Simulation/Regression Pricing Schemes for CVA Computations on CDO Tranches

Stéphane Crépey*
Laboratoire Analyse et Probabilités
Université d'Évry Val d'Essonne
91037 Évry Cedex, France
stephane.crepey@univ-evry.fr

Abdallah Rahal
Bank Audi, Beyrouth, Lebanon
abourach@idm.net.lb

January 29, 2013

Abstract

We devise simulation/regression numerical schemes for pricing the CVA on CDO tranches, where CVA stands for Credit Valuation Adjustment, or price correction accounting for the defaultability of a counterparty in an OTC derivatives transaction. This is done in the setup of a continuous-time Markov chain model of default times, in which dependence between credit names is represented by the possibility of simultaneous defaults. The main idea of the paper is to perform the nonlinear regressions which are used for computing conditional expectations, in the time variable for a given state of the model, rather than in the space variables at a given time in diffusive setups. This idea is formalized as a lemma which is valid in any continuous-time Markov chain model. It is then implemented on the targeted application of CVA computations on CDO tranches.

Keywords: Continuous-time Markov chains, Monte Carlo simulation, regression, counterparty risk, CVA, credit derivatives, CDO.

1 Introduction

Counterparty risk is the risk of default of a party in an OTC derivatives transaction, The pricing (so called Credit Valuation Adjustment, CVA for short) of this risk as well as its management are very topical since the 2007-09 credit crisis; see for instance [1] and other related papers and references in the same volume. A key CVA issue is wrong-way risk, referring to a positive dependence between the amount of counterparty risk exposure and the risk of default of the counterparty. Wrong-way risk is a special concern regarding counterparty risk on credit derivatives, via the strong dependence contagion effects that

*The research of this author benefited from the support of the 'chaire risque de crédit' and the 'chaire marchs en mutation', Fédération Bancaire Française.

may occur between the default risk of the different credit names involved, including the counterparty. In particular, an important practical issue is for banks is that of counterparty risk on CDOs. Note that even if there are barely no new CDO issuances since the crisis, however CDOs are still relevant in terms of risk-management as they are still present in banks portfolios.

Portfolio credit risk is an intrinsically high-dimensional issue, so that from the curse of dimensionality, CVA computations on credit derivatives can only be done by Monte Carlo simulation. Moreover with CVA computations the underlying contract (a CDO tranche in this paper) needs to be priced at every future point of every simulated trajectory. This requires to be able to compute the pricing function of the tranche in a Markov model on all the state space, or at least, on a (random) mesh discretizing a "significant" part of the state space.

Now, for computing by simulation a pricing function as a whole, standard "European" Monte Carlo loops are inappropriate (since these would involve resimulation from every state). A possible approach consists in simulation/regression schemes, such as the ones which are used for pricing by simulation American options (even though no early exercise features are involved here, but for the same motivation that prices are needed everywhere on a grid, in a high-dimensional state space [14]). The contribution of this paper is to devise such kinds of schemes in the context of credit risk applications involving jumps and systems of equations. The ***main idea*** of the paper is then to perform the nonlinear regressions which are used for computing conditional expectations, in the time variable for a given state of the model, rather than in the space variables for a given time in more common diffusive setups (see for instance [19]). This idea is formalized in Section 2 as a generic ***lemma*** which is valid in any continuous-time Markov chain model. It is then implemented and tested on the targeted CVA application, in the common shock model of counterparty credit risk of [2, 3, 4, 1, 18]. As developed in Section 3 in this model a CDO tranche can be priced explicitly at any future time point, so that an exact CVA Monte Carlo scheme is also available and can be used for validating the results of a CVA simulation/regression scheme. We report on numerics in Section 4.

1.1 Standing Notation

The evolution of a financial market is modeled throughout in terms of stochastic processes defined on a standard stochastic basis $(\Omega, \mathcal{F}, \mathbb{F}, \mathbb{P})$. Here $\mathbb{F} = (\mathcal{F}_t)_{t\in[0,T]}$ with $\mathcal{F}_T = \mathcal{F}$ is a continuous-time filtration with finite horizon T, and $\mathbb{P}$ denotes a risk-neutral probability measure such that prices of traded assets are $\mathbb{P}$-martingales, assuming nil funding rates for simplicity. A process on $[0,T]$ (resp. a random variable) has to be $\mathbb{F}$-adapted (resp. $\mathcal{F}$-measurable), by definition. We denote by $\mathbb{E}_t$ (resp. $\mathbb{E}$) the $\mathbb{P}$-conditional expectation given $\mathcal{F}_t$ (resp. $\mathbb{P}$-expectation).

2 General Setup

2.1 CDOs

As a prerequisite let us recall the portfolio credit risk problem of pricing a CDO tranche. Let N_t represent the cumulative default process, or number of firms defaulted by time t within a reference pool of n credit names, with n "large" (of the order of 100 or more in applications, see [1] and other references in the same volume). Let also $L_t = (1-R)N_t$

represent the cumulative portfolio loss process, assuming a constant fractional recovery R (typically taken as $R = 40\%$ before the crisis, often modeled as random since then). We consider protection legs of CDO tranches, with cumulative payoff process of the form, for $t \in [0, T]$,

$$\pi(N_t) = (\frac{L_t}{n} - a)^+ \wedge (b - a) \tag{1}$$

where the attachment and detachment points a and b are such that $0 \leq a \leq b \leq 100\%$. We shall refer to such stylized loss derivatives as CDO tranches, thus ignoring the fees legs of the actual products. Our CDO tranches are thus European options with payoff $\xi = \pi(N_T)$ and price process

$$\Pi_t = \mathbb{E}_t \xi. \tag{2}$$

Remark 2.1 Note that the price Π thus defined is in fact the cumulative price of the tranche, including past dividends. The ex-dividend price $\check{\Pi}_t$ corresponds to $\mathbb{E}_t(\xi - \pi(N_t))$.

2.2 Key Lemma

Let be given a continuous-time d-variate Markov chain $\mathcal{N} = (N^1, \ldots, N^d)$ living in a finite set $\mathcal{I}$. The cumulative default process N_t of a credit portfolio can then be modeled as $N_t = \Lambda(\mathcal{N}_t)$ for some integer-valued function Λ, in the filtration $\mathcal{F}_t = \mathcal{F}_t^{\mathcal{N}}$. Many dynamic models of credit risk can be cast in this framework, directly or after discretisation in space. A concrete specification will be used in Sections 3 and 4.

Given a payoff $\xi = \pi(N_T) = \phi(\mathcal{N}_T)$ with $\phi = \pi \circ \Lambda$, the corresponding price process satisfies for $t \in [0, T]$, by the Markov property of $\mathcal{N}$:

$$\Pi_t = \mathbb{E}_t \xi = \mathbb{E}_t \phi(\mathcal{N}_T) = u(t, \mathcal{N}_t). \tag{3}$$

Here $u(t, \imath)$ or $u_\imath(t)$, for $t \in [0, T]$ and $\imath = (i_1, \ldots, i_d) \in \mathcal{I}$, defines the pricing function, solution to a related system of Kolmogorov equations. In low-dimensional cases with small d, it is possible to devise efficient deterministic computational schemes for the pricing function u (see for instance [5]):

- ODE schemes for the related Kolmogorov equations;
- or, in the time-homogenous case, numerical matrix exponentiation schemes.

However, due to the curse of dimensionality, these schemes cease to be practical as soon as d exceeds a few units. Our aim in this paper is to compute the pricing function u (or part of it) in high dimension, numerically by simulation/regression. The following lemma will be key for this purpose.

