Credit Derivatives Project

Bilateral counterparty risk valuation with stochastic dynamical models and application to CDS

Damiano Brigo and Agostino Capponi, Sept 23, 2008

Students:
Jeremy Marck
Johan Macq
Antoine Lefebvre

Professor:
Caroline Hillairet

April 22, 2020

École nationale
de la statistique
et de l'administration
économique
université
PARIS-SACLAY

Contents

Introduction

As more and more credit derivatives are traded over the counter, the question of whether a counterparty may default is of utmost importance. In a classical credit default swap (CDS) contract, three entities may be subject to default : the reference credit entity (who might not be able to reimburse its loan to the CDS buyer), the CDS buyer (who might not be able to pay its coupons to the CDS seller) and the CDS seller (who might not be able to offer protection to the CDS buyer in case the reference credit entity defaults). Therefore, a fair pricing of such contract should include all three sources of risk.

In their paper *Bilateral counterparty risk valuation with stochastic dynamical models and application to Credit Default Swaps*, Brigo and Capponi propose a method for evaluating bilateral counterparty risk. The bilateral risk credit value adjustment (BRCVA) of a contract is defined as the difference between the price of this contract assuming no counterparty default and the price of this contract assuming default might happen on both sides of the contract. The authors provide mathematical proof that the BRCVA of a contract, from the point of view of the investor, can be rewritten as the difference between two options with specific exercise conditions :

- a zero-strike call option on the contract, with maturity equal to the default time of the counterparty, and that can only be exercised if the counterpaty defaults before the investor and during the period of the contract ;
- a zero-strike put option on the contract, with maturity equal to the default time of the investor, and that can only be exercised if the investor defaults before the counterparty and during the period of the contract.

This result is very important since it allows one to interpret the sign of the BRCVA. In case the counterparty always defaults before the investor, the put option can never be exercised and the resulting BRCVA is positive. Therefore, a positive BRCVA from the point of view of the investor means the investor supports the risk of the contract, whereas a negative one means the counterparty bears the risk. In reality, the sign of the BRCVA also depends on the default time of the reference credit entity, and all three default times might be correlated. An investor who buys protection to a counterparty would not want the default time of its counterparty to be positively correlated to the default time of the reference credit entity. This kind of situation is called wrong-way risk and can only be captured in complex mathematical frameworks.

Brigo and Capponi model the default intensities through Cox-Ingersoll-Ross shifted integrated processes and allow for correlation between default times via gaussian copulas. After providing a few mathematical results, the authors present an algorithm for the Monte Carlo computation of the BRCVA. They then compute the BRCVA of the investor when he is on one or the other side of a CDS contract, and for different market scenarios corresponding to different default correlations and different entity risk parameters.

Part B of the present document consists of a linear summary of the article. Part C details the different steps we went through to build our Monte Carlo BRCVA algorithm, based on the article of Brigo and Capponi. The reader must have been emailed the associated python code, otherwise he/she should warn us.

Linear summary of the article

1 Arbitrage-free valuation of bilateral couterparty risk

First, we introduce the following notations for the different stakeholders:

$$\text{investor} \longrightarrow \text{name "0"}$$
$$\text{counterparty} \longrightarrow \text{name "2"}$$
$$\text{reference credit} \longrightarrow \text{name "1"}$$

with τ_0, τ_2 and τ_1 the default times associated. We define the probability space $(\Omega, G, G_t, \mathbb{Q})$ with G_t the filtration of the flow of information of the whole market, $\mathbb{Q}$ the risk neutral probability. We define F_t as the sub-filtration representing all the information but the default event ($F_t \in G_t$), and if τ is a stopping time, we denote F_τ the stopped filtration associated ($F_\tau = \sigma(F_t\ U\ \{t \leq \tau\}, t \geq 0)$). Finally, T is the maturity of the payoff.

At each time t, there can be a default event of either the counterparty, the investor or both. We have six exhaustive and exclusive events:

$$A = \{\tau_0 \leq \tau_2 \leq T\},\ B = \{\tau_0 \leq T \leq \tau_2\},\ C = \{\tau_2 \leq \tau_0 \leq T\},\ D = \{\tau_2 \leq T \leq \tau_0\},\ E = \{T \leq \tau_0 \leq \tau_2\}, F = \{T \leq \tau_2 \leq \tau_0\}$$

We denote $\Pi^D(t,T)$ our discounted payoff valuated at t and $C_{ashflows}(u,s)$ the net cash flows without default between u and s discounted at u. All payoffs are seen from the investor. With the net present value $NPV(\tau_i) = \mathbb{E}_{\tau_i}(C_{ashflows}(\tau_i, T))$, $i = 1, 2$, and $D(t,T)$ the price at t of a zero coupon with maturity T, we have:

$$\begin{aligned}\Pi^D(t,T) = 1_{EUF} C_{ashflows}(t,T) &+ 1_{CUD}[\ C_{ashflows}(t,\tau_2) + D(t,\tau_2)(\ R_{ec,2}(NPV(\tau_2))^+ - (-NPV(\tau_2))^+\ )] \\ &+ 1_{AUB}[\ C_{ashflows}(t,\tau_0) + D(t,\tau_0)(\ (NPV(\tau_0))^+ - R_{ec,0}(-NPV(\tau_0))^+\ )]\end{aligned}$$

where $R_{ec,i}$ is the recovery rate of i after default.
This equation means that if there if no default we have the normal cash flow (first term). If the counterparty defaults before the investor (1_{CUD}), then between t and τ_2 we have the normal cash flow, in τ_2 the investor will receive the recovery of the NPV (if positive) and will pay the full NPV (if negative), all being discounted back to t (second term). If the investor defaults before the counterparty (third term), the same apply as in the second term: if the NPV is positive the investor will receive the full value, if the NPV is negative it will pay only the recovery.
We can simplify this formula and obtain the general bilateral counterparty risk pricing formula:

$$\begin{aligned}\mathbb{E}_t[\Pi^D(t,T)] = \mathbb{E}_t[\Pi(t,T)] &+ \mathbb{E}_t[LGD_0 * 1_{AUB} * D(t,\tau_0) * (-NPV(\tau_0))^+\ ] \\ &- \mathbb{E}_t[LGD_2 * 1_{CUD} * D(t,\tau_2) * (NPV(\tau_2))^+\ ] \\ &= \mathbb{E}_t[\Pi(t,T)] - \text{BR-CVA}(t,T,LGD_{0,1,2})\end{aligned}$$

where $\mathbb{E}_t[\Pi(t,T)]$ is the same payoff without default risk, $LGD = 1 - Rec$ is the loss given default and BR-CVA is the bilateral risk credit valuation adjustment. Indeed, we have:

$$\Pi(t,T) = 1_{AUB} C_{ashflows}(t,T) + 1_{CUD} C_{ashflows}(t,T) + 1_{EUF} C_{ashflows}(t,T)$$

and

$$1_{AUB}C_{ashflows}(t,T) = 1_{AUB}(C_{ashflows}(t,\tau_0) + D(t,\tau_0)C_{ashflows}(\tau_0,T)$$

We obtain the desired result by the linearity of the expectation. We can see that the BR-CVA can be either positive or negative, depending on whether the counterparty is more or less subject to default than the investor. The BR-CVA have an interesting property of symmetry: if the counterparty 2 were to calculate the risk of its position towards the investor 1, it would find the exact opposite ($-1 *$ BR-CVA$(t,T,LGD_{0,1,2})$ of what the investor found. This is obviously not the case when we consider a default-free investor (as in most models). The statement of Citigroup is explained by a degradation of the credit quality of "0", increasing the term in LGD_0.

