HAL open science

Toward a coherent Monte Carlo simulation of CVA

Lokman Abbas-Turki, Ayech Bouselmi, Mohammed Mikou

To cite this version:

Lokman Abbas-Turki, Ayech Bouselmi, Mohammed Mikou. Toward a coherent Monte Carlo simulation of CVA. 2013. hal-00873200v2

HAL Id: hal-00873200

https://hal.science/hal-00873200v2

Preprint submitted on 12 Dec 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Toward a coherent Monte Carlo simulation of CVA

L. A. Abbas-Turki[*], **A. I. Bouselmi**[†] and **M. A. Mikou**[‡]

[*]TU Berlin, Stochastik und Finanzmathematik,

Building MA, Str. des 17. Juni 136, 10623 Berlin, Germany

[†]Université Paris-Est, Laboratoire d'Analyse et de Mathématiques Appliquées,

Champs-sur-Marne, 77454 Marne-la-Vallée Cedex2, France

[‡] EISTI, Laboratoire de Mathématiques,

avenue du Parc, 95011 Cergy-Pontoise Cedex, France

Abstract

This paper is devoted to the simulation of the Credit Valuation Adjustment (CVA) using a pure Monte Carlo technique with Malliavin Calculus (MCM). The procedure presented is based on a general theoretical framework that includes a large number of models as well as various contracts, and allows both the computation of CVA and its sensitivity with respect to the different assets. Moreover, we provide the expression of the backward conditional density of assets vector that can be simulated off-line in order to reduce the variance of the CVA estimator. Using the suitability of MCM to parallel architectures and thus to a Graphic Processing Unit (GPU) implementation, we show that the results obtained are accurate once sufficient number of trajectories are simulated. Both complexity and accuracy are studied for MCM and regression methods and compared to the square Monte Carlo benchmark.

1 Introduction

After the 2007 economic crisis, several laws were issued for better financial regulation. Among the most important measures are those taken at Basel III that include the calculation of the CVA (Credit Valuation Adjustment) as an important part of the prudential rules. In a financial transaction between a party A that has to pay another party B some amount V, the CVA value is the price of the insurance contract that covers the default of party A to pay the whole sum V. In other words, in the absence of arbitrage opportunities, the CVA is the value of liquid products that must be saved to deal with counterparty default (see [13, 14]).

Formally speaking, the CVA is given by the following equality

$$\mathrm{CVA}_{t,T} = (1-R)E_t\left(V_\tau^+ 1_{t<\tau\leq T}\right), \tag{1}$$

where R (assumed equal to zero in this paper) is the recovery to make on the portfolio if the counterparty defaults, E_t denotes the conditional expectation knowing all the available information at t, V_t is the process of the value exposure to the counterparty, τ is the random

default time of the counterparty, T is the protection time horizon and the positive part function is denoted either by $^+$ or $_+$.

What makes difficult the numerical approximation of (1) is the expression of V_t that does not only includes assets, but also European contracts and American contracts. Said differently, once the stochastic model of the assets is fixed, one needs to simulate contracts before simulating the value of the CVA. The approximation of (1) is performed through three steps: First, simulating the assets $S_t = (S_t^1, ..., S_t^d)$ trajectories, then simulating the contracts trajectories to get V_t trajectories as a sum over the whole exposure:

$$V_t = \sum_{ie} \phi_{ie}^{exp}(S_t) + \sum_{ii} \phi_{ii}^{eui}(S_t) + \sum_{id} \phi_{id,t}^{eud}(S_t) + \sum_{ia} \phi_{ia,t}^{am}(S_t), \tag{2}$$

where ie, ii, id and ia are the exposure indices and:

ϕ^{exp} is an explicit function that represents pure assets transaction, for example: $\phi^{exp}(S_{t_k}) = S_{t_k}^1 - S_{t_k}^2$.

ϕ^{eui} is a path-independent European contract. It is a contract involving only the simulation of the assets S_t at $t = T$ and whose expression is given thanks to an explicit payoff function f^{eui} by

$$\phi^{eui}(S_{t_k}) = E(f^{eui}(S_T)|S_{t_k}), \text{ with } t_k \in [0, T], \tag{3}$$

for example: $f^{eui}(S_T) = (S_T^1 - S_T^2)_+$.

ϕ_t^{eud} is a path-dependent European contract. It is a contract involving the simulation of the whole discretized path of S_t at $t \in \{t_0, t_1, ..., T\}$ and whose expression is given thanks to a path-dependent payoff function f_t^{eud}, at each time t_k by

$$\phi_{t_k}^{eud}(S_{t_k}) = E(f_{t_k}^{eud}(S_{t_{k+1}})|S_{t_k}), \tag{4}$$

for example: $f_{t_k}^{eud}(S_{t_{k+1}}) = (\max_{i=0,..,k} S_{t_i}^1 \vee S_{t_{k+1}}^1 - S_{t_{k+1}}^2)_+$ where $\vee$ is the maximum operator. In the previous example, the dependence of the payoff according to the information available at time t_k is illustrated by $\max_{i=0,..,k} S_{t_i}^1$.

ϕ_t^{am} is an American contract. It is a contract that depends on the assets path through an optimal stopping problem implemented by the dynamic programming algorithm

$$\phi_{t_k}^{am}(S_{t_k}) = f(S_{t_k}) \vee E(\phi_{t_{k+1}}^{am}(S_{t_{k+1}})|S_{t_k}) \tag{5}$$

with f an explicit payoff that generally does not depend on the asset path.

Without loss of generality, assuming $t = 0$ and $R = 0$ in (1), the last step of approximating $\text{CVA}_{0,T}$ is based on a time discretization, to get

$$\text{CVA}_{0,T} \approx \sum_{k=0}^{N-1} E\left(V_{t_k}^+ 1_{\tau \in (t_k, t_{k+1}]}\right) \tag{6}$$

and N must be smaller or equal to the number of time steps used to approximate the trajectories of the assets.

The last point that has to be specified is the model used for the default time τ and how it should be related to the V_t dynamics. In this paper, the two main families of modeling default time are studied: i) The reduced form family, ii) The structural family. For each family, a specific model involving dependence between the exposure and the default time (so called WWR/RWR) is considered with its associated expressions of the computation of $\text{CVA}_{0,T}$ and $\partial_{S_0^i}\text{CVA}_{0,T}$, where S_0^i is the ith spot price. The latter quantities will be expressed only as a function of the exposure and its gradient vector.

The common point between the expressions (3), (4) and (5) is the computation of a conditional expectation that should be simulated before the approximation (6). In this paper, we reexpress both the conditional expectation involved and its gradient using Malliavin Calculus. Denoting $E(f(S_{t_{k+1}})|S_{t_k}) = \varphi(S_{t_k})$, we aim at computing

$$\varphi(x) = E(f(S_{t_{k+1}})|S_{t_k} = x) \text{ and } \partial_{x_i}\varphi(x) \text{ with } x_i \in \{x_1, ..., x_d\}. \tag{7}$$

References [15, 20] were first to propose the use of Malliavin Calculus respectively to express the conditional expectation in finance and to employ it in American contract simulation. While [19, 7] were first to study deeply the use of localization technique with Malliavin Calculus, respectively in one dimension and in the multidimensional case, and its application for density simulation and optimal portfolio selection. The results presented in this paper are more related to [2]. In fact, in the latter reference, the results presented go beyond Markov diffusions setting and it can be applied to a large family of multidimensional stochastic volatility models. Using Malliavin Calculus, we rewrite (7) as a quotient of expectations that can be simulated by Monte Carlo. Moreover, we express the backward conditional density and its gradient as a quotient of expectations that can be simulated by Monte Carlo. When this density and its gradient are known off-line for some trajectories (after discretization), the values given in (7) can be efficiently approximated for any path-dependent or path-independent function f.

One could wonder why it is more advantageous to use Malliavin Calculus with Monte Carlo simulation (MCM) when compared to the direct use of a square[1] Monte Carlo simulation. We will see that MCM is justified by its computational complexity which is smaller than square Monte Carlo, in particular for the CVA that involves American contracts. Moreover, in contrast to regression methods used in [14], we will justify that MCM is a nonparametric method whose accuracy depends only on the number of simulated trajectories. Consequently, one can increase "indefinitely" the accuracy by simulating more trajectories and using more computational resources thanks to an efficient parallel implementation. Besides, MCM does not have the drawback of using different regression basis when different contracts are involved.

The easy adaptability to various models is a key advantage of our procedure, then we present in Section 2 a brief summary of the different models that can be used. In Section 3, we provide, on the one hand, the expression of the conditional expectation (7) as well as its partial derivative and we introduce, on the other hand, the value of the backward conditional density that can be simulated off-line to speedup the convergence of the CVA estimator. In Section 4, we compare the computational complexity of MCM with the method that involves regressions and with the square Monte Carlo method used as a benchmark. In addition to

[1]This is a two levels Monte Carlo simulation, one for simulating assets trajectories and another one for simulating contracts trajectories.

the MCM general framework, its adaptability to parallel architecture and its good accuracy[2] studied in Section 5 suffice to justify its use as a real alternative to regression methods.

2 Model families

In this Section, we present the general modelling framework of asset prices and counterparty default for which MCM could be applied. We also express $\mathrm{CVA}_{0,T}$ and $\partial_{S_0^i}\mathrm{CVA}_{0,T}$ using V_t and $\partial_{S_0^i}V_t$ whose values are given by (2) and

$$\begin{aligned}\partial_{S_0^i}V_t = \sum_{ie}\partial_{S_0^i}\phi_{ie}^{exp}(S_t) \quad &+ \quad \sum_{ii}\partial_{S_0^i}\phi_{ii}^{eui}(S_t) \quad + \quad \sum_{id}\partial_{S_0^i}\phi_{id,t}^{eud}(S_t) \\ &+ \quad \sum_{ia}\partial_{S_0^i}\phi_{ia,t}^{am}(S_t).\end{aligned} \tag{8}$$

The value of each ϕ term in (2) and its derivative in (8) has the general formulation (7) given by φ and its derivative which are both computed thanks to the results presented in Section 3. Without loss of generality, we remind that we assume $R = 0$ in (1).

Let T be the protection time horizon, $(\Omega, \mathcal{F}, P)$ a probability space on which we define a d-dimensional standard Brownian motion $W = (W^1, ..., W^d)$ and $\mathbb{F} = \{\mathcal{F}_s\}_{s\leq T}$ the P-completion of the filtration generated by W until T. We denote by S_t the vector of asset prices $S_t^1, ..., S_t^d$. which are the solutions of the following stochastic differential equations

$$\frac{dS_t^i}{S_t^i} = r_i dt + \sum_{j=1}^{i}\sigma_{ij}(t)dW_t^j, \quad S_0^i = z_i, \quad i = 1, .., d, \tag{9}$$

where r_i are constants and $\sigma(t) = \{\sigma_{ij}(t)\}_{1\leq i,j\leq d}$ is a deterministic triangular matrix ($\{\sigma_{ij}(t)\}_{i<j} = 0$). We suppose that the matrix $\sigma(t)$ is invertible, bounded and uniformly elliptic which ensures the existence of the inverse matrix $\rho(t) = \sigma^{-1}(t)$ and its boundedness. Dynamics (9) is widely used for equity models, HJM interest rate models and variance swap models. One should note that in the case where the dynamics of S is given by local volatility model, we can use a discretization scheme to reduce it to an SDE of type (9) on subintervals. The methodology developed in Section 3 can be extended to jump diffusion and stochastic volatility models, Indeed:

i) We can replace (9) by the following SDE

$$\frac{dS_t^i}{S_t^i} = r_i dt + \sum_{j=1}^{i}\sigma_{ij}(t)dW_t^j + dJ_t^i, \quad S_0^i = z_i, \quad i = 1, .., d,$$

where $J = (J^1, ..., J^d)$ is a jump process independent from W. Then the conditional expectation in (7) is given by

$$\varphi(x) = E\left(E[f(S_{t_{k+1}})|\sigma\left((J_u)_{0\leq u\leq t}\right), S_{t_k} = x]|S_{t_k} = x\right), \; x = (x_1, ..., x_d). \tag{10}$$

The computations performed in Section 3 can be implemented to the inner expectation in (10).

