditional quantile. We know there exist a function s such that:

$$\begin{aligned} s(X) &= \mathrm{ES}^{\alpha}\left(\Phi(X, Z) \mid \mathcal{G}\right) && (128)\\ &= \mathbb{E}\left[p(Y, \psi(X)) \mid X\right], && (129) \end{aligned}$$

where $Y \coloneqq \Phi(X, Z)$, $\psi(\cdot)$ is the conditional quantile function defined in equation (105), and $p(\cdot, \cdot)$ is the pinball loss of equation (106). Thus, it is easy to see that:

$$\begin{aligned} s(X) &= \mathbb{E}\left[p(Y, \psi(X)) \mid X\right] && (130)\\ \Leftrightarrow (s - \psi)(X) &= \mathbb{E}\left[p(Y, \psi(X)) - \psi(X) \mid X\right] && (131)\\ \Leftrightarrow r(X) &= \mathbb{E}\left[p(Y, \psi(X)) - \psi(X) \mid X\right], && (132) \end{aligned}$$

where $r \coloneqq s - \psi$. Learning the residual r is then equivalent to learning a conditional expectation as in Appendix A.1:

$$r = \arg\min_{h \in \mathcal{B}} \mathbb{E}\left[\left(p(Y, \psi(X)) - \psi(X) - h(X)\right)^2\right], \tag{133}$$

so that[30] $s = \psi + r$. If we have access to iid. samples $\{(X_1, Y_1), \cdots, (X_M, Y_M)\}$, we first estimate the function ψ by $\hat{\psi}$, and we then approximate the expectation in equation (133) by its empirical counterpart:

$$r \approx \arg\min_{h \in \mathcal{B}} \frac{1}{M} \sum_{m=1}^{M} \left(p(Y_m, \hat{\psi}(X_m)) - \hat{\psi}(X_m) - h(X_m)\right)^2. \tag{134}$$

[30] There may be several functions that satisfy equations (105)-(133), but one can take any of them to determine the expected shortfall function.

As in Appendix A.1, we can perform minimization over functions $h(\cdot)$ defined as in equation (98):

$$\hat{a} = \arg\min_{a\in\mathbb{R}^{K+1}} \frac{1}{M}\sum_{m=1}^{M}\left(p(Y_m,\hat{\psi}(X_m)) - \hat{\psi}(X_m) - L(X_m)a\right)^2 \tag{135}$$

$$= \left(\mathbf{L}^T\mathbf{L}\right)^{-1}\mathbf{L}^T\chi, \tag{136}$$

where $\chi \coloneqq (\chi_1,\cdots,\chi_M) \in \mathbb{R}^M$ with $\chi_m \coloneqq p(Y_m,\hat{\psi}(X_m)) - \hat{\psi}(X_m)$. Then, we would have:

$$r(X) \approx \hat{r}(X) = \sum_{k=0}^{K}\hat{a}_k L_k(X), \tag{137}$$

so that $\hat{s} = \hat{\psi} + \hat{r}$. Notice that we usually learn the residual $r = s - \psi$ instead of the expected shortfall s directly, because it may speed up convergence if one uses neural network techniques.

It is also useful to recall the following property between superquantile and quantile, derived from Bayes theorem.

Proposition 2. *The following property holds.*

$$\mathbb{E}\left[Y\cdot\mathbf{1}_{Y>VaR^{\alpha}(Y|\mathcal{G})} \mid \mathcal{G}\right] = (1-\alpha)ES^{\alpha}(Y\mid\mathcal{G}). \tag{138}$$

B On Neural Networks and Stochastic Gradient Descent

In this Appendix, we recall the main theoretical results on feed-forward neural networks and stochastic gradient descent (SGD). Assume we want to solve the following stochastic optimization problem over the set of Borel functions $\mathcal{B}$ from $\mathbb{R}^d$ to $\mathbb{R}$:

$$\min_{h\in\mathcal{B}}\mathbb{E}\left[L(Y,h(X))\right], \tag{139}$$

where $L(\cdot,\cdot):\mathbb{R}\times\mathbb{R}\mapsto\mathbb{R}$ is a given function, and (X,Y) is a couple of random variables with values in $\mathbb{R}^d\times\mathbb{R}$, defined on a given probability space. As this problem has infinite dimension, assume we minimize the loss over the set of neural networks functions $\mathcal{H}_\theta$ parameterized by θ, such that:

$$\theta^\star \coloneqq \arg\min_{\theta}\mathbb{L}(\theta), \tag{140}$$

where $\mathbb{L}(\theta) \coloneqq \mathbb{E}\left[\mathcal{L}(X,Y;\theta)\right]$ and $\mathcal{L}(x,y;\theta) \coloneqq L\left(y,h_\theta(x)\right)$, which leads to:

$$h_{\theta^\star} = \arg\min_{h_\theta\in\mathcal{H}_\theta}\mathbb{E}\left[L\left(Y,h_\theta(X)\right)\right]. \tag{141}$$