Lemma 2.1 *Given a fixed state $\imath \in \mathcal{I}$, let τ denote the first hitting time of $\imath$ by $\mathcal{N}$, with the convention that $\tau = +\infty$ if $\tau > T$ (so that τ is a $[0, T] \cup \{+\infty\}$-valued stopping time). Then, almost surely:*

$$\mathbb{1}_{\tau < T} u_\imath(\tau) = \mathbb{E}(\mathbb{1}_{\tau < T} \xi | \tau). \tag{4}$$

Proof. An application of Doob's optional sampling theorem to the martingale[1] $u(t, \mathcal{N}_t)$ and to the bounded stopping time $\bar{\tau} = \tau \wedge T$ yields that

$$\mathbb{E}_{\bar{\tau}}(\xi) = \mathbb{E}_{\bar{\tau}}(\phi(\mathcal{N}_T)) = u(\bar{\tau}, \mathcal{N}_{\bar{\tau}}) \tag{5}$$

where $\mathcal{N}_{\bar{\tau}} = \imath$ on $\{\tau < T\}$. Multiplying (5) by $\mathbb{1}_{\tau<T}$ one gets

$$\mathbb{E}_{\bar{\tau}}(\mathbb{1}_{\tau<T}\xi) = \mathbb{1}_{\tau<T} u_\imath(\bar{\tau}). \tag{6}$$

Since $\sigma(\bar{\tau}) \subseteq \mathcal{F}^{\mathcal{N}}_{\bar{\tau}}$, taking conditional expectations given $\bar{\tau}$ in (6) yields:

$$\mathbb{E}(\mathbb{1}_{\tau<T}\xi|\bar{\tau}) = \mathbb{1}_{\tau<T} u_\imath(\bar{\tau}).$$

The proof is concluded by noting that on $\{\tau < T\}$ we have $\bar{\tau} = \tau$ and $\mathbb{E}(\mathbb{1}_{\tau<T}\xi|\bar{\tau}) = \mathbb{E}(\mathbb{1}_{\tau<T}\xi|\tau)$. □

2.3 Generic Simulation Pricing Scheme

The practical output of Lemma 2.1 is the following algorithm, based on (4), for estimating the function $t \mapsto u_\imath(t)$ over $(0, T)$:

- simulate m ($= 10^4$ to 10^6, say) trajectories of $\mathcal{N}$ over $[0, T]$ and related values of the payoff $\xi = \phi(\mathcal{N}_T)$, storing the pairs $(\tau^j, \xi^j)^{1\leq j\leq m}$, where the superscript j refers to the j^{th} trajectory. Note that this can typically be done exactly by standard simulation methods, without any discretisation error (see the following Sections for a concrete example);

- estimate the function $u_\imath(\cdot)$ over $(0, T)$ by nonlinear regression (see Section A) of the ξ^j with respect to the τ^j, where the nonlinear regression is performed over the subset of the indices j such that $\tau^j < T$ (for which $u(\tau, \mathcal{N}_\tau) = u_\imath(\tau)$).

The nonlinear regression provides an estimator of the following regression function (cf. Section A)

$$(0, T) \ni t \mapsto \mathbb{E}(\xi|\tau = t) = u_\imath(t) \in \mathbb{R}_+, \tag{7}$$

where the equality holds by (4). Moreover we have $u_\imath(T) = \phi(\imath)$ and $\Pi_0 = u_{\mathbf{0}}(0)$ can be estimated in a standard Monte Carlo way by the sample average $\widehat{\Pi}_0$ of the ξ^j. We finally end up with an estimator of the function $u_\imath(\cdot)$ over $[0, T]$, which is exact up to the error involved in the estimation of the conditional expectations in (7). By exact, we mean here that this estimator does not involve any time-discretisation error, but only Monte Carlo simulation/regression errors, whereas in a deterministic ODE scheme for instance, default times need to be "discretized" on a fixed time-grid.

The simulation step (first bullet item above) is only performed once. The simulated paths are then used in a second step for estimating the functions $u_\imath(\cdot)$ corresponding to the different states $\imath$ of interest. Given a state $\imath \in \mathcal{I}$, let us denote by:

- τ_i^j in $[0, T] \cup \{+\infty\}$, the first hitting time of the state $\imath \in \mathcal{I}$ on the j^{th} trajectory of $\mathcal{N}$, with the convention that $\tau_i^j = +\infty$ if $\tau_i^j > T$;

[1] In a càdlàg version as always by default throughout the paper.

- $\Omega_\imath$, the subset of indices j of the simulated trajectories such that $\tau_\imath^j < T$, so that $u(\tau_\imath^j, \mathcal{N}^j_{\tau_\imath^j}) = u_\imath(\tau_\imath^j)$ for $j \in \Omega_\imath$.

Of course, the set $\Omega_\imath$ will typically be very small, even actually empty, for most states $\imath \in \mathcal{I}$. So, for most states $\imath \in \mathcal{I}$, the nonlinear regression related to the state $\imath$ will not be tractable. Now the key point of the method[2] is that instead of looping over all states $\imath \in \mathcal{I}$ and tentatively performing the regression over each set $\Omega_\imath$, we can instead loop over the trajectories $j = 1, \ldots, m$, make a record of the states $\imath \in \mathcal{I}$ for which the set $\Omega_\imath$ is of cardinality greater then a given threshold μ, and only perform the regressions corresponding to those $\imath$.

The rationale for this procedure is that the states $\imath$ which are not hit (or not hit a sufficient number of times) in the simulation, are low probability $\imath$, starting from the initial condition $\mathcal{N}_0 = \mathbf{0}$. So the values of the pricing functions at these states are essentially irrelevant, and we do not need, nor would we be able, to compute them.

3 Common Shocks Model of Portfolio Credit Risk

In the remaining two Sections, we deal with counterparty risk on CDO tranches (see [1] and other references in the same volume). Let us consider a default-free bank holding a buy-protection CDO tranche from a risky counterparty. The position of the bank is thus subject to counterparty risk. For simplicity we only consider unilateral counterparty risk here, which corresponds to the case of a default-free bank.

Remark 3.1 This also covers the case of a bank possibly risky but disregarding, in a unilateral CVA perspective, its own default risk, when assessing the counterparty risk of its position with another party. The more general situation of bilateral counterparty risk, including also a related nonlinear funding issue [16, 17, 12], becomes very involved in the case of counterparty risk on portfolio credit derivatives.

We denote by $\tau_0 = \tau$ the default time of the counterparty, and by τ_i, for $i \in \mathbb{N}_n^* = \{1, \ldots, n\}$, the default times of n credit names underlying the CDO. For $i \in \mathbb{N}_n = \{0, \ldots, n\}$, we let N^i stand for the default indicator process of τ_i, so $N_t^i = \mathbb{1}_{\tau_i \le t}$, and we denote $\mathcal{N} = (N^i)_{i \in \mathbb{N}_n}$, $N_t = \sum_{i \in \mathbb{N}_n^*} N_t^i$. For the targeted application of CVA computations on CDO tranches, we need a model of counterparty credit risk with the following key features:

- It should be a dynamic model. Indeed, as it is well known (and will be apparent on the formulas (13)-(14) below), the CVA is an option on the future values of the so-called clean price (price not accounting for the counterparty risk) of the underlying, a CDO tranche in our case. Thus we need a model with tractable clean valuation of CDO tranches at any future time point.

- The model should be calibratable jointly to CDS and CDO data. It should therefore be a bottom-up [6] model of portfolio credit risk so that individual names are represented in the model. But for tractability of the calibration the model should also enjoy a copula-like separation property between the individual and the dependence parameter.

[2] This is the interest of a simulation/regression schemes with respect to deterministic ones; see also for instance [19].