[2]Demonstrated here at least till dimension $d = 3$.

ii) We can replace (9) by the following stochastic volatility model,

$$\frac{dS_t^i}{S_t^i} = r_i dt + \sum_{j=1}^{i} \sigma_{ij}(t, \widetilde{W}_t) dW_t^j, \quad S_0^i = z_i, \quad i = 1, .., d,$$

where $\widetilde{W}$ is a multidimensional Brownian motion correlated to W as it is done in [2]. Then the conditional expectation in (7) is given by

$$\varphi(x) = E\left(E[f(S_{t_{k+1}})|\sigma\left((\widetilde{W}_u)_{0\leq u\leq t}\right), S_{t_k} = x]|S_{t_k} = x\right), \; x = (x_1, ..., x_d) \tag{11}$$

and the inner expectation in (11) and its partial derivative according to each x_i can be computed as explained in Section 3.

In addition to the large number of asset models that can be used, when assuming independence between V_u and τ in (1), one has a wide choice of the counterparty default time distribution $P_\tau(du)$. In fact, under the independence assumption, we have

$$\text{CVA}_{0,T} = \int_0^T E\left(V_u^+|\tau\right) P_\tau(du) = \int_0^T E\left(V_u^+\right) P_\tau(du),$$

$$\partial_{S_0^i}\text{CVA} = \int_0^T E\left(\partial_{S_0^i} V_u 1_{\{V_u>0\}}\right) P_\tau(du).$$

The permutation of the differentiation $\partial_{S_0^i}$ and the expectation is possible thanks to Remark 3.1 ii).

In the following, we consider CVA models involving WWR or RWR, this implies that V_u and τ are no longer assumed to be independent. ISDA (the International Swaps and Derivatives Association) defines Wrong Way Risk as the risk that occurs when the "exposure to a counterparty is adversely correlated with the credit quality of that counterparty", when the Right Way Risk (RWR) refers to the opposite correlation. Consequently, the choice of the counterparty default model will influence the CVA and $\partial_{S_0^i}$CVA expressions.

Using the literature [6, 5, 12, 13], we distinguish two main ways to model the default time: i) The structural family (firm value) and ii) The reduced form (intensity) family. However, as pointed out by the authors of [18], there is no standard way to specify the dependence between the counterparty default and the exposure. Subsequently, we will only give an example for each default model with its CVA expressions.

2.1 CVA intensity models including WWR/RWR

In these models, we assimilate the default time as the first jumping time of a Poisson process, where we denote by λ its intensity. We point out that λ can be even considered as deterministic, either constant or time dependent, and even stochastic like in Cox model. If λ is deterministic, we have $P(\tau > \theta) = e^{-\lambda\theta}$ and more precisely

$$P(\tau \in (t, t+dt]|\tau > t) = \frac{P(\tau \in (t, t+dt])}{P(\tau > t)} = \lambda dt.$$

The same formula is true when λ_t is $\mathcal{F}_t$-adapted by conditioning with respect to $\mathcal{F}_t$. Defined by

$$\Lambda(t) = \int_0^t \lambda_s ds,$$

the function Λ is commonly known as the hazard function or cumulated intensity and λ represents the intensity or hazard rate.
When Λ is deterministic[3], the Poisson process properties imply that $\Lambda(\tau)$ is an exponential random variable with $E(\Lambda(\tau)) = 1$. Notice also that $\Lambda(\tau)$ is independent from the default-free market information $\mathbb{F}$. Thus we obtain,

$$P(\tau > t) = P(\Lambda(\tau) > \Lambda(t)) = e^{-\Lambda(t)} = e^{-\int_0^t \lambda_s ds}.$$

If the hazard function is stochastic, we get

$$P(\tau > t) = E\left(e^{-\int_0^t \lambda_s ds}\right).$$

Specific example with its CVA and CVA sensitivity estimation
We present the model proposed in [18], involving WWR or RWR. The intensity model is assumed to have a stochastic $\mathcal{F}_t$-adapted hazard rate λ_t. Particularly, we suppose that

$$\lambda(t) = f(t, V_t),$$

where f is some "known" positive function and V_t represents the exposure at t. In some cases, one can take

$$\lambda(t) = g(t, S_t),$$

where S_t is the $\mathbb{R}^d$ process describing the underlying asset prices. In [18], the authors assume that either

$$\lambda(t) = \exp\left(a(t) + bV_t\right) \text{ or } \lambda(t) = \ln\left[1 + \exp\left(a(t) + bV_t\right)\right],$$

where $a(t)$ is a deterministic function that is useful for the calibration and b represents the dependence (WWR or RWR). In our work, we test a simpler model that takes into account only WWR and given by

$$\lambda_t = \alpha'' + \alpha'(V_t)_+, \text{ with } \alpha' \geq 0 \text{ and } \alpha'' \geq 0. \tag{12}$$

Assuming that we are able to estimate the process V_t on a time discretized grid $\{t_0 < t_1 < t_2 < .. < t_n = T\}$ (see Section 3), the value of the $\text{CVA}_{0,T}$ will be given by

$$\text{CVA}_{0,T} = E\left(\sum_{k=0}^{n-1} V_{t_{k+1}}^+ P\left(\tau \in (t_k, t_{k+1}] | \mathcal{F}_{t_{k+1}}\right)\right). \tag{13}$$

[3]If Λ is $\mathcal{F}_t$-adapted, we obtain the same result by conditioning with respect to $\mathcal{F}_t$.

The sensitivity of the $\mathrm{CVA}_{0,T}$ according to the ith spot price S_0^i is as follows,

$$\begin{aligned}\partial_{S_0^i}\mathrm{CVA}_{0,T} &= E\left(\sum_{k=0}^{n-1}\partial_{S_0^i}V_{t_{k+1}}1_{\{V_{t_{k+1}}>0\}}P(\tau\in(t_k,t_{k+1}]|\mathcal{F}_{t_{k+1}})\right)\\ &\quad +E\left(\sum_{k=0}^{n-1}V^+_{t_{k+1}}\partial_{S_0^i}P(\tau\in(t_k,t_{k+1}]|\mathcal{F}_{t_{k+1}})\right).\end{aligned}$$

The permutation of the operator $\partial_{S_0^i}$ and the expectation is justified in Remark 3.1 ii). Regarding the derivative $\partial_{S_0^i}P(\tau\in(t_k,t_{k+1}])$, we have

$$\begin{aligned}\partial_{S_0^i}P(\tau\in(t_k,t_{k+1}]|\mathcal{F}_{t_{k+1}}) &= \partial_{S_0^i}\left(P(\tau>t_k|\mathcal{F}_{t_{k+1}})-P(\tau>t_{k+1}|\mathcal{F}_{t_{k+1}})\right)\\ &= \partial_{S_0^i}\left(e^{-\int_0^{t_k}f(s,V_s)ds}-e^{-\int_0^{t_{k+1}}f(s,V_s)ds}\right).\end{aligned}$$

Using the chain rule and the expression of the hazard rate given in (12), this derivative becomes

$$\begin{aligned}\partial_{S_0^i}P(\tau\in(t_k,t_{k+1}]|\mathcal{F}_{t_{k+1}}) &= -P(\tau>t_k|\mathcal{F}_{t_{k+1}})\int_0^{t_k}\partial_{V_s}f(s,V_s)\partial_{S_0^i}V_s ds\\ &\quad +P(\tau>t_{k+1}|\mathcal{F}_{t_{k+1}})\int_0^{t_{k+1}}\partial_{V_s}f(s,V_s)\partial_{S_0^i}V_s ds\\ &= -P\left(\tau\in(t_k,t_{k+1}]|\mathcal{F}_{t_{k+1}}\right)\int_0^{t_k}\partial_{V_s}f(s,V_s)\partial_{S_0^i}V_s ds\\ &\quad +P(\tau>t_{k+1}|\mathcal{F}_{t_{k+1}})\int_{t_k}^{t_{k+1}}\partial_{V_s}f(s,V_s)\partial_{S_0^i}V_s ds\\ &= -P\left(\tau\in(t_k,t_{k+1}]|\mathcal{F}_{t_{k+1}}\right)\int_0^{t_k}\alpha'\partial_{S_0^i}V_s 1_{\{V_s>0\}}ds\\ &\quad +P(\tau>t_{k+1}|\mathcal{F}_{t_{k+1}})\int_{t_k}^{t_{k+1}}\alpha'\partial_{S_0^i}V_s 1_{\{V_s>0\}}ds.\end{aligned}$$

Both V_t and $\partial_{S_0^i}V_t$ are provided in (2) and (8).

2.2 CVA structural models including WWR/RWR

First introduced by Merton [21], the default time in these models is defined according to the behavior of the positive firm value process $(X_t)_{t\geq 0}$. Merton's example assumes that default occurs if, at the final time T, the firm value X_T is below a given threshold L which generally represents a promised terminal payoff.
Inspired by this model, Black and Cox proposed to modelize the default time by

$$\tau=\inf\{t\geq 0 \quad | \quad X_t\leq L_t\}$$

where

$$L_t=\begin{cases} e^{-\gamma(T-t)}K & \text{if} \quad t<T\\ L & \text{if} \quad t=T\end{cases}$$

with γ smaller than the risk neutral interest rate and $K \leq L$. In this situation, the "critical" threshold L_t must not be crossed by the firm value process. For more details, we refer the reader to [11]. In the structural model presented in Section 5, we will assume that L_t is constant.

Specific example with its CVA and CVA sensitivity estimation
The dependence between the default time variable τ and the exposure V_t is modelized thanks to the correlation between some Brownian motion W_t^0 that drives the process X_t and $\left(W_t^1, .., W_t^d\right)$ which drive the asset prices S_t.
Thus, $\mathrm{CVA}_{0,T}$ is given by

$$\begin{aligned} \mathrm{CVA}_{0,T} &= E\left(\sum_{k=0}^{n-1} V_\tau^+ 1_{\{\tau\in(t_k,t_{k+1}]\}}\right) \\ &\approx E\left(\sum_{k=0}^{n-1} V_{t_k}^+ 1_{\{\tau\in(t_k,t_{k+1}]\}}\right). \end{aligned} \tag{14}$$

The sensitivity of $\mathrm{CVA}_{0,T}$ according to the ith spot price S_0^i is given by

$$\begin{aligned} \partial_{S_0^i}\mathrm{CVA}_{0,T} &= \sum_{k=0}^{n-1} \partial_{S_0^i} E\left(V_{t_k}^+ | \tau\in(t_k,t_{k+1}]\right) P(\tau\in(t_k,t_{k+1}]) \\ &\quad + \sum_{k=0}^{n-1} E\left(V_{t_k}^+ | \tau\in(t_k,t_{k+1}]\right) \partial_{S_0^i} P(\tau\in(t_k,t_{k+1}]) \\ &= \sum_{k=0}^{n-1} E\left(\partial_{S_0^i} V_{t_k} 1_{\{V_{t_k}>0\}} 1_{\{\tau\in(t_k,t_{k+1}]\}}\right) \end{aligned}$$

which is ensured by the assumption $\partial_{S_0^i} P(\tau\in(t_k,t_{k+1}]) = 0$ and Remark 3.1 ii) that allows the permutation of the operator $\partial_{S_0^i}$ and the expectation.

