



HAL
open science

Conditional Monte Carlo Learning for Diffusions II: extended methodology and application to risk measures and early stopping problems

Lokman Abbas-Turki, G. Pagès, Babacar Diallo

► **To cite this version:**

Lokman Abbas-Turki, G. Pagès, Babacar Diallo. Conditional Monte Carlo Learning for Diffusions II: extended methodology and application to risk measures and early stopping problems. 2020. hal-02959494

HAL Id: hal-02959494

<https://hal.science/hal-02959494>

Preprint submitted on 6 Oct 2020

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.

Conditional Monte Carlo Learning for Diffusions II: extended methodology and application to risk measures and early stopping problems

L. Abbas-Turki*, B. Diallo[†] and G. Pagès[‡]

October 6, 2020

Abstract

In Conditional Monte Carlo Learning for Diffusions part I (CMCLDI) [2], we presented a One-layered Nested Monte Carlo (1NMC) to simulate functionals U of a Markov process X . Based on a judicious combination between regression and 1NMC used for localization purpose, this methodology allows to simulate $U_{t \geq s}$ conditionally on X_s . The parallel suitability and scalability of 1NMC makes this algorithm very competitive to simulate quantities that are almost impossible to simulate with other methods. In this paper, using the double layer of trajectories, we explain further the mathematical background of the control on the bias propagation. With this double layer structure, we also detail how to adjust the variance to get a better approximation of the second moment from the regression. In normal and log-normal models, this variance adjustment allows a better description of tail events. Since we applied this algorithm on Backward Stochastic Differential Equations in CMCLDI, we show here its strength for the simulation of risk measures and optimal stopping problems. Two highly dimensional numerical examples are executed in few minutes on one Graphics Processing Unit (GPU).

1 Introduction

Various contributions like [1, 8, 15, 20, 24] show an increasing interest in Nested Monte Carlo (NMC) simulation. NMC used to be avoided due to the heavy computations that it involves, but it becomes relevant as it can be very efficiently implemented on massively parallel architectures like Graphics Processing Units (GPUs). Its perfect scalability with respect to an increasing computing capability makes it a sustainable solution for various applications. In CMCLDI [2], we proposed local analytical approximations of functionals of a diffusion using various regressions within One-layered Nested Monte Carlo (1NMC) simulation. As 1NMC is two layers of Monte Carlo simulations, the inner trajectories

*Email: lokmane.abbas_turki@sorbonne-universite.fr. LPSM (UMR 8001), Sorbonne Université, 4 Place Jussieu 75005 Paris, France & CMAP (UMR 7641), École Polytechnique, Route de Saclay 91128 Palaiseau cedex, France.

[†]Email: babacardiallopro@gmail.com. LaMME (UMR 8071), Université Paris-Saclay, 23 Boulevard de France 91037 Evry, France.

[‡]Email: gilles.pages@sorbonne-universite.fr. LPSM (UMR 8001), Sorbonne Université, 4 Place Jussieu 75005 Paris, France.

are used for regressions when the outer ones are used for localization purpose. The conditional learning stands then for regressions conditional to various starting points on the outer trajectories.

In the present work as well as in CMCLDI [2], we choose local regressions even though it can be replaced by local Neural Networks (NNs). Unlike [10, 12, 21] where NNs are global on the whole Monte Carlo realizations, regressions are sufficient when made local using 1NMC and their implementation is also very efficient using batch parallel processing [3]. We already demonstrated the effectiveness of our methodology on a large number of Backward Stochastic Differential Equations (BSDEs) examples presented in CMCLDI. We also showed that the bias backpropagation can be removed using a new simple trick that is only possible because of the nested nature of the proposed algorithm.

In this contribution, CMCLD part II (CMCLDII), we push further the capabilities of CMCLDI (cf. [2]) using a small modification that allows to get accurate approximations of the variance of the simulated functionals of diffusions. This modification plays an important role to simulate tail events necessary to risk measures and optimal stopping strategies. To make as simple as possible the explanation of the introduced modifications, we use the same generic example presented in [2].

On a filtered probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{0 \leq t \leq T}, \mathbb{P})$, the generic example is driven by an \mathcal{F}_t -Markov process $(X_t)_{t \in [0, T]}$ taking its values on \mathbb{R}^{d_1} . Given the fine time discretization $\mathbb{S} = \{t_0, \dots, t_{2^L}\} = \{0, T/2^L, \dots, T\}$, the functional U_s of X is defined for $s \in \mathbb{S}$ by

$$(f) \quad U_s = u_s(X_s) = \mathbb{E}_s \left(\sum_{k=s2^L/T}^{2^L} f(t_k, X_{t_k}, X_{t_{k+1}}) \right) = \mathbb{E} \left(\sum_{s \leq t_k \leq T} f(t_k, X_{t_k}, X_{t_{k+1}}) \middle| \mathcal{F}_s \right),$$

where $\mathbb{E}_s(\cdot) = \mathbb{E}(\cdot | \mathcal{F}_s)$, the expectation is always considered under \mathbb{P} , each deterministic function $f(t_k, \cdot, \cdot)$ is $\mathcal{B}(\mathbb{R}^{d_1}) \otimes \mathcal{B}(\mathbb{R}^{d_1})$ -measurable and assumed to satisfy the square integrability condition $\mathbb{E}(f^2(t_k, X_{t_k}, X_{t_{k+1}})) < +\infty$ with convention $f(t_{2^L}, X_{t_{2^L}}, X_{t_{2^L+1}}) = f(t_{2^L}, X_{t_{2^L}})$. From the Markov assumption, for each $s \in \mathbb{S}$, the $\mathcal{B}(\mathbb{R}^{d_1})$ -measurable function u_s is deterministic.

1NMC requires the simulation of a first layer $(X^{m_0})_{m_0=1, \dots, M_0}$ of outer trajectories of the process X that are kept in the machine's random-access memory, then a second $(X^{m_0, m_1})_{m_1=1, \dots, M_1}$ unstored layer of inner trajectories, on the top of each outer one indexed by m_0 . Introducing a discretization set $\mathcal{S} \subset \mathbb{S}$ for the process U , it is then possible to define an inner simulation $U^{m_0, m_1} = \bar{u}^{m_0, \mathcal{S}}(X^{m_0, m_1})$ and an outer simulation $U^{m_0} = \tilde{u}^{m_0, \mathcal{S}}$. Defined in CMCLDI, $\bar{u}^{m_0, \mathcal{S}}$ and $\tilde{u}^{m_0, \mathcal{S}}$ are respectively the coarse and the fine approximations of the deterministic function u introduced in (f). The approximation $\tilde{u}^{m_0, \mathcal{S}}$ is considered as an average value on the realizations $\bar{u}^{m_0, \mathcal{S}}(X^{m_0, m_1})$ and thus it averages out and reduces the approximation errors of $\bar{u}^{m_0, \mathcal{S}}$.

To improve further the coarse approximation, we replace the expression

$$\bar{u}_{s_j, s_k}^{m_0, \mathcal{S}}(x) = \tilde{u}_{s_k, \bar{s}_k}^{m_0, \mathcal{S}} + {}^t\mathcal{T}_{s_j, s_k, M_1'}^{m_0}(x) B_{s_j, s_k}^{m_0, \mathcal{S}}, \quad (1.1)$$

given in CMCLDI by

$$\bar{u}_{s_j, s_k}^{m_0, \mathcal{S}}(x) = \ell \left[\bar{u}_{s_j, s_k}^{m_0, \mathcal{S}} \right] + {}^t\mathcal{T}_{s_j, s_k, M_1'}^{m_0}(x) \widehat{B}_{s_j, s_k}^{m_0, \mathcal{S}}, \quad (1.2)$$

with

$$\widehat{B}_{s_j, s_k}^{m_0, \mathcal{S}} = \gamma_{s_j, s_k}^{m_0, \mathcal{S}} B_{s_j, s_k}^{m_0, \mathcal{S}} \quad (1.3)$$

and

$$\ell\left[\bar{u}_{s_j, s_k}^{m_0, \mathcal{S}}\right] = \tilde{u}_{s_k, \bar{s}_k}^{m_0, \mathcal{S}} + \frac{(1 - \gamma_{s_j, s_k}^{m_0, \mathcal{S}})}{M_1} \sum_{m_1=1}^{M_1} {}^t \mathcal{T}_{s_j, s_k, M_1}^{m_0} (X_{s_j, s_k}^{m_0, m_1}) B_{s_j, s_k}^{m_0, \mathcal{S}}, \quad (1.4)$$

where t is used for the transpose operator, $s_j < s_k \leq \bar{s}_k$ are increments from \mathcal{S} , $\mathcal{T}_{s_j, s_k, M_1}^{m_0}(x)$ is the considered regression basis, $B_{s_j, s_k}^{m_0, \mathcal{S}}$ is the regression vector and $\gamma_{s_j, s_k}^{m_0, \mathcal{S}}$ is the variance adjustment parameter.

