Pricing and hedging of XVAs : from classic numerical methods to supervised learning algorithms with applications in finance and insurance

Samy Mekkaoui

ENSAE Paris

Forvis Mazars

6 November 2024

Contents

Context and motivations :

- *XVAs* are a generic name for *X*-valuation adjustments which gained a lot of interest since the global financial crisis of 2008. They now represent a significant part of the risk department of financial institutions.
- *XVAs* are linked with high computational costs due to a nested Monte-Carlo structure in the pricing formulas.
- Banking and Insurance industries are looking for efficient numerical methods to manage their risks associated with the computation of *XVAs*.

Objectives :

- Implement new numerical methods based on supervised learning algorithms to compute efficiently *XVAs* and overcome the principal weaknesses of the Monte-Carlo approach.
- Show the potential applications of these numerical methods in finance and actuarial fields.

Table: Different Types of XVA

XVA	valuation adjustment	Expected Cost of the Bank
CVA	Credit Valuation Adjustment	Client Default Losses
DVA	Debt Valuation Adjustment	Bank Default Losses
FVA	Funding Valuation Adjustment	Funding expenses for variation margin
MVA	Margin Valuation Adjustment	Funding expenses for initial margin
KVA	Capital Valuation Adjustment	Remuneration of Shareholder capital at risk

- *CVA* and *DVA* refer to credit valuation adjustments. When both quantities are computed, we use the term *BCVA* as *Bilateral Credit Valuation Adjustment*.
- *FVA* and *MVA* refer to funding valuation adjustments and are still under debate in the industry in how they should be evaluated.
- *KVA* refers to the capital valuation adjustment and highly depends in the institution's policy.

Contents

Assuming a probability space $(\Omega, \mathcal{F})$ with Q a risk-neutral probability measure associated with a numeraire $B = (B_t)_{t \geqslant 0}$ with dynamics $dB_t = B_t r_t dt$ with r_t the short rate, the *CVA* can be computed as follows :

$$CVA_t = (1 - R^C)\mathbb{E}^Q[\mathbb{1}_{t \leqslant \tau^C \leqslant T}(V_{\tau^C})^+ \frac{B_t}{B_{\tau^C}}|\mathcal{G}_t] = (1 - R^C)\mathbb{E}^Q[\int_t^T \frac{B_t}{B_s}(V_s)^+ dH_s|\mathcal{G}_t]. \quad (1)$$

with :

- R^C the *recovery rate* for the counterparty C such as $LGD = 1 - R^C$.
- V_t the product/portfolio value at time t such that $(V_t)^+$ refers to counterparty *Exposure*.
- T the maturity of the product/portfolio.
- τ^C the time default of the counterparty C and $H_t = \mathbb{1}_{\tau^C \leqslant t}$.

Remark

The computation of CVA involves the computation of the portfolio value at any time which in the most common case needs to be performed using a numerical method like a Monte − Carlo procedure resulting in a nested Monte-Carlo.

By noting $G(t) = Q(\tau^C > t)$ and by supposing that τ^C admits a density probability function, we can rewrite CVA_0 as follows :

$$CVA_0 = -(1 - R^C) \int_0^T \mathbb{E}^Q[\frac{(V_t)^+}{B_t} | \tau = t] dG(t). \tag{2}$$

Under independance between exposure value of the portfolio and default time, equation (2) can be rewritten over a timegrid $0 = t_0 < t_1 < \ldots < t_N = T$ by :

$$CVA_0 \approx -(1 - R^C) \sum_{i=0}^{N-1} \mathbb{E}^Q[\frac{(V_{t_i})^+}{B_{t_i}}](G(t_{i+1}) - G(t_i)). \tag{3}$$

- $\mathbb{E}^Q[\frac{(V_t)^+}{B_t}]$ is called *Expected Positive Exposure* and is noted $EPE(t)$.
- $\mathbb{E}^Q[\frac{(V_t)^-}{B_t}]$ is called *Expected Negative Exposure* and is noted $ENE(t)$.

Remark

We recover the 3 *components of the credit risk in the* CVA_0 *expression with the the Loss Given Default (LGD) , the Probability of Default (PD) and the Exposure at Default (EAD).*

The *Margin Valuation Adjustment* is expected to capture the cost associated with the deposit of an initial margin in collateralized contracts and can be defined as follows :

$$DIM(t) = \mathbb{E}^Q[\frac{1}{B_t} IM(t)|\mathcal{F}_0]. \tag{4}$$

$$MVA_0 = \int_0^T f(s) DIM(s) ds. \tag{5}$$

with :

- $IM(t)$ the initial margin to be posted at t calculated according to the recommandations of the regulator *International Swaps and Derivatives Association* (*ISDA*) which is seen as a *VaR* calculation over the portfolio value V_t.
- f a funding spread between the collateralized rate and the risk free rate.

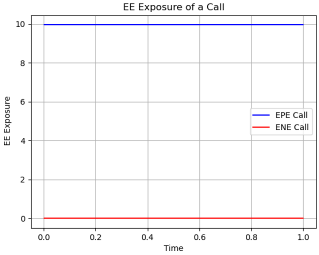
MVA_0 can therefore be approximated over a timegrid $0 = t_0 < t_1 < \ldots < t_N = T$ by :

$$MVA_0 \approx \sum_{i=0}^{N-1} f(t_i) DIM(t_i)(t_{i+1} - t_i). \tag{6}$$

Contents

An application under the *Black-Scholes* $(B-S)$ model with the following dynamics :

$$dS_t = S_t(rdt + \sigma dW_t), \quad S_0 \in \mathbb{R}_*^+.$$



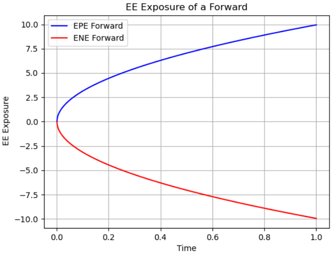


Figure: *EPE* and *ENE* profiles of a call (left) and a forward (right) in the $B-S$ model with the following parameters : $(S_0 = 100,\ K = 100,\ r = 0$ and $\sigma = 0.25)$

- For European derivatives, it can be shown that $EPE(t) = V_0, \quad \forall t \in [0, T]$.
- For forward contracts, an analytic formula can be derived in the $B-S$ model.

An application under the *Hull & White* model with the following dynamics :

$$dr_t = \kappa(\theta(t) - r_t)dt + \sigma dW_t, \quad r_0 \in \mathbb{R}.$$

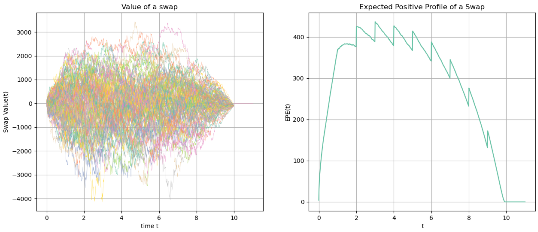


Figure: Value of a swap on a notional of $N = 10^5$ and associated EPE profile under Hull & White model with the following parameters : ($\kappa = 0.5$, $\sigma = 0.06$, $r_0 = 0.01$ with fictious initial zero-coupon bond curve given by $B(0,t) = e^{-r_0 t}$) with 50000 M-C simulations

- The sawtooth profile for a swap can be explained due to the payment dates which create this *EPE* profile.