Both V_t and $\partial_{S_0^i} V_t$ are provided in (2) and (8). Using the same argument presented in (10) and (11), the dependence according to τ is not an important issue for computations performed in Section 3. In fact, the conditional expectation (7) is equal to

$$\varphi(x) = E\left(E[f(S_{t_{k+1}})|\sigma\left((W_u^0)_{0\leq u\leq t}\right), S_{t_k}=x]|S_{t_k}=x\right),\ x=(x_1,...,x_d) \tag{15}$$

and the inner expectation can be computed as if the trajectory of $\{X_u\}_{0\leq u\leq t}$ is completely known. For more details, we refer the reader to Section 5 in which a more specific example is presented.

3 Computing the value exposure, its sensitivity and the backward conditional density

Estimating the value exposure to the counterparty V_t is crucial in the CVA computation. In order to calculate V_t using (2), one has to express the conditional expectation involved

in each contract. Using Malliavin calculus for American contracts pricing, this conditional expectation was expressed as a ratio of two expectations (see for example [2, 3]). We aim here to adapt the previous results to the CVA problem. Moreover, we give an explicit formulation of the sensitivity with respect to the initial value of the stock price. In Section 3.2, we will be interested by rather a theoretical result that provides the value of the backward conditional density of the process (9) and of multidimensional stochastic volatility and jump diffusion models that extends (9). The backward transition probability does not depend on the payoff, then it should be computed off-line and stored to be re-used, in the same fashion as it is done in [4, 22].

In this section we suppose that the stock price S is given by (9). To simplify the notations, we denote $H_i^x(S_s^i) = H(S_s^i - x_i) := 1_{S_s^i \geq x_i}$ for the Heaviside function of the difference between the ith stock and the ith coordinate of the positive vector x. Throughout this article, we assume that $g \in \mathcal{E}_b(\mathbb{R}^d)$ is a measurable function with polynomial growth

$$\mathcal{E}_b(\mathbb{R}^d) = \Big\{ f \in \mathcal{M}(\mathbb{R}^d) \ : \ \exists C > 0 \text{ and } m \in \mathbb{N};\ |f(y)| \leq C(1 + |y|_d)^m) \Big\} \tag{16}$$

where $\mathcal{M}(\mathbb{R}^d)$ is the set of measurable functions on $\mathbb{R}^d$ and $|\cdot|_d$ is the euclidean norm. The elements of the set $\mathcal{E}_b(\mathbb{R}^d)$ satisfy the finiteness of the expectations computed in this article. Besides, we usually use Malliavin derivative D_u^j for the differentiation with respect to the jth Brownian motion.

3.1 The conditional expectation value and its gradient

We have already seen that V_t and $\partial_{S_0^i} V_t$ are given by (2) and (8) where the value of each contract is expressed using (3), (4) and (5). The only point that remains to be specified is the conditional expectation and its partial derivative in (7). Theorem 3.1 deals with the latter issue, but before that we need to introduce some definitions.

Definition 3.1 *We define the random variable $\Gamma_{s,t} = \Gamma_{s,t}^1$ and $\Gamma_{s,t}^1$ can be computed by the following induction scheme*

$$\Gamma_{s,t}^d = \pi_{s,t}^{d,d},\ \Gamma_{s,t}^k = \Gamma_{s,t}^{k+1}\pi_{s,t}^{k,d} - \sum_{j=k+1}^{d} \int_0^t D_u^j \Gamma_{s,t}^{k+1} D_u^j \pi_{s,t}^{k,d} du,\ \ k \in \{1, ..., d-1\},$$

where $\pi_{s,t}^{k,d}$ is given by

$$\pi_{s,t}^{k,d} = 1 + \sum_{j=k}^{d} \int_0^t \varphi_{jk}(u) dW_u^j,\ \varphi_{jk}(u) = \frac{1}{s}\rho_{jk}(u) 1_{u \in (0,s)} - \frac{1}{t-s}\rho_{jk}(u) 1_{u \in (s,t)}.$$

with ρ is the inverse of the volatility matrix σ.

Theorem 3.1 *For any $s \in (0,t)$, $g \in \mathcal{E}_b(\mathbb{R}^d)$ and $x = (x_1, ..., x_d)$ with $x_i > 0$,*

$$E\left(g(S_t) \Big| S_s = x \right) = \frac{T_{s,t}[g](x)}{T_{s,t}[1](x)}, \tag{17}$$

and its partial derivative

$$\partial_{x_i} E\left(g(S_t)\Big| S_s = x\right) = \frac{R^i_{s,t}[g](x)T_{s,t}[1](x) - T_{s,t}[g](x)R^i_{s,t}[1](x)}{T_{s,t}[1](x)^2}, \tag{18}$$

where $T_{s,t}[f](x)$ and $R^i_{s,t}[f](x)$ are defined for every function[4] *$f \in \mathcal{E}_b(\mathbb{R}^d)$ by*

$$T_{s,t}[f](x) = E\Big(f(S_t)\Gamma_{s,t}\widehat{H}^x(S_s)\Big), \tag{19}$$

$$R^i_{s,t}[f](x) = -E\Big(\frac{f(S_t)}{S^i_s}\widehat{H}^x(S_s)\Big(\Gamma_{s,t}(1+\pi^{i,d}_{s,t}) - \sum_{j=i}^{d}\int_0^t D^j_u \pi^{i,d}_{s,t} D^j_u \Gamma_{s,t} du\Big)\Big), \tag{20}$$

with $\widehat{H}^x(S_s) = \prod_{k=1}^{d} \frac{H^x_k(S^k_s)}{S^k_s}$, $\Gamma_{s,t}$ and $\pi^{k,d}_{s,t}$ are given in Definition 3.1.

$H^x_k(S^k_s)$ is the Heaviside function of the difference between the kth stock and the kth coordinate of the positive vector x, $\mathcal{E}_b(\mathbb{R}^d)$ is defined in (16).

Using Theorem 3.1, the conditional expectation in (7) and its derivative are given by (17) and (18). To prove Theorem 3.1, we need the following two lemmas which are proved in [2]. It follows from Lemma 3.1 that the sum $\sum_{j=i}^{d} \rho_{ji}(u) D^j_u f(S_t)$ does not depend on u.

Lemma 3.1 *For any $u \in (0,t)$, $f \in \mathcal{C}^1(\mathbb{R}^d)$ and S given by the SDE (9), we have*

$$\sum_{j=i}^{d} \rho_{ji}(u) D^j_u f(S_t) = S^i_t \partial_{x_i} f(S_t). \tag{21}$$

The second lemma is based on the duality property of the Malliavin calculus.

Lemma 3.2 *For any interval $I \subset (0,t)$, $h \in \mathcal{C}^\infty_b(\mathbb{R})$, $F \in Dom(D)$ and S given by the SDE (9), we have*

$$\begin{aligned} E\left(\int_I \frac{F D^i_u h(S^i_s)}{\sigma_{ii}(u)} du\right) &= E\Big(h(S^i_s) F \sum_{j=i}^{d} \int_I \rho_{ji}(u) dW^j_u\Big) \\ &- E\Big(h(S^i_s) \sum_{j=i}^{d} \int_I \rho_{ji}(u) D^j_u F du\Big). \end{aligned} \tag{22}$$

<u>Proof of Theorem 3.1</u>: The equalities (17) and (19) are proved in [2], the new result of this theorem is the partial derivative value (18). Regarding this part, it is sufficient to prove that

$$\partial_{x_i} T_{s,t}[f](x) = R^i_{s,t}[f](x).$$

[4] In our case $f = g$ or $f = 1$

Let $\phi \in \mathcal{C}_c^\infty(\mathbb{R})$ be a mollifier function with support equal to $[-1,1]$ and such that $\int_{\mathbb{R}} \phi(u)du = 1$, then for any $u \in \mathbb{R}$ we define

$$h_{mi}(u) = (H_i^x * \phi_m)(u) \in \mathcal{C}_b^\infty(\mathbb{R}), \quad \phi_m(u) = m\phi(mu), \quad m \in \mathbb{N}.$$

The dependence with respect to x_i can be dominated and the differentiation under the integral sign provides

$$\partial_{x_i} E\left(f(S_t)\Gamma_{s,t}\frac{h_{mi}(S_s^i)}{S_s^i}\widehat{H}^i(S_s)\right) = -E\left(f(S_t)\Gamma_{s,t}\frac{h'_{mi}(S_s^i)}{S_s^i}\widehat{H}^i(S_s)\right), \tag{23}$$

where $\widehat{H}^i(S_s) = \prod_{k=1;k\neq i}^{d} \frac{H_k^x(S_s^i)}{S_s^i}$.

Under our assumptions, the distribution of the vector $(S_s^1, ..., S_s^d, S_t^1, ..., S_t^d)$ admits a lognormal joint distribution density $p_{s,t}$ with respect to the Lebesgue measure on $\mathbb{R}_+^d \times \mathbb{R}_+^d$. Similar to the argument presented in proof of Theorem 2.1 in [2], using $p_{s,t}$ one gets the limit as $m \longrightarrow \infty$

$$\partial_{x_i} E\left(f(S_t)\Gamma_{s,t}\frac{h_{mi}(S_s^i)}{S_s^i}\widehat{H}^i(S_s)\right) \longrightarrow \partial_{x_i} T_{s,t}[f](x),$$

that provides

$$\partial_{x_i} T_{s,t}[f](x) = -\lim_{m\to+\infty} E\left(f(S_t)\Gamma_{s,t}\frac{h'_{mi}(S_s^i)}{S_s^i}\widehat{H}^i(S_s)\right). \tag{24}$$

We introduce the following notations

$$\Pi(S_s) = \frac{\widehat{H}^i(S_s)}{(S_s^i)^2}, \quad \widehat{h}_{mi}(S_s) = \frac{h_{mi}(S_s^i)}{S_s^i}\widehat{H}^i(S_s^i).$$

We have by the chain rule $h'_{mi}(S_s^i) = \frac{D_u^i h_{mi}(S_s^d)}{D_u^i S_s^i}$ and $D_u^i S_s^i = \sigma_{ii}(u)S_s^i$ for every $u \in (0,s)$, thus

$$\begin{aligned} E\left(f(S_t)\Gamma_{s,t}\frac{h'_{mi}(S_s^i)}{S_s^i}\widehat{H}^i(S_s)\right) &= E\left(\frac{1}{s}\int_0^s f(S_t)\Gamma_{s,t}\widehat{H}^i(S_s)\frac{D_u^i h_{mi}(S_s^i)}{S_s^i D_u^i S_s^i}du\right) \\ &= E\left(\frac{1}{s}\int_0^s f(S_t)\Gamma_{s,t}\widehat{H}^i(S_s)\frac{D_u^i h_{mi}(S_s^i)}{\sigma_{ii}(u)(S_s^i)^2}du\right). \end{aligned}$$