This motivates the use of the Markovian copula common shock model of [2, 4] (to our knowledge, this is in fact the only model in the currently existing literature with all the required features). One first defines a certain number ν (typically small: a few units) of groups $I_l \subseteq \mathbb{N}_n$, of obligors who are likely to default simultaneously, for $l = 0, \ldots, \nu$. The idea is that at any time t, there is a positive probability that the survivors of the group of obligors I_l (obligors of group I_l still alive at time t) can default simultaneously. Let then $\mathcal{I} = \{I_0, \ldots, I_\nu\}$, $\mathcal{Y} = \{\{0\}, \ldots, \{n\}, I_0, \ldots, I_\nu\}$. Given the so-called shock intensities λ_Y, $Y \in \mathcal{Y}$ (assumed constant for simplicity in this paper), we model $\mathcal{N} = (N^i)_{i \in \mathbb{N}_n}$ as a Markov chain with respect to its natural filtration $\mathbb{F}$, with infinitesimal generator $\mathcal{A}$ of $\mathcal{N}$ given in functional form as, for $u = u(t, \imath)$ with $t \in [0, T]$ and $\imath = (i_j)_{j \in \mathbb{N}_n}$:

$$\mathcal{A}_t u(t, \imath) = \sum_{Y \in \mathcal{Y}} \lambda_Y \delta_Y u(t, \imath) \tag{8}$$

where $\delta_Y u(t, \imath) = u(t, \imath^Y) - u(t, \imath)$, in which $\imath^Y$ is the vector obtained from $\imath = (i_0, \ldots, i_n)$ by replacing the components i_j, $j \in Y$, by numbers one. As shown in [2], this structure of the generator implies the following expression for the intensity of a jump of $\mathcal{N} = (N^i)_{i \in \mathbb{N}_n}$ from $\mathcal{N}_{t-} = \imath$ to $\mathcal{N}_t = \jmath$, with $\jmath \neq \imath$ in $\{0, 1\}^{n+1}$:

$$\sum_{\{Y \in \mathcal{Y};\, \imath^Y = \jmath\}} \lambda_Y. \tag{9}$$

The intensity of a jump of $\mathcal{N}$ from $\imath$ to $\jmath$ at time t is thus equal to the sum of the intensities of the shocks $Y \in \mathcal{Y}$ such that, if the default of the survivors in set Y occurred at time t, the state of $\mathcal{N}$ would move from $\imath$ to $\jmath$. In particular, the pre-default intensity of a given name i is given as

$$\lambda_i := \sum_{\mathcal{Y} \ni Y \ni i} \lambda_Y = \lambda_{\{i\}} + \sum_{\mathcal{I} \ni I \ni i} \lambda_I,$$

which does not depend on the default status of other obligors. As shown in [2], this implies the so-called Markov copula property [9, 7, 8] that for every $i \in \mathbb{N}_n$, N^i is a $\{0, 1\}$-valued Markov chain admitting the following generator, for $u_i = u_i(t, j)$ with $(t, j) \in [0, T] \times \{0, 1\}$:

$$\mathcal{A}^i_t u_i(t, j) = \lambda_i \big(u_i(t, 1) - u_i(t, j)\big),$$

or in an equivalent matrix-form:

$$\mathcal{A}^i \equiv \begin{bmatrix} -\lambda_i & \lambda_i \\ 0 & 0 \end{bmatrix}.$$

3.1 Example

The following illustrative example is drawn from [2]. Take $n = 2$ (case of three credit names labelled 0 to 2), so that the state space of $\mathcal{N}$ contains the following 8 elements:

$$\{(0,0,0), (1,0,0), (0,1,0), (0,0,1), (1,1,0), (1,0,1), (0,1,1), (1,1,1)\}.$$

Let $\mathcal{Y}$ be given as $\mathcal{Y} = \{\{0\}, \{1\}, \{2\}, \{0,1\}, \{0,1,2\}\}$. There are five different shocks, corresponding to the elements of $\mathcal{Y}$. In particular, obligors 0 and 1 can default simultaneously if either the shock corresponding to $\{0, 1\}$ arrives, or the shock corresponding to $\{0, 1, 2\}$

arrives. This is an example of a nested structure of $\mathcal{I}$: $I_0 = \{0,1\} \subset I_1 = \{0,1,2\}$. Here the generator of the chain $\mathcal{N}$ takes the following matrix-form:

$$\mathcal{A} \equiv \begin{bmatrix} \cdot & \lambda_{\{0\}} & \lambda_{\{1\}} & \lambda_{\{2\}} & \lambda_{\{0,1\}} & 0 & 0 & \lambda_{\{0,1,2\}} \\ 0 & \cdot & 0 & 0 & \lambda_{\{1\}} + \lambda_{\{0,1\}} & \lambda_{\{2\}} & 0 & \lambda_{\{0,1,2\}} \\ 0 & 0 & \cdot & 0 & \lambda_{\{0\}} + \lambda_{\{0,1\}} & 0 & \lambda_{\{2\}} & \lambda_{\{0,1,2\}} \\ 0 & 0 & 0 & \cdot & 0 & \lambda_{\{0\}} & \lambda_{\{1\}} & \lambda_{\{0,1,2\}} + \lambda_{\{0,1\}} \\ 0 & 0 & 0 & 0 & \cdot & 0 & 0 & \lambda_{\{2\}} + \lambda_{\{0,1,2\}} \\ 0 & 0 & 0 & 0 & 0 & \cdot & 0 & \lambda_{\{1\}} + \lambda_{\{0,1,2\}} + \lambda_{\{0,1\}} \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdot & \lambda_{\{0\}} + \lambda_{\{0,1,2\}} + \lambda_{\{0,1\}} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

where a "$\cdot$" represents minus the sum of all the other elements in the same row. Now, consider group $\{0,1,2\}$. Suppose that at some point of time obligor 1 is defaulted, but obligors 0 and 2 are still alive, so that process $\mathcal{N}$ is in state $(0,1,0)$. In this case the two survivors in the group $\{0,1,2\}$ may default simultaneously with intensity $\lambda_{\{0,1,2\}}$. Note that here $\lambda_{\{0,1,2\}}$ cannot be interpreted as intensity of all three defaulting simultaneously, as obligor 1 has already defaulted. The only state of the model in which $\lambda_{\{0,1,2\}}$ can be interpreted as the intensity of all three defaulting, is the origin $(0,0,0)$. Regarding the Markov copula property observe that as long as obligor 0 is alive, her default intensity is $\lambda_{\{0\}} + \lambda_{\{0,1\}} + \lambda_{\{0,1,2\}}$, regardless of the state of the pool. Similar remarks apply to obligors 1 and 2.

3.2 Marshall-Olkin Representation

In [2] a connection is established between the above Markov chain model $\mathcal{N}$ with default times τ_i (actually, an extension of this model involving diffusive factor processes), and a static Marshall-Olkin model [24].