The factor $\gamma_{s_j, s_k}^{m_0, \mathcal{S}}$ is used to adjust the variance of $\bar{u}_{s_j, s_k}^{m_0, \mathcal{S}}$ defined in (1.2) without changing its average value. Indeed, the expression of $\ell\left[\bar{u}_{s_j, s_k}^{m_0, \mathcal{S}}\right]$ makes the value $\sum_{m_1=1}^{M_1} \frac{\bar{u}_{s_j, s_k}^{m_0, \mathcal{S}}(X_{s_j, s_k}^{m_0, m_1})}{M_1}$ invariant with respect to $\gamma_{s_j, s_k}^{m_0, \mathcal{S}}$ because equality (1.2) yields

$$\sum_{m_1=1}^{M_1} \frac{\bar{u}_{s_j, s_k}^{m_0, \mathcal{S}}(X_{s_j, s_k}^{m_0, m_1})}{M_1} = \tilde{u}_{s_k, \bar{s}_k}^{m_0, \mathcal{S}} + \frac{1}{M_1} \sum_{m_1=1}^{M_1} {}^t \mathcal{T}_{s_j, s_k, M_1}^{m_0} (X_{s_j, s_k}^{m_0, m_1}) B_{s_j, s_k}^{m_0, \mathcal{S}}.$$

Consequently, for each $m_0 = 1, \dots, M_0$, the benefit of (1.2) compared to (1.1) would be to better simulate the second moment $u.(X^{m_0, m_1})$ without changing the accuracy of the first moment given that the latter is generally very well approximated using regressions.

In the same fashion as for stopping bias backpropagation (cf. CMCLDI [2]), the variance adjustment is based on the tower property used to obtain the equality

$$\mathbb{E}(\text{Var}_s(U_{s'})) = \mathbb{E}(\mathbb{E}_s([U_{s'} - \mathbb{E}_s(U_{s'})]^2)) = \mathbb{E}([U_{s'} - \mathbb{E}_s(U_{s'})]^2)$$

true for $s' > s > 0$. For $s = s_j$ and $s' = s_k$ with (s_j, s_k) as above, this equality allows us to choose the adjustment parameter $\gamma_{s, s'}^{m_0, \mathcal{S}}$ that makes the average variance $\mathbb{E}(\text{Var}_s(U_{s'}))$ estimated by $\frac{1}{M_0} \sum_{m_0=1}^{M_0} \left(\frac{1}{M_1} \sum_{m_1=1}^{M_1} \left[\bar{u}_{s, s'}^{m_0, \mathcal{S}}(X_{s, s'}^{m_0, m_1}) - \sum_{m_1=1}^{M_1} \bar{u}_{s, s'}^{m_0, \mathcal{S}}(X_{s, s'}^{m_0, m_1}) / M_1 \right]^2 \right)$ equal to $\mathbb{E}([U_{s'} - \mathbb{E}_s(U_{s'})]^2)$ estimated by $\frac{1}{M_0} \sum_{m_0=1}^{M_0} \left(\left[\tilde{u}_{s', \bar{s}'}^{m_0, \mathcal{S}} - \sum_{m_1=1}^{M_1} \bar{u}_{s, s'}^{m_0, \mathcal{S}}(X_{s, s'}^{m_0, m_1}) / M_1 \right]^2 \right)$.

Thus, the value of $\gamma_{s, s'}^{m_0, \mathcal{S}}$ improves significantly the accuracy of the former estimator using the latter one. Using numerically the tower property to get the variance adjustment parameter $\gamma^{m_0, \mathcal{S}}$ is made possible by the nested nature of the proposed simulation. We take fully advantage of this (nested structure, tower property) combination for the proposed variance adjustment trick. Other applications could benefit from this combination to improve the simulation result.

The rest of this paper is organized as follows. In Section 2, we recall the main steps of the algorithm including the variance adjustment modification and its adaptation to an optimal stopping problem. In Section 3, we express the value of $\gamma^{m_0, \mathcal{S}}$, provide an iterative construction of the discretization set \mathcal{S} and explore in-depth the use of the tower property for local bias control and local variance adjustment. Previously presented in CMCLDI, stopping the bias backpropagation comes at the expense of an additional error term studied in Section 4. Section 5 concludes this paper with a successful simulation of both an American option and a risk measure examples in very high dimensions.

2 Methodology extension and application to optimal stopping problems

In [2], we presented the algorithm steps and what should be done to stabilize it. We explained the whole methodology on a simple problem and we showed its benefits when one has to simulate the composition of functionals of a Markov process. Large number of BSDEs and Reflected BSDEs (RBSDEs) are specific composition of functionals of a forward process assumed Markov in various situations. In [2], we already presented the adaptation of our methodology on the One step forward Dynamic Programming (ODP) discretization scheme (cf. [16, 25]) of BSDEs. In this section, we start by briefly introducing the needed notations then explain the new modified version of our algorithm on the generic example defined in (f). We finally apply the new modified version to a RBSDE without a Z term.

Starting with notations, we introduce $\mathbb{S} = \{t_0, \dots, t_{2L}\} = \{0, \Delta_t, 2\Delta_t, \dots, T\}$ as a fine discretization set and $\mathcal{S} \subset \mathbb{S}$ as a coarser one with $\Delta_t = T/2^L$. We then simulate the outer trajectories $(X^{m_0})_{m_0=1, \dots, M_0}$ as well as the inner ones $(X_{s_j, \cdot}^{m_0, m_1})_{s_j \in \mathcal{S}}^{m_0=1, \dots, M_0, m_1=1, \dots, M_1}$ that start at the top of each realization $X_{s_j}^{m_0}$ with $s_j \in \mathcal{S}$. Given the Borel-measurable functions $(\mathcal{E}_{t_k})_{k=0, \dots, 2L-1} : \mathbb{R}^{d_1+d_2} \rightarrow \mathbb{R}^{d_1}$, we define

$$X_{t_k}^{m_0} = \mathcal{E}_{t_{k-1}}(X_{t_{k-1}}^{m_0}, \xi_{t_k}^{m_0}), \text{ when } k \geq 1 \text{ and } X_{t_0}^{m_0} = x_0, \quad (2.1)$$

$$X_{s_j, t_k}^{m_0, m_1} = \mathcal{E}_{t_{k-1}}(X_{s_j, t_{k-1}}^{m_0, m_1}, \xi_{s_j, t_k}^{m_0, m_1}) \text{ and } X_{s_j, s_j}^{m_0, m_1} \Big|_{m_1=1, \dots, M_1+M'_1} = X_{s_j}^{m_0}. \quad (2.2)$$

Since we focus essentially on the diffusion case, the functions $(\mathcal{E}_{t_k})_{k=0, \dots, 2L-1}$ can be obtained with a discretization scheme to get for instance

$$\mathcal{E}_{t_k}(x, \xi) = x + \Delta_t b(t_k, x) + \sigma(t_k, x)\xi \quad (2.3)$$

with the usual (cf. [28]) Lipschitz continuity condition on the coefficients $b(t, x)$ and $\sigma(t, x)$ uniformly with respect to $t \in [0, T]$. Introducing outer W^{m_0} and inner W^{m_0, m_1} realizations of an \mathbb{R}^{d_2} -Brownian motion W , we define $\xi_{t_k}^{m_0} = W_{t_k}^{m_0} - W_{t_{k-1}}^{m_0}$ and $\xi_{s_j, t_k}^{m_0, m_1} = W_{s_j, t_k}^{m_0, m_1} - W_{s_j, t_{k-1}}^{m_0, m_1}$ where W^1, \dots, W^{M_0} are independent with

$$W_{s_j, t_k}^{m_0, m_1} = W_{s_j, t_{k-1}}^{m_0, m_1} + \Delta W_{s_j, t_k}^{m_0, m_1} \text{ and } W_{s_j, s_j}^{m_0, m_1} \Big|_{m_1=1, \dots, M_1+M'_1} = W_{s_j}^{m_0}$$

and $(\Delta W_{s_j, t_k}^{m_0, m_1})_{k \in \{j, \dots, 2L\}, j \in \{1, \dots, 2L\}}^{(m_0, m_1) \in \{1, \dots, M_0\} \times \{1, \dots, M_1+M'_1\}}$ are independent Brownian motion increments independent from W^1, \dots, W^{M_0} with $\mathbb{E}([\Delta W_{s_j, t_k}^{m_0, m_1}]^2) = \Delta_t$. The resulting components of the vector $\left((\xi_{s_j, t_k}^{m_0, m_1})_{k \in \{j, \dots, 2L\}, j \in \{1, \dots, 2L\}}^{(m_0, m_1) \in \{1, \dots, M_0\} \times \{1, \dots, M_1+M'_1\}}, (\xi_{t_k}^{m_0})_{k=1, \dots, 2L}^{m_0=1, \dots, M_0} \right)$ are independent realizations with the same normal distribution.