At this point, we should specify what we exactly mean by a neural network function h_θ. Here, we consider a simple feedforward neural network structure mapping an input in $\mathbb{R}^d$ to an output

in $\mathbb{R}$:

$$\mathcal{H}_\theta = \left\{ \mathbb{R}^d \ni x \mapsto W_{l+1}\sigma\left(W_l\sigma\left(\cdots\sigma\left(W_1\tilde{x}\right)\right)\right) \in \mathbb{R} \right\}, \tag{142}$$

where $\tilde{x} = [x,\, 1]^T \in \mathbb{R}^{d+1}$ is the augmented input vector to account for bias terms, and $W_i \in \mathbb{R}^{n_i \times n_{i-1}}$ is the weight matrix of the i-th layer, with $n_0 = d+1$ and $n_{l+1} = 1$. Moreover, $\sigma(\cdot)$ is a (potentially) non-linear activation function (eg. ReLU, sigmoid, tanh) that is applied element-wise to a given vector. A stylized representation of such neural network function is displayed in Figure 20.

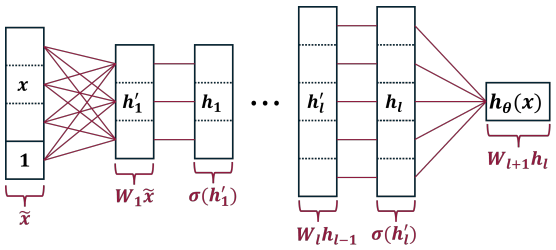

Figure 20: Stylized representation of a feedforward neural network.

The parameter θ is then defined as:

$$\theta = \{W_1, W_2, \cdots, W_{l+1}\}. \tag{143}$$

Coming back to our stochastic optimization problem, one usually doesn't know the joint law of (X, Y), so that it is not possible to explicitly compute $\mathbb{L}(\theta)$ or $\nabla\mathbb{L}(\theta)$. When writing gradient with respect to θ, we consider θ to be the flattened array of all weight matrices coordinates. If we have access to iid. samples $\{(X_1, Y_1), \cdots, (X_M, Y_M)\}$, we can use the trick of stochastic gradient descent given in its simplest version by Algorithm 3.

Algorithm 3: SGD for a feedforward neural network

Input: a sequence of learning rates $(\alpha_k)_{k\geq 1}$ and a number of epochs K.

Initialize $\hat{\theta}$

for epoch $k = 1$ **to** K **do**

  Shuffle the dataset $\{(X_1, Y_1), \cdots, (X_M, Y_M)\}$

  for $m = 1$ **to** M **do**

    $\hat{\theta} \leftarrow \hat{\theta} - \alpha_k \nabla_\theta \mathcal{L}\left(X_m, Y_m; \hat{\theta}\right)$

The trick is that $\nabla_\theta \mathcal{L}\left(X_m, Y_m; \hat{\theta}\right)$ is an unbiased estimator of $\nabla\mathbb{L}(\hat{\theta})$ at each step, so that we may converge to a local minimum of the loss function $\mathbb{L}(\cdot)$ with an appropriate learning rate sequence. As neural network functions are non-convex in θ, there is no guarantee to converge to a global minimum but the noisy property of SGD (through dataset shuffling) helps escaping saddle points of the loss function landscape. Notice that this algorithm belongs to the broader

class of stochastic approximation algorithms. This approach is effective in practice and has been extended to have a better control over the variance, numerical stability, or global convergence. For instance, we can penalize the loss (Lasso, Ridge), use mini-batches in SGD, or choose more advanced techniques such as Adam (see [KB17]) and others. The mini-batch SGD is presented in Algorithm 4.

Algorithm 4: Mini-batch SGD for a feedforward neural network

Input: a sequence of learning rates $(\alpha_k)_{k\geq 1}$, a number of epochs K, and a partition $\mathcal{J}$ of $\{1, \cdots, M\}$.

Initialize $\hat{\theta}$

for epoch $k = 1$ **to** K **do**

  Shuffle the dataset $\{(X_1, Y_1), \cdots, (X_M, Y_M)\}$

  for batch $J \in \mathcal{J}$ **do**

    $\hat{\theta} \leftarrow \hat{\theta} - \alpha_k \frac{1}{\text{Card}(J)} \sum_{m\in J} \nabla_\theta \mathcal{L}\left(X_m, Y_m; \hat{\theta}\right)$

It enables to reduce the variance of the gradient unbiased estimator:

$$\mathbb{V}\left[\frac{1}{\text{Card}(J)} \sum_{m\in J} \nabla_\theta \mathcal{L}\left(X_m, Y_m; \hat{\theta}\right)\right] = \frac{1}{\text{Card}(J)} \mathbb{V}\left[\nabla_\theta \mathcal{L}\left(X, Y; \hat{\theta}\right)\right]. \tag{144}$$

Although these smoother iterates may increase the quality of convergence, taking a too large batch size $\text{Card}(J)$ may prevent the algorithm to escape local minima and may decrease convergence speed since updating the parameters is more costly. Notice that the classic SGD corresponds to the case where $\mathcal{J} = \{\{1\}, \cdots, \{M\}\}$, ie. each batch contains only one observation.

