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

1. Introduction
2. Mathematical framework for XVAs
   - The CVA Pricing framework
   - An MVA Pricing framework
3. Numerical results on *EE* profile computation
   - EE Profile of equity products
   - EE Profile of an interest rate swap
   - EE Profile of a bermudan option
4. Review of Machine and Deep Learning Algorithms for XVA computations
   - *Gaussian Process Regression* and application to CVA_0 computation
   - *Deep Conditional Expectation Solver* and application to MVA_0 computation
5. Conclusion

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

1. Introduction
2. Mathematical framework for XVAs
   - The CVA Pricing framework
   - An MVA Pricing framework
3. Numerical results on *EE* profile computation
   - EE Profile of equity products
   - EE Profile of an interest rate swap
   - EE Profile of a bermudan option
4. Review of Machine and Deep Learning Algorithms for XVA computations
   - *Gaussian Process Regression* and application to CVA_0 computation
   - *Deep Conditional Expectation Solver* and application to MVA_0 computation
5. Conclusion

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

1 Introduction

2 Mathematical framework for XVAs
- The CVA Pricing framework
- An MVA Pricing framework

3 Numerical results on *EE* profile computation
- EE Profile of equity products
- EE Profile of an interest rate swap
- EE Profile of a bermudan option

4 Review of Machine and Deep Learning Algorithms for XVA computations
- *Gaussian Process Regression* and application to CVA_0 computation
- *Deep Conditional Expectation Solver* and application to MVA_0 computation

5 Conclusion

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

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

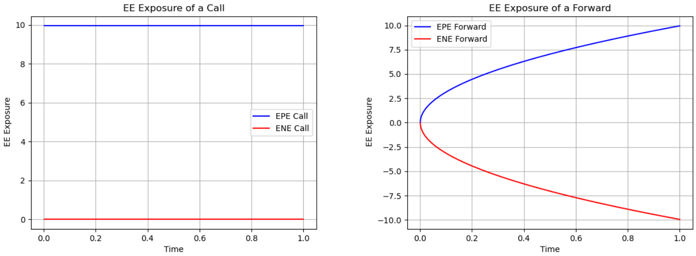

Figure: *EPE* and *ENE* profiles of a call (left) and a forward (right) in the $B - S$ model with the following parameters : $(S_0 = 100,\ K = 100,\ r = 0$ and $\sigma = 0.25)$

- 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 :

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

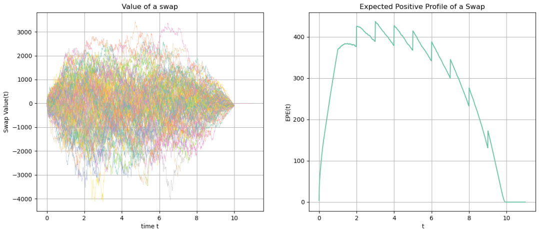

Figure: Value of a swap on a notional of $N = 10^5$ and associated EPE profile under Hull & White model with the following parameters : ($\kappa = 0.5$, $\sigma = 0.06$, $r_0 = 0.01$ with fictious initial zero-coupon bond curve given by $B(0,t) = e^{-r_0 t}$) with 50000 M-C simulations

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

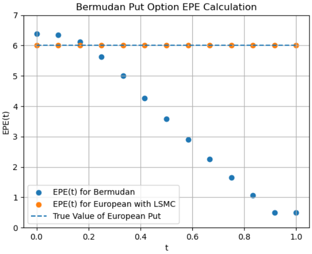

Figure: Calculation of the *EPE* profile of a bermudan put under $B-S$ model with the following parameters : ($S_0 = 100$, $K = 100$, $r = 0.04$, $\sigma = 0.2$, $T = 1$ and $N = 13$) with $N^{MC} = 100000$

- 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

1. Introduction
2. Mathematical framework for XVAs
   - The CVA Pricing framework
   - An MVA Pricing framework
3. Numerical results on *EE* profile computation
   - EE Profile of equity products
   - EE Profile of an interest rate swap
   - EE Profile of a bermudan option
4. Review of Machine and Deep Learning Algorithms for XVA computations
   - *Gaussian Process Regression* and application to CVA_0 computation
   - *Deep Conditional Expectation Solver* and application to MVA_0 computation
5. Conclusion

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 $f : \mathbb{R}^d \to \mathbb{R}$ 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 :

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

Our Aim :

- 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 :

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

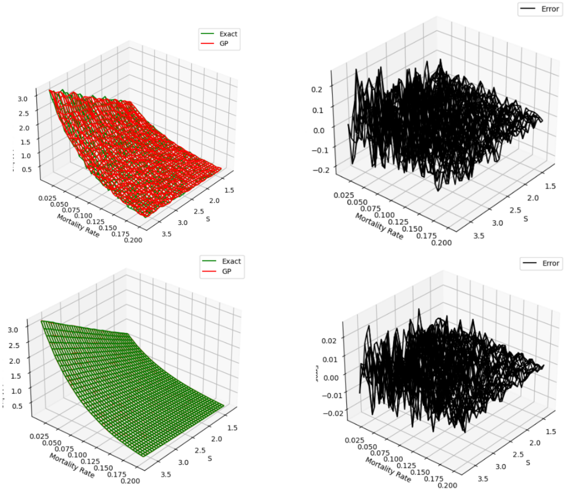

Figure: 1000 vs 100000 *MC* simulations to learn the price surface P_0^{GMMB} as a function of (λ_0, S_0) under the model (7) with the parameters : $(c = 7,50.10^{-2},\ \xi = 5,97.10^{-4}, r = 0.02,\ \sigma = 0.2,\ \rho = -0.7,\ K = 1)$