We use $X_{\cdot, \cdot}^{m_0, m_1, 1}, \dots, X_{\cdot, \cdot}^{m_0, m_1, d_1}$ to denote the d_1 components of $X_{\cdot, \cdot}^{m_0, m_1}$ and use the notation $X_{s_j, s_k}^{m_0, m_1}$ for $(X_{s_j, s_l}^{m_0, m_1}, X_{s_j, s_l+\Delta_t}^{m_0, m_1}, \dots, X_{s_j, s_k-\Delta_t}^{m_0, m_1}, X_{s_j, s_k}^{m_0, m_1})$ for $s_j, s_k \in \mathcal{S}$ with $s_j < s_k$. We remind also the notation of the stabilized regression basis $\mathcal{T}_{t_j, t_k, M'_1}^{m_0}$ defined in CMCLDI (cf. [2]) for any couple $t_j < t_k$ from the fine discretization grid. The cardinal of the basis $\mathcal{T}_{\cdot, \cdot, M'_1}^{m_0}$ is equal to d'_1 with $d'_1 \leq d_1$. $\mathcal{T}_{\cdot, \cdot, M'_1}^{m_0}$ is also indexed by the number M'_1 of trajectories needed for the approximation of regression matrices and that are independent from those used in the backward induction. We also reduce any superfluous dependence structure

as well as the memory occupation by re-simulating on the fly each realization that has not to be constantly stored. For example, if an inner trajectory is needed α times in the backward induction, we simulate α independent copies of it and use each copy once.

Finally, introducing \wedge and \vee notations for

$$x \wedge y = \min(x, y) \text{ and } x \vee y = \max(x, y),$$

we remind the definition of the time operator $\delta^{\mathcal{S}}$ that associates to each $s \in \mathcal{S}$ the next increment in \mathcal{S} . For a fixed index $j \in \{1, \dots, 2^L\}$, we define $\delta_{s_j}^{\mathcal{S}}(\cdot)$ on $(s_k)_{k \leq j}$, taken its values in $\mathcal{S} \cap [s_j, \bar{s}_j]$, by

$$\delta_{s_j}^{\mathcal{S}}(s_k) = \bar{s}_j \wedge \min\{s \in \mathcal{S}; s_k < s \leq \bar{s}_j\} \quad (2.4)$$

with $\min(\emptyset) = \infty$ and the couple (s_j, \bar{s}_j) sets the minimal and the maximal time increment of the performed regressions conditionally on the realization $X_{s_j}^{m_0}$ of the Markov process X . The couple (\underline{s}, \bar{s}) , mainly \bar{s} , sets the learning depth conditional to the information available at time s . Given a coarse discretization set \mathcal{S} , definitions 2.1 and 2.2 express the value of $(\underline{s}, \bar{s})_{s \in \mathcal{S}}$.

When \mathcal{S} is fixed, the notation $\delta_{s_j}^{\mathcal{S}}$ can be replaced by δ_{s_j} . When $s_k < \bar{s}_j$, the notation $\delta_{s_j}^{\mathcal{S}}$ can be replaced by $\delta^{\mathcal{S}}$. When \mathcal{S} is fixed and $s_k < \bar{s}_j$, we can simplify both indices and use δ instead of $\delta_{s_j}^{\mathcal{S}}$. Given a coarse discretization set \mathcal{S} , the operator δ has the following properties

Pr1. $\underline{s}_j = \delta_{s_j}(s_j) = \delta(s_j)$.

Pr2. As long as $s_{j_1} \vee s_{j_2} \leq s_k < \bar{s}_{j_1} \wedge \bar{s}_{j_2}$, $\delta_{s_{j_1}}(s_k) = \delta_{s_{j_2}}(s_k) = \delta(s_k)$.

Pr3. The n th composition of δ_{s_j} denoted $\delta_{s_j}^n(\cdot)$ is equal to \bar{s}_j when $n \geq |\mathcal{S} \cap]s_j, \bar{s}_j]|$ where $|\cdot|$ denotes the cardinal.

2.1 Variance adjustment modification on the generic example

When applied for the simulation of U defined by (f), the proposed learning procedure associates to each scenario m_0 and to each discretization set \mathcal{S} a couple of function families $(\tilde{u}^{m_0, \mathcal{S}}, \bar{u}^{m_0, \mathcal{S}})$. Given $s_j < s_k \leq \bar{s}_k$ increments from \mathcal{S} , equations (1.2), (1.3) and (1.4) parametrizes the definition of $\bar{u}^{m_0, \mathcal{S}}$ with respect to the variance adjustment parameter $\gamma_{s_j, s_k}^{m_0, \mathcal{S}}$. Thus, we generalize the definition of $\bar{u}^{m_0, \mathcal{S}}$ introduced in CMCLDI (cf. [2]) which is obtained straightforwardly if we take $\gamma_{s_j, s_k}^{m_0, \mathcal{S}} = 1$. Regarding the regression vector $B_{s_j, s_k}^{m_0, \mathcal{S}}$ used in (1.3), its value is obtained from an estimation of the vector $a \in \mathbb{R}^{d_1}$ that minimizes the quadratic error given by

$$\mathbb{E} \left[\mathbb{B}_{s_j, s_k}^{m_0, \mathcal{S}, \delta_{s_j}(s_k)} (X_{s_j, s_k: \delta_{s_j}(s_k)}^{m_0, m_1}) - {}^t a \mathcal{T}_{s_j, s_k, M_1'}^{m_0} (X_{s_j, s_k}^{m_0, m_1}) \right]^2 \quad (2.5)$$

with $X_{s_j, s_k: \delta_{s_j}(s_k)}^{m_0, m_1} = \left(X_{s_j, s_k}^{m_0, m_1}, X_{s_j, s_k + \Delta t}^{m_0, m_1}, \dots, X_{s_j, \delta_{s_j}(s_k) - \Delta t}^{m_0, m_1}, X_{s_j, \delta_{s_j}(s_k)}^{m_0, m_1} \right)$ and

$$\mathbb{B}_{s_j, s_k}^{m_0, \mathcal{S}, \delta_{s_j}(s_k)}(x) = \bar{u}_{s_j, \delta_{s_j}(s_k)}^{m_0, \mathcal{S}} \left(x \frac{\delta_{s_j}(s_k) - s_k}{\Delta t} \right) - \tilde{u}_{s_k, s_k}^{m_0, \mathcal{S}} + \sum_{l=1}^{\frac{\delta_{s_j}(s_k) - s_k}{\Delta t} - 1} f(t_{k_{s_k} + l}, x_l, x_{l+1}) \quad (2.6)$$

plus the projection of $\sum_{t_{l+1} > s_k}^{t_2L} f(t_l, X_{s_j, t_l}^{m_0, m_1}, X_{s_j, t_{l+1}}^{m_0, m_1}) + f(T, X_{s_j, T}^{m_0, m_1})$ as shown on Figure 1.

$\tilde{u}_{s_k, \bar{s}_k}^{m_0, \mathcal{S}}$ is equal to $\frac{1}{M_1} \sum_{m_1=1}^{M_1} \left[\sum_{t_{l+1} > s_k}^{t_2L} f(t_l, X_{s_k, t_l}^{m_0, m_1}, X_{s_k, t_{l+1}}^{m_0, m_1}) + f(T, X_{s_k, T}^{m_0, m_1}) \right]$ as $\delta(s_k) = T$.

Once we replaced (1.1) by (1.2), Definition 2.1 hereafter is similar to Definition 3.1 presented in CMCLDI.