The SGD algorithm (or any of its variants) then produces an estimator $\hat{\theta}$ of the optimal parameter $\theta^\star$, so that we can define the learned neural network function by $\hat{h} \coloneqq h_{\hat{\theta}}$. The function $\theta \mapsto \mathcal{L}(x, y; \theta)$ may not be differentiable everywhere[31] for all coordinates of θ, eg. if we have ReLu activation functions or if $\hat{y} \mapsto L(y, \hat{y})$ is not differentiable everywhere, like the pinball loss in equation (106) of Appendix A.2. In this case, one simply replaces the gradient by a given subgradient in the SGD algorithm. Fortunately, most deep learning libraries such as `pytorch` or `tensorflow` implement this twist in their SGD and backpropagation (see later) algorithms, so that we do not have to worry about it.

One may argue that we could have taken the empirical version of $\mathbb{L}(\cdot)$ based on our dataset of iid. observations, as we have done in Appendix A.1 or Appendix A.2:

$$\theta^\star \approx \arg\min_\theta \bar{L}(\theta), \tag{145}$$

where $\bar{L}(\theta) \coloneqq \frac{1}{M} \sum_{m=1}^M \mathcal{L}(X_m, Y_m; \theta)$. Then, we would perform the minimization over θ using

[31] The function should be at least continuous in all the coordinates of θ.

a simple gradient descent algorithm as presented in Algorithm 5.

Algorithm 5: Gradient Descent for a feedforward neural network

Input: a sequence of learning rates $(\alpha_k)_{k\geq 1}$ and a number of epochs K.

Initialize $\hat{\theta}$

for epoch $k = 1$ **to** K **do**

  $\hat{\theta} \leftarrow \hat{\theta} - \alpha_k \nabla \bar{L}(\hat{\theta})$

However, as guessed before for the limits of mini-batch SGD, this approach is not suitable for neural networks and large datasets for several reasons. First, as there is no random effect in the classic gradient descent, we have less chance to reach the global minima (or a good local minima) since $\bar{L}(\cdot)$ is not convex. Second, the gradient of $\bar{L}(\cdot)$ is given by:

$$\nabla \bar{L}(\theta) = \frac{1}{M} \sum_{m=1}^{M} \nabla_\theta \mathcal{L}(X_m, Y_m, \theta) \tag{146}$$

which requires to evaluate the gradient for each observation of the dataset *before* making a weight update, so that the cost of a weight update is $\mathcal{O}(M)$ versus $\mathcal{O}(1)$ for SGD. Hence, SGD provides rapid and noisy updates, speeding up convergence and potentially escaping shallow minima, contrary to the classic gradient descent.

Finally, one should mention how to calculate the gradient of the loss function $\mathcal{L}(x, y; \theta)$ with respect to the parameters θ. First, one has to compute $h_\theta(x)$ by a "forward pass" through the neural network, starting from the input layer and going through each layer until the output layer. Then, one calculates the value of $\nabla_\theta \mathcal{L}(x, y; \theta)$ using the so-called "backpropagation" technique, which is the Automatic Adjoint Differentiation (AAD) method applied to neural networks. The idea is that every evaluation of the neural network (and hence of the loss function) can be seen as a directed acyclic graph of elementary operations, and the backpropagation algorithm allows to compute the gradient of the loss function with respect to the parameters θ by traversing this graph backwards. This technique also enables to accomodate immediatly for different neural network architectures contrary to a symbolic (ie. analytical) differentiation approach. The cost of backpropagation is only a small constant multiple of the cost of a forward pass, making large-scale neural networks training tractable. Moreover, this algorithm is implemented in most deep learning libraries, so that one does not have to focus on the details of the backpropagation algorithm. For more information on AAD, we refer the interested reader to [Sav18].