2 Application to Credit Default Swaps

This section is dedicated to the evaluation of the BR-CVA (bilateral risk credit value adjustment) in credit default swaps contracts. As a reminder, a credit default swap (CDS) is a financial swap agreement that the seller of the CDS will compensate the buyer in the event of a debt default (by the debtor) or other credit event. That is, the seller of the CDS insures the buyer against some reference asset defaulting. The buyer of the CDS makes a series of payments (the CDS "fee" or "spread") to the seller and, in exchange, may expect to receive a payoff if the asset defaults.
This section will consist of the presentation of the CDS payoff, the default correlation, the presentation of the stochastic intensity process and the bilateral risk credit valuation adjustment for receiver CDS.

2.1 CDS Payoff

Interest rates are assumed to be deterministic. The main implication of this assumption is the independence between τ_1, $D(0,T)$ (zero coupon with maturity T) and deterministic recovery rates. It should be noted that the results hold true as far as we use stochastic interest rates independent of default times. The receiver CDS valuation, in the case of a CDS selling protection LGD_1 at time 0 for default of the reference entity between times T_a and T_b in exchange of a periodic premium is provided by:

$$\begin{aligned}\mathrm{CDS}_{a,b}(0,S_1,\mathrm{LGD}_1) =& S_1\left[-\int_{T_a}^{T_b} D(0,t)(t-T_{\gamma(t)-1})d\mathbb{Q}(\tau_1>t) + \sum_{i=a+1}^{b}\alpha_i D(0,T_i)\mathbb{Q}(\tau_1>T_i)\right] \\ &+ \mathrm{LGD}_1\left[\int_{T_a}^{T_b} D(0,t)d\mathbb{Q}(\tau_1>t)\right]\end{aligned}$$

Interpretation: the first term corresponds to the accrued coupon, the second term corresponds to the payment of the coupons and the last term corresponds to the payment at the time of default.
Note that $\gamma(t)$ is the first payment period T_j following time t. The net present value of the CDS contract evaluated at an intermediate date $T_a < T_j < T_b$ is defined as the residual net present value of a receiver CDS between T_a and T_b. One can write: $\mathrm{NPV}(T_j,T_b) := \mathrm{CDS}_{a,b}(T_j,S,LGD_1)$. Hence, one can write the CDS evaluation at an intermediate time T_j by conditioning on the information available at time T_j. This gives:

$$\begin{aligned}&\mathrm{CDS}_{a,b}(T_j,S_1,LGD_1) = \mathbf{1}_{\tau_1>T_j}\overline{\mathrm{CDS}}_{a,b}(T_j,S_1,LGD_1) \\ &:= \mathbf{1}_{\tau_1>T_j}\left\{S_1\left[-\int_{\max(T_a,T_j)}^{T_b} D(T_j,t)(t-T_{\gamma(t)-1})d\mathbb{Q}(\tau_1>t/\mathcal{G}_{T_j}) + \sum_{i=\max(a,b)+1}^{b}\alpha_i D(T_j,T_i)\mathbb{Q}\left(\tau_1>T_i/\mathcal{G}_{T_j}\right)\right]\right\} \\ &+ \mathbf{1}_{\tau_1>T_j}\left\{\mathrm{LGD}_1\left[\int_{\max(T_a,T_j)}^{T_b} D(T_j,t)d\mathbb{Q}(\tau_1>t/\mathcal{G}_{T_j})\right]\right\}\end{aligned}$$

2.2 Default correlation

The objective is to give an expression to the intensities of defaults. These intensities are assumed to be stochastic and correlated through a Gaussian copula. In particular, the default correlation between the three names is defined through a dependence structure on the exponential random variables characterizing the default times of the three names.

Notations and some considerations about the default intensity: recall that $i = 0, 1, 2$ refers to the three entities (0 is the investor, 1 the reference credit and 2 the counterparty). $\lambda_i(t)$ will denote the default intensity for i at time t. The associated cumulated intensity will be written $\Lambda_i(t) = \int_0^t \lambda_i ds$.
It is important to understand that default times are random variables in our context. Many extensions exist regarding the intensity of these default times (the intensity characterizes the rate of arrivals of new events i.e. the rate of arrivals of new default times). In standard models, defaults intensity λ is deterministic (constant or depending on time). However, some models allow this intensity to be stochastic: it is the case when considering λ is drawn by a CIR process (our context). Other processes exist: Hawkes process for instance (presented below).

Note that λ constant would involve a standard homogeneous Poisson process. Hence, the associated counting process $N_t = \sum_{j\geq 1} \mathbf{1}_{T_j<t}$ would be Poisson distributed as follows: $N_t \sim P(\lambda t)$. In such a case, interarrival times $S_i = T_i - T_{i-1}$ would be exponentially distributed with parameter λ. Under the hypothesis of a Poisson counting process, increments of the counting process are independent and stationnary.
Taking $\lambda(\cdot)$ (a deterministic function) would imply a standard non-homogeneous poisson process whose associated counting process N_t would follow a Poisson distribution $N_t - N_s \sim P(\Lambda(t) - \Lambda(s))$. Increments would be independent but not stationary anymore.
One could also consider a stochastic default intensity. Hawkes processes constitute a popular alternative. The process is self-exciting and drawn by the following equation (exponentiel Kernel):

$$\lambda(t) = \lambda_0(t) + \int_0^t v(t-s)dN_s = \lambda_0(t) + \sum_{0<t_i<t} v(t-t_i) = \lambda_0(t) + \sum_{0<t_i<t} \sum_{j=1}^{P} \alpha_j e^{-\beta_j} \mathbf{1}_{\mathbb{R}_+}(t-t_i).$$

Where v expresses the positive influence of the past events t_i on the current value of the intensity process and λ_0 is the baseline intensity. .
The following graphs represent simulations of default times with a standard Poisson process and with a Hawkes process (with the same horizon time and identical baseline intensity). Our simulators are available through our **'processes_simulators.py'** file.

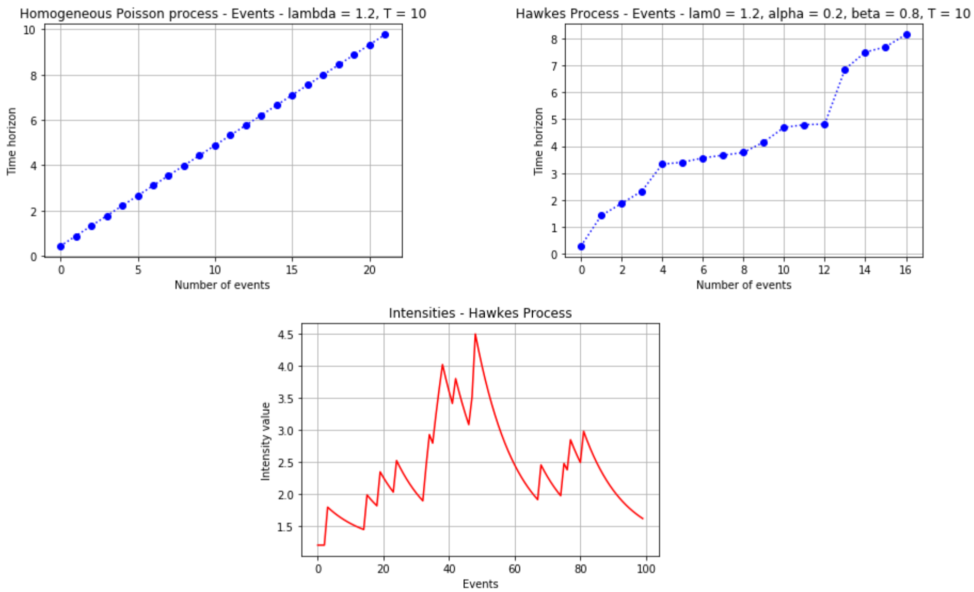

Figure 1: Comparing Poisson and Hawkes default times simulators. Intensity plot is performed with $T = 100$.