EE Profile computation

An application to a bermudan option using the *Least Square Monte Carlo* algorithm

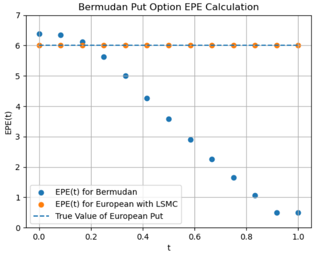


Figure: Calculation of the *EPE* profile of a bermudan put under $B-S$ model with the following parameters : ($S_0 = 100$, $K = 100$, $r = 0.04$, $\sigma = 0.2$, $T = 1$ and $N = 13$) with $N^{MC} = 100000$

- We can see that the exposure at $t_0 = 0$ of the Bermudan is higher than her european counterparty which is expected due to the potential early exercise of the product.
- We also see that the profile decreases over time compared with the European one which is also normal as during the lifetime of the product, the buyer of the option can exerce the option, the exposure becoming 0 on the residual time.

Contents

Supervised Learning Methods for *XVAs*

In the following, we will introduce 2 supervised learning methods for *XVA*s computations and we will discuss for each how they can be helpful for theses computations. For this, we will consider the following methods :

- ***Gaussian Process Regression***, a machine learning (ML) method which will help us to calculate efficiently prices surfaces for markovian processes. We will apply this ML method for EE profile and efficient CVA_0 computation to avoid the nested Monte-Carlo procedure.
- ***Deep Conditional Expectation Solver***, a deep learning method which will help us to compute MVA_0 in an efficient manner by using the conditional expectation representation as a minimization problem.

Remark

An other deep learning algorithm called ***Deep XVA Solver*** *has been studied and presented in the dissertation. It is a deep learning method based on the Deep BSDE Solver introduced in [1] and which we illustrated for high dimensionnal computation of exposure profile and associated* CVA_0.

Definition

We say that a function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is distributed by a $\mathcal{GPR}(\mu, K_{X,X})$ if $\forall n \in \mathbb{N}^*$ $\forall\ x_1, x_2, \ldots, x_n \in \mathbb{R}^d$, we have that :

$$[f(x_1), f(x_2), \ldots, f(x_n)] \sim \mathcal{N}(\mu_X, K_{X,X})$$

with $\mu \in \mathbb{R}^n$ and $K_{X,X} \in \mathcal{M}_n(\mathbb{R})$ symetric semi-definite positive matrix with general term defined by :

$$\mu_i = \mu(x_i)$$
$$K_{X,X}(i,j) = K(x_i, x_j)$$

<u>Our Aim :</u>

- Use of $\mathcal{GPR}$ to learn efficiently surface prices with training data $(X_i, Y_i)_{i \in [\![1;N]\!]}$ with N beeing really low (X representing the Markov State and Y the price) at different times over the lifetime of the product/portfolio to avoid a nested Monte-Carlo procedure.
- Combine the $\mathcal{GPR}$ methodology with a classic simple $Monte - Carlo$ to calculate CVA_0.

We present the case of a Guaranteed Minimum Maturity Benefit (*GMMB*) contract with payoff given by :

$$\mathbb{1}_{\tau > T} \max(S_T, K).$$

where :

- τ denotes the mortality date of the insured starting from 0 at age x.
- S_T is the value of the underlying stock at time T with $S_0 \in \mathbb{R}_+^*$.
- K is a minimum guarantee for the insured.

We assume the following dynamics for the underlying stock and the mortality rate λ for someone aged of x at $t = 0$:

$$\begin{aligned} dS_t &= S_t(rdt + \sigma dW_t^1), \\ d\lambda_t &= c\lambda_t dt + \xi\sqrt{\lambda_t} dW_t^2, \\ d < W^1, W^2 >_t &= \rho dt. \end{aligned} \tag{7}$$

The fair value of the *GMMB* contract is defined as $t = 0$ by :

$$P_0^{GMMB}(S_0, \lambda_0) = \mathbb{E}^Q[e^{-rT} \mathbb{1}_{\tau > T} \max(S_T, K)]. \tag{8}$$

Gaussian Process Regression

$\mathcal{GPR}$ to learn a *GMMB* price surface

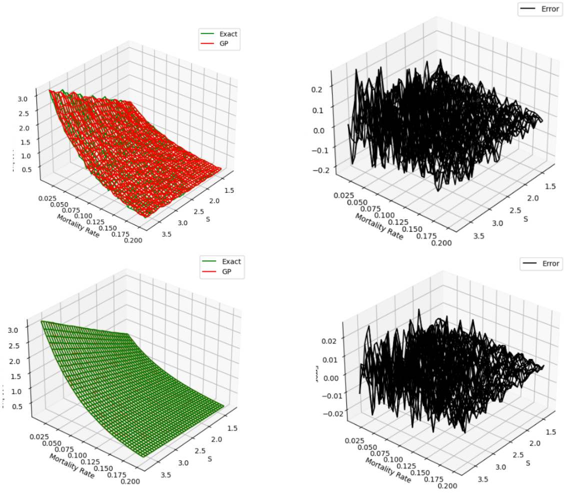


Figure: 1000 vs 100000 *MC* simulations to learn the price surface P_0^{GMMB} as a function of (λ_0, S_0) under the model (7) with the parameters : $(c = 7,50.10^{-2}$, $\xi = 5,97.10^{-4}$,$r = 0.02$, $\sigma = 0.2$, $\rho = -0.7$, $K = 1)$

Using M samples of Monte-Carlo, CVA_0 from equation (3) can be approximated as :

$$CVA_0 \approx -\frac{(1-R^C)}{M} \sum_{j=1}^{M} \sum_{i=0}^{N-1} \frac{V(t_i, X^j_{t_i})^+}{B^j_{t_i}} (G(t_{i+1}) - G(t_i)) \tag{9}$$

In a standard nested Monte-Carlo framework, the quantity $V(t_i, X_{t_i^j})^+$ should be itself calculated using a MC procedure. The goal of the $\mathcal{GPR}$ will be to learn price surfaces at different dates t_i and evaluate efficiently the quantity $V(t_i, X_{t_i^j})^+$ to save one level of the nested Monte-Carlo. Our $\mathcal{GPR} - \mathcal{MC}$ estimator can therefore be defined as :

$$C\hat{V}A_0 = \frac{(1-R^C)}{M} \sum_{j=1}^{M} \sum_{i=0}^{N-1} \frac{(\mathbb{E}[V_*|X, Y, x^* = X^j_{t_i}])^+}{B^j_{t_i}} (G(t_{i+1}) - G(t_i)) \tag{10}$$

Remark

The calculation of $\mathbb{E}[V_*|X, Y, x^* = X_{t_i^j}]$ at each time-date $(t_i)_{i \in [\![0;N]\!]}$ is performed using $\mathcal{GPR}$. Therefore, we will have to train as much $\mathcal{GPR}$ as number of timesteps in the discretization of $[0, T]$. As we combined 2 numerical methods, we can take advantage of each of them. $\mathcal{GPR}$ will provide an error on EPE profile and MC an error on CVA_0.