Using Lemma 3.2 with

$$F = f(S_t)\Gamma_{s,t}\frac{\widehat{H}^i(S_s)}{(S_s^i)^2} = f(S_t)\Gamma_{s,t}\Pi(S_s),$$

we get

$$E\left(f(S_t)\Gamma_{s,t}\frac{h'_{mi}(S^i_s)}{S^i_s}\widehat{H}^i(S_s)\right) = E\left(h_{mi}(S^i_s)F\frac{1}{s}\sum_{j=i}^{d}\int_0^s \rho_{ji}(u)dW^j_u\right)$$

$$- E\left(h_{mi}(S^i_s)\frac{1}{s}\sum_{j=i}^{d}\int_0^s \rho_{ji}(u)\left[\begin{array}{c}\Gamma_{s,t}\Pi(S_s)D^j_u f(S_t) + f(S_t)\Gamma_{s,t}D^j_u\Pi(S_s)\\ +f(S_t)\Pi(S_s)D^j_u\Gamma_{s,t}\end{array}\right]du\right)$$

$$= E\left(\frac{\widehat{h}_{mi}(S^i_s)}{S^i_s}f(S_t)\frac{1}{s}\Big(\Gamma_{s,t}(\sum_{j=i}^{d}\int_0^s \rho_{ji}(u)dW^j_u + 2s) - \sum_{j=i}^{d}\int_0^s \rho_{ji}(u)D^j_u\Gamma_{s,t}du\Big)\right)$$

$$- E\left(\frac{\widehat{h}_{mi}(S^i_s)}{S^i_s}\Gamma_{s,t}\frac{1}{s}\Big(\sum_{j=i}^{d}\int_0^s \rho_{ji}(u)D^j_u f(S_t)du\Big)\right), \tag{25}$$

since, using Lemma 3.1, we have

$$\sum_{j=i}^{d}\rho_{ji}(u)D^j_u\Pi(S_s) = S^i_s\partial_{x_i}\Pi(S_s) = -2\Pi(S_s).$$

Let us develop the last term in (25), using Lemma 3.1

$$E\left(\frac{\widehat{h}_{mi}(S^i_s)}{S^i_s}\Gamma_{s,t}\frac{1}{s}\Big(\sum_{j=i}^{d}\int_0^s \rho_{ji}(u)D^j_u f(S_t)du\Big)\right)$$

$$= E\left(\frac{\widehat{h}_{mi}(S^i_s)}{S^i_s}\frac{1}{t-s}\Big(\sum_{j=i}^{d}\Gamma_{s,t}\int_s^t \rho_{ji}(u)D^j_u f(S_t)du\Big)\right)$$

$$= E\left(\frac{\widehat{h}_{mi}(S^i_s)}{S^i_s}\frac{1}{t-s}\sum_{j=i}^{d}E\Big(\Gamma_{s,t}\int_s^t \rho_{ji}(u)D^j_u f(S_t)du\Big|\mathcal{F}_s\Big)\right)$$

$$= E\left(\frac{\widehat{h}_{mi}(S^i_s)}{S^i_s}\frac{1}{t-s}\sum_{j=i}^{d}E\Big(f(S_t)\Gamma_{s,t}\int_s^t \rho_{ji}(u)dW^j_u\Big|\mathcal{F}_s\Big)\right)$$

$$= E\left(\frac{\widehat{h}_{mi}(S^i_s)}{S^i_s}\frac{1}{t-s}\sum_{j=i}^{d}E\Big(f(S_t)(\Gamma_{s,t}\int_s^t \rho_{ji}(u)dW^j_u - \int_s^t \rho_{ji}(u)D^j_u\Gamma_{s,t}du)\Big|\mathcal{F}_s\Big)\right).$$

Thus, (25) becomes

$$E\left(f(S_t)\Gamma_{s,t}\frac{h'_{mi}(S^i_s)}{S^i_s}\widehat{H}^i(S_s)\right) \tag{26}$$

$$= E\left(\frac{\widehat{h}_{mi}(S^i_s)}{S^i_s}f(S_t)\Big(\Gamma_{s,t}(1+\pi^{i,d}_{s,t}) - \sum_{j=i}^{d}\int_0^t D^j_u\pi^{i,d}_{s,t}D^j_u\Gamma_{s,t}du\Big)\right).$$

Using a dominated convergence argument, from (24) and (26) we get

$$\partial_{x_i}T_{s,t}[f](x) = -E\left(\frac{f(S_t)}{S^i_s}\prod_{k=1}^{d}\frac{H^x_k(S^k_s)}{S^k_s}\Big(\Gamma_{s,t}(1+\pi^{i,d}_{s,t}) - \sum_{j=i}^{d}\int_0^t D^j_u\pi^{i,d}_{s,t}D^j_u\Gamma_{s,t}du\Big)\right).$$

■

Remark 3.1 *i) Its is* **important** *to point out that* $\Gamma_{s,t}$ *and* $\sum_{j=i}^{d}\int_0^t D_u^j \pi_{s,t}^{i,d} D_u^j \Gamma_{s,t} du$ *can be simulated efficiently using the trick given in [2] which will be detailed in Remark 3.2.*

ii) For every $g \in \mathcal{E}_b(\mathbb{R}^d)$*, one can show that the functions* $R_{s,t}^i[g]$ *and* $T_{s,t}[g]$ *are continuous since the Heaviside function is continuous except on the negligible set* $\{S_s = x\}$*. The latter fact implies that the partial derivative of the conditional expectation (18) is locally bounded when its payoff is in* $\mathcal{E}_b(\mathbb{R}^d)$*.*

3.2 Backward conditional density estimation

The authors of [22] proposed a Karhunen-Loeve expansion of Brownian motion and provide in [26] an optimal choice[5] of the couple (space discretization, probability weights) to approximate the standard Gaussian distribution. Known as quantization, this method of using preloaded files should be the method of choice for problems involving Brownian motion or Brownian bridge. Indeed, for this model, the effectiveness of quantization and its good accuracy for dimensions bigger than 3 make it relevant for various problems. Nevertheless, it is not straightforward to use this method for other multidimensional stochastic volatility or jump diffusion models.

Developed for various models, our method of computing (7) employing Malliavin Calculus is more complex than using directly preloaded files as done with the quantization method. However, one can use also Malliavin Calculus to express the backward conditional density. Provided that we are employing the same pseudo random number generator, one can first approximate the backward conditional density off-line for some points of assets trajectories and store it, then re-use them directly as preloaded files to have the distribution of each contract which is sufficient to compute the CVA or develop any other risk hedging strategy. Moreover, when the backward conditional density is already stored, using it to compute (7) is better, from a variance reduction point of view, than using (17) and (18). In fact, we provide in (29) and (30) the value of the backward conditional density and of its partial derivative. First, let us introduce some notations.

Definition 3.2 *The random variables* $\overline{\Gamma}_{s,t}$ *and* $\widehat{\Gamma}_{s,t}^i$ *are the solution* $X_{s,t}$ *of the following induction scheme*

$$X_{s,t} = X_{s,t}^1,\ X_{s,t}^k = X_{s,t}^{k+1}\bar{\pi}_{s,t}^{k,d} - \sum_{j=k}^{d}\int_0^t D_u^j X_{s,t}^{k+1} D_u^j \bar{\pi}_{s,t}^{k,d} du, \quad k \in \{1, ..., d\},$$

with the terminal values

$$\overline{\Gamma}_{s,t}^{d+1} = \Gamma_{s,t}, \quad \text{and} \quad \widehat{\Gamma}_{s,t}^{d+1,i} = \Gamma_{s,t}(1 + \pi_{s,t}^{i,d}) - \sum_{j=i}^{d}\int_0^t D_u^j \pi_{s,t}^{i,d} D_u^j \Gamma_{s,t} du,$$

where $\Gamma_{s,t}$*,* $\pi_{s,t}^{i,d}$ *are defined in Defintion 3.1 and* $\bar{\pi}_{s,t}^{k,d}$ *is given by*

$$\bar{\pi}_{s,t}^{k,d} = 1 + \frac{1}{t-s}\sum_{j=k}^{d}\int_s^t \rho_{jk}(u) dW_u^j, \text{ with } \rho \text{ is the inverse of the volatility matrix } \sigma.$$

[5] In a sense explained in their paper.

Theorem 3.2 *For any $s \in (0,t)$, $g \in \mathcal{E}_b(\mathbb{R}^d)$, $x = (x_1, ..., x_d)$ and $y = (y_1, ..., y_d)$ with $x_i > 0$ and $y_i > 0$,*

$$T_{s,t}[f](x) = E\left(f(S_t)\bar{h}(x, S_t)\right), \tag{27}$$

$$R^i_{s,t}[f](x) = E\left(f(S_t)\hat{h}(x, S_t)\right), \tag{28}$$

where

$$\bar{h}(x,y) = E\left(\Gamma_{s,t}\widehat{H}^x(S_s)\Big|S_t = y\right) = \frac{E\left(\overline{\Gamma}_{s,t}\widehat{H}^x(S_s)\widehat{H}^y(S_t)\right)}{E(\Gamma_{s,t}\widehat{H}^y(S_t))} \tag{29}$$

$$\begin{aligned}\hat{h}(x,y) &= E\left(\left(\Gamma_{s,t}(1+\pi^{i,d}_{s,t}) - \sum_{j=i}^{d}\int_0^t D^j_u \pi^{i,d}_{s,t} D^j_u \Gamma_{s,t} du\right)\frac{\widehat{H}^x(S_s)}{S^i_s}\Big|S_t = y\right) \\ &= \frac{E\left(\frac{\widehat{\Gamma}_{s,t}}{S^i_s}\widehat{H}^x(S_s)\widehat{H}^y(S_t)\right)}{E(\Gamma_{s,t}\widehat{H}^y(S_t))},\end{aligned} \tag{30}$$

with $\widehat{H}^y(S_t) = \prod_{j=1}^{d}\frac{H^y_j(S^j_t)}{S^j_t}$, $\Gamma_{s,t}$, $\pi^{i,d}_{s,t}$, $\overline{\Gamma}_{s,t}$ and $\widehat{\Gamma}_{s,t}$ are given in Definition 3.1 and Definition 3.2.

<u>Proof of Theorem 3.2</u>: In the same fashion as in the beginning of the proof of Theorem 3.1, we regularize the heaviside function H^y_i by $h_i \in \mathcal{C}^\infty_b(\mathbb{R})$. In order to lighten the notations, we remove in this proof the dependence on m in h_{mi} ($m \to \infty$). Then, we need to prove that for $0 \le k \le d$, we have

$$E\left(\Gamma_{s,t}\widehat{H}^x(S_s)\prod_{i=1}^{d}h'_i(S^i_t)\right) = E\left(\overline{\Gamma}^{k+1}_{s,t}\widehat{H}^x(S_s)\prod_{i=1}^{k}h'_i(S^k_t)\prod_{i=k+1}^{d}\frac{h_i(S^k_t)}{S^k_t}\right), \tag{31}$$

with (27) obtained directly from (31) when $k = 0$. In fact, $E\left(\Gamma_{s,t}\prod_{j=1}^{d}\frac{H_j(S^j_s)}{S^j_s}\Big|S_t = y\right)$ can be viewed heuristically as $\frac{E\left(\Gamma_{s,t}\prod_{j=1}^{d}\frac{H_j(S^j_s)}{S^j_s}\varepsilon_y(S_t)\right)}{E\left(\varepsilon_y(S_t)\right)}$ where ε_y is the Dirac distribution at y and we know that $\varepsilon_{y_i} = (H^y_i)'$. In order to make the reasoning rigorous, one replace the expectation of $\varepsilon_y(S_t)$ by the density of S_t evaluated at y.