Let $\mathrm{supp}^c(\mathcal{N}_t)$ denote the set of the surviving names in our model $\mathcal{N}$ at time t, so that

$$\mathrm{supp}^c(\mathcal{N}_t) = \{i \in \mathbb{N}_n;\, N_t^i = 0\}.$$

One can define a Marshall-Olkin model of default times $(\widetilde{\tau}_i)_{i \in \mathrm{supp}^c(\mathcal{N}_t)}$ of the surviving names in model $\mathcal{N}$ at time t, such that given $\mathcal{N}_t$, $\tau = (\tau_i)_{i \in \mathrm{supp}^c(\mathcal{N}_t)}$ and $\widetilde{\tau} = (\widetilde{\tau}_i)_{i \in \mathrm{supp}^c(\mathcal{N}_t)}$ are equal in law. Denoting $\mathcal{Y}_t = \{Y \in \mathcal{Y}; Y \not\subseteq \mathrm{supp}(\mathcal{N}_t)\}$, let us thus define, for every $Y \in \mathcal{Y}_t$,

$$\tau_Y = t + \frac{\epsilon_Y}{\lambda_Y} \tag{10}$$

for i.i.d. exponential random variables ϵ_Y. Let us set for every $i \in \mathrm{supp}^c(\mathcal{N}_t)$

$$\widetilde{\tau}_i = \bigwedge_{Y \in \mathcal{Y}_t\,;\, i \in Y} \tau_Y. \tag{11}$$

The τ_i denoting as before the default times of the credit names in the Markovian model $\mathcal{N}$, we have for every $t_1, \ldots, t_n \geq t$ (see [2]):

$$\mathbb{P}\left(\tau_i > t_i,\, i \in \mathrm{supp}^c(\mathcal{N}_t) \,\middle|\, \mathcal{N}_t\right) = \mathbb{P}\left(\widetilde{\tau}_i > t_i,\, i \in \mathrm{supp}^c(\mathcal{N}_t) \,\middle|\, \mathcal{N}_t\right). \tag{12}$$

Crucially for the targeted application of CVA computation on CDO tranches, the Marshall-Olkin representation can be used for deriving fast exact recursive procedures for pricing a CDO tranche, conditionally on $\mathcal{N}_t$ (see [2, 3]). The Marshall-Olkin representation is also used for simulation purposes. The simulation of a set of random times $(\tau_i)_{i\in\mathbb{N}_n}$, or, equally in law by (12), $(\widetilde{\tau}_i)_{i\in\mathbb{N}_n}$, is very fast: all one needs is to simulate the i.i.d. exponential random variables ϵ_Y in (10), for Y in $\mathcal{Y}$. Finally, thanks to the Markov copula property of the model, the marginal survival probabilities can essentially be considered as inputs to the model (they can be bootstrapped independently in a preprocessing step from individual CDS quotes); the only parameters to calibrate are then the few model dependence parameters λ_I.

3.3 CVA Computations on CDO Tranches

The numerical results of [2, 3] show that the model is indeed jointly calibratable to CDS and CDO real data sets. Moreover, as this is a Markovian model of portfolio credit risk, it is suitable to deal with intrinsic dynamic issues like counterparty risk (see [1, 4]).

The Credit Valuation Adjustment (CVA) refers to the correction in value accounting for the credit risk of a counterparty. Let us thus consider the default-free bank buying credit protection from her defaultable counterparty, modeled as name 0, through a buy-protection CDO tranche with maturity T on names 1 to n. By application of the results of [11, 4], one can represent the CVA of the bank by the following process Θ_t, for $0 \le t \le \tau \wedge T$:

$$\Theta_t = \mathbb{E}_t(\mathbb{1}_{\tau<T}\chi) \tag{13}$$

for an $\mathcal{F}_\tau$-measurable counterparty exposure χ of the bank given as

$$\chi = (1-R_0)\left(\Pi_\tau - \pi(N_{\tau-}) - \Gamma_\tau\right)^+ \tag{14}$$

in which:

- R_0 is the recovery rate of the counterparty,
- $\Pi_\tau = \mathbb{E}_\tau\xi = u(\tau,\mathcal{N}_\tau)$ is the counterparty clean (not accounting for counterparty risk) price of the CDO tranche at the time of default τ of the counterparty,
- π is the tranche (cumulative) payoff function, and
- Γ represents a non-negative (cumulative) margin amount, value of the accumulated collateral posted by the counterparty to the bank in order to mitigate counterparty risk.

Remark 3.2 As seen in Remark 2.1, Π defined by (2) is in fact the cumulative price of the (protection leg of a) CDO tranche. This explains the expression (14) for χ, consistent with the corresponding expression of [4] in terms of the ex-dividend price process $\check{\Pi} = \Pi - \pi(N)$:

$$\chi = (1-R_0)\left(\check{\Pi}_\tau + \pi(N_\tau) - \pi(N_{\tau-}) - \Gamma_\tau\right)^+. \tag{15}$$

Thus, we have under the so-called naked (no collateralization) scheme $\Gamma = 0$ and the (collateralized) scheme $\Gamma = \Pi_- - \pi_-$, respectively:

$$\begin{aligned}
&\Theta_t = \Theta^0_t = \mathbb{E}_t(\mathbb{1}_{\tau<T}\chi^0) \text{ with } \chi^0 = (1-R_0)\left(u(\tau,\mathcal{N}_\tau) - \pi(N_{\tau-})\right)^+ \\
&\Theta_t = \Theta^1_t = \mathbb{E}_t(\mathbb{1}_{\tau<T}\chi^1) \text{ with } \chi^1 = (1-R_0)\left(u(\tau,\mathcal{N}_\tau) - u(\tau,\mathcal{N}_{\tau-})\right)^+.
\end{aligned} \tag{16}$$

Remark 3.3 Accounting for various frictions and delays (notably the so-called cure period [4]) regarding formation and delivery of the collateral, the case where $\Gamma = \Pi_- - \pi_-$ can be considered as the most extreme case of collateralization, sometimes referred to as continuous collateralization (with an "infinitesimal" cure period reflected by the left-limit in $\Gamma = \Pi_- - \pi_-$, whereas more realistic models of collateral would account for a positive cure period; see [15]).

Using the Marshall-Olkin representation of Subsection 3.2, the simulation of the counterparty's default time $\tau = \tau_0$ and the computation of $\Pi_\tau = \mathbb{E}_\tau \xi = u(\tau, \mathcal{N}_\tau)$ in χ^0, and also of $\Pi_{\tau-} = u(\tau, \mathcal{N}_{\tau-})$ in χ^1, can be done in a very fast and exact way. One thus has the following exact Monte Carlo estimates $\widetilde{\Theta}_0^0$ and $\widetilde{\Theta}_0^1$ for the naked and collateralized CVA at time 0 of a CDO tranche, based on m simulated trajectories of $\mathcal{N}$:

$$\begin{aligned} \widetilde{\Theta}_0^0 &= (1-R_0)\frac{1}{m}\sum_{j=1}^m \mathbb{1}_{\tau^j<T}\left(u(\tau^j,\mathcal{N}_{\tau^j}) - \pi(N_{\tau^j-})\right)^+, \\ \widetilde{\Theta}_0^1 &= (1-R_0)\frac{1}{m}\sum_{j=1}^m \mathbb{1}_{\tau^j<T}\left(u(\tau^j,\mathcal{N}_{\tau^j}) - u(\tau^j,\mathcal{N}_{\tau^j-})\right)^+. \end{aligned} \tag{17}$$

By exact Monte Carlo estimates, we mean that these involve no time-discretisation error, nor any approximation or error regarding the computation of the pricing function u. The only error is the statistical $\mathrm{O}(m^{-\frac{1}{2}})$-Monte Carlo error.

Alternatively to the above exact Monte Carlo estimates, the approximate Monte Carlo estimates $\widehat{\Theta}_0^0$ and $\widehat{\Theta}_0^1$ are defined as

$$\begin{aligned} \widehat{\Theta}_0^0 &= (1-R_0)\frac{1}{m}\sum_{j=1}^m \mathbb{1}_{\tau^j<T}\left(\widehat{u}(\tau^j,\mathcal{N}_{\tau^j}) - \pi(N_{\tau^j-})\right)^+ \\ \widehat{\Theta}_0^1 &= (1-R_0)\frac{1}{m}\sum_{j=1}^m \mathbb{1}_{\tau^j<T}\left(\widehat{u}(\tau^j,\mathcal{N}_{\tau^j}) - \widehat{u}(\tau^j,\mathcal{N}_{\tau^j-})\right)^+, \end{aligned} \tag{18}$$

in which $\widehat{u}(\tau^j,\mathcal{N}_{\tau^j})$ and $\widehat{u}(\tau^j,\mathcal{N}_{\tau^j-})$ are numerical approximations of $\Pi_{\tau^j} = u(\tau^j,\mathcal{N}_{\tau^j})$ and $\Pi_{\tau^j-} = u(\tau^j,\mathcal{N}_{\tau^j-})$, computed by the simulation/regression approach of Subsection 2.3.