Definition 2.1. Given a discretization set $\mathcal{S} \subset \mathbb{S}$

- For any $s \in \mathcal{S}$, \underline{s} is set to be equal to $\delta(s)$ and \bar{s} is set backwardly to be the largest discretization time $u \in \mathcal{S} \cap]s, \delta(s)]$ that satisfies

$$\left| \frac{1}{M_0} \sum_{m_0=1}^{M_0} \left(\tilde{u}_{s, u}^{m_0, \mathcal{S}} - \tilde{u}_{\delta(s), \delta(s)}^{m_0, \mathcal{S}} - \sum_{t_{l+1} > s}^{\delta(s)} f(t_l, X_{t_l}^{m_0}, X_{t_{l+1}}^{m_0}) \right) \right| < \epsilon_s^{\mathcal{S}} \quad (2.10)$$

where $\{\epsilon_s^{\mathcal{S}}\}_{s \in \mathcal{S}}$ is a family of positive bias tuning parameters.

- For $k < j \in \{1, \dots, 2^L\}$ that satisfy $s_j < s_k \leq \bar{s}_j < t_2L = T$, the simulation $U_{s_j, s_k}^{m_0, m_1}$ of U around $X_{s_k}^{m_0}$ conditionally on $X_{s_j}^{m_0}$ is set to be equal to $\bar{u}_{s_j, s_k}^{m_0, \mathcal{S}}(X_{s_j, s_k}^{m_0, m_1})$ where \bar{u} is given in (1.2) and (2.8).
- For $k \in \{1, \dots, 2^L\}$ and $s \in \{s_k, s_k + \Delta_t, \dots, \delta(s_k) - \Delta_t\} - \{0\}$, the simulation $U_s^{m_0}$ of U at $X_s^{m_0}$ is set to be equal to $\tilde{u}_{s, \bar{s}_k}^{m_0, \mathcal{S}}$ with \tilde{u} expressed in (2.7).
- The average U_0^{lear} of learned values on U_0 is equal to

$$U_0^{\text{lear}} = \frac{1}{M_0} \sum_{m_0=1}^{M_0} \tilde{u}_{0, \bar{0}}^{m_0, \mathcal{S}} \quad (2.11)$$

and the simulated value U_0^{sim} of U_0 is equal to

$$U_0^{\text{sim}} = \frac{1}{M_0} \sum_{m_0=1}^{M_0} \left[\tilde{u}_{\delta(0), \delta(0)}^{m_0, \mathcal{S}} + \sum_{t_{l+1} > 0}^{\delta(0)} f(t_l, X_{t_l}^{m_0}, X_{t_{l+1}}^{m_0}) \right] \quad (2.12)$$

with \tilde{u} expressed in (2.7).

In contrast to the inner layer approximation U^{m_0, m_1} which is only defined on \mathcal{S} , the outer layer approximation U^{m_0} is defined on the whole fine discretization set \mathbb{S} . At $t = 0$, inner and outer trajectories coincide and yield U_0^{lear} and U_0^{sim} as two possible approximations of U_0 . The value of \bar{s} depends on $\delta(s)$ whose value involves the one of $\delta^2(s)$ and so on till T . In Section 3.1, we provide an example of how to update iteratively $(\underline{s}, \bar{s})_{s \in \mathcal{S}}$ with respect to the discretization set \mathcal{S} .

It is remarkable that the term on the left of inequality (2.10) makes possible the estimation of the average bias. In Section 3.2, we introduce a local bias estimation that allows a more stringent bias control. None of these bias estimations is possible without the double layer structure of NMC1. The choice of the tuning parameters $\{\epsilon_s^{\mathcal{S}}\}_{s \in \mathcal{S}}$ reflects a compromise between the targetted accuracy and the required complexity. This choice has to take into account the standard deviation of the estimator

$$\frac{1}{M_0} \sum_{m_0=1}^{M_0} \left(\tilde{u}_{s, u}^{m_0, \mathcal{S}} - \tilde{u}_{\delta(s), \delta(s)}^{m_0, \mathcal{S}} - \sum_{t_{l+1} > s}^{\delta(s)} f(t_l, X_{t_l}^{m_0}, X_{t_{l+1}}^{m_0}) \right).$$

2.2 Application on RBSDEs with a Markov forward process

The variance adjustment trick introduced in this paper was not needed for the BSDE examples studied in CMCLDI. However, when the considered example requires a better description of the tail distribution, the choice $\gamma_{\cdot, \cdot}^{m_0, \mathcal{S}} = 1$ becomes insufficient. Here, we consider an application to RBSDEs like the one presented in [7] with the final condition function $g(\cdot)$ and the driver $\{f(t_k, \cdot)\}_{k=0}^{2^L-1}$ assumed to be \mathcal{L} -Lipschitz. We want to propose a double layer approximation V^{m_0} and V^{m_0, m_1} of the Snell envelope V , solution to

$$(Snl) \quad V_T = g(X_T) \text{ and for } k < 2^L : V_{t_k} = g(X_{t_k}) \vee \mathbb{E}_{t_k}[V_{t_{k+1}} + \Delta_t f(t_k, V_{t_{k+1}})],$$

that can be done using straightforwardly the recipe of Section 2.1 combined with a maximization by g . We introduce the coarse $\bar{w}^{m_0, \mathcal{S}}$ and the fine $\tilde{w}^{m_0, \mathcal{S}}$ approximations of the process \underline{V} defined by

$$\underline{V}_{t_k} = \mathbb{E}_{t_k}[V_{t_{k+1}} + \Delta_t f(t_k, V_{t_{k+1}})]. \quad (2.13)$$

Then, given a discretization set \mathcal{S} and indices $k < j \in \{1, \dots, 2^L\}$ that satisfy $s_j < s_k < \bar{s}_j$ and using $\delta_{s_j}(s_k)$ defined in (2.4), we set the coarse approximation \bar{v}_{s_j, s_k} around $X_{s_k}^{m_0}$ conditionally on $X_{s_j}^{m_0}$ to

$$\bar{v}_{s_j, s_k}^{m_0, \mathcal{S}}(x) = \bar{w}_{s_j, s_k}^{m_0, \mathcal{S}}(x) \vee g(x), \quad (2.14)$$

and the fine approximation \tilde{v}_{s, \bar{s}_k} at $X_s^{m_0}$, $s \in \{s_k, s_k + \Delta_t, \dots, \delta_{s_j}(s_k) - \Delta_t\}$, to

$$\tilde{v}_{s, \bar{s}_k}^{m_0, \mathcal{S}} = \tilde{w}_{s, \bar{s}_k}^{m_0, \mathcal{S}} \vee g(X_s^{m_0}). \quad (2.15)$$

Denoting $\Delta_s = \delta_{s_j}(s_k) - s$ and $\Delta_{s_k} = \delta_{s_j}(s_k) - s_k$, we define

$$\tilde{w}_{s, \bar{s}_k}^{m_0, \mathcal{S}} = \frac{1}{M_1} \sum_{m_1=1}^{M_1} \left(\Delta_s f(s_k, \bar{v}_{s_k, \delta_{s_j}(s_k)}^{m_0, \mathcal{S}}(X_{s, \delta_{s_j}(s_k)}^{m_0, m_1})) + \bar{v}_{s_k, \delta_{s_j}(s_k)}^{m_0, \mathcal{S}}(X_{s, \delta_{s_j}(s_k)}^{m_0, m_1}) \right), \quad (2.16)$$

$$\bar{w}_{s_j, s_k}^{m_0, \mathcal{S}}(x) = \ell \left[\bar{w}_{s_j, s_k}^{m_0, \mathcal{S}} \right] + {}^t \mathcal{T}_{s_j, s_k, M_1'}^{m_0}(x) \widehat{C}_{s_j, s_k}^{m_0, \mathcal{S}}, \quad (2.17)$$

where

$$\ell \left[\bar{w}_{s_j, s_k}^{m_0, \mathcal{S}} \right] = \tilde{w}_{s_k, \bar{s}_k}^{m_0, \mathcal{S}} + \frac{(1 - \gamma_{s_j, s_k}^{m_0, \mathcal{S}})}{M_1} \sum_{m_1=1}^{M_1} {}^t \mathcal{T}_{s_j, s_k, M_1'}^{m_0}(X_{s_j, s_k}^{m_0, m_1}) C_{s_j, s_k}^{m_0, \mathcal{S}}, \quad (2.18)$$