C On Johnson Distributions

C.1 The types of Johnson distributions

The Johnson distribution family is a flexible family of probability distributions that can model a wide range of shapes, including those with skewness and kurtosis. More precisely, it is possible to match any value of skewness and kurtosis. It is particularly useful for modeling financial data, which often exhibit non-normal characteristics. The Johnson distribution family consists of four types: S_L for *log-normal*, S_U for *unbounded*, S_B for *bounded*, S_T for *two-ordinate* (which is a special case of S_B), and S_N for *normal*. If X follows a Johnson distribution, then we denote $X \sim \text{Johnson}\left(J(\cdot), \gamma, \delta, \xi, \lambda\right)$ such that:

$$\gamma + \delta \cdot J\left(\frac{X-\xi}{\lambda}\right) \sim \mathcal{N}\left(0,1\right), \tag{147}$$

where the parameters $(\gamma, \delta, \xi, \lambda)$ control the location, scale, and shape of the distribution, while $J(\cdot)$ is a transformation function that depends on the type of Johnson distribution:

$$J(u) \coloneqq \begin{cases} \ln(u) & (S_L), \\ \ln(u + \sqrt{u^2+1}) = \sinh^{-1}(u) & (S_U), \\ \ln(u/(1-u)) & (S_B) \text{ or } (S_T), \\ u & (S_N). \end{cases}$$

Noticing that the cdf. of X is given by:

$$F_X(x) = \Phi\left(\gamma + \delta \cdot J\left(\frac{x-\xi}{\lambda}\right)\right), \tag{148}$$

where $\Phi(\cdot)$ is the cdf. of $\mathcal{N}\left(0,1\right)$, it is easy to show that the pdf. of X is given by:

$$f_X(x) = \frac{\delta}{\lambda} J'\left(\frac{x-\xi}{\lambda}\right) \cdot \phi\left(\gamma + \delta \cdot J\left(\frac{x-\xi}{\lambda}\right)\right), \tag{149}$$

where $\phi(\cdot)$ is the pdf. of $\mathcal{N}\left(0,1\right)$. The derivative of J is given by:

$$J'(u) = \begin{cases} 1/u & (S_L), \\ \frac{1}{\sqrt{1+u^2}} & (S_U), \\ \frac{1}{u(1-u)} & (S_B) \text{ or } (S_T), \\ 1 & (S_N). \end{cases} \tag{150}$$

Moreover, from equation (148), the quantile at level α of X is given by:

$$q = \xi + \lambda J^{-1}\left(\frac{\Phi^{-1}(\alpha) - \gamma}{\delta}\right), \tag{151}$$

where the inverse of $J(\cdot)$ is given by:

$$J^{-1}(u) = \begin{cases} e^u & (S_L), \\ \sinh(u) & (S_U), \\ \frac{1}{1+e^{-y}} & (S_B) \text{ or } (S_T), \\ u & (S_N). \end{cases} \tag{152}$$

One can set in practice $\xi = 0$ when the family is S_N without loss of generality, implying an appropriate rescaling of γ. Indeed in this case, $X \sim \mathcal{N}\left(\frac{\xi-\gamma\lambda}{\delta}, \frac{\lambda^2}{\delta^2}\right)$ so that ξ and γ only act on one parameter of the distribution. Moreover, one should be very careful when using the `scipy.stats` functions available in `Python` as they do not follow the same convention as the one used in this report. One should use them in the following way:

$$\begin{cases} \texttt{lognorm.xxx}\left(\text{yyy}, \texttt{s=}\frac{1}{|\delta|}, \texttt{loc=}\xi, \texttt{scale=}\lambda e^{\frac{-\gamma}{\delta}}\right) & (S_L), \\ \texttt{johnsonsu.xxx}\left(\text{yyy}, \texttt{a=}\gamma, \texttt{b=}\delta, \texttt{loc=}\xi, \texttt{scale=}\lambda\right) & (S_U), \\ \texttt{johnsonsb.xxx}\left(\text{yyy}, \texttt{a=}\gamma, \texttt{b=}\delta, \texttt{loc=}\xi, \texttt{scale=}\lambda\right) & (S_B) \text{ or } (S_T), \\ \texttt{norm.xxx}\left(\text{yyy}, \texttt{loc=}\frac{\xi-\gamma\lambda}{\delta}, \texttt{scale=}\frac{|\lambda|}{|\delta|}\right) & (S_N). \end{cases}$$

The function $J(\cdot)$ is uniquely determined by the square of the skewness γ and the kurtosis κ defined as:

$$\begin{cases} \gamma & := \dfrac{\mathbb{E}\left[(X-\mathbb{E}[X])^3\right]}{\left(\mathbb{E}\left[(X-\mathbb{E}[X])^2\right]\right)^{3/2}}, \\ \kappa & := \dfrac{\mathbb{E}\left[(X-\mathbb{E}[X])^4\right]}{\left(\mathbb{E}\left[(X-\mathbb{E}[X])^2\right]\right)^{2}}. \end{cases} \tag{153}$$

Figure 21 shows the regions of Johnson functions (and so Johnson types) in the (γ^2, κ) plane. We see that the (S_T) type is indeed a special case of the (S_B) type when $\gamma^2 = \kappa + 1$. Moreover, we recover $(\gamma, \kappa) = (0, 3)$ for the normal distribution corresponding to the (S_N) type. The impossible region corresponds to the case where $\gamma^2 < \kappa + 1$ which is theoretically impossible for any distribution, but it can be observed in practice due estimation issues.