Note that in a real life CVA setup, one deals not with a single CDO tranche, but with a netted portfolio of thousands of contracts, collateralized in a path-dependent way (see [4, 15]). The interest of the estimates (18) is that they may still be practical in more realistic situations where the exact Monte Carlo estimates (17) will not be available anymore (see Cesari et al. in [14]). The estimates (18) are also typically faster to compute (see the "CPU+" rows of the numerical tables in the next section).

4 Numerical Results

We implemented in Matlab the exact and approximate Monte Carlo CVA estimates in the naked and collateralized cases, so $\widetilde{\Theta}_0^0$, $\widetilde{\Theta}_0^1$, $\widehat{\Theta}_0^0$ and $\widehat{\Theta}_0^1$, on a stylized credit portfolio of 100 obligors (including the counterparty), for CDO tranches with maturity $T = 2$ years, and for recovery rates $R_0 = R = 40\%$. The individual default intensities were taken as $\lambda_{\{i\}} = 10^{-4} \times (200 - i)$ for $i = 0, \ldots, n = 99$. Moreover we considered four nested groups

of joint defaults (so $\nu = 3$) respectively consisting of the riskiest 3, 9, 21 and 100 (so all) names, with corresponding joint default intensities $\lambda_I = 10^{-3} \times \frac{2}{1+l}$, for $l = 0, \ldots, \nu$. The counterparty was taken as the fiftieth riskiest name (name with median risk) in the portfolio. Also, regarding the approximate Monte Carlo estimates $\widehat{\Theta}_0^0$ and $\widehat{\Theta}_0^1$, various values were tried for the degree ρ of the polynomial regression in the time variable which is used[3] for computing the conditional expectations ($\widehat{u}$-terms in (18)) and for the threshold μ in Subsection 2.3.

The results are presented in Tables 1 (naked CVA) and 2 (collateralized CVA). With the approximate Monte Carlo estimate $\widehat{\Theta}_0^1$ in the collateralized case, one needs to estimate not only the $u(\tau^j, \mathcal{N}_{\tau^j})$, but also the $u(\tau^j, \mathcal{N}_{\tau^j -})$. This requires a greater number of simulated trajectories to achieve a given level of accuracy. All our naked, respectively collateralized, Monte Carlo estimates, were thus based on a common set of $m = 1.5 \times 10^5$, respectively $m = 3 \times 10^5$, simulated trajectories. The times needed to simulate these m trajectories and construct the tables of simulated paths and payoffs which are used in all the corresponding Monte Carlo estimates were 362s and 2700s, respectively. In Tables 1 and 2:

- The CVA numbers correspond to a nominal of one hundred per obligor, e.g. the maximal loss on the equity tranche in this scale is Nom $\times\, n \times 5\% = 500$,
- σ is the standard error associated with a Monte Carlo CVA estimate,
- $\%\sigma$ is the percentage error in the sense of $10^2 \times \frac{\sigma}{\text{CVA}}$,
- %Err is the percentage relative error of the approximate Monte Carlo estimate with respect to the exact Monte Carlo estimate,
- CPU+ is the additional time (in seconds) needed to compute a specific CVA estimate, on the top of the common time of 362s or 2700s which were required for constructing the tables of payoffs corresponding to the $m = 1.5 \times 10^5$ and 3×10^5 trajectories,
- μ and ρ are the values of the threshold and of the order of polynomials which are used in the simulation/regression estimates of the conditional expectations (see Subsection 2.3 and Section A).

First observe from Tables 1-2 that the impact of the collateralization, as assessed by the difference between the exact Monte Carlo CVA results of Tables 1 and 2, is mainly significant for the equity tranche. So, for the equity tranche 0-5, the exact Monte Carlo value of the naked CVA is $\widetilde{\Theta}_0^0 = 4.78$, whereas the exact Monte Carlo estimate of the collateralized CVA is $\widetilde{\Theta}_0^1 = 3.41$. For the 35+ tranche these numbers respectively become 2.44 and 2.26.

These results reflect the fact that in (15) the senior tranche CVA is less due to the clean price term $\check{\Pi}_\tau$, and more to the joint defaults term $\pi(N_\tau) - \pi(N_{\tau-})$, where the latter is hardly collateralizable (see [1]). Collateralization having little impact on the senior tranches conveys the important message that due to wrong-way risk (which is represented in our model by the possibility of joint defaults), counterparty risk on credit derivatives may be scarcely collateralizable.

In order to keep the construction of the payoff tables under reasonable CPU budget, our approach consists in first simulating the default times following the approach of Subsection 3.2, and then, only constructing the trajectories j of $\mathcal{N}$ with $\tau^j < T$ (therefore

[3] See also Subsection 4.2 for a different kind of regressor which can be used.

in particular, ignoring in the regressions the "rejected" paths for which $\tau^j > T$). Indeed simulating the default times is very fast, whereas reconstructing the sequence of configurations successively reached at the subsequent default times on the corresponding trajectory of $\mathcal{N}$, is much more time-consuming. Moreover even in case $\tau < T$ we do not store the full trajectory, but only the last two states $\mathcal{N}_\tau$ and $\mathcal{N}_{\tau-}$. These reductions drastically diminish the CPU budget of the tables construction, which still took 362s and 2700s above for $m = 1.5 \times 10^5$ and 3×10^5, but would have been intractable otherwise for $n = 100$ obligors and such values of m (see however Subsection 4.2 for possible improvements in this regard).

Note that this rejection of paths for which $\tau^j > T$ induces a specific bias into the $\widehat{u}(\tau^j, \mathcal{N}_{\tau^j-})$ which intervene in $\widehat{\Theta}_0^1$ in (18). Indeed, these are estimates of $\widehat{u}$ at states $\imath = \mathcal{N}_{\tau^j-}$ also possibly attained on some of the rejected paths, which should then have be taken into account in the time-regression estimate of such $\widehat{u}_\imath(\cdot)$. The computation of the $\widehat{u}(\tau^j, \mathcal{N}_{\tau^j-})$ is therefore slightly biased with respect to that of the $\widehat{u}(\tau^j, \mathcal{N}_{\tau^j})$, hence a specific bias in $\widehat{\Theta}_0^1$ follows.

However, in either the naked case of Table 1 or the collateralized case of Table 2, the accuracy of the approximate Monte Carlo estimates $\widehat{\Theta}_0^0$ and $\widehat{\Theta}_0^1$ is satisfactory, in both sense of $\%\sigma$ or %Err. This is at least true provided one appropriately chooses the threshold μ and the degree ρ of the regression. The accuracy of the approximate Monte Carlo estimates is best and of the order of 1%, both in the sense of $\%\sigma$ and %Err, for $\mu = 1$ and any ρ in the naked case, respectively for $\mu = 1$ or 2 and $\rho = 3$ in the collateralized case. The additional times CPU+ needed to compute the approximate Monte Carlo estimates are always less than for the exact estimates.

Even in the collateralized case of Table 2, the accuracy of the approximate Monte Carlo estimate[4] is quite satisfactory, since for well-chosen μ and ρ, the error %Err is typically within the confidence interval error $\%\sigma$; this holds even though we chose for the latter is a relatively narrow one-standard-error confidence interval.

Regarding the threshold μ, the value $\mu = 1$ (or equally $\mu = 2$ in the collateralized case), gives the best results in both cases. It thus seems preferable to retain as much states as possible in the estimates, including states only reached a very small number of times (in fact, even only once) on the simulated trajectories.