$\widehat{C}_{s_j, s_k}^{m_0, \mathcal{S}} = \gamma_{s_j, s_k}^{m_0, \mathcal{S}} C_{s_j, s_k}^{m_0, \mathcal{S}}$ with

$$C_{s_j, s_k}^{m_0, \mathcal{S}} = \frac{(\tilde{\Lambda}_{s_j, s_k, M_1'}^{m_0})^{-1}}{M_1} \sum_{m_1=1}^{M_1} C_{s_j, s_k, M_1'}^{m_0, \mathcal{S}, \delta_{s_j}(s_k)}(X_{s_j, s_k}^{m_0, m_1}, X_{s_j, \delta_{s_j}(s_k)}^{m_0, m_1})$$

and $C_{s_j, s_k, M_1'}^{m_0, \mathcal{S}, \delta_{s_j}(s_k)}(x', x) = \mathcal{T}_{s_j, s_k, M_1'}^{m_0}(x') C_{s_j, s_k}^{m_0, \mathcal{S}, \delta_{s_j}(s_k)}(x)$ with

$$C_{s_j, s_k}^{m_0, \mathcal{S}, \delta_{s_j}(s_k)}(x) = \begin{bmatrix} \Delta_{s_k} f(s_k, \bar{v}_{s_j, \delta_{s_j}(s_k)}^{m_0, \mathcal{S}}(x)) \\ + \bar{v}_{s_j, \delta_{s_j}(s_k)}^{m_0, \mathcal{S}}(x) - \tilde{w}_{s_k, \bar{s}_k}^{m_0, \mathcal{S}} \end{bmatrix}, \quad (2.19)$$

with a final coarse approximation given by

$$\bar{v}_{s_j, \bar{s}_j}^{m_0, \mathcal{S}}(x) = \begin{cases} g(x) & \text{if } \bar{s}_j = t_{2^L}, \\ \bar{v}_{\underline{s}_j, \bar{s}_j}^{m_0, \mathcal{S}}(x) = \bar{v}_{\delta_{s_j}(s_j), \bar{s}_j}^{m_0, \mathcal{S}}(x) & \text{if } \bar{s}_j < t_{2^L}. \end{cases} \quad (2.20)$$

Definition 2.2. Given a discretization set $\mathcal{S} \subset \mathbb{S}$

- For any $s \in \mathcal{S}$, \underline{s} is set to be equal to $\delta(s)$ and \bar{s} is set backwardly to be the largest discretization time $u \in \mathcal{S} \cap]s, \delta(s)]$ that satisfies

$$\left| \frac{1}{M_0} \sum_{m_0=1}^{M_0} \left(\tilde{w}_{s,u}^{m_0, \mathcal{S}} - \tilde{v}_{\delta(s), \delta(s)}^{m_0, \mathcal{S}} - (\delta(s) - s) f(s, \tilde{v}_{\delta(s), \delta(s)}^{m_0, \mathcal{S}}) \right) \right| < \epsilon_s^{\mathcal{S}} \quad (2.21)$$

where $\{\epsilon_s^{\mathcal{S}}\}_{s \in \mathcal{S}}$ is a family of positive bias tuning parameters.

- For $k < j \in \{1, \dots, 2^L\}$ that satisfy $s_j < s_k \leq \bar{s}_j < t_{2^L} = T$, the simulation $V_{s_j, s_k}^{m_0, m_1}$ of V around $X_{s_k}^{m_0}$ conditionally on $X_{s_j}^{m_0}$ is set to be equal to $\bar{v}_{s_j, s_k}^{m_0, \mathcal{S}}(X_{s_j, s_k}^{m_0, m_1})$ where \bar{v} is given in (2.14), (2.17) and (2.20).
- For $k \in \{1, \dots, 2^L\}$ and $s \in \{s_k, s_k + \Delta_t, \dots, \delta(s_k) - \Delta_t\} - \{0\}$, the simulation $V_s^{m_0}$ of V at $X_s^{m_0}$ is set to be equal to $\tilde{v}_{s, \bar{s}_k}^{m_0, \mathcal{S}}$ with \tilde{v} expressed in (2.15) and (2.16).
- The average V_0^{lear} of the learned values on V_0 is equal to

$$V_0^{\text{lear}} = \frac{1}{M_0} \sum_{m_0=1}^{M_0} \tilde{v}_{0, \bar{0}}^{m_0, \mathcal{S}}, \quad (2.22)$$

and the simulated value V_0^{sim} of V_0 is equal to

$$V_0^{\text{sim}} = g(x_0) \vee \frac{1}{M_0} \sum_{m_0=1}^{M_0} \left[\delta(0) f\left(\delta(0), \tilde{v}_{\delta(0), \delta(0)}^{m_0, \mathcal{S}}\right) + \tilde{v}_{\delta(0), \delta(0)}^{m_0, \mathcal{S}} \right]. \quad (2.23)$$

Similar to Definition 2.1, the inner layer approximation V^{m_0, m_1} is only defined on \mathcal{S} when the outer layer approximation V^{m_0} is defined on the whole fine discretization set \mathbb{S} . At $t = 0$, inner and outer trajectories coincide and yield V_0^{lear} and V_0^{sim} as two possible approximations of V_0 . Similar also to Definition 2.1, it is remarkable that the term on the left of inequality (2.21) makes possible the estimation of the average bias.

Although using an optimal stopping formulation [11] of the dynamic programming is known to provide better numerical results [26], we preferred here to use 1NMC on the top of the original algorithm [30] since its error estimates remains similar to the one presented for BSDEs. As a future work, we would like to apply variance reduction methods with 1NMC and provide very accurate double layer estimations of the optimal stopping strategy.

3 Learning the values of \mathcal{S} and $\left\{ \gamma_{s_j, s_k}^{m_0, \mathcal{S}} \right\}_{m_0=1, \dots, M_0, s_j \in \mathcal{S}, s_k \in \mathcal{S} \cap]s_j, \bar{s}_j]}$

Till now, the proposed learning methodology introduced M_0 , M_1 , M'_1 , \mathcal{S} , $\epsilon_s^{\mathcal{S}}$ and $\gamma_{\cdot, \cdot}^{\mathcal{S}}$ as hyperparameters. In a Monte Carlo simulation, the standard and efficient way to know what is the appropriate number of trajectories that have to be simulated is through the estimated confidence interval. The same can be done for the number of outer and inner trajectories in 1NMC. We only draw attention to the number of trajectories M_1 and M'_1 that have to be sufficient given the number of terms d'_1 in $\mathcal{T}_{\cdot, \cdot, M'_1}^{m_0}$ used for the regression.

We refer the reader to [27] for asymptotic relations between the size of the projection basis and the number of needed realizations. Consequently, the choice of M_0 , M_1 and M'_1 have to be specific to each application and set during the numerical simulation. The same can be said for the tuning parameter $\epsilon^{\mathcal{S}}$ as its value should be proportional to $1/\sqrt{M_0}$. Indeed, from the left term in inequalities (2.10) and (2.21), the user has to make a numerical compromise between accuracy and complexity when picking the value of $\epsilon^{\mathcal{S}}$.

In this section, we want to relieve the use of \mathcal{S} and of $\gamma_{\cdot, \cdot}^{\mathcal{S}}$ as hyperparameters. We start by specifying iteratively the coarse discretization set \mathcal{S} . At each iteration, we show also how to update the couple (\underline{s}, \bar{s}) for any $s \in \mathcal{S}$. The iterative choice of \mathcal{S} and of $(\underline{s}, \bar{s})_{s \in \mathcal{S}}$ is made to keep the bias under control and thus to conserve the first moment of simulations. Moreover, we detail how to conserve the second moment of simulations using an appropriate value for the variance adjustment parameter $\gamma_{\cdot, \cdot}^{\mathcal{S}}$. The latter choice is further refined when the number M_0 of simulated outer trajectories makes possible a local variance adjustment.

The iterative construction of \mathcal{S} and $(\underline{s}, \bar{s})_{s \in \mathcal{S}}$ presented in Section 3.1 as well as the expression of $\gamma_{\cdot, \cdot}^{\mathcal{S}}$ presented in Section 3.2 are only given as an example and thus are not restrictive.

3.1 An example of iterative definition of \mathcal{S} and update of $(\underline{s}, \bar{s})_{s \in \mathcal{S}}$

Like definitions 3.1 and 3.2 in CMCLDI, definitions 2.1 and 2.2 in Section 2 are intentionally announced without specifying the coarse discretization set \mathcal{S} . Indeed, the presented methodology can be applied for any \mathcal{S} . It is then possible to use it for an iterative family $\{\mathcal{S}^i\}_{i=0, \dots, L-L'}$ where L' is an integer chosen in this section $\in]L/2, L]$. When i increases the coarse discretization gets finer in the sense of inclusion $\mathcal{S}^0 \subseteq \mathcal{S}^1 \subseteq \dots \subseteq \mathcal{S}^{L-L'}$.