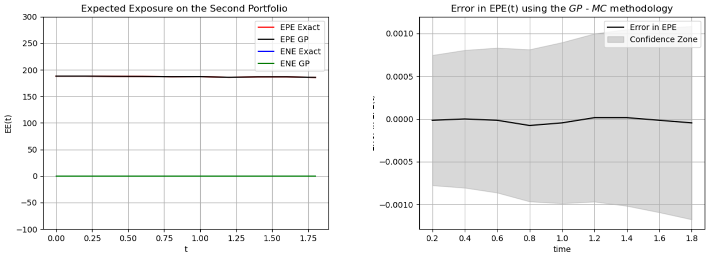


Figure: Expected Exposure Profile on a Portfolio of 10 long positions in European Call and 5 long positions in European Put using the $GP - MC$ methodology with 10 timesteps discretization for the $\mathcal{GPR}$

Table: CVA_0 using the $GP - MC$ methodology on the Second equity Portfolio with $M = 10000$ simulations

	True Value	$GP - MC$ estimation	Upper Bound	Lower Bound
CVA_0	2.2333603	2.2333624	2.2654195	2.2013054

Gaussian Process Regression

An application to an Equity Portfolio of European Options

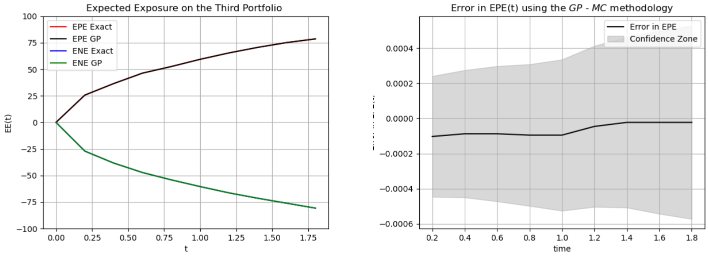
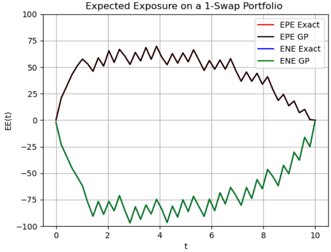


Figure: Expected Exposure Profile on a portfolio of 5 long positions in calls and 5 short positions in puts using the $GP - MC$ methodology with 10 timesteps discretization for the $\mathcal{GPR}$

Table: CVA_0 using the $GP - MC$ methodology on the Third equity Portfolio with $M = 10000$ simulations

	True Value	$GP - MC$ estimation	Upper Bound	Lower Bound
CVA_0	0.6092085	0.6092076	0.61602855	0.6023867

We give below the numerical results for a 1-swap portfolio :



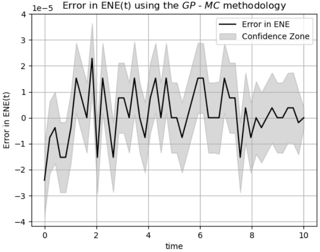


Figure: Expected Exposure Profile of a single swap using the $GP - MC$ methodology with 50 timesteps discretization for the $\mathcal{GPR}$

Table: CVA_0 using the $GP - MC$ methodology on the first swap Portfolio with $M = 10000$ simulations

	True Value	$GP - MC$ estimation	Upper Bound	Lower Bound
CVA_0	2.6152343	2.6152344	2.6974686	2.5330003

Pros :

- Require a really low number of training samples $(X_i, Y_i)_{i \in \mathbb{N}*}$ to learn the price surface as a function of the Markov state X.
- Provide a really accurate estimation of the EE profile with a confidence interval
- The error in the CVA_0 computation is almost fully based on the simple Monte-Carlo loop and not in the $\mathcal{GPR}$ algorithm.

Cons :

- The learning process can be difficult when the output labels $(Y_i)_{i \in \mathbb{N}*}$ are noisy which can lead to an inefficient learning algorithm.

The method is based on the following proposition :

Proposition

Consider 2 random variables Y and X such as $\mathbb{E}[Y|X]$ is in $L^2(X)$. Then, $\mathbb{E}[Y|X]$ is the unique solution to the following optimization problem :

$$argmin_{f \in L^2(X)} \mathbb{E}[(Y - f(X))^2]$$

As the space $L^2(X)$ leads to an infinite dimension problem, we will replace this space by the space of functions generated by neural networks parametrized by a vector θ of finite dimension denoted by f^θ. The problem can therefore be rewritten by

$$argmin_\theta \mathbb{E}[(Y - f^\theta(X))^2]$$

From the definition of the problem, we see that the appropriate loss to consider is the *MSE* loss and **then we can train the neural network by sampling** $((X_i, Y_i))_{i \in [\![1;N]\!]}$.

We illustrate the methodology with the calculation of a vector **DIM** $\in \mathbb{R}^{N+1}$ such as **DIM** $= (DIM(t_0), \ldots, DIM(t_N))$. Following (4) and defining an appropriate **IM** vector, we have **DIM**$=\mathbb{E}^{Q}[$**IM**$|\mathcal{F}_0]$. We will therefore compute **DIM** for an interest rate swap in the $G2++$ model which is parametrized by 6 parameters beeing our initial vector X. The outputs **IM** are computed using the *ISDA* methodology given in [9].