4.1 Reduced versus Full Payoff Tables

To further investigate the impact, in the collateralized case, of working with reduced tables of simulated paths and payoffs, we now present comparative results obtained on a portfolio of $n = 60$ obligors, with reduced payoff tables as above, or with the full payoff tables. For $m = 2 \times 10^4$ simulated trajectories, the construction of the reduced and full payoff tables took 3s and 3800s. Note that in our experience, for a given number of trajectories, the time of construction of the full payoffs table increases very fast ("exponentially") with the number of obligors, whereas the time of construction of the reduced payoff tables only increases "linearly" with the number of obligors. With a portfolio of $n = 100$ obligors previously, the construction of the full payoff tables was thus far out of scope.

It is apparent in Tables 3 (reduced payoffs table) and 4 (full payoffs table) that the accuracy of the CVA results is less robust with the reduced payoffs table than with the full payoffs table. Indeed in the first case accurate results are only obtained for $\mu = 3$ and $\rho = 2$, whereas the accuracy of the results is much more uniform in μ and ρ in the second case. To get better results with a reduced payoffs table, one needs to use a greater number

[4]Though endowed with a specific bias as explained above.

$\mu = 1$	MC ρ =1			MC ρ =2			MC ρ =3			Exact		
Tranche	0-5	5-35	35+	0-5	5-35	35+	0-5	5-35	35+	0-5	5-35	35+
CVA	4.75	2.93	2.42	4.75	2.93	2.42	4.76	2.93	2.42	4.78	2.96	2.44
σ	0.08	0.24	0.20	0.08	0.24	0.20	0.08	0.24	0.20	0.08	0.24	0.20
$\%\sigma$	1.6	8.2	8.3	1.6	8.2	8.3	1.6	8.2	8.3	1.6	8.1	8.2
%Err	-0.6	-1.1	-0.8	-0.6	-1.0	-0.8	-0.6	-1.0	-0.7			
CPU+	190s			200s			210s			308s		
$\mu = 3$	MC ρ =1			MC ρ =2			MC ρ =3			Exact		
Tranche	0-5	5-35	35+	0-5	5-35	35+	0-5	5-35	35+	0-5	5-35	35+
CVA	3.42	2.91	2.42	3.42	2.91	2.42	3.42	2.91	2.42	4.78	2.96	2.44
σ	0.07	0.24	0.20	0.07	0.24	0.20	0.07	0.24	0.20	0.08	0.24	0.20
$\%\sigma$	2.0	8.2	8.3	2.0	8.2	8.3	2.0	8.2	8.3	1.6	8.1	8.2
%Err	-28.5	-1.6	-0.8	-28.5	-1.6	-0.8	-28.4	-1.6	-0.7			
CPU+	190s			200s			210s			308s		
$\mu = 5$	MC ρ =1			MC ρ =2			MC ρ =3			Exact		
Tranche	0-5	5-35	35+	0-5	5-35	35+	0-5	5-35	35+	0-5	5-35	35+
CVA	3.36	2.91	2.42	3.36	2.91	2.42	3.36	2.91	2.42	4.78	2.96	2.44
σ	0.07	0.24	0.20	0.07	0.24	0.20	0.07	0.24	0.20	0.08	0.24	0.20
$\%\sigma$	2.0	8.2	8.3	2.0	8.2	8.3	2.0	8.2	8.3	1.6	8.1	8.2
%Err	-29.8	-1.6	-0.8	-29.9	-1.6	-0.8	-29.8	-1.6	-0.7			
CPU+	3.2s			3.2s			3.2s			308s		
$\mu = 10$	MC ρ =1			MC ρ =2			MC ρ =3			Exact		
Tranche	0-5	5-35	35+	0-5	5-35	35+	0-5	5-35	35+	0-5	5-35	35+
CVA	3.35	2.91	2.42	3.35	2.91	2.42	3.35	2.91	2.42	4.78	2.96	2.44
σ	0.07	0.24	0.20	0.07	0.24	0.20	0.07	0.24	0.20	0.08	0.24	0.20
$\%\sigma$	2.0	8.2	8.3	2.0	8.2	8.3	2.1	8.2	8.3	1.6	8.1	8.2
%Err	-29.9	-1.6	-0.8	-29.9	-1.6	-0.8	-29.9	-1.6	-0.7			
CPU+	3.2s			3.2s			3.2s			308s		

Table 1: *Naked CVA ($n = 100$ obligors, $m = 1.5 \times 10^5$ trajectories).*

$\mu = 1$	MC ρ =1			MC ρ =2			MC ρ =3			Exact		
Tranche	0-5	5-35	35+	0-5	5-35	35+	0-5	5-35	35+	0-5	5-35	35+
CVA	2.76	2.73	2.27	2.91	2.76	2.27	3.37	2.77	2.27	3.41	2.73	2.26
σ	0.09	0.16	0.14	0.14	0.17	0.14	0.23	0.17	0.14	0.05	0.16	0.14
$\%\sigma$	3.2	6.0	6.0	4.7	6.0	6.0	6.8	6.0	6.0	1.4	6.0	6.0
%Err	-19.1	-0.2	0.4	-14.6	1.2	0.4	-1.0	1.3	0.4			
CPU+	168s			168s			168s			1033s		
$\mu = 2$	MC ρ =1			MC ρ =2			MC ρ =3			Exact		
Tranche	0-5	5-35	35+	0-5	5-35	35+	0-5	5-35	35+	0-5	5-35	35+
CVA	2.76	2.73	2.27	2.91	2.76	2.27	3.37	2.77	2.27	3.41	2.73	2.26
σ	0.09	0.16	0.14	0.14	0.17	0.14	0.23	0.17	0.14	0.05	0.16	0.14
$\%\sigma$	3.2	6.0	6.0	4.7	6.0	6.0	6.8	6.0	6.0	1.4	6.0	6.0
%Err	-19.1	-0.2	0.4	-14.6	1.2	0.4	-1.0	1.3	0.4			
CPU+	168s			168s			168s			1033s		
$\mu = 3$	MC ρ =1			MC ρ =2			MC ρ =3			Exact		
Tranche	0-5	5-35	35+	0-5	5-35	35+	0-5	5-35	35+	0-5	5-35	35+
CVA	2.53	2.73	2.27	2.58	2.76	2.27	2.87	2.77	2.27	3.41	2.73	2.26
σ	0.04	0.16	0.14	0.05	0.17	0.14	0.07	0.17	0.14	0.05	0.16	0.14
$\%\sigma$	1.7	6.0	6.0	1.9	6.0	6.0	2.4	5.8	6.0	1.4	6.0	6.0
%Err	-25.7	-0.2	0.4	-24.3	1.2	0.4	-15.8	5.0	0.4			
CPU+	168s			168s			168s			1033s		
$\mu = 5$	MC ρ =1			MC ρ =2			MC ρ =3			Exact		
Tranche	0-5	5-35	35+	0-5	5-35	35+	0-5	5-35	35+	0-5	5-35	35+
CVA	2.53	2.73	2.27	2.53	2.73	2.27	2.81	2.73	2.27	3.41	2.73	2.26
σ	0.04	0.16	0.13	0.04	0.16	0.13	0.06	0.16	0.13	0.05	0.16	0.14
$\%\sigma$	1.8	6.0	5.7	1.8	6.0	5.7	2.3	6.0	5.7	1.4	6.0	6.0
%Err	-25.7	-0.2	0.4	-25.7	-0.2	0.4	-17.4	0.0	0.4			
CPU+	5.3s			5.3s			5.3s			1033s		
$\mu = 10$	MC ρ =1			MC ρ =2			MC ρ =3			Exact		
Tranche	0-5	5-35	35+	0-5	5-35	35+	0-5	5-35	35+	0-5	5-35	35+
CVA	2.53	2.73	2.27	2.53	2.73	2.27	2.81	2.73	2.27	3.41	2.73	2.26
σ	0.04	0.16	0.13	0.04	0.16	0.13	0.06	0.16	0.13	0.05	0.16	0.14
$\%\sigma$	1.8	6.0	5.7	1.8	6.0	5.7	2.3	6.0	5.7	1.4	6.0	6.0
%Err	-25.7	-0.2	0.4	-25.7	-0.2	0.4	-17.4	0.0	0.4			
CPU+	5.3s			5.3s			5.3s			1033s		

Table 2: *Collateralized CVA ($n = 100$ obligors, $m = 3 \times 10^5$ trajectories).*

m of simulated trajectories. But for a given level of accuracy this is still much less costly numerically than using a full payoffs table with less trajectories.