Modeling default correlation: in our case, we will consider independent default intensities and assume λ_i to be strictly positive almost everywhere, implying Λ_i invertible. We place ourselves in a Cox process setting. The procedure for simulating default times is the following:

1. Simulate an uniform triplet through a trivariate Gaussian copula $C_{\mathbf{R}}(u_0, u_1, u_2) = \mathbb{Q}(U_0 < u_0, U_1 < u_1, U_2 < u_2)$. This can be easily done by using `scipy.stats` in Python. Such a simulator is available in our Python Notebook. $\mathbf{R} = [r_{i,j}]_{i,j=0,1,2}$ is the correlation matrix parametrizing the Gaussian copula. It is important to keep in mind that a trivariate Gaussian copula implies bivariate gaussian copulas. Once again, such a task is performed in our Python Notebook.

2. Using $U_i = 1 - \exp(\xi_i)$, recuperate the associated exponential random variables (with parameter 1) by $\xi_i = \ln(1 - U_i)$ (an application of the standard inverse method).

3. Get the random default times with $\tau_i = \Lambda_i^{-1}(\xi_i) \quad i = 0, 1, 2$ where τ_i is the random default time associated to the name i.

Let us now specify the Cox dynamics associated to a Cox process.

2.3 CIR stochastic intensity model

Through the paper, the following stochastic intensity is assumed:

$$\lambda_j(t) = y_j(t) + \psi_j(t, \beta_{\mathbf{j}}) \qquad t \geq 0 \quad j = 0, 1, 2$$

Note that $\psi(\cdot)$ is a deterministic function depending on the parameter vector β, considered to be integrable on closed intervals. Note also that y_j is drawn with a standard CIR process, that is:

$$dy_j(t) = \kappa_j(\mu_j - y_j(t))dt + \nu_j\sqrt{y_j(t)}dZ_j(t) + J_{M,j}dM_j(t) \qquad j = 0, 1, 2$$

Z_t is a standard brownian motion, J's are iid positive jumps sizes that are exponentially distributed with mean χ_j and M_j are poisson processes with intensity m_j measuring the interarrival times of jumps with intensity λ_j. By the way, $\beta_j = (\kappa_j, \mu_j, \nu_j, y_j(0), \chi_j, m_j)$ with each vector component being a positive deterministic constant. Integrated quantities are thus given by the following relationships:

$$\Lambda_j(t) = \int_0^t \lambda_j(s)ds \qquad Y_j(t) = \int_0^t y_j(s)ds \qquad \Psi_j(t, \beta_j) = \int_0^t \psi_j(s, \beta_j)ds$$

Note that the analysis made through the article is done under the hypothesis that $m_j = 0$ (no-jumps model).

2.4 Bilateral risk credit valuation adjustment for receiver CDS

The BR-CVA at time t for a receiver CDS contract (protection seller) runing from time T_a to time T_b with premium S is given by:

$$\begin{aligned}
&\text{BR-CVA-CDS}_{a,b}(t, S, \text{LGD}_{0,1,2}) = \mathbb{E}_t\left\{\text{LGD}_2.\mathbf{1}_{C\cup D}.D(t,\tau_2).\,[\text{NPV}(\tau_2)]^+\right\} - \mathbb{E}_t\left\{\text{LGD}_0.\mathbf{1}_{A\cup B}.D(t,\tau_0).\,[-\text{NPV}(\tau_0)]^+\right\}\\
&= \text{LGD}_2\mathbb{E}_t\left\{\mathbf{1}_{C\cup D}.D(t,\tau_2).\,[\text{CDS}_{a,b}(\tau_2, S, \text{LGD}_1)]^+\right\} - \text{LGD}_0\mathbb{E}_t\left\{\mathbf{1}_{A\cup B}.D(t,\tau_0).\,[\text{CDS}_{a,b}(\tau_0, S, \text{LGD}_1)]^+\right\}\\
&= \text{LGD}_2\mathbb{E}_t\left\{\mathbf{1}_{C\cup D}.D(t,\tau_2).\,\left[\mathbf{1}_{\tau_1>\tau_2}\overline{CDS}_{a,b}(\tau_2, S, \text{LGD}_1)\right]^+\right\}\\
&- \text{LGD}_0\mathbb{E}_t\left\{\mathbf{1}_{A\cup B}.D(t,\tau_0).\,\left[-\mathbf{1}_{\tau_1>\tau_0}\overline{CDS}_{a,b}(\tau_0, S, \text{LGD}_1)\right]^+\right\}
\end{aligned}$$

Now the only terms which need to be known in order to compute the previous calculus are:

$$\mathbf{1}_{C\cup D}\mathbf{1}_{\tau_1>\tau_2}\mathbb{Q}(\tau_1 > t/\mathcal{G}_{\tau_2}) \qquad \mathbf{1}_{A\cup B}\mathbf{1}_{\tau_1>\tau_0}\mathbb{Q}(\tau_1 > t/\mathcal{G}_{\tau_0})$$

Section 4 purposes a method to calculate these quantities.

3 Monte-Carlo Evaluation of the BR-CVA adjustment

This section is dedicated to the Monte-Carlo simulation of our problem. The first section compares some discretizations methods in order to simulate a CIR process. The second section provides a method to calculate $\mathbf{1}_{C\cup D}\mathbf{1}_{\tau_1>\tau_2}\mathbb{Q}(\tau_1 > t/\mathcal{G}_{\tau_2})$. Then, we provide a discussion about the way we implemented our algorithm.

3.1 Several ways of simulating a CIR process

Chi-square distribution of a CIR process: it is a well-know result that a path of a CIR process can be drawn with a Chi-square distribution, up to a scale factor. Consider $u < t$, then a discretization of the CIR process writes down:

$$y(t) = \frac{\nu^2(1-e^{-\kappa(t-u)})}{4\kappa}\chi'_d\left(\frac{4\kappa e^{-\kappa(t-u)}}{v^2(1-e^{-\kappa(t-u)})}y(u)\right) \qquad d = \frac{4\kappa\mu}{\nu^2}$$

Note that $\chi'_d(v)$ denotes a non-central chi-square random variable with u degrees of freedom and non centrality parameter v. The unique thing we need to know in order to simulate the process is $y(0)$. The simulation has to be done over a grid $t_0 = 0 < t_1 < ... < t_n = T$.

Euler scheme discretization: our goal is to simulate the following CIR process:

$$dy(t) = \kappa(\mu - y(t))dt + \nu\sqrt{y(t)}dZ(t)$$

Where Z is a standard Brownian motion. Over a grid $t_0 = 0 < t_1 < ... < t_n = T$ with M time steps, we have $\Delta_t = T/M$. The associated Euler schema is:

$$y(t+\Delta_t) = y(t) + \kappa(\mu - y(t))\Delta_t + \nu\sqrt{y(t)}\underbrace{\left[Z(t+\Delta_t) - Z(t)\right]}_{\mathcal{N}(0,\Delta t)}$$

Milstein scheme discretization: although the regularity conditions that ensure a better convergence for the Milstein scheme are not satisfied (the diffusion coefficient in non Lipschitz), one can try to apply it:

$$y(t+\Delta_t) = y(t) + \kappa(\mu - y(t))\Delta_t + \nu\left[Z(t+\Delta_t) - Z(t)\right] + \frac{1}{4}\nu^2\left[\left(Z(t+\Delta_t) - Z(t)\right)^2 - \Delta t\right]$$

3.2 Comparing CIR simulators

This section is dedicated to perform two comparisons. The first one consists in comparing 100 simulations realized using the Chi-square distributions for different parameters values. These values are the ones provided through the article and correspond to different profiles of risk.

Comparing 100 simulations for Chi-square Discretization with different risk levels.