For $i = 0, \dots, L - L'$, we define the set \mathcal{S}^i through the values taken by the decreasing sequence $(T - s_j^i)_{j=0, \dots, 2^L}$. When $i = 0$, $(s_j^0)_{j=0, \dots, 2^L}$ is a homogeneously distributed sequence where each term is repeated $2^{L-L'}$ times as follows

$$s_j^0 = \left\lceil \frac{j2^{L'}}{2^L} \right\rceil \frac{T}{2^{L'}} \text{ where } \lceil \cdot \rceil \text{ is the ceiling function.} \quad (3.1)$$

that yields $\mathcal{S}^0 = \{0, T/2^{L'}, \dots, (2^{L'} - 1)T/2^{L'}, T\}$. The value of s_0^i stays the same for any iteration and is then given by $s_0^i \Big|_{i=1, \dots, L-L'} = 0$.

To define $(s_j^i)_{j=1, \dots, 2^L}$ for $i = 1, \dots, L - L'$, we introduce a scale Q_i and a translation q_i parameters. Since we want to keep each \mathcal{S}^i as coarse as possible, the translation parameter q_i sets the starting index from which we should refine the discretization. $q_0 = 0$ and q_i is given by

$$q_i = q_{i-1} \vee \max \left(\{j \in \{q_{i-1} + 1, \dots, 2^L\}; \mathcal{K}_j^{i-1}\} \right) \quad (3.2)$$

where the condition \mathcal{K}_j^{i-1} has to be specific to each application. Expression (3.2) ensures an increasing translation parameter q_i with respect to the iteration index i . The scale parameter $Q_i = 2^{L-L'-i}$ is used to know the number of terms that should be equal to each other. Thus, for indices bigger than q_i , when the iteration index i increases $Q_i = 2^{L-L'-i}$ decreases and the discretization becomes finer.

For $i = 1, \dots, L - L'$, we choose the following induction

$$s_j^i = s_j^{i-1} \mathbf{1}_{j \leq q_i} + \hat{s}_j^{i-1} \mathbf{1}_{j > q_i}, \quad (3.3)$$

where the values of $(\hat{s}_j^{i-1})_{j=0, \dots, 2^L}$ are given by

$$\begin{cases} \text{when } j \leq q_{i-1} \text{ define } \hat{s}_j^{i-1} = s_j^{i-1} \\ \text{otherwise, for } j' > q_{i-1}/Q_i \text{ set } \hat{s}_j^i \Big|_{j=Q_i(j'-1)+1}^{Q_i j'} = \frac{s_{Q_i j'}^{i-1} + s_{Q_i(j'-1)}^{i-1}}{2}. \end{cases} \quad (3.4)$$

The notation $s_j^i \Big|_{j=Q_i(j'-1)+1}^{Q_i j'}$ is used for $s_{Q_i(j'-1)+1}^i, \dots, s_{Q_i j'}^i$. Similar to \mathcal{S}^i defined through $(T - s_j^i)_{j=0, \dots, 2^L}$, we introduce $\hat{\mathcal{S}}^i$ to be the set of values taken by $(T - \hat{s}_j^i)_{j=0, \dots, 2^L}$. From (3.4), one can easily see that $\mathcal{S}^i \subseteq \hat{\mathcal{S}}^i$ and thus for any $s \in \mathcal{S}^i$ we have the inequality

$$\delta^{\hat{\mathcal{S}}^i}(s) \leq \delta^{\mathcal{S}^i}(s). \quad (3.5)$$

In Figure 2, we illustrate how this discretization strategy is implemented when $L' > L/2$.

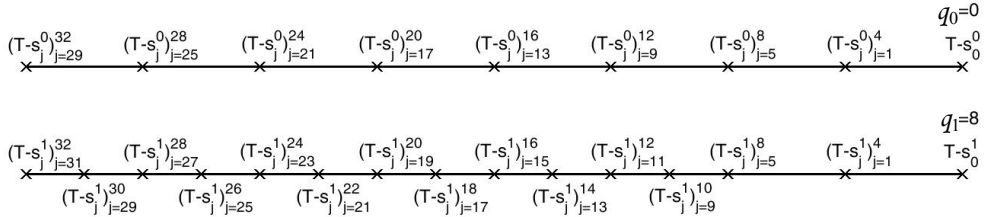


Figure 2: An example for (3.3) when $L = 5$, $L' = 3$, $\mathcal{S}^0 = \{0, \frac{T}{8}, \frac{T}{4}, \frac{3T}{8}, \frac{T}{2}, \frac{5T}{8}, \frac{3T}{4}, \frac{7T}{8}, T\}$ and $\mathcal{S}^1 = \{0, \frac{T}{16}, \frac{T}{8}, \frac{3T}{16}, \frac{T}{4}, \frac{5T}{16}, \frac{3T}{8}, \frac{7T}{16}, \frac{T}{2}, \frac{9T}{16}, \frac{5T}{8}, \frac{11T}{16}, \frac{3T}{4}, \frac{7T}{8}, T\}$.

As shown on Figure 2, because \mathcal{S}^i is the set of the values taken by $(T - s_j^i)_{j=0, \dots, 2^L}$, the increment $\delta(s) - s$ decreases as the value of the increment $s \in \mathcal{S}^i$ decreases. To counterbalance a rapid decrease of the time increment $\delta(s) - s$ when s gets smaller, the condition \mathcal{K}_j^{i-1} in (3.2) has to take into account the accumulation of bias through backpropagation. The value of \mathcal{K}_j^i given in Definition 3.1 allows then to increase the discretization steps only when the estimated gained accuracy is larger than the maximum accumulation of bias.

Definition 3.2 provides an updating strategy for $(\underline{s}_j, \overline{s}_j)_{(s_j)_{j=0, \dots, 2^L} \in \mathcal{S}^i}$ associated to the iterative construction of \mathcal{S}^i given in Definition 3.1. We will use the notation $(\underline{s}_j(s_j^i), \overline{s}_j(s_j^i))$ to show an explicit dependence of the updating strategy on the sequence $(s_j^i)_{j=0, \dots, 2^L}$ involved in the definition of \mathcal{S}^i .

Definition 3.1. Given an iteration step $i = 0, \dots, L - L' - 1$ for which the values of $(s_j^i)_{j=0, \dots, 2^L}$, \mathcal{S}^i , $\hat{\mathcal{S}}^i$, $\{\epsilon_s^{\mathcal{S}^i}\}_{s \in \mathcal{S}^i}$ and q_i are known, given $j \in \{q_i + 1, \dots, 2^L\}$ we define the condition $\mathcal{K}_j^i = "e_s^{\mathcal{S}^i} < \epsilon_s^{\mathcal{S}^i} \text{ for } s \in (T - s_k^i)_{q_i+1 \leq k \leq j}"$ where for $s \in (T - s_k^i)_{q_i+1 \leq k \leq j}$ we have $\delta^{\mathcal{S}^i}(s) = \delta^{\hat{\mathcal{S}}^i}(\delta^{\hat{\mathcal{S}}^i}(s))$ and the definition of the couple $(e_s^{\mathcal{S}^i}, \epsilon_s^{\mathcal{S}^i})$ is specified for the simulation of (f) by

$$e_s^{\mathcal{S}^i} = \frac{1}{M_0 M_1} \sum_{m_0=1}^{M_0} \sum_{m_1=1}^{M_1} \left[\overline{u}_{\delta^{\hat{\mathcal{S}}^i}(s), \delta^{\mathcal{S}^i}(s)}^{m_0, \hat{\mathcal{S}}^i} (X_{\delta^{\hat{\mathcal{S}}^i}(s), \delta^{\mathcal{S}^i}(s)}^{m_0, m_1}) - \overline{u}_{s, \delta^{\mathcal{S}^i}(s)}^{m_0, \mathcal{S}^i} (X_{\delta^{\hat{\mathcal{S}}^i}(s), \delta^{\mathcal{S}^i}(s)}^{m_0, m_1}) \right],$$

$$\varepsilon_s^{\mathcal{S}^i} = \sum_{u \in \mathcal{S}^i \cap]s, \bar{s}] } \epsilon_u^{\mathcal{S}^i}$$

and specified for the simulation of (Snl) by

$$e_s^{\mathcal{S}^i} = \frac{1}{M_0 M_1} \sum_{m_0=1}^{M_0} \sum_{m_1=1}^{M_1} \left[\bar{w}_{\delta^{\widehat{\mathcal{S}}^i}(s), \delta^{\mathcal{S}^i}(s)}^{m_0, \widehat{\mathcal{S}}^i} (X_{\delta^{\widehat{\mathcal{S}}^i}(s), \delta^{\mathcal{S}^i}(s)}^{m_0, m_1}}) - \bar{w}_{s, \delta^{\mathcal{S}^i}(s)}^{m_0, \mathcal{S}^i} (X_{\delta^{\widehat{\mathcal{S}}^i}(s), \delta^{\mathcal{S}^i}(s)}^{m_0, m_1}}) \right],$$

$$\varepsilon_s^{\mathcal{S}^i} = \sum_{u \in \mathcal{S}^i \cap]s, \bar{s}] } \mathcal{L}^{|\mathcal{S}^i \cap]s, u|} \epsilon_u^{\mathcal{S}^i}$$

with $|\mathcal{S}^i \cap]s, u|$ denotes the cardinal of the finite set $\mathcal{S}^i \cap]s, u|$.