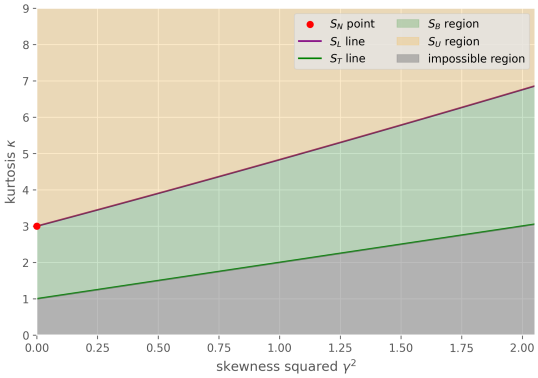

Figure 21: Johnson distribution types in the (γ^2, κ) plane.

C.2 Johnson parameters estimation

From the theoretical developments explained above, it is necessary to have a procedure to estimate the parameters of a Johnson distribution. We do not focus here on maximum likelihood estimation (MLE) for two reasons. First, it is not possible to compute analytically the maximum likelihood estimates, and maximizing the likelihood function is difficult because of its non-regularity [GR11]. Second, MLE methods involve computing various statistics based on iid. samples of X. However, in our case of forward initial margin and counterparty credit risk, we do not have access to iid. samples of $X = (\Delta V_{t+\delta} \mid v)$ for a given t and a given v. More precisely, we may generate iid. samples of X at a high cost provided that the MLE numerical method is fast enough, but it is not the case. We refer the reader to [GR11] for a detailed discussion on MLE methods for Johnson distributions.

C.2.1 Moment matching

Putting maximum likelihood methods aside, the most natural way to estimate the parameters of a Johnson distribution is to use the method of moments (also called moment fitting or moment matching). It consists in finding the parameters $(\gamma, \delta, \xi, \lambda)$ such that the first four theoretical Johnson moments match the moments we have specified. The function $J(\cdot)$ is entirely determined by the squared skewness and kurtosis computed based on our first four raw moments. In our case of initial margin and counterparty credit risk, the first four moments of $X = (\Delta V_{t+\delta} \mid v)$ are estimated by regression. However, the procedure of matching moments is non trivial and costly, with little control on the running time. It is given by the algorithm AS99 of [HHH76] which requires the mean, variance, skewness, and kurtosis of X as inputs. It was initially written in `Fortran` in 1976, then translated into an unofficial `Matlab` package in the 2010's, and finally implemented in `Python` through aGithub project in the last years. However, it is important to note that the `Matlab` code contains some errors that are also repeated in the `Python` implementation. I have corrected those I have seen in my `Python` code, which is available upon request. In practice, it may be possible that some parameters are not well defined because of the estimation of raw moments and higher order metrics such as variance, skewness, and kurtosis. Moreover, the moment fitting procedure could simply fail. In these case, we say that we have an impossible region and cancel the estimation.

C.2.2 Percentile matching

Because of the drawbacks of the moment matching method described above, it is interesting to explore other approaches. We consider here the percentile matching method of [SS80]. The first step is to compute four quantiles of X denoted x_{3z}, x_z, x_{-z}, and x_{-3z} respectively at the levels $\Phi(3z)$, $\Phi(z)$, $\Phi(-z)$, and $\Phi(-3z)$ for a given $z > 0$, where $\Phi(\cdot)$ is the cdf. of $\mathcal{N}(0,1)$. In our

case of forward initial margin and counterparty credit risk, it involves simulating iid. samples of $X = (\Delta V_{t+\delta} \mid v)$ at a high cost, so that the quantiles can be estimated by the corresponding empirical quantiles, potentially considering smoother definitions than the one of Definition 1. Based on these quantiles, we define the following metrics:

$$\begin{cases} m & \coloneqq x_{3z} - x_z, \\ n & \coloneqq x_{-z} - x_{-3z}, \\ p & \coloneqq x_z - x_{-z}, \end{cases} \tag{154}$$

The Johnson distribution type is then (S_B) if $d < 0.999$, (S_U) if $d > 1.001$, and (S_L) otherwise, where $d \coloneqq \frac{m \cdot n}{p^2}$. Notice that theoretically, the (S_L) type is set iif. $d = 1$. As this event has probability zero, we use in practice a tolerance threshold of ± 0.001. Moreover, the case of (S_T) is embedded in (S_B) while the case of (S_N) is not possible here as we assume that X is not normally distributed (otherwise we would have used a Gaussian distribution directly). Then, we can compute the parameters $(\gamma, \delta, \xi, \lambda)$ from Table 7, by denoting $q \coloneqq m/p$, $\bar{q} \coloneqq 1/q$, $r \coloneqq n/p$, and $\bar{r} \coloneqq 1/r$.