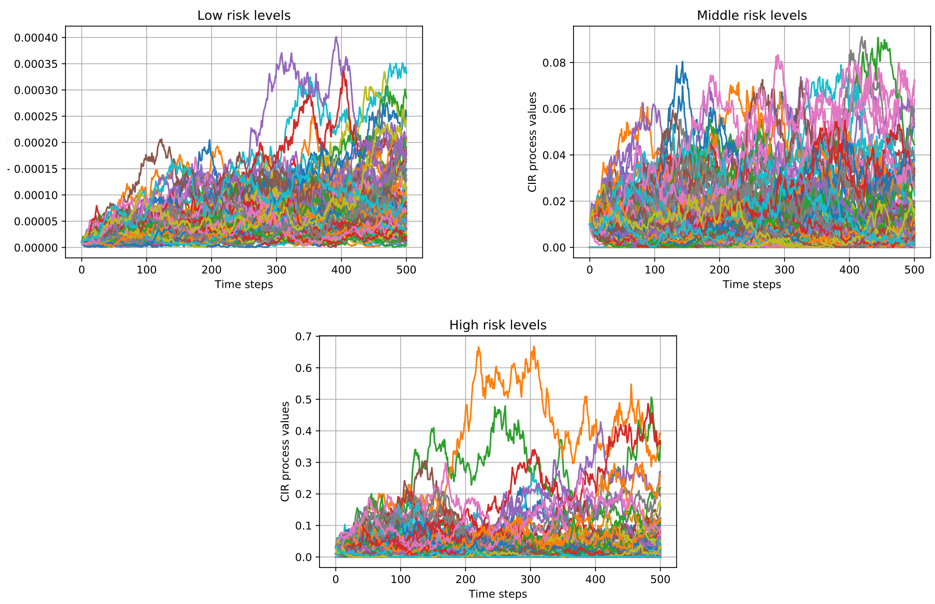

Figure 2: Comparing $n = 100$ simulations for different levels of risk with Chi-square discretization.

Comparing 100 simulations for Euler-Scheme with different risk levels.

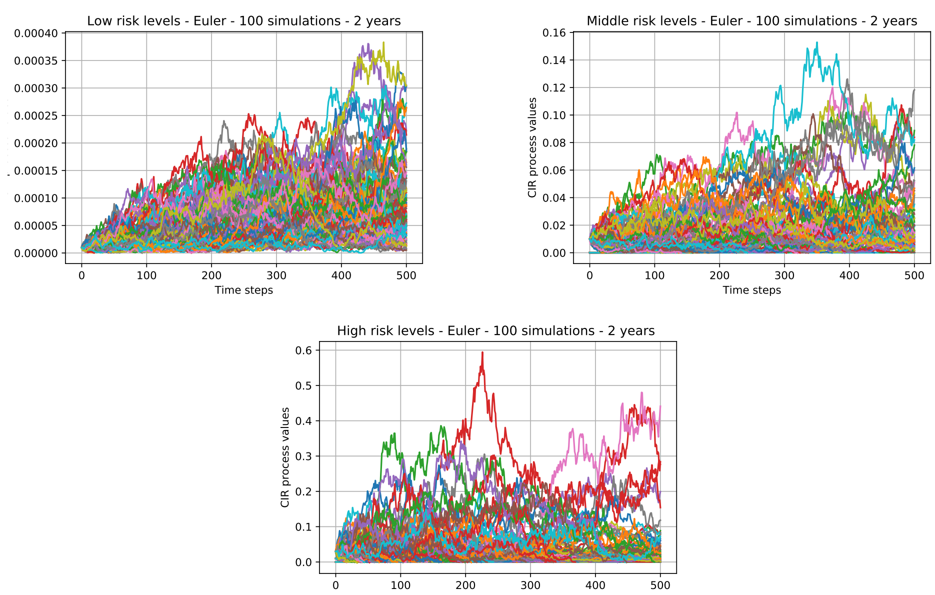

Figure 3: Comparing $n = 100$ simulations for different levels of risk with Euler-Scheme discretization.

Comparing 100 simulations for Milstein-Scheme with different risk levels.

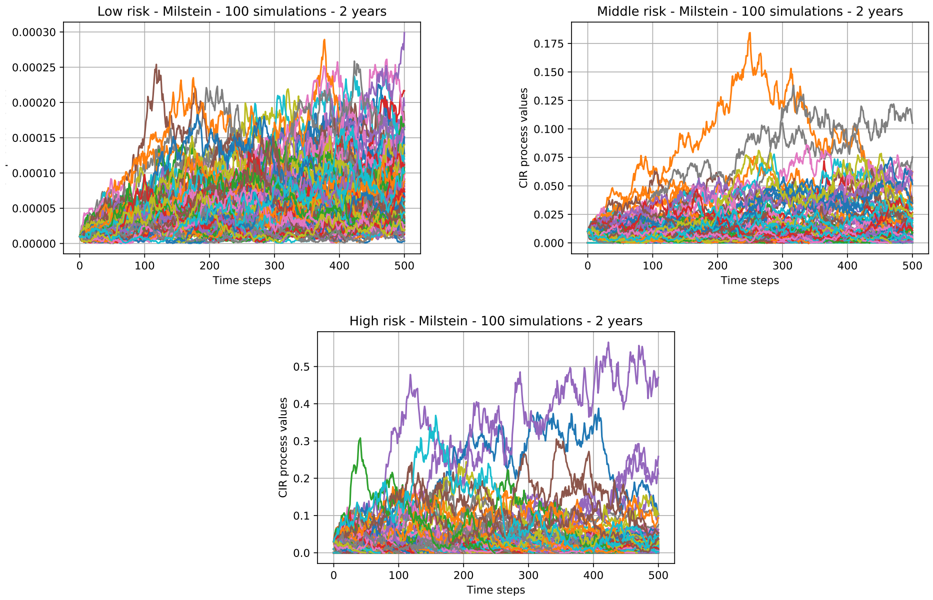

Figure 4: Comparing $n = 100$ simulations for different levels of risk with Milstein-Scheme discretization.

Let us now compare the different levels of risks when simulating paths with the three available discretization schemes.

Comparing 100 simulations low-risk levels and for different discretization schemes.

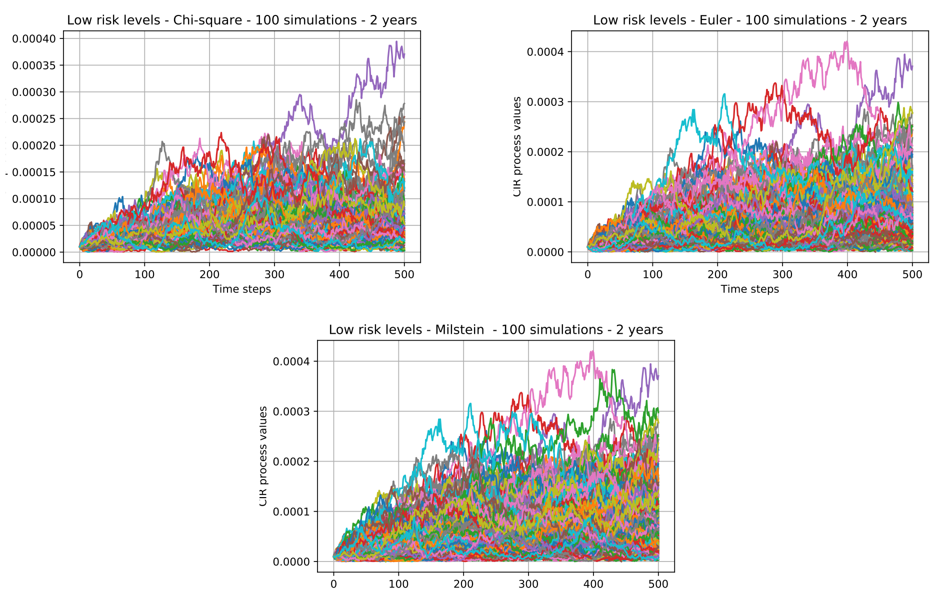

Figure 5: Comparing 100 simulations low-risk levels and for different discretization schemes.