Table: Neural Network Architecture for the DIM calculation in the $G2++$ model

Number of Inputs	6
Number of Outputs	101
Number of Hidden Layers	3
Number of Neurons per Layer	256
Activation Function	$\phi(x) = x^+$ (ReLu)
Weight Initialization	Xavier/Goriot
Gradient Descent Algorithm	Adam Optimizer (learning rate $= 0.001$)

Table: Lower and Upper Bounds for market state variable in the $G2++$ model

X	κ_x	σ_x	κ_y	σ_y	ρ	r_0
min(X)	2.4%	0.5%	3%	0.5%	-0.999	-3%
max(X)	12%	2.5%	15%	2.5%	0.999	6%

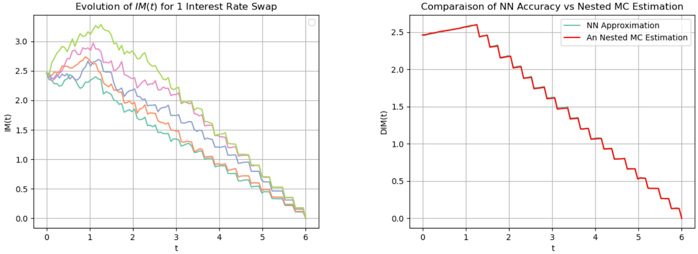


Figure: Noisy Labels for the following set of parameters ($\kappa_x = 0.10$, $\sigma_x = 0.02$, $\kappa_y = 0.12$, $\sigma_y = 0.02$, $\rho = -0.3$ and $r_0 = 0.03$) and *NN* accuracy with the nested Monte-Carlo procedure

- We can see that the neural network is fed with samples from the left figure showing that from noisy labels, he is able to reproduce a form which is really similar to the ouput given from the nested $M - C$ procedure. The *MSE* Loss is given by $6.28.10^{-5}$.
- We see a sawtooth behaviour which is expected due to the payment cashflows of the swap we considered and with the initial margin beeing 0 at terminal date which is $T = 6Y$ here.

Pros :

- The neural network doesn't require *DIM* output labels but only *IM* which helps to reduce the computational cost by computing only noisy labels.
- Once trained, the neural network provides immediate *DIM* profiles whereas the nested Monte-Carlo took more than half an hour for a single computation for a given choice of parameters.

Cons :

- The methodology based on neural networks doesn't provide an error control unlike Monte-Carlo methods which makes the final output complicated to interpret.
- The choice of the hyperparameters of the neural network are highly subjective and several choice of architectures could lead to better results in the computation of the *DIM* profile. There is still no rule to make a good choice of architecture.

Contents

Sum up of the presentation :

- Review of the mathematical framework for *XVAs*, mainly *CVA*, *FVA* and *MVA* and the computational challenged associated with the computations of theses *XVAs*.
- Computation of *EE* profile for some Bermudan Options using the **Least Square Monte Carlo method** and study of the algorithm efficiency for exposure calculation.
- Study of the **GPR-MC** methodology for the fast computation of *EPE* profile and CVA_0 computation to avoid the nested Monte-Carlo procedure showing great accuracy on the *EE* profile and on the CVA_0 computation.
- Study of the **Deep Conditional Learning** algorithm for MVA_0 computation to avoid the nested Monte-Carlo procedure showing great accuracy once the neural network is trained with immediate computations.

To go further :

- Study of the *Wrong Way Risk* impact on the *EE* profile.
- Study of the **Deep BSDE Solver** for a computation of high-dimensional *EE* profile and XVA_0 computations deriving from a *PDE* representation of *XVAs*.
- Study of a dynamic hedging strategy of the counterparty exposure based on the **Mean-Variance Minimization** quadratic hedging method with analytic formulas in a simple framework.

References

E.Weinan, J.Han, A.Jentzen, 2017, *Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations*, Springer, Commun. Math. Stat. 5, 349–380.

A.Gnoatto, A.Picarelli, C.Reisinger, 2022, *Deep XVA Solver – A Neural Network Based CounterParty Credit Risk Management Framework*, SIAM, Volume 14.

C.Ceci, K.Colanery, R.Frey, V.Köck, 2019, *Value Adjustments And Dynamic Hedging of Reinsurance Counterparty Risk*, SIAM, Volume 11.

D.Brigo, F.Vrins, 2016, *Disentangling wrong-way risk: pricing CVA via change of measures and drift adjustment*, European Journal of Operational Research, Volume 269, 1154-1164.

S.Crépey, M.F.Dixon, 2019, *Gaussian Process Regression for Derivative Portfolio modelling and Application to CVA Computations*, arXiv : 1901.11081.

J.P. Villarino, A.Leitao, 2024, *On Deep Learning for computing the Dynamic Initial Margin and Margin Value Adjustment*, arXiv : 2407.16435.

F.Longstaff, E.Schwartz, 2001, *Valuing American options by simulation: a simple least-squares approach*, Review of Financial Studies, 113-147.

J.D.B Cano, S.Crépey, E.Gobet, H-D.Nguyen, B.Saadeddine, 2022, *Learning Value-at-Risk and Expected Shortfall*, arXiv : 2209.06476.

ISDA SIMM Methodology version 2.6

D.Brigo, F.Mercurio, 2001, *Interest Rate Models – Theory and Practice*, Springer.

S.Becker, P.Cheridito, A.Jentzen, 2020, *Deep Optimal Stopping*, Journal of Machine Learning Research.

K.Andersson, C.W.Oosterlee, 2020, *A deep learning approach for computations of exposure profiles for high-dimensional Bermudan options* Applied Mathematics and Computation, Volume 408, 126332.

P. Glasserman, 2003, *Monte-Carlo methods in financial engineering*, Springer.

K.Barigou, L.Delong, 2021, *Pricing equity-linked life insurance contracts with multiple risk factors by neural networks*, Journal of Computational and Applied Mathematics, Volume 404, 113922.

K.Hornik, 1988, *Multilayer feedforward networks are universal approximators*, Neural Networks, Volume 2 Issue 5, 359-366.

Let's return to equation 2. If we no longer assume independance between the value of the exposure at default and the time at default, then we must be able to manage the term $\mathbb{E}^Q[\frac{(V_t)^+}{B_t}|\tau = t]$. To do this and based on [4] considering the process $S = (S_t)_{t\geqslant 0} = Q[\tau^C > t|\mathcal{F}_t]$ called *F-supermartingale* of Azéma, we can show that :

$$CVA_0 = -(1 - R^C)\mathbb{E}^Q[\int_0^T \frac{(V_t)^+}{B_t} dS_t]$$

If we suppose that the process S takes the following form :

$$S_t = e^{-\int_0^t \lambda_s ds}$$

with $\lambda = (\lambda_t)_{t\in[0,T]}$ a positive stochastic process and $\mathbb{F}$-adapted. Then we can write CVA_0 as follows :

$$CVA_0 = -(1 - R^C)\int_0^T \mathbb{E}^Q[\frac{(V_t)^+}{B_t}\xi_t]dG(t) \tag{11}$$

With :

- $G(t) = e^{-\int_0^t h(s)ds}$ and $\mathbb{E}^Q[S_t] = G(t)$ called calibration equation
- $\xi_t = \frac{\lambda_t S_t}{h(t)G(t)}$

Basé sur [4] et l'équation 11, on va calculer la CVA_0 en calculant sur une grille temporelle $0 = t_0 < t_1 < \ldots < t_N = T$ CVA_0 comme suit :

$$CVA_0 \approx -(1 - R^C) \sum_{i=0}^{N-1} \mathbb{E}^Q[\frac{(V_{t_i})^+}{B_{t_i}} \xi_{t_i}](G(t_{i+1}) - G(t_i)) \tag{12}$$