Now let us prove (31) by induction, we introduce the following notations

$$\widehat{h}^d_k(S_t) = \prod_{i=k}^{d}\frac{h_i(S^i_t)}{S^i_t}, \quad \widehat{h'}_k(S_t) = \prod_{i=1}^{k}h'_i(S^i_t).$$

When $k=d$, we have by the chain rule $h'_d(S_t^d)=\frac{D_u^d h_d(S_t^d)}{D_u^d S_t^d}$ and $D_u^d S_t^d=\sigma_{dd}(u)S_t^d$ for every $u\in(s,t)$, thus

$$\begin{aligned}E\left(\Gamma_{s,t}\widehat{H}^x(S_s)\widehat{h'}_d(S_t)\right) &= E\left(\frac{1}{t-s}\int_s^t \Gamma_{s,t}\widehat{H}^x(S_s)\widehat{h'}_{d-1}(S_t)\frac{D_u^d h_d(S_t^d)}{D_u^d S_t^d}du\right)\\ &= E\left(\frac{1}{t-s}\int_s^t \Gamma_{s,t}\widehat{H}^x(S_s)\widehat{h'}_{d-1}(S_t)\frac{D_u^d h_d(S_t^d)}{\sigma_{dd}(u)S_t^d}du\right).\end{aligned}$$

Using Lemma 3.2 with

$$F=\frac{\Gamma_{s,t}}{S_t^d}\widehat{H}^x(S_s)\widehat{h'}_{d-1}(S_t)$$

and the fact that $\widehat{h'}_{d-1}(S_s)$ does not depend on the dth coordinate of the Brownian motion yields

$$\begin{aligned}&E\left(\Gamma_{s,t}\widehat{H}^x(S_s)\widehat{h'}_d(S_t)\right)\\ &= E\left(Fh_d(S_t^d)\frac{1}{t-s}\int_s^t\frac{dW_u^d}{\sigma_{dd}(u)}\right)-E\left(h_d(S_t^d)\frac{1}{t-s}\int_s^t D_u^d F\frac{du}{\sigma_{dd}(u)}\right)\\ &= E\left(h_d(S_t^d)\widehat{H}^x(S_s)\widehat{h'}_{d-1}(S_t)\Big(\frac{\Gamma_{s,t}}{S_t^d}(\overline{\pi}_{s,t}^{d,d}-1)-\frac{1}{t-s}\int_s^t D_u^d\frac{\Gamma_{s,t}}{S_t^d}\frac{du}{\sigma_{dd}(u)}\Big)\right)\\ &= E\left(\frac{h_d(S_t^d)}{S_t^d}\widehat{H}^x(S_s)\widehat{h'}_{d-1}(S_t)\Big(\Gamma_{s,t}\overline{\pi}_{s,t}^{d,d}-\frac{1}{t-s}\int_s^t D_u^d\Gamma_{s,t}\frac{du}{\sigma_{dd}(u)}\Big)\right) \qquad (32)\\ &= E\left(\overline{\Gamma}_{s,t}^d\widehat{H}^x(S_s)\widehat{h'}_{d-1}(S_t)\frac{h_d(S_t^d)}{S_t^d}\right)\end{aligned}$$

where the equality (32) comes from the fact that for $u\in(s,t)$

$$\frac{1}{\sigma_{dd}}D_u^d\frac{\Gamma_{s,t}}{S_t^d}=\frac{1}{S_t^d\sigma_{dd}}D_u^d\Gamma_{s,t}-\frac{\Gamma_{s,t}}{S_t^d}.$$

Now, let us suppose that (31) is satisfied for k and prove it for $k-1$. We have by the chain rule $h'_k(S_s^k)=\frac{D_u^k h_k(S_s^k)}{D_u^k S_s^k}$ and $D_u^k S_s^k=\sigma_{kk}(u)S_s^k$, thus

$$\begin{aligned}E\left(\Gamma_{s,t}\widehat{H}^x(S_s)\widehat{h'}_d(S_t)\right) &= E\left(\overline{\Gamma}_{s,t}^{k+1}\widehat{H}^x(S_s)\widehat{h'}_k(S_t)\widehat{h}_{k+1}^d(S_t)\right)\\ &= E\left(\frac{1}{t-s}\int_s^t\overline{\Gamma}_{s,t}^{k+1}\widehat{H}^x(S_s)\widehat{h'}_{k-1}(S_t)\widehat{h}_{k+1}(S_t)\frac{D_u^k h_k(S_t^k)}{\sigma_{kk}(u)S_t^k}du\right).\end{aligned}$$

As before, using Lemma 3.2 with in this time

$$F=\frac{\overline{\Gamma}_{s,t}^{k+1}}{S_t^k}\widehat{H}^x(S_s)\widehat{h'}_{k-1}(S_t)\widehat{h}_{k+1}(S_t)$$

and the fact that $\widehat{h}'_{k-1}(S_t)$ and $\widehat{h}_{k+1}(S_t)$ are not depend on the kth coordinate of the Brownian motion, we obtain

$$
\begin{aligned}
&E\left(\Gamma_{s,t}\widehat{H}^x(S_s)\widehat{h}'_d(S_t)\right)\\
&=\sum_{i=k}^{d}E\Bigg(h_k(S_s^k)\widehat{h}_{k+1}(S_t)\widehat{h}'_{k-1}(S_t)\frac{\widehat{H}^x(S_s)}{t-s}\Big(\frac{\overline{\Gamma}_{s,t}^{k+1}}{S_t^k}\int_s^t\rho_{ik}(u)dW_u^i-\int_s^t\rho_{ik}(u)D_u^i\frac{\overline{\Gamma}_{s,t}^{k+1}}{S_t^k}du\Big)\Bigg)\\
&=E\Bigg(\widehat{H}^x(S_s)\widehat{h}'_{k-1}(S_t)\widehat{h}_k(S_t)\Big(\overline{\Gamma}_{s,t}^{k+1}\bar{\pi}_{s,t}^{k,d}-\frac{1}{t-s}\int_s^t D_u^i\overline{\Gamma}_{s,t}^{k+1}D_u^i\bar{\pi}_{s,t}^{k,d}du\Big)\Bigg)\\
&=E\Bigg(\overline{\Gamma}_{s,t}^{k}\widehat{H}^x(S_s)\widehat{h}'_{k-1}(S_t)\widehat{h}_k(S_t)\Bigg)
\end{aligned}
$$

Similarly, one can prove (28).

■

In Remark 3.2, we use the set of the second order permutations $\overline{\mathcal{P}}_{1,d}$ defined by

$$\overline{\mathcal{P}}_{1,d}=\{p\in\mathcal{P}_{1,d};\ p\circ p=Id\},\tag{33}$$

where $\mathcal{P}_{1,d}$ is the set of permutations on $\{1,...,d\}$ and Id is the identity application.

Remark 3.2 *In order to make easier the implementation of* (19)*, it was shown in [2] that* $\Gamma_{s,t}$ *given in Definition 3.1 can be computed as a determinant of a given matrix* A

$$\Gamma_{s,t}=\sum_{p\in\overline{\mathcal{P}}_{1,d}}\epsilon(p)\prod_{i=1}^{d}A_{i,p(i)},\tag{34}$$

with $\epsilon(p)$ *as the signature of the permutation* $p\in\overline{\mathcal{P}}_{1,d}$*, and*

$$A=\begin{pmatrix}\pi_{s,t}^{1,d} & C_{1,2} & C_{1,3} & \cdots & C_{1,d}\\ 1 & \pi_{s,t}^{2,d} & C_{2,3} & \cdots & C_{2,d}\\ \vdots & \ddots & \ddots & \ddots & \vdots\\ 1 & \cdots & 1 & \pi_{s,t}^{d-1,d} & C_{d-1,d}\\ 1 & 1 & \cdots & 1 & \pi_{s,t}^{d,d}\end{pmatrix},$$

where

$$C_{k,l}=Cov(\pi_{s,t}^{k,d},\pi_{s,t}^{l,d})=\sum_{j=k\vee l}^{d}\Big(\frac{1}{s^2}\int_0^s\rho_{jk}(u)\rho_{jl}(u)du+\frac{1}{(t-s)^2}\int_s^t\rho_{jk}(u)\rho_{jl}(u)du\Big).$$

Using the same idea, we can deduce a generating method for the computation of $\overline{\Gamma}_{s,t}$ *and* $\widehat{\Gamma}_{s,t}$*. In fact, the solution* $X_{s,t}$ *of the induction scheme in Definition 3.2 can be expressed in a similar way as follow*

$$X_{s,t}=\sum_{p\in\overline{\mathcal{P}}_{1,d+1}}\epsilon(p)\prod_{i=1}^{d+1}\bar{A}_{i,p(i)},$$

with $\epsilon(p)$ is the signature of the permutation $p \in \overline{\mathcal{P}}_{1,d+1}$ and

$$\bar{A} = \begin{pmatrix} \bar{\pi}_{s,t}^{1,d} & \bar{C}_{1,2} & \bar{C}_{1,3} & \cdots & \bar{C}_{1,d+1} \\ 1 & \bar{\pi}_{s,t}^{2,d} & \bar{C}_{2,3} & \cdots & \bar{C}_{2,d+1} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 1 & \cdots & 1 & \bar{\pi}_{s,t}^{d,d} & \bar{C}_{d,d+1} \\ 1 & 1 & \cdots & 1 & X_{s,t}^{d+1,d} \end{pmatrix},$$

where $\bar{C}_{k,d+1} = \left\langle \bar{\pi}_{s,\cdot}^{k,d}, X_{s,\cdot}^{d+1,d} \right\rangle_t$ and

$$\bar{C}_{k,l} = Cov(\bar{\pi}_{s,t}^{k,d}, \bar{\pi}_{s,t}^{l,d}) = \sum_{j=k\vee l}^{d} \frac{1}{(t-s)^2} \int_s^t \rho_{jk}(u)\rho_{jl}(u)du, \ k,l \in \{1,..,d\}.$$

Let denote $\Gamma_{s,t}^{-k}$ the second order permutation determinant of the matrix A^{-k} that comes from A by suppressing both line and column "k"

$$\Gamma_{s,t}^{-k} = \sum_{p \in \overline{\mathcal{P}}_{1,d}^{k}} \epsilon(p) \prod_{i=1; i\neq k}^{d} A_{i,p(i)},$$

where $\overline{\mathcal{P}}_{1,d}^{k} = \{p \in \overline{\mathcal{P}}_{1,d}, \quad p(k) = k\}$. The following lemma gives a generating way to implement the term " $\sum_{j=i}^{d} \int_0^t D_u^j \pi_{s,t}^{i,d} D_u^j \Gamma_{s,t} du$" which appear in both Theorem 3.1 and Theorem 3.2.