Comparing 100 simulations Middle-risk levels and for different discretization schemes.

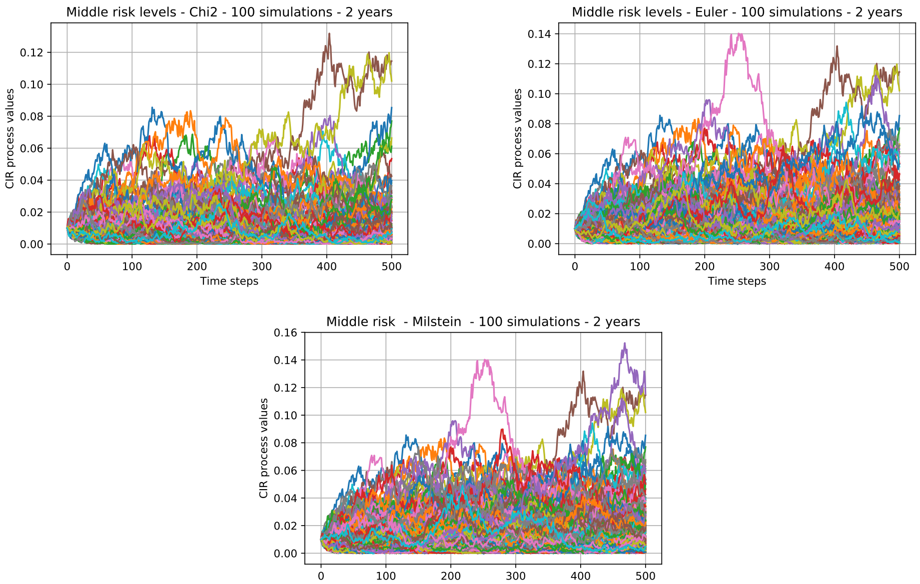

Figure 6: Comparing 100 simulations Middle-risk levels and for different discretization schemes.

Comparing 100 simulations High-risk levels and for different discretization schemes.

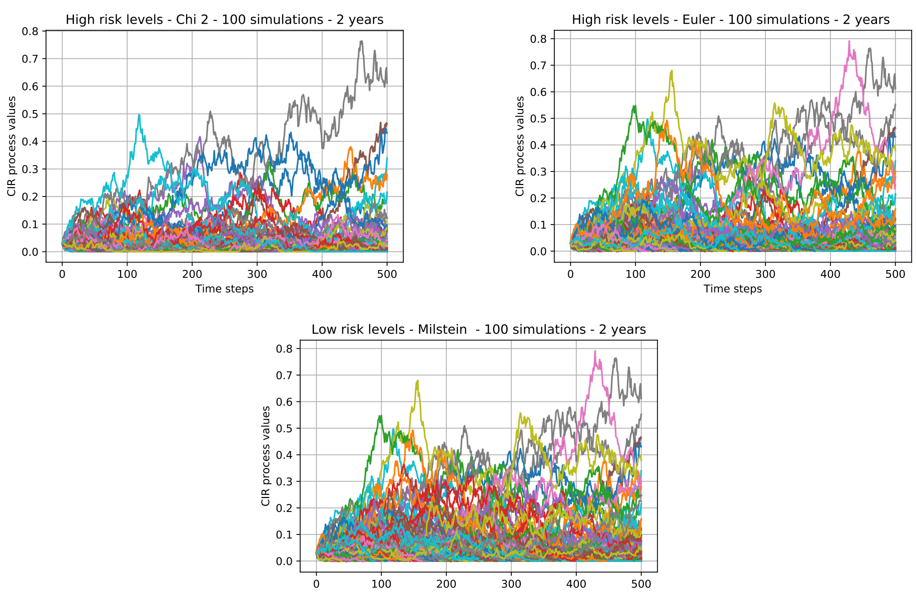

Figure 7: Comparing 100 simulations High-risk levels and for different discretization schemes.

The Python code performing these simulations is available through our **'comparisons.py'** file.

3.3 Calculation of Survival Probability

The section is dedicated to the formulas used for the computation of the survival probability given in 3.4:

$$\mathbf{1}_{C\cup D}\mathbf{1}_{\tau_1>\tau_2}\mathbb{Q}(\tau_1 > t/\mathcal{G}_{\tau_2}) \qquad \mathbf{1}_{A\cup B}\mathbf{1}_{\tau_1>\tau_0}\mathbb{Q}(\tau_1 > t/\mathcal{G}_{\tau_0})$$

We define:

$$\mathbf{1}_{C\cup D}\mathbf{1}_{\tau_1>\tau_2}\mathbb{Q}(\tau_1 > t/\mathcal{G}_{\tau_2}) = \mathbf{1}_{\tau_2\leq T}\mathbf{1}_{\tau_2\leq\tau_0}\left(\mathbf{1}_{\bar{A}} + \mathbf{1}_{\tau_2<t}\mathbf{1}_{\tau_1\geq\tau_2}\int_{\bar{U}_{1,2}}^{1} F_{\Lambda_1(t)-\Lambda_1(\tau_2)}(-\log(1-u_1)-\Lambda_1(\tau_2))dC_{1/0,2}(u1,U2)\right)$$

With $\bar{A} = \{t < \tau_2 < \tau_1\}$ and:

$$C_{1/0,2}(u_1,U_2) = \frac{\left.\frac{\partial C_{1,2}(u_1,u_2)}{\partial u_2}\right|_{u_2=U_2} - \left.\frac{\partial C\left(\bar{U}_{0,2},u_1,u_2\right)}{\partial u_2}\right|_{u_2=U_2} - \left.\frac{\partial C_{1,2}\left(\bar{U}_{1,2},u_2\right)}{\partial u_2}\right|_{u_2=U_2} + \left.\frac{\partial C\left(\bar{U}_{0,2},\bar{U}_{1,2},u_2\right)}{\partial u_2}\right|_{u_2=U_2}}{1 - \left.\frac{\partial C_{0,2}\left(\bar{U}_{0,2},u_2\right)}{\partial u_2}\right|_{u_2=U_2} - \left.\frac{\partial C_{1,2}\left(\bar{U}_{1,2},u_2\right)}{\partial u_2}\right|_{u_2=U_2} + \left.\frac{\partial C\left(\bar{U}_{0,2},\bar{U}_{1,2},u_2\right)}{\partial u_2}\right|_{u_2=U_2}}$$

Similarly, one obtains:

$$\mathbf{1}_{A\cup B}\mathbf{1}_{\tau_1>\tau_0}\mathbb{Q}(\tau_1 > t/\mathcal{G}_{\tau_0}) = \mathbf{1}_{\tau_0\leq T}\mathbf{1}_{\tau_0\leq\tau_2}\left(\mathbf{1}_{\bar{B}} + \mathbf{1}_{\tau_0<t}\mathbf{1}_{\tau_1\geq\tau_0}\int_{\bar{U}_{1,0}}^{1} F_{\Lambda_1(t)-\Lambda_1(\tau_0)}(-\log(1-u_1)-\Lambda_1(\tau_0))dC_{1/2,0}(u1,U_0)\right)$$