Type	S_L	S_U	S_B
γ	$\delta \ln\left(\frac{q-1}{p\sqrt{q}}\right)$	$\delta \sinh^{-1}\left(\frac{r-q}{2\sqrt{qr-1}}\right)$	$\delta \sinh^{-1}\left(\frac{(\bar{r}-\bar{q})\sqrt{(1+\bar{q})(1+\bar{r})-4}}{2(\bar{q}\bar{r}-1)}\right)$
δ	$\frac{2z}{\ln q}$	$\frac{2z}{\cosh^{-1}\left(\frac{q+r}{2}\right)}$	$\frac{z}{\cosh^{-1}\left(\frac{1}{2}\sqrt{(1+\bar{q})(1+\bar{r})}\right)}$
ξ	$\frac{x_z+x_{-z}}{2} - \frac{p}{2} \cdot \frac{q+1}{q-1}$	$\frac{x_z+x_{-z}}{2} + \frac{p(r-q)}{2(q+r-2)}$	$\frac{x_z+x_{-z}}{2} - \frac{\lambda}{2} + \frac{p(\bar{r}-\bar{q})}{2(\bar{q}\bar{r}-1)}$
λ	1	$\frac{2p\sqrt{qr-1}}{(qr-2)\sqrt{q+r-2}}$	$\frac{p\sqrt{((1+\bar{q})(1+\bar{r})-2)^2-4}}{\bar{q}\bar{r}-1}$

Table 7: Johnson parameters estimation in the percentile matching method.

As for the moment-matching method of Appendix C.2.1, it may be possible that some parameters are not well defined, because of the estimation of the quantiles x_{3z}, x_z, x_{-z}, and x_{-3z}. In this case, we can use the following heuristics:

- If $d \in [0.999, 1)$ and $q \leq 1$, we set the type to (S_B) instead of (S_L).
- If $d \in (1, 1.001]$ and $q \leq 1$, we set the type to (S_U) instead of (S_L).
- If $d = 1$ and $q \leq 1$, or if some parameters are not well defined for a given type, we say that it is an impossible region and we cancel the estimation.

An important question is how to choose the value of z. We have found that for our application, $z = 0.524$ or $z = 1$ usually yield robust results. Finally, notice that the percentile matching

method has been shown empirically to be more robust than the moment matching method, see [SS80].

D On One Factor Hull-White Model and Swaption Pricing

D.1 The one factor Hull-White model

The one factor Hull-White model (HW1F) of [WH90], also called extended Vasicek Hull-White model, is a popular model used to describe the evolution of the short rate (r_t) over time. In this section, we recall its main properties that we will use for the pricing of swaption in Appendix D.2 and we recall the expectation under $\mathbb{Q}$ is denoted $\mathbb{E}[\cdot]$. The interested reader can find more details in [BM13] although the notations are not exactly the same. More precisely, it assumes the following dynamics under the risk-neutral probability measure $\mathbb{Q}$:

$$dr_t = (\theta(t) - k \cdot r_t)dt + \sigma dW_t, \tag{155}$$

where $\theta(t)$ is a deterministic long-term mean function chosen to match the initial curve of zero-coupon (ZC) prices, k is the mean reversion speed, σ is the volatility of the short rate, and W is a $\mathbb{Q}$-Brownian motion. Let us denote $T \mapsto \hat{B}_0(T)$ the observed ZC price curve, and $B_t(T)$ the price of the ZC with maturity T at time t in our HW1F model. Notice that we don't specify here how to construct the observed ZC price curve from available market instruments with the so-called bootstrapping technique, see eg. [HW08]. Instead, we assume the ZC price curve is given, for instance from a market data provider. It can be shown that we calibrate perfectly to the observed ZC price curve, ie. $B_0(T) = \hat{B}_0(T)$, by setting:

$$\theta(t) = \partial_t \hat{f}(0,t) + k \cdot \hat{f}(0,t) + \frac{\sigma^2}{2k}\left(1 - e^{-2k(T-t)}\right), \tag{156}$$

where $T \mapsto \hat{f}(0,T) = -\partial_T \ln \hat{B}_0(T)$ is the observed instantaneous forward rate curve[32]. Moreover, the random variable r_t is Gaussian conditional on $\mathcal{F}_s$ under $\mathbb{Q}$:

$$r_t \mid \mathcal{F}_s \sim \mathcal{N}\left((r_s - \beta(s))e^{-k(t-s)} + \beta(t), \frac{\sigma^2}{2k}\left(1 - e^{-2k(t-s)}\right)\right), \tag{157}$$

where $\beta(t) = \hat{f}(0,t) + \frac{\sigma^2}{2k}\left(1 - e^{-k(T-t)}\right)^2$.We also know that the HW1F model is an affine structure model:

$$B_t(T) = \mathbb{E}\left[e^{-\int_t^T r_s ds} \mid \mathcal{F}_t\right] = e^{m(t,T) - n(t,T)r_t}, \tag{158}$$