The bias backpropagation of simulation (f) is linear when the one associated to simulation (Snl) is not. Thus, the definition of the maximum bias accumulation $\varepsilon_s^{\mathcal{S}^i}$ associated to (Snl) involves a sum on the Lipschitz constant \mathcal{L} to the power of lag index, given here by $|\mathcal{S}^i \cap]s, u|$, multiplied by the maximal accepted bias $\epsilon_u^{\mathcal{S}^i}$ per increment $u \in \mathcal{S}^i \cap]s, \bar{s}]$.

$e^{\mathcal{S}^i}$ is defined as the average difference between the estimation $\bar{w}^{m_0, \mathcal{S}^i}$ (resp. $\bar{w}^{m_0, \widehat{\mathcal{S}}^i}$) that involves the discretization set \mathcal{S}^i and the estimation $\bar{w}^{m_0, \widehat{\mathcal{S}}^i}$ (resp. $\bar{w}^{m_0, \widehat{\mathcal{S}}^i}$) that involves a finer discretization set $\widehat{\mathcal{S}}^i$. Using actualization (3.3) given (3.2) and \mathcal{K}_j^i expressed in Definition 3.1, we basically make the discretization set finer only when the difference between approximations is superior to the maximal accumulation of bias $\varepsilon^{\mathcal{S}^i}$.

Definition 3.2. Given an iteration step $i = 1, \dots, L-L'$ for which the values of $(s_j^i)_{j=0, \dots, 2^L}$, $(s_j^{i-1})_{j=0, \dots, 2^L}$, \mathcal{S}^i and q_i are known, introducing

$$\overline{\mathcal{S}}_s^{i, (f)} = \{u \in \mathcal{S}^i \cap]s, \overline{\delta^{\mathcal{S}^i}(s)}] \text{ that satisfies (2.10) when } \mathcal{S} = \mathcal{S}^i\}$$

$$\text{and } \overline{\mathcal{S}}_s^{i, (Snl)} = \{u \in \mathcal{S}^i \cap]s, \overline{\delta^{\mathcal{S}^i}(s)}] \text{ that satisfies (2.21) when } \mathcal{S} = \mathcal{S}^i\}$$

we set $\underline{s}_j(s_j^i) = \delta^{\mathcal{S}^i}(T - s_j^i)$ and $\overline{s}_j(s_j^i) = \overline{s}_j(s_j^{i-1}) \wedge \max(\overline{\mathcal{S}}_{T-s_j^i}^{i, (f)})$ for the simulation of (f) or $\overline{s}_j(s_j^i) = \overline{s}_j(s_j^{i-1}) \wedge \max(\overline{\mathcal{S}}_{T-s_j^i}^{i, (Snl)})$ for the simulation of (Snl) .

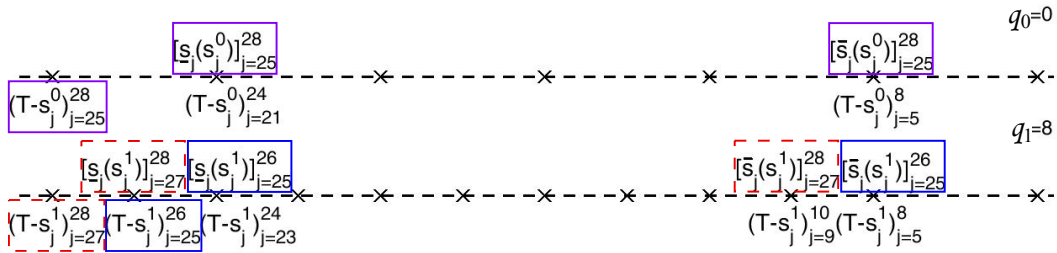


Figure 3: An example for Definition 3.2 based on the example of Figure 2. Here, for $j = 27, 28$ we have $\overline{s}_j(s_j^0) > \max(\overline{\mathcal{S}}_{T-s_j^1}^{1, (\cdot)})$.

As illustrated on Figure 3 and on Figure 4, the value of $\overline{s}_j(s_j^i)$ given in Definition 3.2 can change depending on $\overline{s}_j(s_j^{i-1}) > \max(\overline{\mathcal{S}}_{T-s_j^i}^{i, (\cdot)})$ is true or not. Indeed, since \mathcal{S}^i is finer than \mathcal{S}^{i-1} , the bias backpropagation associated to \mathcal{S}^i can be larger than the one associated to \mathcal{S}^{i-1} and thus $\max(\overline{\mathcal{S}}_{T-s_j^i}^{i, (\cdot)})$ can happen to be significantly smaller than $\overline{s}_j(s_j^{i-1})$.

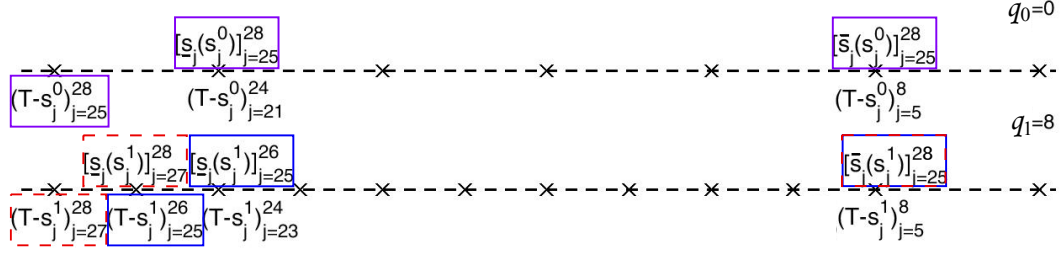


Figure 4: An example for Definition 3.2 based on the example of Figure 2. Here, for $j = 27, 28$ we have $\bar{s}_j(s_j^0) \leq \max(\bar{\mathcal{S}}_{T-s_j^1}^{1,(\cdot)})$.

3.2 From global to local bias control and variance adjustment

3.2.1 Local bias control

Given $s \in \mathcal{S}$, the couple (\underline{s}, \bar{s}) was expressed in Definition 2.1 as well as in Definition 2.2 and an updating strategy for (\underline{s}, \bar{s}) was established in Definition 3.2 when \mathcal{S} is constructed iteratively. In order to reduce the bias backpropagation, these definitions used the double layer Monte Carlo to control the average bias. Indeed, as $\frac{1}{M_0} \sum_{m_0=1}^{M_0} \tilde{u}_{s,u}^{m_0, \mathcal{S}}$ and $\frac{1}{M_0} \sum_{m_0=1}^{M_0} \left(\tilde{u}_{\delta(s), \delta(s)}^{m_0, \mathcal{S}} + \sum_{t_{l+1} > s}^{\delta(s)} f(t_l, X_{t_l}^{m_0}, X_{t_{l+1}}^{m_0}) \right)$ are both approximations of the same expectation $\mathbb{E}(U_s) = \mathbb{E} \left(U_{\delta(s)} + \sum_{t_{l+1} > s}^{\delta(s)} f(t_l, X_{t_l}^{m_0}, X_{t_{l+1}}^{m_0}) \right)$, it is natural to have them almost equal. The same can be said for $\frac{1}{M_0} \sum_{m_0=1}^{M_0} \left(\tilde{v}_{\delta(s), \delta(s)}^{m_0, \mathcal{S}} + (\delta(s) - s) f(s, \tilde{v}_{\delta(s), \delta(s)}^{m_0, \mathcal{S}}) \right)$ and for $\frac{1}{M_0} \sum_{m_0=1}^{M_0} \tilde{w}_{s,u}^{m_0, \mathcal{S}}$ that are both approximations of $\mathbb{E} \left(V_{\delta(s)} + (\delta(s) - s) f(s, V_{\delta(s)}) \right) = \mathbb{E} \left(\mathbb{E}_s [V_{\delta(s)} + (\delta(s) - s) f(s, V_{\delta(s)})] \right)$. For large values of M_0 , the difference between these approximations is due to bias. A judicious method to reduce this bias backpropagation is to adjust the number of successive regressions through the appropriate choice of u involved in $\frac{1}{M_0} \sum_{m_0=1}^{M_0} \tilde{u}_{s,u}^{m_0, \mathcal{S}}$ and in $\frac{1}{M_0} \sum_{m_0=1}^{M_0} \tilde{w}_{s,u}^{m_0, \mathcal{S}}$. In definitions 2.1 and 2.2, we select \bar{s} as the largest value of u for which the bias remains under some threshold. This choice of \bar{s} , with $\underline{s} = \delta(s)$, sets the learning depth conditional to the information available at $s \in \mathcal{S}$.