With $\bar{B} = \{t < \tau_0 < \tau_1\}$ and:

$$C_{1/2,0}(u_1,U_0) = \frac{\left.\frac{\partial C_{0,1}(u_0,u_1)}{\partial u_0}\right|_{u_0=U_0} - \left.\frac{\partial C\left(,u_0,u_1,\bar{U}_{2,0}\right)}{\partial u_0}\right|_{u_0=U_0} - \left.\frac{\partial C_{0,1}\left(u_0,\bar{U}_{1,0}\right)}{\partial u_0}\right|_{u_0=U_0} + \left.\frac{\partial C\left(u_0,\bar{U}_{1,0},\bar{U}_{2,0}\right)}{\partial u_0}\right|_{u_0=U_0}}{1 - \left.\frac{\partial C_{0,2}\left(u_0,\bar{U}_{2,0}\right)}{\partial u_0}\right|_{u_0=U_0} - \left.\frac{\partial C_{0,1}\left(u_0,\bar{U}_{1,0}\right)}{\partial u_0}\right|_{u_0=U_0} + \left.\frac{\partial C\left(u_0,\bar{U}_{1,0},\bar{U}_{2,0}\right)}{\partial u_0}\right|_{u_0=U_0}}$$

3.4 Calculation of the deterministic function Ψ

This part is dedicated to determine the deterministic function $\Psi_i(t,\beta_i)$. Its role is to calibrate our process to the market data. The CDS spreads quoted on the market (here theoretical) are given by the following table, where t is the maturity in year:

Maturity	Low risk	Middle risk	High risk
1y	0	92	234
2y	0	104	244
3y	0	112	248
4y	1	117	250
5y	1	120	251
6y	1	122	252
7y	1	124	253
8y	1	125	253
9y	1	126	254
10y	1	127	254

Table 1: CDS spreads for different risk levels

The survival probabilities associated with a CIR intensity process are given by the following relation:

$$\mathbb{Q}(\tau_i > t) := \mathbb{E}\left[e^{-Y_i(t)}\right] = P^{CIR}(0,t,\beta_i)$$

Note that $P^{CIR}(0,t,\beta_i)$ is the price at time 0 of a zero coupon bond maturing at time t under a stochastic interest rate dynamics given by the CIR process where $\beta_i = (y_i(0),\kappa_i,\mu_i,\nu_i)$ being the vector of CIR parameters, $i = 0,1,2$. Parameters used for running the simulation are given in the article. One recovers the integrated shift $\Psi(t,\beta)$ which makes the model survival probabilities consistent with the market survival probabilities (presented in the article). In mathematical terms, it means that $\forall i$:

$$\mathbb{Q}(\tau_i > t)_{model} := \mathbb{E}\left[e^{-\Lambda_i(t)}\right] = \mathbb{Q}(\tau_i > t)_{market}$$

Construction of the market survival probability: for name i, it is bootstrapped from the market CDS quotes presented in Table 1. Such bootstrap procedure is realized assuming a piecewise linear hazard rate function.
From the two equations above, we deduce:

$$\Psi_i(t, \beta_i) = \log\left(\frac{\mathbb{E}\left[e^{-Yi(t)}\right]}{\mathbb{Q}(\tau_i > t)_{market}}\right) = \log\left(\frac{P^{CIR}(0, t, \beta_i)}{\mathbb{Q}(\tau_i > t)_{market}}\right)$$

Performing such a calculus requires computing both the numerator and the denominator (section 4.3. provides the way to follow). However, as far as the numerator is in fact the price of a zero-coupon in a CIR model, let us provide some additional details about CIR processes (useful for simulating/calibrating this type of processes).

Reminder about CIR processes. The dynamics for a process $(y)_{t\geq 0}$ is given by:

$$dy_t = a(b - y_t)dt + \sigma\sqrt{y_t}dW_t$$

Where b is the long run value of the process $(y)_t$ and a is the speed of adjustment. The parameter σ characterizes the volatility of the process, as far as W_t is the standard Brownian motion. Note that $y_t \neq 0$ if $2ab \geq \sigma^2$.

Properties of CIR processes. This family of processes exhibits mean reversion, level dependent volatility ($\sigma\sqrt{y_t}$). For a given y_0, one obtains:

$$\mathbb{E}[y_t/y_0] = r_0 e^{-at} + b(1 - e^{-at}) \qquad V(y_t/y_0) = r_0\frac{\sigma^2}{2}\left(e^{-at} - e^{-2at}\right) + \frac{b\sigma^2}{2a}\left(1 - e^{-at}\right)^2$$

Calibration: two main approaches for calibrating such a model.
(1) Ordinary least squares. From $dy_t = a(b - y_t)dt + \sigma\sqrt{y_t}dW_t$ one has the following discretization (where Δ_t is the discretization step):

$$y_{t+\Delta_t} - y_t = a(b - y_t)\Delta_t + \sigma\sqrt{y_t\Delta_t}\varepsilon_t \iff \frac{y_{t+\Delta_t} - y_t}{\sqrt{y_t}} = \frac{ab\Delta_t}{\sqrt{y_t}} - a\Delta_t\sqrt{y_t} + \sigma\sqrt{\Delta_t}\varepsilon_t$$

Where ε_t is i.i.d $\mathcal{N}(0, 1)$.
(2) Maximum likelihood estimation.

Zero Coupon Bond price: let us suppose that there exists a function $B(t, y, T)$ depending both on time and y (y represents interest rates) with the following terminal condition $B(T, r, T) = 1$:

$$\frac{1}{2}\sigma^2\partial_{r^2}B(t, r, T) + a(b - r)\partial_r B(t, r, T) - rB(t, r, T) + \partial_t B(t, r, T) = 0$$

Then the price of a zero-coupon in a CIR model is:

$$B(t, T) = A(T - t)\exp\left[-r_t C(T - t)\right]$$

where $C(\theta) = \dfrac{2(e^{p\theta} - 1)}{(p + a)(e^{p\theta} - 1) + 2p}$ and $A(\theta) = \left(\dfrac{2pe^{\theta\frac{(p+a)}{2}}}{(p + a)(e^{p\theta} - 1) + 2p}\right)^{\frac{2ab}{\sigma^2}}$ and $p = \sqrt{a^2 + 2\sigma^2}$

Five-step implementation of the BRCVA Monte Carlo algorithm

4 Five-step implementation of the BRCVA Monte Carlo algorithm

In this section we present the five main steps we went through for implementing the pseudo-algorithms proposed by the authors to calculate the BR-CVA of a CDS contract. Then, we will compare our results with the ones of the authors and give a few tracks to be developed for further work.
The pseudo-algorithms of the paper is given in appendix. A Python notebook with detailed comments on the code was also sent to the reader.

4.1 Computation of the Cumulative Distribution Function of an integrated CIR process

We aim here to calculate the CDF of the equations (4.4) and (4.6) page 12 of the article, which will become the $p_k = \text{CDF}(x_k)$ in the third algorithm on page 14.
To do so, we imply the CDF from the characteristic function using a numerical approach proposed by Teng, Ehrhardt and Gunther in their paper *Numerical evaluation of complex logarithms in the Cox–Ingersoll–Ross model* (2012). One can find bellow the illustration of CDF for different risk profiles:

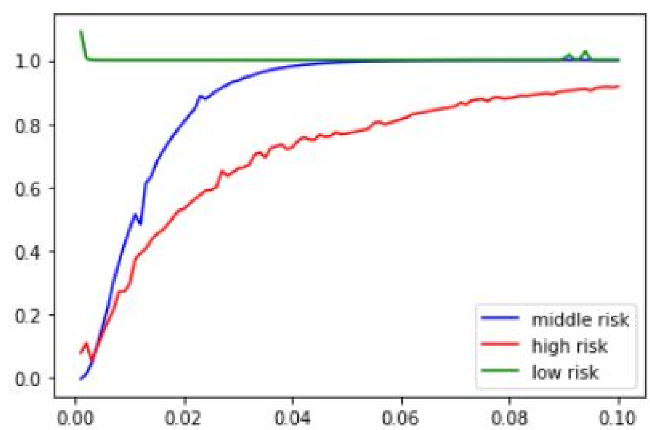

Figure 8: Comparing three CDF for high, medium and low risk levels.