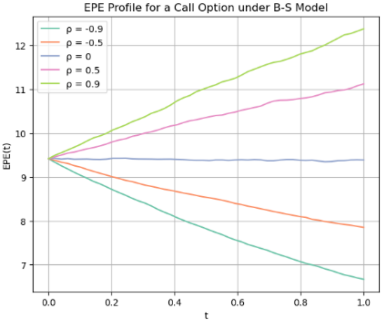
Dans les exemples portant sur des produits Equity, on supposera le modèle suivant :

$$\begin{aligned} dS_t &= S_t(rdt + \sigma dW_t^1), \quad S_0 \in \mathbb{R}_*^+ \\ d\lambda_t &= \kappa^\lambda(\theta^\lambda - \lambda_t)dt + \%sigma^\lambda \sqrt{\lambda_t} dW_t^2, \quad \lambda_0 \in \mathbb{R}_*^+ \\ d &< W_t^1, W_t^2 >_t = \rho dt \end{aligned}$$

Dans le cas des swaps de taux, on supposera un modèle Hull & White :

$$\begin{aligned} dr_t &= (\theta(t) - \kappa r_t)dt + \sigma dW_t^1, \quad r_0 \in \mathbb{R} \\ d\lambda_t &= \kappa^\lambda(\theta^\lambda - \lambda_t)dt + \sigma^\lambda \sqrt{\lambda_t} dW_t^2, \quad \lambda_0 \in \mathbb{R}_*^+ \\ d &< W_t^1, W_t^2 >_t = \rho dt \end{aligned}$$

- Le processus intensité de défault λ est supposé suivre un modèle *CIR*.
- Le paramètre ρ capture le paramètre de *Wrong Way Risk*



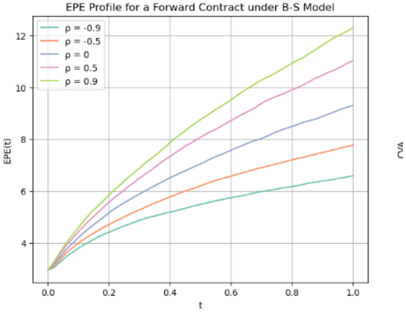


Figure: EPE Profile of Equity products under *WWR* with the following parameters ($S_0 = 100$, $K = 100$, $r = 0.03$, $\sigma = 0.2$, $\lambda_0 = \theta^\lambda = \sigma^\lambda = 0.12$ and $\kappa^\lambda = 0.35$) with $N^{MC} = 50000$

- We can see the impact of the *Wrong Way Risk* with the parameter ρ in the expected postive exposure profile as it globally increases the profile over time making the overall *CVA* higher.

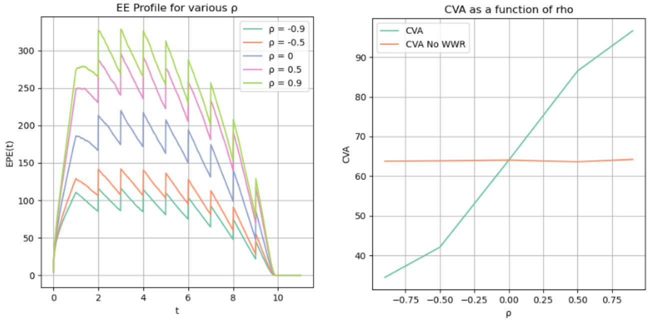


Figure: EPE Profile of an IRS under *WWR* with the following parameters ($r_0 = 0.01$, $\kappa = 0.5$, $\sigma = 0.03$, $\lambda_0 = \theta^\lambda = \sigma^\lambda = 0.12$ and $\kappa^\lambda = 0.35$ and $R^C = 0.4$) with $N^{MC} = 50000$

- We also observe that the impact of the *Wrong Way Risk* on the expected positive exposure profile of a swap is really important and the impact on the *CVA* cannot be neglected

L'idée de la méthode présentée dans [4] consiste en un changement de mesure et dans un ajustement de drift pour pouvoir calculer CVA_0 d'une manière analogue à **??**. Pour celà, on définit le processus $(C_s^{\mathcal{F},t})_{s\in[0,t]}$ de la manière suivante :

$$C_s^{\mathcal{F},t} = \mathbb{E}^Q[\frac{B_s}{B_t}\lambda_t S_t|\mathcal{F}_s] = B_s\mathbb{E}^Q[\frac{1}{B_t}\lambda_t S_t|\mathcal{F}_s] \tag{13}$$

Dès lors, en notant $(M_s^t)_{s\in[0,t]} = (\mathbb{E}^Q[\frac{1}{B_t}\lambda_t S_t|\mathcal{F}_s])_{s\in[0,t]}$, ce processus définit une $\mathbb{F}$-martingale positive et on peut alors définir une mesure de probabilité $Q^{C^{\mathcal{F},t}}$ telle que $\left.\frac{dQ^{C^{\mathcal{F},t}}}{dQ}\right|_{\mathcal{F}_s} = Z_s^t$ avec Z_s^t défini par :

$$Z_s^t = \frac{C_s^{\mathcal{F},t} B_0}{C_0^{\mathcal{F},t} B_s} = \frac{M_s^t}{M_0^t} = \frac{\mathbb{E}^Q[\frac{\lambda_t S_t}{B_t}|\mathcal{F}_s]}{\mathbb{E}^Q[\frac{\lambda_t S_t}{B_t}]} \tag{14}$$

$$\mathbb{E}^Q[\frac{(V_t)^+}{B_t}\xi_t] = \mathbb{E}^{C^{\mathcal{F},t}}[\frac{C_0^{\mathcal{F},t}}{C_t^{\mathcal{F},t}}\xi_t(V_t)^+] = \mathbb{E}^{C^{\mathcal{F},t}}[(V_t)^+]\mathbb{E}^Q[\frac{\xi_t}{B_t}] \tag{15}$$

Si on suppose l'indépendance entre le taux sans risque et le crédit, on peut écrire en notant que $\mathbb{E}^Q[\xi_t] = 1$ et $\mathbb{E}^Q[\frac{1}{B_t}] = B^r(0,t) = \mathbb{E}^Q[e^{-\int_0^t r_s ds}]$:

$$CVA = -(1-R)\int_0^T \mathbb{E}^{C^{\mathcal{F},t}}[(V_t)^+]B^r(0,t)dG(t) \tag{16}$$