Lemma 3.3 *Let $i \in \{1, ..., d\}$ and $s \in (0,t)$. We have*

$$\sum_{j=i}^{d} \int_0^t D_u^j \pi_{s,t}^{i,d} D_u^j \Gamma_{s,t} du = \sum_{l=1}^{d} \Gamma_{s,t}^{-l} C_{l,i}.$$

<u>Proof of Lemma 3.3</u>: First, using (34), we have for every $u \in (0,t)$

$$\begin{aligned} D_u^j \Gamma_{s,t} &= \sum_{p \in \overline{\mathcal{P}}_{1,d}} \epsilon(p) D_u^j \left(\prod_{i=1}^{d} A_{i,p(i)} \right) \\ &= \sum_{p \in \overline{\mathcal{P}}_{1,d}} \epsilon(p) \sum_{l=1}^{d} \left(\prod_{i=1; i\neq l}^{d} A_{i,p(i)} \right) D_u^j A_{l,p(l)} \\ &= \sum_{l=1}^{d} \sum_{p \in \overline{\mathcal{P}}_{1,d}^{l}} \epsilon(p) \left(\prod_{i=1; i\neq l}^{d} A_{i,p(i)} \right) D_u^j \pi_{s,t}^{l,d}, \end{aligned}$$

where the last equality is due to the fact that $A_{l,p(l)}$ is deterministic except for $p(l) = l$ for which $A_{l,l} = \pi_{s,t}^{l,d}$. Now, since $\pi_{s,t}^{l,d}$ does not depend on the Brownien motion W^j

when $l > j$, using Fubini theorem we get

$$\begin{aligned}
\sum_{j=i}^{d} \int_0^t D_u^j \pi_{s,t}^{i,d} D_u^j \Gamma_{s,t} du &= \sum_{j=i}^{d} \sum_{l=1}^{j} \sum_{p \in \overline{\mathcal{P}}_{1,d}^{l}} \epsilon(p) \prod_{i=1; i \neq l}^{d} A_{i,p(i)} \int_0^t D_u^j \pi_{s,t}^{i,d} D_u^j \pi_{s,t}^{l,d} du \\
&= \sum_{l=1}^{d} \sum_{j=l \vee i}^{d} \sum_{p \in \overline{\mathcal{P}}_{1,d}^{l}} \epsilon(p) \prod_{i=1; i \neq l}^{d} A_{i,p(i)} \int_0^t D_u^j \pi_{s,t}^{i,d} D_u^j \pi_{s,t}^{l,d} du \\
&= \sum_{l=1}^{d} \Gamma_{s,t}^{-l} C_{l,i}.
\end{aligned}$$

■

4 Complexity comparison of different approaches

4.1 Square Monte Carlo benchmark and regression method

As we saw in the introduction, to approximate the CVA value using (6), one needs first to simulate assets trajectories, also known as simulating scenarios [14], that corresponds to the outer expectation, then compute the conditional expectation given by (7) involved in (3), (4) and (5). For some models and some contracts, this computation could be explicit or semi-explicit using Fourier transform [12]. When the dimension is 2 or 3 and for a large choice of models and contracts, numerical methods for PDEs[6] become more appropriate because faster than a Monte Carlo simulation. However, for the most general models and contracts and for dimensions bigger than 4, the only option that remains is the use of Monte Carlo for both simulating asset trajectories and the conditional expectation in (7).

When only exchanging European contracts with the counterparty and the number of simulated trajectories is big enough, using a square Monte Carlo for CVA computation gives reference results. The parallel suitability of Monte Carlo allows to increase the accuracy using more computing resources including many-cores architectures as done in [1]. Nevertheless, the complexity of square Monte Carlo remains too high. Indeed, if we denote by d the dimension of the problem in terms of assets, by M the number of simulated trajectories and by N the number of time steps of the simulation[7], then we shall simulate exactly $dNM[1 + (M-1)(N-1)/2]$ segments of trajectory (see Figure 1). The proof of this result is built on the assumption that we use the same number of trajectories for both Monte Carlo simulations involved. In fact, we have to draw dMN segments for the first Monte Carlo and at each time step t_k, we should simulate $(M-1)(N-k)$ segments of trajectories at each point from the first dMN set, which leads to

$$dMN + \sum_{k=1}^{N} dM(M-1)(N-k) = dMN\left(1 + \frac{(M-1)(N-1)}{2}\right).$$

[6]Partial Differential Equations

[7]Supposed here equal to the number of time steps of the SDE discretization

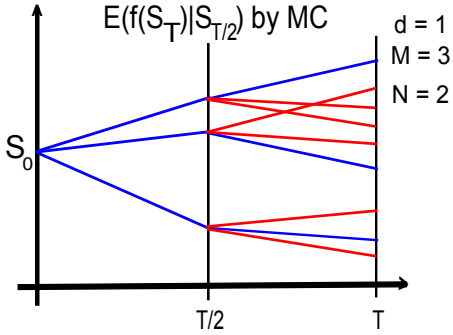

Figure 1: The price of the contract at $T/2$ is approximated by a Monte Carlo simulation.

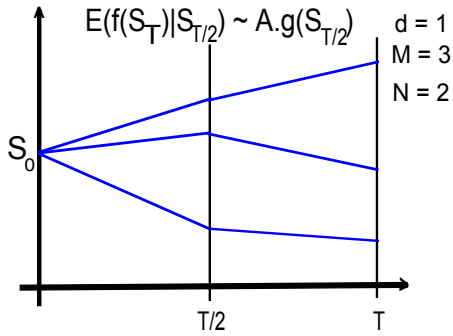

Figure 2: The price of the contract at $T/2$ is approximated by a linear regression on a finite basis.

Thus, if we denote by $\mathcal{T}$ the complexity of the simulation of a segment of a trajectory, the complexity of square Monte Carlo is of order $O(\mathcal{T}dM^2N^2)$. Another inconvenient of this method comes from the fact that it has an exponential complexity according to the number of trajectories when employed for American contracts. Indeed, according to induction (5), to get $\phi^{am}_{t_k}(S_{t_k})$ we must have M trajectories of the function $\phi^{am}_{t_{k+1}}(S_{t_{k+1}})$ and so on.

For all the previous reasons, practitioners prefer the use of methods based on regression whose complexity can be decomposed into two terms: A part associated to the scenarios generation which is of order $O(\mathcal{T}dMN)$ and another part for the approximation of the conditional expectation by a regression on a polynomial basis $b(S_{t_k})$: $E\left(f(S_{t_{k+1}})|S_{t_k}\right) \approx A.b(S_{t_k})$ (see Figure 2). The vector A minimizes the quadratic error

$$\left|\left|f(S_{t_{k+1}}) - A.b(S_{t_k})\right|\right|_{L^2} \tag{35}$$

and thus equal to

$$A = \Psi^{-1}E\left(f(S_{t_{k+1}})b(S_{t_k})\right), \tag{36}$$

where the matrix $\Psi = E\left(b(S_{t_k})b^t(S_{t_k})\right)$. Consequently, at each time step, the matrix inversion (36) can be implemented by the Singular Value Decomposition (SVD) explained in [23] and the expectations are approximated by an arithmetic average

$$E\left(f(S_{t_{k+1}})b(S_{t_k})\right) \approx \frac{1}{N}\sum_{l=1}^{N} f(S^{(l)}_{t_{k+1}})b(S^{(l)}_{t_k}),$$

$$E\left(b(S_{t_k})b^t(S_{t_k})\right) \approx \frac{1}{N}\sum_{l=1}^{N} b(S^{(l)}_{t_k})b^t(S^{(l)}_{t_k}).$$

Due to the SVD inversion performed at each time step after an expectation approximation, the complexity of the regression phase is of order $O(K_M^3MN)$, where K_M is the cardinal of the vector $b(S_{t_k})$. The overall regression method for CVA computation has a complexity of order $O((\mathcal{T}d + K_M^3)MN)$ even when American contracts are involved. Although this

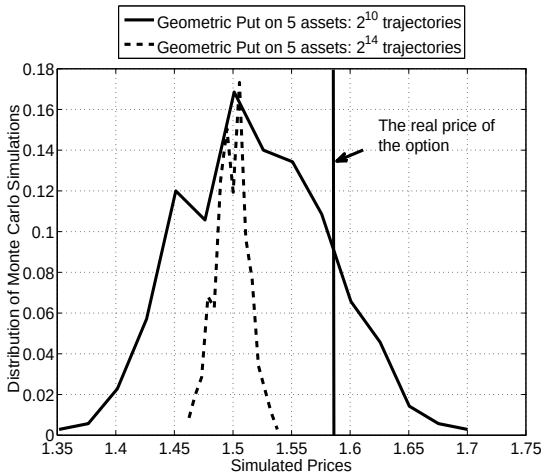

Figure 3: Histogram of prices obtained by Monte Carlo using Longstaff-Schwartz algorithm with maturity $T = 1$ and thirty exercise dates.

method is less complex than square Monte Carlo, it is not well suited to parallel implementation (see [1]) because of the matrix inversion phase. Moreover, computing CVA with regression method involves generally the implementation of several regressions, each one uses a different vector $b(S_{t_k})$ for each contract. Finally, the accuracy cannot be increased only by increasing the number of trajectories, because one has also to take bigger values of K_M. The latter fact becomes a real issue when the dimension d is big enough which could be seen in Figure 3 for $d = 5$, as the obtained prices remain far from the real price even when taking more trajectories. This limitation was studied by the authors of [17] that recommend to have a number of polynomials $K_M \sim O(\sqrt{\log(M)})$. Despite of these drawbacks and because this method is quite fast, practitioners, like in [14], prefer to use it with some heuristics to improve the simulated results.

Although we discussed here the basic implementation of the regression method, we point out the existence of different versions of it. Among those versions, the authors of [16] propose the use of a local regression basis and derived a full convergence analysis of their method. The algorithm based on their idea can be considered in the midway between a basic global regression and MCM which is a pure local method. Indeed, the algorithm proposed uses regressions iteratively that keeps the overall complexity similar to a basic regression method and it improves the results locally. Nevertheless, this algorithm inherits the problem of choosing the right local regression, especially when dealing with American contracts, in which case one has to perform iterative regressions for each contract.

4.2 MCM as an alternative

From a theoretical point of view, MCM is really interesting because it is general enough to a large choice of models and contracts. Thus, unlike regression methods, there is no heuristic choice of the vector $b(S_{t_k})$ that should be specified according to the model and to the contract. Indeed, as already mentioned in Section 2 as well as in [2] and will be discussed also in Section

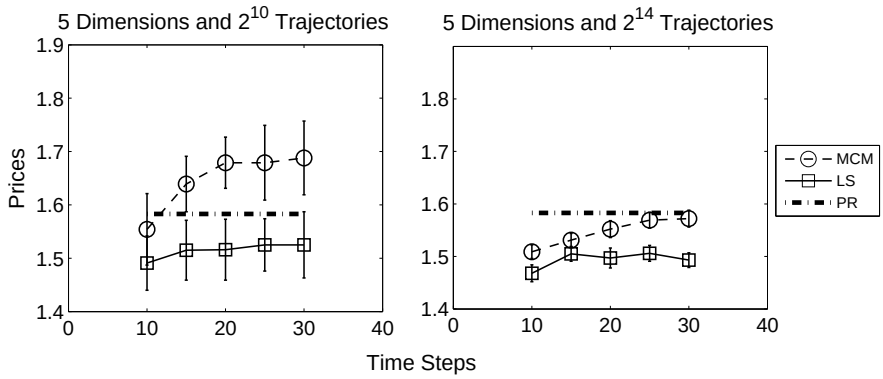

Figure 4: LS vs. MCM for 5 dimensional American option, the real price is given by PR.