Some interpretation:

- For the low-risk profile (green curve), the distribution function is 1 very quickly. It can be said that the integrated CIR of the low-risk profile is "almost certainly small". Therefore, its default instant, inversely proportional to the intensity (and the integrated CIR), will be very large! The relative position of the curves is therefore intuitively correct.
- The small oscillations are due to the oscillations of the characteristic functions and can be lowered with Gauss-Lobatto adaptive numerical integration (as mentioned by Teng, Ehrhardt and Gunther).

4.2 Copulas and partial copulas

To run our algorithm, we needed to compute $C_{1|0,2}$ and $C_{1|2,0}$ defined in part 4.3. Doing so required computing the derivative of a copula. So, after having created functions these calculate bivariate and trivariate copulas, we just computed numerically their derivatives with an increment. We used the so-called `scipy` Python library in order to run the computation of our Gaussian copulas. This package is very tractable as far as it allows to draw some Gaussian picks and then transform these picks in uniform ones.

4.3 Adjustment of the CIR intensity deterministic component with market data

In order to adjust the intensity to the reality of the market, we need to calibrate its deterministic component $\psi_j(t, \beta_j)$. We recall the formula we use:

$$\Psi_i(t, \beta_i) = \log\left(\frac{\mathbb{E}[e^{-Y_i(t)}]}{\mathbb{Q}(\tau_i > t)_{market}}\right) = \log\left(\frac{P^{CIR}(0, t, \beta_i)}{\mathbb{Q}(\tau_i > t)_{market}}\right)$$

Two terms are to be computed:

1. $\mathbb{Q}(\tau_i > t) = \exp\left(\frac{-s(i,t)}{\text{LGD}(i)}\right)$ (see chapter 24, 9th edition, *Options, futures and other derivatives*, John C. Hull), which is the survival probability. Where $s(i, t)$ is the spread quoted on the market for entity i, on which we perform a linear interpolation for maturity we do not observe. LGD is the loss given default.

2. $P^{CIR}(0, t, \beta_i)$ is the price of a zero coupon bond at time 0 with maturity t, in the CIR model. It is computed using the closed formula given by the model:
   $P^{CIR}(0, t, \beta_i) = A\exp(-By_i(0))$ where:
   $A = \left(\frac{2he^{\frac{(\kappa_i+h)(T-t)}{2}}}{2h + (\kappa_i + h)(e^{(T-t)h} - 1)}\right)^{\frac{2\kappa_i\mu_i}{\nu_i^2}}$ and $B = \frac{2(e^{(T-t)h}-1)}{2h+(\kappa_i+h)(e^{(T-t)h}-1)}$ and $h = \sqrt{\kappa_i^2 + 2\nu_i^2}$

Some additional considerations about CIR processes.

4.4 Stopping times simulation

The stopping times are computed using the inverse of the integrated stochastic intensity and are correlated via a trivariate gaussian copula. We simulate them in three steps:

1. Pick three numbers U_i following uniform law and correlated with a copula (we use a Cholesky decomposition of the covariance matrix).

2. We apply the formula (3.5) given in the article to obtain $\xi_i = -\log(1 - U_i)$, following an exponential law.

3. compute τ_i by generating and integrating a shifted CIR (with the shiting function Ψ above), returning τ_i when the integrated process is equal to ξ_i (as prescribed in *On Cox Processes and Credit Risky Securities*, Lando, 1998). We use the Chi-square simulation to compute the CIR process because of its simplicity of implementation.

We have the following formula for the default times $\tau_i = \Lambda_i^{-1}(\xi_i) \implies \Lambda_i(\tau_i) = \xi_i$. The following code works on the following principle: since it is difficult to find an explicit function (form) for $\Lambda_i^{-1}(\cdot)$, we compute $\Lambda_i(\cdot)$ for each time step t and when $\Lambda_i(t) = \xi_i$ then $t = \tau_i$. The next figure is an illustration of the trajectory of a CIR process until stopping time is reached.

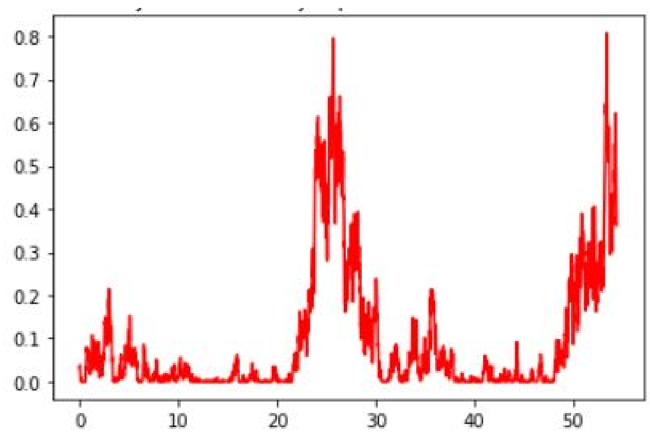

Figure 9: CIR trajectory until stopping time is reached.

4.5 Monte Carlo simulation of the BR-CVA

Using the previous implemented functions to compute required elements, we are able to implement the three algorithms presented pages 13 and 14 of the article.
The results below are computed with the following parameters:

	Investor	Reference Underlying	Counterparty
y_0	0.03	0.05	0.03
κ	0.5	0.3	0.5
μ	0.05	0.08	0.05
ν	0.5	0.8	0.5
LGD	0.7	0.8	0.7

Table 2: Parameters of the investor, the reference underlying and the counterparty

So we have an investor and a counterparty that have the same risk level: high. The reference underlying is considered as very high risk. We chose these parameters for computation reasons, for the algorithms not being too slow. The following graphs represent the computation of our algorithm and in particular the evolution of the BR-CVA when running our simulation.

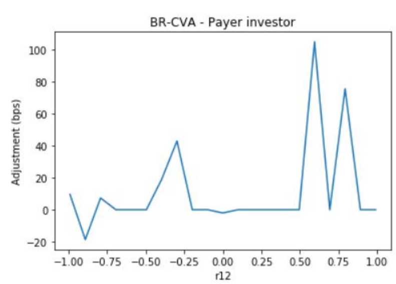

Figure 10: Bilateral risk credit valuation adjustment for a payer CDS.

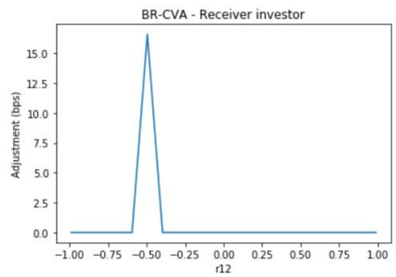

Figure 11: Bilateral risk credit valuation adjustment for a receiver CDS.

We represent here the evolution of the adjustment with the correlation between the reference underlying and the counterparty. We can see that for a payer CDS, the more the correlation increases the more the BR-CVA seems to become larger. The inverse seems to be observed on a Receiver CDS. Here the results are quite poor. There are a lot of zeros returning from our algorithm because we observe no default from neither the investor nor the couterparty. This is because of technical limitations. The algorithm is computationaly intensive: around 5min for 1 trajectory with our resources. As a consequence, it was quite difficult to have a high number of simulations and perform satisfying Monte Carlo approximations. Getting more RAM should have allowed our algorithm produce similar graphs as the ones presented in the article (the two following graphs available).