On a donc une expression similaire à **??** mais on doit spécifier la dynamique de $(V_t)^+$ sous la nouvelle mesure $C^{\mathcal{F},t}$. Pour se faire, on va supposer que la dynamique de V_t sous Q est donnée par (en notant W_t^V un brownien sous Q) :

$$dV_t = \mu_t dt + \sigma_t dW_t^V \tag{17}$$

Par le théorème de Girsanov, on peut alors montrer que en notant $\tilde{W}^V$ un mouvement brownien sous $Q^{C^{\mathcal{F},t}}$ défini par :

$$\tilde{W_s^V} = W_s^V - \int_0^s d\langle W^V, (lnM^t)\rangle_u = W_s^V - \int_0^s d\langle W^V, ln(C^{\mathcal{F},t})\rangle_u$$

La dynamique de V_t sous $Q^{C^{\mathcal{F},t}}$ est alors donnée par :

$$dV_s = (\mu_s + \theta_s^t)ds + \sigma_s d\tilde{W}_s^V \tag{18}$$

Avec θ_s^t l'ajustement de drift dont l'expression est donnée par :

$$\theta_s^t ds = \sigma_s d\langle W^V, ln(C^{\mathcal{F},t})\rangle_s$$

Si on suppose des structures affines pour les processus $B^\lambda(s,t) = \mathbb{E}^Q[e^{-\int_s^t \lambda_u du}|\mathcal{F}_s]$ et $B^r(s,t) = \mathbb{E}^Q[e^{-\int_s^t r_u du}|\mathcal{F}_s]$, c'est à dire en notant :

$$B^\lambda(s,t) = \mathbb{E}^Q[e^{-\int_s^t \lambda_u du}|\mathcal{F}_s] = A^\lambda(s,t)e^{-D^\lambda(s,t)\lambda_s}$$
$$B^r(s,t) = \mathbb{E}^Q[e^{-\int_s^t r_u du}|\mathcal{F}_s] = A^r(s,t)e^{-D^r(s,t)r_s}$$

On peut en déduire la forme explicite de l'ajustement de drift θ_s^t (cf [4])

$$\theta_s^t = \rho_s^\lambda \sigma_s \sigma_s^\lambda \left(\frac{A^\lambda(s,t)\frac{\partial D^\lambda(s,t)}{\partial t}}{A^\lambda(s,t)\frac{\partial D^\lambda(s,t)}{\partial t}\lambda_s - \frac{\partial A^\lambda(s,t)}{\partial t}} - D^\lambda(s,t) \right) \tag{19}$$

A ce stade, on voit que l'ajustement de drift a toujours un comportement stochastique de part le terme λ_s. Dans [4], ils proposent 2 approximations déterministes :

- Remplacer λ_s dans 19 par sa valeur moyenne $\bar{\lambda}(s) = \mathbb{E}^Q[\lambda_s]$
- Remplacer λ_s par 19 par le taux de hasard $h(s)$

Ils justifient la connexion entre les 2 approximations par le fait que si on suppose que $Cov^Q[\lambda_t, S_t] = o(\mathbb{E}^Q[S_t])$, alors on a :

$$h(s) = -\frac{d}{ds} ln(G(s)) = -\frac{G'(s)}{G(s)} = \frac{\mathbb{E}^Q[\lambda_s S_s]}{\mathbb{E}^Q[S_s]} = \bar{\lambda}(s) + \frac{Cov^Q[\lambda_s, S_s]}{\mathbb{E}^Q[S_s]} \approx \bar{\lambda}(s) \tag{20}$$

On va supposer une dynamique de la forme suivante pour la valeur du portefeuille :

$$dV_s = (\gamma(T-s) - \frac{V_s}{T-s})ds + \nu dW_s^V$$

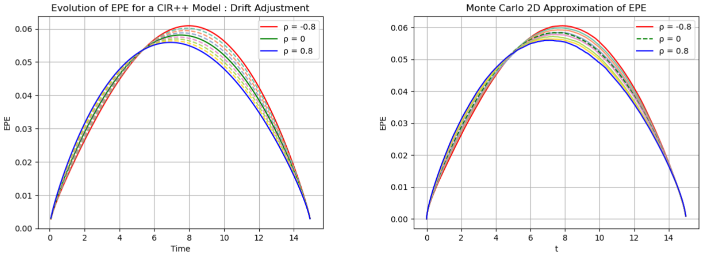


Figure: Comparaison of swap exposure profile between *2D Monte-Carlo* and the *Drift Adjustment* methods (Parameters used : $T = 15Y$, $y_0 = h = 0.30$, $\gamma = 0.001$, $\nu = 0.08$)

On va supposer une dynamique de la forme suivante pour la valeur du portefeuille :

$$dV_s = \nu dW_s^V$$

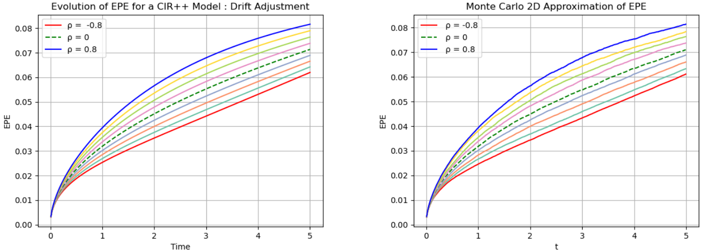


Figure: Comparaison of forward exposure profile between *2D Monte-Carlo* and the *Drift Adjustment* methods (Parameters used : $T = 10Y$, $y_0 = h = 0.15$ and $\nu = 0.08$)

Proposition

Let's considerer the following FBSDE with classic assumptions for existence and unicity of X and (Y, Z)

$$\begin{aligned} X_t &= x + \int_0^t b(s, X_s)ds + \int_0^t \sigma(s, X_s)^T dW_s^Q \\ Y_t &= g(X_T) + \int_t^T f(s, X_s, Y_s, Z_s)ds - \int_t^T Z_s^T dW_s^Q \end{aligned} \tag{21}$$

Let's consider the semilinear parabolic PDE of which $u : [0, T] \times \mathbb{R}^d -> \mathbb{R}$ is solution :

$$\begin{aligned} (\partial_t + \mathcal{L})u(t,x) + f(t,x,u(t,x),\sigma^T(t,x)D_x u(t,x)) = 0 \quad \forall (t,x) \in [0, T[\times \mathbb{R}^d \\ u(T,x) = g(x) \quad \forall x \in \mathbb{R}^d \end{aligned} \tag{22}$$

where the operation $\mathcal{L}$ is the one of the diffusion that is to say :

$$\mathcal{L}(u)(t,x) = \frac{1}{2} Tr(\sigma\sigma^T(t,x)D_x^2 u(t,x)) + \langle b(t,x), D_x u(t,x)\rangle \tag{23}$$

Processes $(Y_t = u(t, X_t))_{t\in[0,T]}$ and $(Z_t = \sigma^T(t, X_t)D_x u(t, X_t))_{t\in[0,T]}$ are solution to 21