[32]Recall that the instantaneous forward rate $f(t,\cdot)$ is such that $B_t(T) = e^{-\int_t^T f(t,s)ds}$.

where:

$$\begin{cases} n(t,T) &= \left(1 - e^{-k(T-t)}\right)/k, \\ m(t,T) &= \ln\left(\frac{\hat{B}_0(T)}{\hat{B}_0(t)}\right) + n(t,T)\hat{f}(0,t) - \frac{\sigma^2}{4k} n(t,T)^2 \left(1 - e^{-2kt}\right). \end{cases} \tag{159}$$

Finally, the HW1F model can be seen as a special case of a Gaussian Heath-Jarrow-Morton (HJM, see [HJM92]) model under $\mathbb{Q}$:

$$df(t,T) = \alpha(t,T)dt + \sigma(t,T)dW_t, \tag{160}$$

where $f(0,T) = \hat{f}(0,T)$, $\sigma(t,T) \coloneqq \sigma e^{-k(T-t)}$, and the drift $\alpha(t,T)$ is entirely determined by the HJM drift condition under absence of arbitrage opportunities:

$$\alpha(t,T) = \Sigma(t,T)\sigma(t,T), \tag{161}$$

where $\Sigma(t,T) \coloneqq \frac{\sigma}{k}(1 - e^{-k(T-t)})$.

D.2 Swaption pricing in HW1F

Assume we want to price a swaption in the HW1F model described in Appendix D.1. We denote T_0 the swaption maturity, T_1 the start of the underlying payer swap with evenly spaced fixing dates $\{T_1, \cdots, T_n\}$ such that $\Delta T \coloneqq T_i - T_{i-1}$. In this swap with a notional amount FV, we pay a fixed rate K and receive the interest rate $L(T_{i-1}, T_i)$ simply compounded on the period $[T_{i-1}, T_i]$ at time T_i, such that:

$$L(t,T) = \frac{1}{T-t}\left(\frac{1}{B_t(T)} - 1\right). \tag{162}$$

We display the stylized payer swap cash flows in Figure 22.

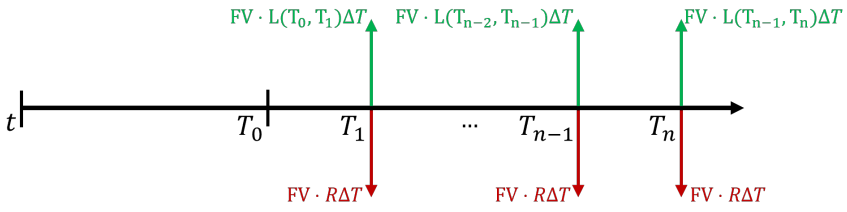

Figure 22: Stylized payer swap cash flows.

Thus, the payoff of the swaption at maturity T_0 is given by:

$$V_{T_0} = \left(V_{T_0}^{\text{swap}}\right)_+, \tag{163}$$

where $V_{T_0}^{\text{swap}}$ is the price at time T_0 of the swap described above. It is easy to show that:

$$V_{T_0}^{\text{swap}} = \text{FV} \cdot \left(1 - \sum_{k=1}^{n} c_k B_{T_0}(T_k)\right), \tag{164}$$

where $c_k = \mathbf{1}_{k=n} + \Delta T \cdot R$ for all $k \in [|1, n|]$. Let us denote the T_0-forward measure $\mathbb{Q}^{T_0}$ such that for all $t \leq T_0$:

$$\left.\frac{d\mathbb{Q}^{T_0}}{d\mathbb{Q}}\right|_{\mathcal{F}_t} \coloneqq \frac{B_t(T_0)}{B_0(T_0)B_t}. \tag{165}$$

The swaption price is a put price on a coupon-bearing bond, given under this probability measure by:

$$V_t = \text{FV} \cdot B_t(T_0) \cdot \mathbb{E}^{\mathbb{Q}^{T_0}}\left[\left(1 - \sum_{k=1}^{n} c_k B_{T_0}(T_k)\right)_+ \mid \mathcal{F}_t\right]. \tag{166}$$