$\mu = 1$	MC ρ =1			MC ρ =2			MC ρ =3			Exact		
Tranche	0-5	5-35	35+	0-5	5-35	35+	0-5	5-35	35+	0-5	5-35	35+
CVA	1.70	1.60	1.32	2.40	1.63	1.35	2.90	1.63	1.35	2.19	1.66	1.36
σ	0.11	0.37	0.31	0.11	0.36	0.30	1.49	0.38	0.32	0.11	0.38	0.32
$\%\sigma$	6.7	23.2	23.4	4.4	22.1	22.2	51.4	23.5	23.6	5.2	23.0	23.4
%Err	-22.2	-3.9	-2.8	10.0	-2.1	-0.8	32.7	-2.1	-0.8			
CPU+	68s			68s			68s			33s		
$\mu = 3$	MC ρ =1			MC ρ =2			MC ρ =3			Exact		
Tranche	0-5	5-35	35+	0-5	5-35	35+	0-5	5-35	35+	0-5	5-35	35+
CVA	1.60	1.60	1.32	2.35	1.63	1.35	2.50	1.63	1.35	2.19	1.66	1.36
σ	0.11	0.37	0.31	0.23	0.36	0.30	0.25	0.26	0.20	0.11	0.38	0.32
$\%\sigma$	7.1	23.2	23.4	9.8	22.1	22.2	9.9	16.1	15.0	5.2	23.0	23.4
%Err	-26.8	-3.9	-2.8	7.6	-2.1	-0.8	14.4	-2.1	-0.8			
CPU+	68s			68s			68s			33s		

Table 3: *Collateralized CVA with reduced payoffs table ($n = 60$ obligors, $m = 2 \times 10^4$ trajectories).*

$\mu = 1$	MC ρ =1			MC ρ =2			MC ρ =3			Exact		
Tranche	0-5	5-35	35+	0-5	5-35	35+	0-5	5-35	35+	0-5	5-35	35+
CVA	2.05	1.60	1.32	2.04	1.53	1.26	2.06	1.68	1.39	2.19	1.66	1.36
σ	0.11	0.37	0.31	0.11	0.36	0.30	0.11	0.41	0.34	0.11	0.38	0.32
$\%\sigma$	5.2	23.2	23.4	5.2	23.5	23.8	5.5	24.5	24.4	5.2	23.0	23.4
%Err	-6.1	-3.9	-2.8	-6.6	-7.7	-7.4	-5.6	1.0	2.3			
CPU+	137s			137s			137s			55s		
$\mu = 3$	MC ρ =1			MC ρ =2			MC ρ =3			Exact		
Tranche	0-5	5-35	35+	0-5	5-35	35+	0-5	5-35	35+	0-5	5-35	35+
CVA	2.05	1.60	1.32	2.05	1.53	1.26	2.09	1.68	1.39	2.19	1.66	1.36
σ	0.11	0.37	0.31	0.11	0.36	0.30	0.11	0.41	0.34	0.11	0.38	0.32
$\%\sigma$	5.2	23.2	23.4	5.2	23.5	23.8	5.4	19.7	24.4	5.2	23.0	23.4
%Err	-6.1	-3.9	-2.8	-6.1	-7.7	-7.4	-4.2	1.0	2.3			
CPU+	137s			137s			137s			55s		

Table 4: *Collateralized CVA with full payoffs table ($n = 60$ obligors, $m = 2 \times 10^4$ trajectories).*

4.2 Further Improvements

One can improve the above results in several ways. First, more elaborate regressions can be used instead of the basic polynomial regressions of degrees one to three above. Table 5 thus shows the results obtained for computing the collateralized CVA with reduced payoffs table on CDO tranches bearing on an underlying portfolio of $n = 120$ obligors, with a probit regression based on $m = 5 \times 10^4$ model trajectories and a threshold $\mu = 1$. By this, we mean that for every state $\imath$ reached by the simulations, we regress the corresponding payoffs of the (a, b)-tranche by $(b - a)\Phi^{-1}(\alpha_\imath + \beta_\imath t)$, where $\alpha_\imath$ and $\beta_\imath$ are the regression parameters

and Φ represents the standard Gaussian cumulative distribution function (Matlab function "gmlfit" with the arguments "binomial, link and probit"). The accuracy of the results is similar to those of Table 2, although six times less trajectories are used and more obligors are considered. Moreover the table construction only took 20s (on top of the additional times CPU+ which appear in the last row of Table 5) instead of 2700s in the case of the experiment of Table 2.

$\mu = 1$	MC Probit			Exact		
Tranche	0-5	5-35	35+	0-5	5-35	35+
CVA	3.70	3.25	2.70	3.66	3.27	2.70
σ	0.23	0.48	0.40	0.13	0.48	0.4
$\%\sigma$	6.2	14.8	14.8	3.6	14.7	14.8
%Err	1.1	-0.6	0.0			
CPU+	46s			241s		

Table 5: *Collateralized CVA with reduced payoffs table and probit regression ($n = 120$ obligors, $m = 5 \times 10^4$ trajectories).*

Second, one can use various variance reduction techniques or simulation add-ons. Table 6 thus shows the results obtained with the same specifications as in Table 5, except that a Latin Hypercube Monte Carlo was used instead of a standard Monte Carlo (Matlab command "lhsdesign" used instead of "rand" for generating the uniform numbers underlying the default times). The accuracy of the results is similar as in Table 5 (at least in terms of %Err), although a lower number of trajectories was used ($m = 3 \times 10^4$ versus 5×10^4 in Table 5). This is however at the cost of an increased table construction time of 220s (on top of the additional times CPU+ in the last row of Table 6), as opposed to 20s in Table 5.

$\mu = 1$	LH Probit			Exact		
Tranche	0-5	5-35	35+	0-5	5-35	35+
CVA	3.72	3.48	2.90	3.70	3.50	2.90
σ	0.25	0.60	0.50	0.16	0.6	0.5
$\%\sigma$	6.7	17.2	17.2	4.3	17.1	17.2
%Err	0.5	-0.6	0.0			
CPU+	40s			159s		

Table 6: *Collateralized CVA computed by Latin Hypercube Monte Carlo with reduced payoffs table and probit regression ($n = 120$ obligors, $m = 3 \times 10^4$ trajectories).*

Conclusions

There is numerical evidence that nonlinear regressions in time can fruitfully be used for CVA computations on CDO tranches, in the context of high-dimensional Markov chain models of portfolio credit risk. More generally the method we propose can be used for solving any Markov chain related system of Kolmogorov ODEs. The work of this paper could be pushed in several directions.

The choice and number of basis functions which should theoretically be used in the

nonlinear regressions is a pending issue. However as demonstrated in this paper the practical side of this issue can only be, to a large extent, experimental.

Having in mind that real life CVA applications bear on portfolios of thousands of contracts, the computation times in Sections 3.3 are still quite high. So the question arises of suitable variance reduction techniques like importance sampling or particle methods [13] which could be used to efficiently address the issue of rare default events simulation.