Based on [6] and following the expression of *MVA* given by equation (6), the idea is to consider a vector **DIM** $\in \mathbb{R}^{N+1}$ such as **DIM** $= (DIM(t_0), \ldots, DIM(t_N))$. According to the equation 4, we can therefore write the vector **DIM** as the following :

$$\mathbf{DIM} = (\mathbb{E}^Q[IM(t_0)|\mathcal{F}_0], \ldots, \mathbb{E}^Q[e^{-\int_0^{t_N} r_s ds} IM(t_N)|\mathcal{F}_0])$$

Now by considering that $\mathcal{F}_0$ is characterized by a vector X of parameters we then know that we can rewrite the vector **DIM** using deterministic functions $(F_{t_i})_{i \in [0,N]}$. If we note $\mathbf{F}=(F_{t_0}, \ldots, F_{t_N})$, we then have :

$$\mathbf{DIM} = (\mathbb{E}^Q[IM(t_0)|\mathcal{F}_0], \ldots, \mathbb{E}^Q[e^{-\int_0^{t_N}} IM(t_N)|\mathcal{F}_0]) = (F_{t_0}(X), \ldots, F_{t_N}(X)) = \mathbf{F}(X) \quad (24)$$

We then now aim to approximate **F** by using the subspace of Neural Networks. Writing down $\mathbf{IM}=(IM(t_0), \ldots, e^{-\int_0^{t_N} r_s ds} IM(t_N))$, we have also the following representation for **DIM** :

$$\mathbf{DIM} = \mathbb{E}^Q[\mathbf{IM}|\mathcal{F}_0] \quad (25)$$

We can aim to learn the vector **DIM** by using Neural Networks by using samples $(X_i, (\mathbf{IM}_i))_{i \in [\![1;N]\!]}$.

The reformulation of the problem in terms of *FBSDE* is related to the following stochastic optimal control problem :

$$min_{y,(Z_t)_{t\in[0,T]}} \mathbb{E}[|g(X_T) - Y_T^{y,Z}|^2] \tag{26}$$

where :

- $X_t = x + \int_0^t b(s, X_s)ds + \int_0^t \sigma(s, X_s)^T dW_s$
- $Y_t^{y,Z} = y - \int_0^t f(s, X_s, Y_s^{y,Z}, Z_s)ds + \int_0^t Z_s dW_s$

The idea of the Deep BSDE is to approximate at each time step n the control process Z_{t_n} by using a *FFNN* by the fact that in the Markovian Setting Z_{t_n} is of the form $\phi_n(X_{t_n})$. As we also aim to learn the optimal parameter y from the stochastic control problem, we will set it y approximated by ξ as a trainable parameter of the neural network which will be optimised during the learning procedure.

Let's denote by θ a vector associated to a specified architecture of a neural network. For sake of simplicity, we will assume that each neural network at each time step has the same structure. Therefore, we can introduce a family of neural networks $(\phi_n^\theta)_{n\in[0,N]}$ valued from $\mathbb{R}^d$ to $\mathbb{R}^d$ such as by defining $Z_{t_n}^\theta = \phi_n^\theta(X_{t_n})$, we can define the following discretisation scheme :

$$Y_{t_{n+1}}^{\xi,\theta} = Y_{t_n}^{\xi,\theta} - h(t_n, X_{t_n}, Y_{t_n}^{\xi,\theta}, Z_{t_n}^\theta)\Delta t + (Z_{t_n}^\theta)^\top (W_{t_{n+1}} - W_{t_n}), \quad Y_0^{\xi,\theta} = \xi \tag{27}$$

Therefore, the global minimization problem becomes :

$$min_{\xi,\theta}\mathbb{E}[(g(X_T) - Y_T^{\xi,\theta})^2] \tag{28}$$

Supposing that $Q(\tau^A > t|\mathcal{F}_t) = e^{-\int_0^t \lambda_s^A ds}$ and $Q(\tau^C > t|\mathcal{F}_t) = e^{-\int_0^t \lambda_s^C ds}$, *CVA* and *FVA* can be rewritten as follows :

$$CVA_t = (1 - R^C)\mathbb{E}^Q[\int_t^T e^{-\int_t^s (r_u + \lambda_u^C + \lambda_u^A)du} (V_s)^+ \lambda_s^C ds|\mathcal{F}_t] FVA_t \quad = \mathbb{E}^Q[\int_t^T e^{-\int_t^s (r_u + \lambda_u^C + \lambda_u^A)du} ($$

According to Feymann-Kac formula, *CVA* can be rewritten as the solution to the following PDE : $\forall (t, x) \in [0, T[\times \Omega$

$$\partial_t \phi^{CVA}(t,x) + \mathcal{L}\phi^{CVA}(t,x) - (r_t + \lambda_t^C + \lambda_t^A)\phi^{CVA}(t,x) + (1 - R^C)(V_t)^+ \lambda_t^C = 0$$
$$\phi^{CVA}(T, .) = 0$$

We then can use the *Deep BSDE Solver* by setting :

- $f(t, X_t, Y_t, Z_t) = (1 - R^C)\lambda_t^C (V_t)^+ - (r_t + \lambda_t^C + \lambda_t^A) Y_t$
- $g(X_T) = 0$

We consider the case of a Basket call option on $d = 100$ assets with payoff given by

$$g(S_T^1, \ldots, S_T^d) = (\textstyle\sum_{i=1}^d S_T^i - dK)^+$$

$$g(S_T^1, \ldots, S_T^d) = (\sum_{i=1}^d S_T^i - dK)^+$$

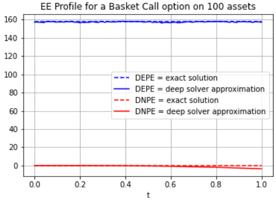


Figure: Exposure Calculation of a Basket Option on $d = 100$ assets under B-S with the following parameters : ($S_0 = 100$,$K = 100$,$r = 0.01$,$\sigma = 0.25$, $\rho = 0$)

Table: CVA_0 computation using *Deep BSDE Solver* with the following parameters : ($R^C = 0.3$, $\lambda^A = 0.01$, $\lambda^C = 0.1$, $s_B = 0.04$ and $s_L = 0$)

Based on [3], we will aim to find an investement strategy in a *CDS* to hedge the counterparty exposure. The payment streams are defined as:

$$C_t = R^{CDS} H_t - \xi \int_0^t (1 - H_u) du \tag{29}$$

With :

- The first term refers to the payment at default
- The second term refers to the premium payment with a supposed continuous spread $\xi > 0$

The present value of the future payments of the *CDS* is given by :

$$D_t = \mathbb{E}^Q [\int_t^T e^{-\int_t^u r_s ds} dC_u | \mathcal{G}_t] \tag{30}$$