5, we can apply MCM to path-dependent contracts and with models as multidimensional Heston models and usual jump diffusion models and others. From a computational point of view, because it is a pure Monte Carlo method, MCM is as suited to parallel implementation as a square Monte Carlo method. MCM could be also used for CVA computation that involves American contracts. Also, in contrast to regression algorithms, MCM aims at a nonparametric algorithm in which the accuracy can be improved by increasing only the number of the simulated trajectories and adding more computing resources to speed up simulations. This fact is demonstrated on Figure 4 in which we can see that MCM performs better[8] than Longstaff-Schwartz (LS) algorithm[9] when we increase the number of simulated trajectories. Moreover, according to Figure 4, MCM is as efficient as LS because it could provide sufficiently good results even when simulating very few number of trajectories, here only 2^{10}. In Section 5, we will see that the accurate results obtained by MCM goes beyond the computation of the CVA, because it allows to have compelling approximation of the delta sensitivity.

Although MCM is both less complex than square Monte Carlo and applicable for CVA involving American contracts, it is more complex than regression methods. Indeed, with MCM we need only to simulate the assets trajectories once, but the approximation of the conditional expectation as a quotient of two expectations is implemented for each path. The complexity of this method is then of order $O(C_d \mathcal{T} d M^2 N)$, where C_d depends on the dimension d. If the backward conditional density, introduced in Section 3, is computed off-line then $C_d = 1$. Otherwise, $C_d = O(n_d)$ where n_d is the number of terms involved in the quasi-determinant $\Gamma_{s,t}$ and expressed in the following proposition.

Proposition 4.1 *For $d \geq 3$, n_d satisfies the following induction*

$$n_d = n_{d-1} + (d-1)n_{d-2} \tag{37}$$

with $n_1 = 1$, $n_2 = 2$.

[8]When only American options are involved. The error should be bigger when simulating the CVA on various American options

[9]An algorithm based on regression.

Proposition 4.1 can be easily proved by induction using the relation

$$\overline{\mathcal{P}}_{1,d} = \{\tau_{d-1}^{d-1} \circ p;\ p \in \overline{\mathcal{P}}_{1,d-1}\} \cup \{\tau_{d-1}^{l} \circ p; p \in \overline{\mathcal{P}}_{1,d-1}, p(l) = l, l \in \{1, ..., d-1\}\},$$

where $\tau_i^j : i \leftrightarrow j$ is the transposition application on $\{1, ..., d-1\}$ that swaps only i to j and j to i. This relation is another version of the relation (2.16) given in [2]. Although the induction (37) provides an n_d which is by far smaller than $d!$, n_d can be quite big for high dimensions, for example: $n_5 = 26$, $n_6 = 76$, $n_7 = 232$ and $n_{10} = 9496$. To overcome this problem, we explain, in the example given in Section 5 how to group contracts together in order to reduce the number of Brownian motions involved and reduce d.

As proposed in [9], using the multidimensional quick sort changes the term M^2 that appears in $O(C_d \mathcal{T} d M^2 N)$ into $M(\ln M)^{(d-1)\vee 1}$. However, we prefer in our work to use the straight implementation because the parallelization of the multidimensional quick sort on GPU make the overall implementation generally either less efficient or barely more efficient for the usual values of M and d.

5 Numerical tests using parallel implementation

In the previous sections, we presented the theoretical framework of CVA estimation using MCM and studied its computational complexity. To finalize this work one has to give some theoretical approximations of the error of the estimation. However, this part is delayed to future work which can be based on works dedicated to American options like [8]. Nevertheless, we demonstrate here the accuracy of MCM by comparing it to square Monte Carlo and to a regression method. When using large number of trajectories, square Monte Carlo provides benchmark values for both path-independent and path-dependent European contracts that will be considered as the real values. Consequently, we study the accuracy of MCM for CVA that involves only European contracts. In order to have an idea of how MCM behaves when dealing with American contracts, we refer the reader to [2, 9]. Before summarizing the results of the different simulations, we start by presenting the considered models and how simulating European path-dependent contracts could be performed.

5.1 Benchmarking setup

Based on Section 2, we implement one example from the intensity family and one example from the structural family. Each model will be completely specified by the assets dynamics and either the dependence between the assets S_t and the default time τ or between the value exposure V_t and τ.

Regarding the intensity model, we take the constant volatility version of (9):

$$\begin{cases} \dfrac{dS_t^i}{S_t^i} = r dt + \sigma_i \displaystyle\sum_{j=1}^{i} \varrho_{ij} dW_t^j, \quad S_0^i = z_i, \quad i = 1, .., d, \\ \lambda_t = \alpha'' + \alpha'(V_t)_+, \text{ with } \alpha' \geq 0 \text{ and } \alpha'' \geq 0 \end{cases} \tag{38}$$

with $T = 1$, the risk neutral interest rate $r = \ln(1.1)$, the time discretization is defined using the time steps that is given as a parameter in each simulation, $S_0^i = 100$ and $\sigma_{ij} = \sigma_i \varrho_{ij}$ with

$\sigma_i = 0.2$ where $\varrho = \{\varrho_{ij}\}_{1\leq i,j\leq d}$ comes from the Cholesky decomposition of the correlation matrix $\delta_{i-j} + \alpha(1-\delta_{i-j})$ such that $\alpha \in [0,1)$ and δ is the Kronecker symbol.

Regarding the structural model, we take the constant volatility version of (9) and we correlate it with the Brownian motion W_t^0 that drives the firm value process X_t:

$$\begin{cases} \dfrac{dX_t}{X_t} = rdt + \sigma_0 dW_t^0, \quad X_0 = z_0, \\ \dfrac{dS_t^i}{S_t^i} = rdt + \sigma_i \displaystyle\sum_{j=0}^{i} \varrho_{ij} dW_t^j, \quad S_0^i = z_i, \quad i = 1,..,d, \end{cases} \tag{39}$$

with $T = 1$, the risk neutral interest rate $r = \ln(1.1)$, the time discretization is defined using the time steps that is given as a parameter in each simulation, $S_0^i = 100$ and $\sigma_{ij} = \sigma_i \varrho_{ij}$ with $\sigma_i = 0.2$ where $\varrho = \{\varrho_{ij}\}_{0\leq i,j\leq d}$ comes from the Cholesky decomposition of the correlation matrix $\delta_{i-j} + \alpha(1-\delta_{i-j})$ such that $\alpha \in [0,1)$ and δ is the Kronecker symbol. With this specific example, one understands better the sense of (15) and how the Malliavin Calculus can be implemented on $W = (W^1, ..., W^d)$ without considering the dependence with respect to W^0.

Using MCM, simulating CVA that involves only European path-independent contracts (3) is quite simple when compared to adding European path-dependent contracts (4). Then, some details should be provided for the implementation of (4). To simplify the explanations, we assume that N involved in (6) is equal to the number of time steps used to approximate the trajectories of the assets. Let us consider the problem of approximating the conditional expectation

$$E\left((\max_{i=0,..,N} S_{t_i}^1 - S_{t_N}^2)_+ | \mathcal{F}_{t_k} \right).$$

Using Markov property and notation $\overline{S}_{t_k}^1 = \max_{i=0,..,k} S_{t_i}^1$, this conditional expectation can be rewritten as

$$\begin{aligned} E\left((\max_{i=0,..,N} S_{t_i}^1 - S_{t_N}^2)_+ | \mathcal{F}_{t_k} \right) &= E\left((\overline{S}_{t_k}^1 \vee \max_{i=k+1,..,N} S_{t_i}^1 - S_{t_N}^2)_+ \Big| \overline{S}_{t_k}^1, S_{t_k}^1, S_{t_k}^2 \right) \\ &= \Theta_{S_{t_k}^1, S_{t_k}^2}(\overline{S}_{t_k}^1) \end{aligned} \tag{40}$$

with:

$$\Theta_{S_{t_k}^1, S_{t_k}^2}(y) = E\left((y \vee \max_{i=k+1,..,N} S_{t_i}^1 - S_{t_N}^2)_+ \Big| S_{t_k}^1, S_{t_k}^2 \right) \tag{41}$$

and for each asset trajectory, we have a specific value y and once it is fixed as a payoff parameter, we can compute the conditional expectation using the result of Theorem 3.1 or Theorem 3.2.

The previous idea for implementing European path-dependent contract can be applied for lookback as well as barrier and Asian contract. In order to have a better approximation of some SDEs and path-dependent contracts, one can also take the number of time steps

bigger than N[10]. Indeed, considering the previous path-dependent example, it is possible to increase the accuracy of the approximation of $\sup_{0\leq s\leq t_k} S_s$ and $\sup_{t_k<s\leq t_N} S_s$ if we have more than k points and $N-k$ points in the intervals $[0,t_k]$ and $[t_k,t_N]$ respectively.

We should also discuss the parameters C_d and n_d introduced in Section 4.2. Previously, we pointed out that the induction (37) provides important values when d is big. However, not all contracts require the d-dimensional information, fact that can be seen with our previous example given in (41). Actually, assuming $d = 10$, to evaluate (40) we need only two Brownian motions if we are using intensity model (38) and three Brownian motions[11] when using structural model (39).

We finish the benchmarking setup by presenting two numerical tricks: The first one should improve the Monte Carlo estimator of (19) and (20) and the second one removes the contributions that are "Wrong".

Localization with truncation

In order to approximate numerically (19) and (20), we remind that we are using the same set of trajectories which makes MCM less complex than a square Monte Carlo. However, some trajectories must not be added [12] to the Monte Carlo estimator of $T_{s,t}[f](x)$ and $R^i_{s,t}[f](x)$ because they are far from the point x. This technique is known by localization and some papers, as [3, 7], propose some "optimal" choice of localizing functions. We use here a simpler idea based on truncation. Without loss of generality, ignoring the dependence on the default time and introducing a subset $\Upsilon \in \mathbb{R}^d$, we replace

$$V_{t_k} = E\left(V_{t_{k+1}}|S_{t_k}\right) = E\left(V_{t_{k+1}}1_{S_{t_{k+1}}\in\Upsilon}|S_{t_k}\right) + E\left(V_{t_{k+1}}1_{S_{t_{k+1}}\in\Upsilon^c}|S_{t_k}\right)$$

in $E\left((V_{t_k})_+\right)$ to get the inequality

$$\begin{aligned} E\left((V_{t_k})_+\right) &\leq E\left(\left[E\left(V_{t_{k+1}}1_{S_{t_{k+1}}\in\Upsilon^c}|S_{t_k}\right)\right]_+ + \left[E\left(V_{t_{k+1}}1_{S_{t_{k+1}}\in\Upsilon}|S_{t_k}\right)\right]_+\right) \\ &\leq \quad E\left(\left[V_{t_{k+1}}\right]_+ 1_{S_{t_{k+1}}\in\Upsilon^c}\right) + \left(\left[E\left(V_{t_{k+1}}1_{S_{t_{k+1}}\in\Upsilon}|S_{t_k}\right)\right]_+\right). \end{aligned} \tag{42}$$

The error value $E\left(\left[V_{t_{k+1}}\right]_+ 1_{S_{t_{k+1}}\in\Upsilon^c}\right)$ can be estimated, and if it has the order of the errors induced by the Central Limit Theorem (95% confidence interval) it can be neglected and we consider:

$$E\left((V_{t_k})_+\right) \approx \left(\left[E\left(V_{t_{k+1}}1_{S_{t_{k+1}}\in\Upsilon}|S_{t_k}\right)\right]_+\right).$$

From an implementation point of view, we choose Υ such that we include the most likely asset trajectories, with probability between $[85\%, 95\%]$ depending on the dimension of the problem. For example, for $d = 1$ or 2, we include only the 95% probability trajectories. Formally speaking, for a fixed $S^i_s = x_i$, we take into account only the trajectories associated to S^i_t that could occur with probability 95%.