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 :

$$\hat{CVA}_0 = \frac{(1-R^C)}{M}\sum_{j=1}^{M}\sum_{i=0}^{N-1}\frac{(\mathbb{E}[V_*|X, Y, x^* = X^j_{t_i}])^+}{B^j_{t_i}}(G(t_{i+1}) - G(t_i)) \tag{10}$$

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*.*

Gaussian Process Regression

An application to an Equity Portfolio of European Options

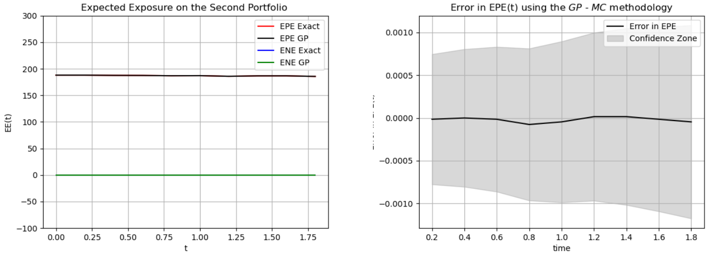

Figure: Expected Exposure Profile on a Portfolio of 10 long positions in European Call and 5 long positions in European Put using the $GP - MC$ methodology with 10 timesteps discretization for the $\mathcal{GPR}$

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

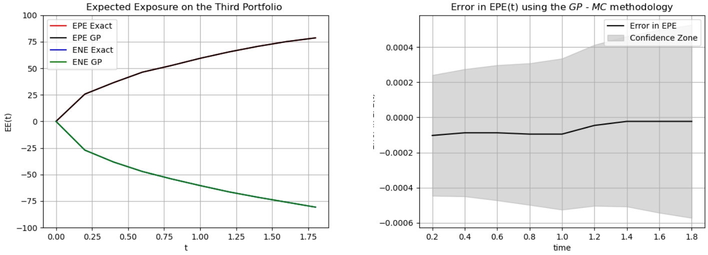

Figure: Expected Exposure Profile on a portfolio of 5 long positions in calls and 5 short positions in puts using the $GP - MC$ methodology with 10 timesteps discretization for the $\mathcal{GPR}$

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

Gaussian Process Regression

An application to a Swap Portfolio

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

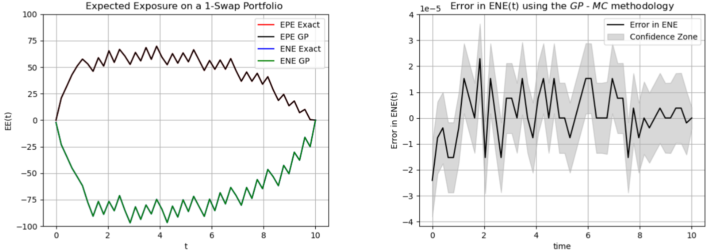

Figure: Expected Exposure Profile of a single swap using the $GP - MC$ methodology with 50 timesteps discretization for the $\mathcal{GPR}$

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** $((X_i, Y_i))_{i \in [\![1;N]\!]}$.

Deep Conditional Expectation Solver

Neural Network settings

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%

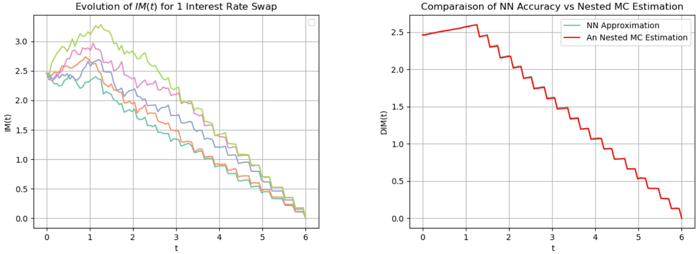

Figure: Noisy Labels for the following set of parameters ($\kappa_x = 0.10$, $\sigma_x = 0.02$, $\kappa_y = 0.12$, $\sigma_y = 0.02$, $\rho = -0.3$ and $r_0 = 0.03$) and *NN* accuracy with the nested Monte-Carlo procedure

- 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

1. Introduction

2. Mathematical framework for XVAs
   - The CVA Pricing framework
   - An MVA Pricing framework

3. Numerical results on *EE* profile computation
   - EE Profile of equity products
   - EE Profile of an interest rate swap
   - EE Profile of a bermudan option

4. Review of Machine and Deep Learning Algorithms for XVA computations
   - *Gaussian Process Regression* and application to CVA_0 computation
   - *Deep Conditional Expectation Solver* and application to MVA_0 computation

5. Conclusion

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.

References

- 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.