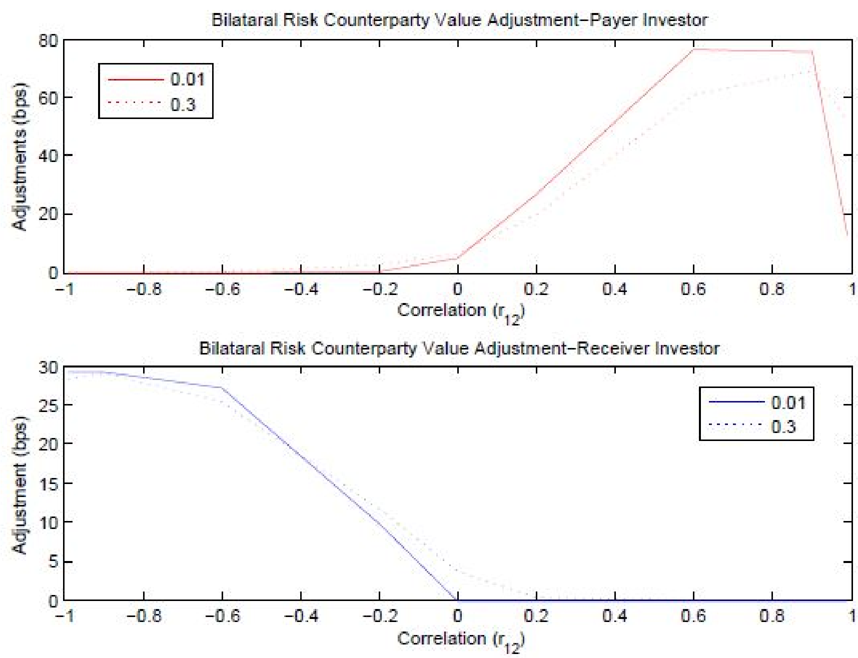

Figure 12: Bilateral risk credit valuation adjustment for payer and receiver CDS.

Although our tendencies are not in contradiction with the ones of Figure 12, we can see that the restricted number of simulations is a real problem. From here, we can not analyze our results without taking a huge risk of misinterpretation. So we must be very careful with the conclusion below. Our results were computed in the scenario 4 of the article (Risky ref): both investor and counterparty have middle credit risk while the reference entity has high credit risk (for computational reasons we placed ourselves with invest./counterparty high risk, and ref. entity very high risk, the interpretation is the same). On Figure 12, the investor and the counterparty are low risk, the reference entity is high risk.

The BR-CVA for a payer CDS increases with the correlation r_{12} (ie the price of the CDS decreases, see 2.) because the risk that the reference entity defaults increases the risk that the counterparty defaults too. Thus,

in case of default, the investor would receive only a fraction of the cover offered by the CDS. On the contrary, if the correlation decreases, the more the reference entity is subject to default the less the counterparty is subject to default: we aim to a riskless counterparty and only a risky investor so the BR-CVA should be negative. This is observed on our results for very low correlation. It is not observed on Figure 12 because the investor is not risky in this simulation, so BR-CVA tends to zero.
The BR-CVA for a receiver CDS evolves in the opposite direction. On our graph we should see a negative BR-CVA for high correlation, but as said above, our results do not represent the all picture due to computation resources.

Conclusion

The goal of this article was to provide a framework for calculating the BR-CVA of any payoff exchanged between two entities. A general formula was provided at the beginning and then an application was performed with a CDS payoff. The innovation of this article was to implement a bilateral risk CVA, taking into account a default risky counterparty and investor. By doing this, we can also modelized the associated correlationThis allows to explain statements that seem contradictory in the first place.
To illustrate these results, we constructed a numerical simulation in order to compare with the authors' ones. Our results were not in contradiction with the latter, but we were limited by computational resources and thus we could not do much Monte Carlo simulation. Therefore, there was a too large random part in our results. We could identify the same tendencies and having coherent results with a different risk scenario.
Here are some improvements that could be done for better simulations:

- Having a greater computational power, which was very limited in our case, in order to perform a good number of Monte Carlo simulations (at least 1,000)
- Having the recommendation above, it is needed to have thinner steps in the discretization algorithms for better accuracy
- Implementing a Gauss-Lobatto procedure to compute the CDF of the integrated CIR process, in order to limit the oscillations

Appendix

- Bilateral counterparty risk valuation with stochastic dynamical models and application to Credit Default Swaps; Brigo and Capponi; 2008
- Numerical evaluation of complex logarithms in the Cox–Ingersoll–Ross model; Teng, Ehrhardt and Gunther; 2012
- Options, futures and other derivatives, 9th edition, chapiter 24 "Credit Risk"; John C. Hull
- Wikipedia, Cox-Ingersol-Ross model
- https://stats.stackexchange.com/questions/37424/how-to-simulate-from-a-gaussian-copula
- On Cox Processes and Credit Risky Securities; Lando; 1998

Pseudo-Algorithm 1

```
[BR–CVA–R,BR–CVA–P] = Calculate Adjustment(N,S)
for i = 1:N do
        Generate tau0, tau1, and tau2 using Eq. (3.5) and Eq. (3.6).
        if tau2 < tau0 and tau2 < Tb then:
                if tau1 > tau2 then:
                [BR–CVA–R-2, BR–CVA–P-2] = CDSAdjust(T(gamma(tau2)), S, LGD1, 2)
                CUM–BR–CVA–R = CUM–BR–CVA–R + LGD2*BR–CVA–R-2
                CUM–BR–CVA–P = CUM–BR–CVA–P + LGD2*BR–CVA–P-2
                end if
        end if
        if tau0 < tau2 and tau0 < Tb then
                if tau1 > tau0 then
                [BR–CVA–R-0, BR–CVA–P-0] = CDSAdjust(T(gamma(tau0)), S, LGD1, 0)
                CUM–BR–CVA–R = CUM–BR–CVA–R - LGD0*BR–CVA–R-0
                CUM–BR–CVA–P = CUM–BR–CVA–P - LGD0*BR–CVA–P-0
                end if
        end if
end for
BR–CVA–R = CUM–BR–CVA–R / N
BR–CVA–P = CUM–BR–CVA–P / N
```

Pseudo-Algorithm 2

```
[CDSR, CDSP] = CDSAdjust(Tj, S, LGD1, index)
Term1 = Term2 = Term3 = 0
tstart = max(Ta; Tj)
Qprev = ComputeProb(tstart, Tj , index)

for t = tstart+delta:delta:Tb
        Qcurr = ComputeProb(t, Tj, index)
        Term1 = Term1 + D(Tj, t-delta)(t-delta–T(gamma(t-delta))-1)(Qcurr - Qprev)
        Term3 = Term3 + D(Tj, t-delta)(Qcurr - Qprev)
        Qprev = Qcurr
end for

for ti = tstart+alpha_i:alpha_i:Tb
        Qcurr = ComputeProb(ti, Tj, index)
```

```
        Term2 = Term2 + alpha_i * D(Tj, ti) * Qcurr
end for

CDSval = S*(Term2 - Term1) + LGD1*Term3
if index == 2 then
        CDSR = D(t, Tj) * max(CDSval, 0)
        CDSP = D(t, Tj) * max(-CDSval, 0)
end if

if index == 0 then
        CDSR = D(t, Tj) * max(-CDSval, 0)
        CDSP = D(t, Tj) * max(CDSval, 0)
end if
```

Pseudo-Algorithm 3

```
Qi = ComputeProb(t, Tj, index)

Uindex = 1 - exp(-Yindex(Tj) - Psi_index(Tj,Beta_index))
U1 = 1 - exp(-Y1(Tj)-Psi_1(Tj,Beta_1))

for xk = 0:Delta:xmax
        pk = CDF(xk)
        uk = 1 - exp(-xk - Psi_1(t))
        if index == 2 then
                Compute fk = C1|0_2(uk, U2) using Eq. (4.5)
        else
                Compute fk = C1|2_0(uk; U0) using Eq. (4.7)
        end if
end for
Qi = SUM _(uk,pk,fk):uk>U1 _ [pk(fk+1 - fk)]
```