A challenging question is how to possibly extend the technique of this paper to a more general CVA setup of bilateral counterparty risk under funding constraints [16, 17]. One then has to deal with implicit, nonlinear CVA equations, not amenable to explicit Monte Carlo loops anymore. Maybe particle simulation schemes could also be useful in this regard.

A Computing Conditional Expectations by Simulation

The pricing schemes of this paper ultimately reduce to the numerical computation of conditional expectations. This can be done by a combination of simulation and regression tools. In this appendix we provide a brief and informal review on this issue.

Let ξ and τ denote non-negative random variables. Under suitable conditions the conditional expectation $\mathbb{E}(\xi|\tau)$ is equal to the L^2-projection of ξ over the vector space of random variables spanned by the deterministic functions of τ. So, using a basis $(\varphi^l)_{l\in\mathbb{N}}$ of the set of the real functions of time t (such as monomials of increasing degree l in t),

$$\mathbb{E}(\xi|\tau) = \mathbb{L}(\xi|(\varphi^l(\tau))_{l\in\mathbb{N}}),$$

where $\mathbb{L}$ stands for the L^2-projection operator. Given pairs $(\tau^j, \xi^j)^{1\leq j\leq m}$ simulated independently according to the law of (τ, ξ), the conditional expectation $\mathbb{E}(\xi|\tau)$ may thus be simulated by linear regression of $(\xi^j)^{1\leq j\leq m}$ against $(\varphi^l(\tau^j))_{1\leq l\leq \rho}^{1\leq j\leq m}$, where the truncation order ρ is a parameter of the method. The computational cost of this regression is $\mathrm{O}(m\rho^2)$ to form the regression matrix, plus the time of solving a linear system of size ρ.

We refer the interested reader to the monograph by Györfi et al. [21] for every detail about these simulation/regression approaches for computing a regression function

$$t \mapsto \mathbb{E}(\xi|\tau = t).$$

Note that there exist alternatives to nonlinear regression for computing conditional expectations by simulation, in particular Malliavin calculus based methods [22, 10] and quantization methods [25]. However the former are typically difficult to implement, and the latter suffer significantly the curse of dimensionality. We only use regression methods in this paper.

References

[1] Assefa, S., Bielecki, T. R., Crépey, S. and Jeanblanc, M.: CVA computation for counterparty risk assessment in credit portfolios. In *Recent advancements in theory and practice of credit derivatives*, Bielecki, T. R., Brigo, D. and Patras, F., eds, Wiley, 2011.

[2] Bielecki, T.R. , Cousin, A., Crépey, S. and Herbertsson, A.: Dynamic Modeling of Portfolio Credit Risk with Common Shocks. *Submitted*.

[3] Bielecki, T.R., Cousin, A., Crépey, S., Herbertsson, A.: A Markov copula model of portfolio credit risk with stochastic intensities and random recoveries. *Submitted*.

[4] Bielecki, T. R. and Crépey, S.: Dynamic Hedging of Counterparty Exposure. Forthcoming in *The Musiela Festschrift*, Zariphopoulou, T., Rutkowski, M. and Kabanov, Y., eds, Springer, 2012.

[5] Bielecki, T. R., Crépey, S. and Herbertsson, A.: Markov Chain Models of Portfolio Credit Risk. In *Oxford Handbook of Credit Derivatives*, Lipton, A. and Rennie, A., eds, Oxford Handbooks in Finance, 2011.

[6] Bielecki, T.R., Crépey, S. and Jeanblanc, M.: Up and Down Credit Risk. *Quantitative Finance* 10 (10), pp. 1137–1151, 2010.

[7] Bielecki, T.R., Jakubowskib, J. and Nieweglowskic, M.: Study of dependence for some stochastic processes: Symbolic Markov copulae, *Stochastic Processes and their Applications* 122 (3), pp. 930–951, 2012.

[8] Bielecki, T.R., Jakubowski, J., Vidozzi, A. and Vidozzi, L.: Study of Dependence for Some Stochastic Processes. *Stochastic analysis and applications*, vol. 26, no4, p. 903-924, 2008.

[9] Bielecki, T.R., Vidozzi, A. and Vidozzi, L.: A Markov Copulae Approach to Pricing and Hedging of Credit Index Derivatives and Ratings Triggered Step–Up Bonds. *Journal of Credit Risk* 4 (1), pp. 47–76, 2008.

[10] Bouchard B., Ekeland I. and Touzi N.: The Malliavin approach to Monte Carlo approximations to conditional expectations. *Finance and Stochastics*, Volume 8, Number 1, 45–71, 2004.

[11] Brigo, D. and Capponi, A.: Bilateral counterparty risk with application to CDSs. *Risk Magazine* March 2010, pp. 85–90.

[12] Pallavicini, A., Perini, D. and Brigo, D.: Funding Valuation Adjustment: a consistent framework including CVA, DVA, collateral, netting rules and rehypothecation. *Working paper*, 2011.

[13] Carmona, R. and Crépey, S.: Importance Sampling and Interacting Particle Systems for the Estimation of Markovian Credit Portfolios Loss Distribution, *International Journal of Theoretical and Applied Finance*, Vol 13, Issue 4, p. 577-602, 2010.

[14] Cesari G., Aquilina J. and Charpillon N.: *Modelling, Pricing, and Hedging Counterparty Credit Exposure*. Springer Finance, 2010.

[15] Crépey, S.: Wrong way and gap risks modeling: a marked stopping times pespective. *In Preparation*.

[16] Crépey, S.: Bilateral Counterparty Risk under Funding Constraints - Part I: Pricing. *Mathematical Finance* online first January 2013.

[17] Crépey, S.: Bilateral Counterparty Risk under Funding Constraints - Part II: CVA. *Mathematical Finance* online first January 2013.

[18] Crépey, S., Jeanblanc, M. and Zargari, B.: Counterparty Risk on a CDS in a Markov Chain Copula Model with Joint Defaults. In *Recent Advances in Financial Engineering 2009*, Kijima, M., Hara, C., Muromachi, Y. and Tanaka, K. (eds.), World Scientific, pp. 91–126, 2010.

[19] Crépey, S. and Rahal, A.: Pricing Convertible Bonds with Call Protection. *Journal of Computational Finance*, Vol.15 Num. 2, 37-75, Winter 2011/12.

[20] Cont, R., Minca, A.: Recovering Portfolio Default Intensities Implied by CDO Quotes. *Mathematical Finance* online first June 2011.

[21] Györfi, L., Kohler, M., Krzyzak, A., Walk, H.: *A Distribution-Free Theory of Nonparametric Regression*, Springer, 2002.

[22] Lions, P.-L., Regnier, H.: Calcul du prix et des sensibilités d'une option américaine par une méthode de Monte Carlo, *Working Paper*, 2001.

[23] Longstaff, F.A., Schwartz, E.S.: Valuing American Options by simulations: A Simple Least-Squares Approach, *Review of Financial Studies*, Volume 14, Number 1, 2001, p. 113-147.

[24] Marshall, A. and Olkin, I.: A multivariate exponential distribution, *J. Amer. Statist. Assoc.*, 2, 84-98, 1967.

[25] Pagès, G. and Bally, V.: A quantization algorithm for solving multidimensional discrete-time optimal stopping problems . *Bernoulli*, Volume 9, Number 6, 1003-1049, 2003.

[26] Schönbucher, P.: Credit Derivatives Pricing Models: Model, Pricing and Implementation, *Wiley Finance*, 2003.

[27] Tsitsiklis, J. and Van Roy, B.: Regression methods for Pricing complex American-style Options, *IEEE Transactions On Neural Networks*, Vol. 12, No. 4, July 2001.