We recall that the HW1F model is a Gaussian HJM model since r_t is Gaussian under $\mathbb{Q}$ from equation (157). Moreover, it has separable volatility so that $\sigma(t,T) = \zeta(t)\rho(T)$ with $\zeta(t) = e^{kt}$ and $\rho(T) = \sigma e^{-kT}$. Using equation (158), it can be shown that:

$$B_{T_0}(T_k) = e^{\mu(T_0,T_k) - \psi(T_0,T_k)Y_{T_0}}, \tag{167}$$

where $Y_{T_0} \sim \mathcal{N}(0,1)$ under $\mathbb{Q}^{T_0}$, and $\psi(T_0,T_k) = \frac{\sigma}{k}(e^{-kt} - e^{-kT})$. Moreover, we have:

$$\mu(t,T) = \ln\left(\frac{B_0(T)}{B_0(t)}\right) - I_1(t,T) + I_2(t,T), \tag{168}$$

where:

$$I_1(t,T) \coloneqq \frac{1}{2}\int_0^t \left(\Sigma(u,T)^2 - \Sigma(u,t)^2\right) du \tag{169}$$

$$= \frac{\sigma^2}{k^3}\left(1 + e^{-kT} - e^{-kt} - e^{-k(T-t)} + \frac{1}{4}\left(e^{-2k(T-t)} - 1\right)\right), \tag{170}$$

and $I_2(t,T) \coloneqq \int_0^t \Sigma(u,T_0)\left(\Sigma(u,T) - \Sigma(u,t)\right) du$, so that:

$$\begin{aligned} I_2(t,T) =& \frac{\sigma^2}{k^3}\Big(e^{-k(T-t)} - e^{-kT} - 1 + e^{-kt} \ - e^{-k(T+T_0-2t)} \\ &+ e^{-k(T+T_0)} + e^{-k(T_0-t)} - e^{-k(t+T_0)}\Big). \end{aligned} \tag{171}$$

Now, we would like to take the sum out of the positive part in equation (166), to get a sum of put prices. We will use the Jamshidian trick described in [Jam89] to do so. Let us define:

$$f(y) \coloneqq \sum_{k=1}^{n} f_k(y), \tag{172}$$

where $f_k(y) \coloneqq c_k e^{\mu(T_0,T_k)-\psi(T_0,T_k)y}$. Let us solve numerically $\bar{y}$ such that $f(\bar{y}) = 1$. Notice that there is a unique solution as f is continuous and strictly decreasing as a sum of strictly decreasing functions $(f_k)_k$ with limits 0 and $+\infty$ at $\pm\infty$. From the monotonicity and positivity of f_k and f, we have:

$$\left(1 - \sum_{k=1}^{n} c_k B_{T_0}(T_k)\right)_+ = \left(f(\bar{y}) - f(Y_{T_0})\right) \mathbf{1}_{f(\bar{y})>f(Y_{T_0})} \tag{173}$$

$$= \left(f(\bar{y}) - f(Y_{T_0})\right) \mathbf{1}_{\bar{y}<Y_{T_0}} \tag{174}$$

$$= \sum_{k=1}^{n} \left(f_k(\bar{y}) - f_k(Y_{T_0})\right) \mathbf{1}_{\bar{y}<Y_{T_0}} \tag{175}$$

$$= \sum_{k=1}^{n} \left(f_k(\bar{y}) - f_k(Y_{T_0})\right) \mathbf{1}_{f_k(\bar{y})>f_k(Y_{T_0})} \tag{176}$$

$$= \sum_{k=1}^{n} c_k \left(K_k - B_{T_0}(T_k)\right)_+, \tag{177}$$

where $K_k \coloneqq f_k(\bar{y})/c_k$. Thus, the swaption price is given by a weighted sum of the prices $\Pi^{\text{put}}(t, T_0, T_k, K_k)$ at time t of put options with maturity T_0 and strike K_k/c_k on the ZC with maturity T_k:

$$V_t = \text{FV} \cdot \sum_{k=1}^{n} c_k \cdot B_t(T_0) \mathbb{E}^{\mathbb{Q}^{T_0}} \left[\left(K_k - B_{T_0}(T_k)\right)_+ \mid \mathcal{F}_t\right] \tag{178}$$

$$= \text{FV} \cdot \sum_{k=1}^{n} c_k \cdot \Pi^{\text{put}}(t, T_0, T_k, K_k). \tag{179}$$

Finally, as the HW1F model is a Gaussian HJM model, it can be shown that:

$$\Pi^{\text{put}}(t, T, S, K) = K B_t(T) \Phi(-d_2) - B_t(S) \Phi(-d_1), \tag{180}$$

where Φ is the cdf. of $\mathcal{N}(0,1)$ and $d_{1,2} \coloneqq \frac{\ln\left(\frac{B_t(S)}{K B_t(T)}\right) \pm \frac{1}{2}\bar{v}(t,T,S)}{\sqrt{\bar{v}(t,S,T)}}$ with:

$$\bar{v}(t, T, S) \coloneqq \frac{\sigma}{k}\left(1 - e^{-k(T-S)}\right)\sqrt{\frac{1 - e^{-2k(T-t)}}{2k}}. \tag{181}$$

Notice that since the ZC price is a an analytical function of (t, r_t) from equation (158), the swaption price is also an analytical function $V_t =: u(t, r_t)$.