[10] We remind the reader that N is associated to the discretization of the time integral.
[11] Because of W^0.
[12] Or added with a much smaller weight.

Removing "Wrong" values of the conditional expectation

For a fixed set of simulated trajectories M and fixed trajectory $l \in \{1, ..., M\}$, this idea is based on the fact that we want the conditional expectation $E\left(V_{t_{k+1}}|S^l_{t_k}\right)$ to be included in $\left[\min_{j\in\{1,...,M\}} V^j_{t_{k+1}}, \max_{j\in\{1,...,M\}} V^j_{t_{k+1}}\right]$. If this condition is not fulfilled, we consider that our estimator gives us a wrong value and we stop taking into account the conditional expectations generated by the trajectory of index l.

5.2 MCM accuracy for CVA and CVA sensitivities

First, we need to point out that we were able to perform our accuracy study thanks to a parallel GPU [13] implementation on an Nvidia 480 GTX card. Indeed, one of the advantageous of MCM algorithm is its suitability to parallel architecture. The goal of this section is to prove the good accuracy of even a basic implementation of MCM without reducing the variance using the backward conditional density introduced in section 3.2. More advanced numerical study of the backward conditional density will be performed in the future.

Table 1: Examples of simulated CVA: MCM (2^{16} trajectories), regression Reg (2^{16} trajectories), square Monte Carlo MC2 (2^{14} trajectories), $N = 10$.

Payoff	$d = 2$					
	$\alpha = 0$			$\alpha = 0.5$		
	MCM	Reg	MC2	MCM	Reg	MC2
Call on average	11.4	12.9	11.6	11.9	14.5	12.6
Call on max	20.9	23.7	22.1	18.3	21.4	19.5
$(\overline{S}^2_T - S^1_T)_+$	14.5	17.8	15.7	12.4	14.8	13.4
Payoff	$d = 3$					
	$\alpha = 0$			$\alpha = 0.5$		
	MCM	Reg	MC2	MCM	Reg	MC2
Call on average	10.0	10.9	10.0	12.1	13.1	11.6
Call on max	30.3	29.2	28.0	25.6	25.4	23.3
$(\overline{S}^3_T - \frac{S^1_T+S^2_T}{2})_+$	15.6	15.5	13.8	13.5	12.3	13.4

We present two parts of the accuracy study: The first one compares MCM to the linear regression method to compute (6) when $1_{\tau\in(t_k,t_{k+1}]}$ is assumed to be equal to 1 and V_{t_k} is the price of each contract associated to each payoff specified in Table 1. Thus, in this part we do not take into account the dependence according to the default time simply because we do not know a standard way to do it for the regression method. In the second part, we study the MCM accuracy for both computing the CVA and its sensitivity for the intensity model (38) and the structural model (39). In tables 1 and 2, the errors associated to the

[13] Graphic Processing Unit

confidence interval of 95% are smaller than 5% of the showed CVA values. As for Table 3, the confidence interval of 95% is specified thanks to the $\pm$ sign.

When considering the values given by the square Monte Carlo (MC2) as the real values, according to Table 1, the values obtained by MCM are almost always better than regression method (Reg). In cases when MCM is less accurate, it is sufficient to increase the number of simulated trajectories. This fact is not true for regression methods because they require the increase of the cardinal of the regression family. Also unlike regression methods, the accurate results obtained by MCM allows to compute Δ sensitivity either by using the finite difference (FD) of two MCM prices or by simulating $R^i_{s,t}[f](x)$ introduced in (20).

Table 2: Examples of CVA and sensitivity computations: MCM (2^{18} trajectories), square Monte Carlo MC2 (2^{14} trajectories).

Payoff	Intensity model			
	$\mathrm{CVA}_{0,T}$		$\partial_{S_0^2}\mathrm{CVA}_{0,T}$	
	MC2	MCM	using (20)	using FD
$(S_T^2 - S_T^1)_+$	1.31	1.39	0.08	0.10
$(\overline{S}_T^2 - S_T^1)_+$	2.55	2.70	0.10	0.13
Call on max	3.55	3.54	0.12	0.10
Payoff	Structural model			
	$\mathrm{CVA}_{0,T}$		$\partial_{S_0^2}\mathrm{CVA}_{0,T}$	
	MC2	MCM	using (20)	using FD
$(S_T^2 - S_T^1)_+$	2.11	2.26	0.04	0.06
$(\overline{S}_T^2 - S_T^1)_+$	3.84	4.13	0.10	0.12
Call on max	0.95	0.90	0.044	0.050

Table 3: Examples of CVA values for $d = 3$: MCM (2^{14} trajectories), square Monte Carlo MC2 (2^{14} trajectories).

Payoff	Intensity model		Structural model	
	MC2	MCM	MC2	MCM
Call on average	0.21 ± 0.04	0.27 ± 0.01	0.03 ± 0.01	0.04 ± 0.01
Call on max	0.67 ± 0.01	0.84 ± 0.05	0.22 ± 0.03	0.31 ± 0.05
$(\overline{S}_T^3 - \frac{S_T^1+S_T^2}{2})_+$	0.80 ± 0.07	0.86 ± 0.09	0.25 ± 0.06	0.30 ± 0.01

Once more, according Table 2, when the number of simulated trajectories is sufficient, we obtain accurate values of the CVA and its sensitivity for both intensity and structural modelling framework. Table 3 shows sufficiently accurate results for $d = 3$ for intensity and structural models even when only 2^{14} trajectories are simulated. For $d = 3$, the results of

the sensitivity using (20) and using FD do not coincides as good as in Table 2 except when we compute the sensitivity according to an asset that is highly involved in the CVA.

Based on what is discussed before, we summarize the comparison of the three methods in the following table

Table 4: Comparison of the different algorithms

	Monte Carlo square	Monte Carlo with regression	Monte Carlo with Malliavin calculus
Complexity	$O(dM^2N^2)$	$O((\mathcal{T}d + K_M^3)MN)$	$O(C_d\mathcal{T}dM^2N)$
Parallelization	Suited	Limited	Suited
American Op.	No	Yes	Yes
Dimensions	Unlimited!	Very limited	Limited
Delta	Expensive	Not possible	Limited

In Table 4, very limited in dimension means limited and not standard in the sense of choosing the vector basis.

References

[1] L. A. Abbas-Turki, S. Vialle, B. Lapeyre and P. Mercier, *Pricing derivatives on graphics processing units using Monte Carlo simulation,* Concurrency and Computation: Practice and Experience, 2012.

[2] L. A. Abbas-Turki and B. Lapeyre, *American Options by Malliavin Calculus and Non-parametric Variance and Bias Reduction Methods,* SIAM Journal on Financial Mathematics, 3(1): 479–510, 2012.

[3] V. Bally, L. Caramellino and A. Zanette, *Pricing American options by Monte Carlo methods using a Malliavin calculus approach,* Monte Carlo Methods and Applications, 11: 97–133, 2005.

[4] V. Bally and G. Pagès, *A Quantization Algorithm for Solving Multidimensional Discrete-Time Optimal Stopping Problems,* Bernoulli, 9(3): 1003–1049, 2003.

[5] T. R. Bielicki, M. Jeanblanc and M. Rutkowski, *Credit Risk Modeling*, Center of the study of finance and Insurance, Osaka University, Japan.

[6] T. R. Bielecki and M. Rutkowski, Credit Risk: Modelling, Valuation and Hedging. *Springer- Verlag*, Berlin Heidelberg New York, 2002.

[7] B. Bouchard, I. Ekeland and N. Touzi, *On the Malliavin approach to Monte Carlo approximation of conditional expectations,* Finance and Stochastics, 8, 45–71, 2004.

[8] B. Bouchard and N. Touzi, *Discrete-Time Approximation and Monte-Carlo Simulation of Backward Stochastic Differential Equations,* Stochastic Processes and their Applications, 111(2), 175–206, 2004.

[9] B. Bouchard and X. Warin, *Monte-Carlo Valuation of American Options: Facts and New Algorithms to Improve Existing Methods,* Numerical Methods in Finance, Springer Proceedings in Mathematics, 12, 215–255, 2012.

[10] A. Boukhobza and J. Maetz, *CVA, Wrong Way Risk, Hedging and Bermudan Swaption*, August 2012.

[11] F. Black and J. C. Cox, *Valuing corporate securities: Some effects of bond indenture provisions*, Journal of Finance, 31: 351–367, 1976.

[12] D. Brigo and F. Mercurio, Interest Rate Models - Theory and Practice: With Smile, Inflation and Credit, *Springer Finance*, 2007.

[13] D. Brigo, M. Morini and A. Pallavicini, Counterparty Credit Risk, Collateral and Funding: With Pricing Cases For All Asset Classes, *John Wiley and Sons*, 2013.

[14] G. Cesari & al, Modelling, Pricing and Hedging Counterparty Credit Exposure, *Springer Finance*, 2009.

[15] E. Fournié, J.-M. Lasry, J. Lebuchoux and P.-L. Lions, *Applications of Malliavin calculus to Monte Carlo methods in finance II*, Finance and Stochastics, 5, 201–236, 2001.

[16] E. Gobet, J.-P. Lemor and X. Warin, *A Regression-based Monte Carlo method to solve Backward Stochastic Differential Equations,* The Annals of Applied Probability, 15(3), 2172–2202, 2005.

[17] P. Glasserman and B. Yu, *Number of Paths Versus Number of Basis Functions in American Option Pricing,* The Annals of Applied Probability, 14(4): 2090–2119, 2004.

[18] J. Hull and A. White, *CVA and Wrong Way Risk,* Financial Analysts Journal, 68(5): 58–69, 2012.

[19] A. Kohatsu-Higa and R. Pettersson, *Variance reduction methods for simulation of densities on Wiener space,* SIAM Journal on Numerical Analysis, 40(2): 431–450, 2002.

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

[21] R. Merton, *On the pricing of corporate debt: The risk structure of interest rates,* Journal of Finance, 3:449–470, 1974.

[22] G. Pagès and J. Printems, *Functional quantization for numerics with an application to option pricing,* Monte Carlo Methods and Applications, 11(11): 407–446, 2005.

[23] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, Numerical Recipes in C++: The Art of Scientific Computing, *Cambridge University Press*, 2002.

[24] M. Pykhtin, *Modeling credit exposure for collateralized counterparties,* The Journal of Credit risk, 5(4): 3–27, Winter 2009/10.

[25] D. Rosen and D. Saunders, *CVA and wrong way,* Journal of Risk Management in Financial Institutions, 5(3): 353–372, 2012.

[26] http://www.quantize.maths-fi.com/gaussian_process_database.