From that, we can define the discounted gain process $CDS = (CDS_t)_{t \in [0,T]}$ as :

$$CDS_t = e^{-\int_0^t r_s ds} D_t + \int_0^t e^{-\int_0^u r_s ds} dC_u$$

We now define a self-financing portfolio strategy if the discounted value of the portfolio $\tilde{V}_t^{\xi} = e^{-\int_0^t r_s ds} V_t^{\xi}$ with $\xi = (\xi^0, \xi^1)$ defines respectively the position in cash and in the CDS, can be rewritten as :

$$\tilde{V}_t^{\xi} = V_0^{\xi} + \int_0^t \xi_s^1 d(CDS)_s, \quad t \in [0, T] \tag{31}$$

The objective is now to minimize the following quantity which will be defined as the tracking error e_T at terminal date T

$$min_{V_0^{\xi}, \xi^1 = (\xi_t^1)_{t \in [0,T]}} \mathbb{E}^Q[(e^{-\int_0^{\tau} r_s ds}(1-R)(V_\tau)^+ \mathbb{1}_{\tau \leqslant T} - (V_0^{\xi} + \int_0^T \xi_t^1 dCDS_t))^2] \tag{32}$$

To find the solution to the problem, the proof is based on the *Föllmer-Schweizer* decomposition

For the numerical illustration, we will suppose the following modeling :

$$dS_t = S_t(rdt + \sigma dW_t^1), \quad S_0 \in \mathbb{R}_*^+$$
$$d\lambda_t = b(\lambda_t)dt + \sigma(\lambda_t)(\rho dW_t^1 + \sqrt{1-\rho^2}dW_t^2), \quad \lambda_0 \in \mathbb{R}_*^+$$

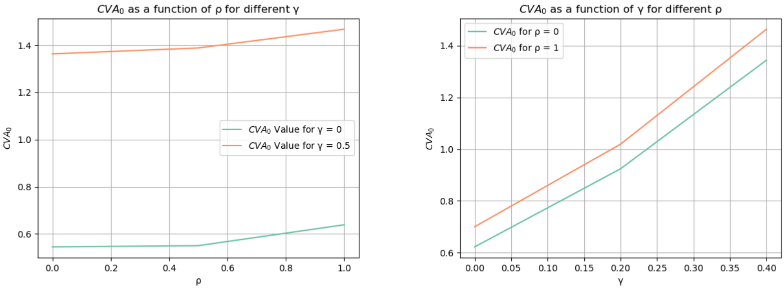


Figure: Comparaison of 3 Hedging Strategies in order to hedge the *CCR* on a Call Option with the following parameters : $\xi = 0.2$, $\lambda = 0.2$, $r = 0$, $\sigma = 0.4$

For the numerical illustration, we will suppose the following modeling :

$$
\begin{aligned}
dS_t &= S_t(rdt + \sigma dW_t^1), \quad S_0 \in \mathbb{R}_*^+ \\
d\lambda_t &= b(\lambda_t)dt + \sigma(\lambda_t)(\rho dW_t^1 + \sqrt{1-\rho^2}dW_t^2), \quad \lambda_0 \in \mathbb{R}_*^+
\end{aligned}
$$

Proposition

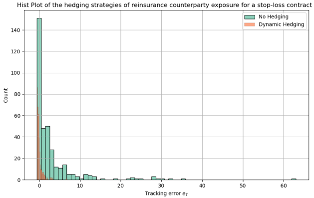
It can be shown that the optimal strategy ξ^1 is such that when $\sigma(\lambda_t) = b(\lambda_t) = 0$:

$$
\begin{aligned}
V_0^{\xi} &= CVA_0 \\
\xi_t^1 &= \frac{d\langle M^{CL}, CDS\rangle_t}{d\langle CDS\rangle_t} = (1-H_t)\frac{(1-R^C)(V(t,S_t)^+ - f^{CVA}(t,S_t,\lambda_0))}{(R^{CDS} - g(t,\lambda_0))} \qquad (33)
\end{aligned}
$$

with by noting $\lambda_0 = \lambda$:

$$
g(t,\lambda) = \mathbb{E}^Q[\int_t^T e^{-\int_t^u (r+\lambda)ds}(R^{CDS}\lambda - \xi)du|\mathcal{F}_t] = R^{CDS}(1-e^{-\lambda(T-t)}) + \frac{\xi}{\lambda}(e^{-\lambda(T-t)} - 1)
$$

$$
f^{CVA}(t,S_t,\lambda_)) = \mathbb{E}^Q[\int_t^T e^{-\int_t^u (r+\lambda)ds}(V_u)^+\lambda du|\mathcal{F}_t] \qquad (34)
$$



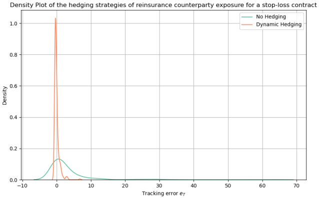
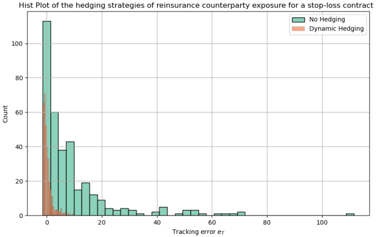


Figure: Dynamic Hedging of the counterparty exposure on a Stop Loss Contract with frequent sinisters but not costly

Table: Norm 2 of e_T in case of a Stop Loss Contract

	No Hedging	Dynamic Hedging
$\mathbb{E}[(e_T)^2]$	50.84	0.52



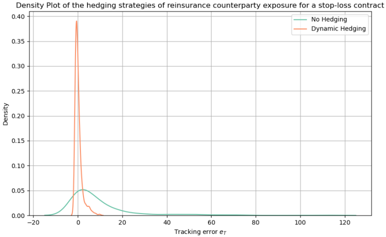


Figure: Comparaison of 2 Hedging Strategies in order to hedge the *CCR* on a Stop Loss Contract with less sinisters but more costly

Table: Norm 2 of e_T in case of a Forward Contract

	No Hedging	Dynamic Hedging
$\mathbb{E}[(e_T)^2]$	283.65	2.90

- The financial industry also seeks to calculate in addition to the average exposure profile *EE* the exposure profile at a given percentile α defined for a level $\alpha \in [0.1]$ given by:

$$PFE_t^{\alpha} = inf\{y : P((V_t)^+ \leqslant y) \leqslant \alpha\}$$

  This complementary measure echoes the definition of *Value-at-Risk* and recently supervised learning methods have emerged for the calculation of these risk measures based on a dual representation of the *Value -at-Risk* and *Expected Shortfall* as minimization problems as introduced in [8] from Cano, Crépey, Gobet, Nguyen and Saadeddine .
- Use of supervised learning algorithms based on neural networks for the valuation of high-dimensional Bermudan options as introduced in [11] from Becker, Cheridito and Jentzen where the optimal exercise time is learned on a sample of data.
- Use of deep neural networks for the valuation of life insurance options indexed to stocks, in particular as introduced in the article [14] *Pricing equity-linked life insurance contracts with multiple risk factors by neural networks* from Barigou and Delong.