The choice of $(\bar{s})_{s \in \mathcal{S}}$ ought to decrease the global average value of the bias. More local approach can be developed using equalities

$$\mathbb{E}(U_s 1_{\{U_s \in [a, b]\}}) = \mathbb{E} \left(1_{\{U_s \in [a, b]\}} \left[U_{\delta(s)} + \sum_{t_{l+1} > s}^{\delta(s)} f(t_l, X_{t_l}^{m_0}, X_{t_{l+1}}^{m_0}) \right] \right) \quad (3.6)$$

$$\mathbb{E} \left([V_{\delta(s)} + (\delta(s) - s) f(s, V_{\delta(s)})] 1_{\{V_s \in [a, b]\}} \right) = \mathbb{E} \left(1_{\{V_s \in [a, b]\}} \mathbb{E}_s [V_{\delta(s)} + (\delta(s) - s) f(s, V_{\delta(s)})] \right) \quad (3.7)$$

which are true for any localizing interval $[a, b]$. Considering the generic example (f) with equality (3.6), when M_0 is large enough, one can sort $\{\tilde{u}_{s,u}^{m_0, \mathcal{S}}\}_{m_0 \leq M_0}$ and define a subdivision of localizing intervals $\{[a_q, a_{q+1}]\}_{q \geq 1}$ then choose $u = \bar{s}$ that does not induce a large difference between $\frac{1}{M_0} \sum_{m_0=1}^{M_0} \left(1_{\{\tilde{u}_{s, \bar{s}}^{m_0, \mathcal{S}} \in [a_q, a_{q+1}]\}} \left[\tilde{u}_{\delta(s), \delta(s)}^{m_0, \mathcal{S}} + \sum_{t_{l+1} > s}^{\delta(s)} f(t_l, X_{t_l}^{m_0}, X_{t_{l+1}}^{m_0}) \right] \right)$ and $\frac{1}{M_0} \sum_{m_0=1}^{M_0} \left(\tilde{u}_{s, \bar{s}}^{m_0, \mathcal{S}} 1_{\{\tilde{u}_{s, \bar{s}}^{m_0, \mathcal{S}} \in [a_q, a_{q+1}]\}} \right)$ for all q s. The local increase of bias can be tracked

for any $s \in \mathcal{S} \cap [\delta(s), \bar{s}[$ and each interval $[a_q, a_{q+1}]$ using the difference

$$\frac{1}{M_0} \sum_{m_0=1}^{M_0} \mathbb{1}_{\{\tilde{u}_{s,\bar{s}}^{m_0,\mathcal{S}} \in [a_q, a_{q+1}]\}} \left(\tilde{u}_{s,\bar{s}}^{m_0,\mathcal{S}} - \tilde{u}_{\delta(s),\delta(s)}^{m_0,\mathcal{S}} - \sum_{t_{l+1} > s}^{\delta(s)} f(t_l, X_{t_l}^{m_0}, X_{t_{l+1}}^{m_0}) \right). \quad (3.8)$$

The same can be done for (Snl) with equality (3.7). Although the local tracking of bias was not necessary in our simulations, it is quite remarkable to point out the strength of bias control induced by 1NMC. Provided that M_0 is sufficiently large, it is basically possible to remove any bias backpropagation.

3.2.2 From global to local value of $\left\{ \gamma_{s_j, s_k}^{m_0, \mathcal{S}} \right\}_{s_j \in \mathcal{S}, s_k \in \mathcal{S} \cap]s_j, \bar{s}_j]}$

Given $s_j \in \mathcal{S}$ and $s_k \in \mathcal{S} \cap]s_j, \bar{s}_j]$, various values can be considered for $\gamma_{s_j, s_k}^{m_0, \mathcal{S}}$. The straight choice is to take $\gamma_{s_j, s_k}^{m_0, \mathcal{S}} = 1$ which reduces the procedure to a standard regression like in CMCLDI. However, this is not the suitable choice for problems that heavily depend on tail distribution. Indeed, given two arbitrary square integrable random variables χ_1 and χ_2 , consider χ_3 to be the regression of χ_1 on a $\sigma(\chi_2)$ -measurable basis. Because generally regression preserves the mean value, it is reasonable to assume $\mathbb{E}(\chi_3) = \mathbb{E}(\chi_1)$. However, projections reduce the value of the second moment i.e. $\mathbb{E}(\chi_3^2) < \mathbb{E}(\chi_1^2)$ and thus $\text{Var}(\chi_3) < \text{Var}(\chi_1)$. The latter fact becomes a real problem for tail distribution in case $\text{Var}(\chi_3) < \text{Var}(\mathbb{E}(\chi_1|\chi_2))$. Some contributions tackle rare event simulation using a change of probability trick [9, 14] and more recent contribution [4] implements reversible shaking transformations.

In case $\mathbb{E}(\chi_3) = \mathbb{E}(\chi_1) = 0$, given that we know a good approximation of $\text{Var}(\mathbb{E}(\chi_1|\chi_2))$, it is possible to adjust the projected value χ_3 with a parameter γ to make $\text{Var}(\gamma\chi_3) \approx \text{Var}(\mathbb{E}(\chi_1|\chi_2))$. Thus, $\gamma\chi_3$ has the appropriate variance for the price of a slight increase of the quadratic error since

$$\mathbb{E}([\chi_1 - \gamma\chi_3]^2) = \mathbb{E}([\chi_1 - \chi_3]^2) + \frac{(1-\gamma)^2}{\gamma^2} \mathbb{E}([\mathbb{E}(\chi_1|\chi_2)]^2)$$

then $\mathbb{E}([\chi_1 - \gamma\chi_3]^2) \approx \mathbb{E}([\chi_1 - \chi_3]^2)$ provided that γ is slightly bigger than 1. This basic idea is adapted hereafter to adjust the variance of the coarse estimators $\bar{u}^{m_0, \mathcal{S}}$ and $\bar{w}^{m_0, \mathcal{S}}$.

Using the global and local bias control presented above, we established strong constraints to make $\tilde{u}^{m_0, \mathcal{S}}$ and $\tilde{v}^{m_0, \mathcal{S}}$ almost unbiased estimators of U and V respectively given in (f) and (Snl). It is then possible to use their values to propose an appropriate adjustment of the variance. For $s_j < s$ with $s = s_k, \delta(s_k)$, the whole idea is based on the following equality

$$\mathbb{E}(\text{Var}_{s_j}(\Theta_s)) = \mathbb{E}\left(\mathbb{E}_{s_j}\left([\Theta_s - \mathbb{E}_{s_j}(\Theta_s)]^2\right)\right) = \mathbb{E}\left([\Theta_s - \mathbb{E}_{s_j}(\Theta_s)]^2\right)$$

true for any square integrable process Θ and thus when $\Theta = U$ or when $\Theta = \underline{V}$ with U defined in (f) and \underline{V} defined in (2.13).

For $\Theta = U, \underline{V}$ we associate $\tilde{\theta}^{m_0, \mathcal{S}} = \tilde{u}^{m_0, \mathcal{S}}, \tilde{w}^{m_0, \mathcal{S}}$ respectively as fine approximations and $\bar{\theta}^{m_0, \mathcal{S}} = \bar{u}^{m_0, \mathcal{S}}, \bar{w}^{m_0, \mathcal{S}}$ respectively as coarse ones. We also set $H^{m_0, \mathcal{S}} = B^{m_0, \mathcal{S}}, C^{m_0, \mathcal{S}}$ which are subsequently the regression vector associated to either $\bar{u}^{m_0, \mathcal{S}}$ or $\bar{w}^{m_0, \mathcal{S}}$. To define

$\gamma^{m_0, \mathcal{S}}$ inductively, we need to introduce three notations starting with the non-adjusted ($\gamma_{s_j, s}^{m_0, \mathcal{S}} = 1$) conditional variance $(\widehat{\sigma}_{s_j, s}^{m_0, \mathcal{S}})^2$ ($s_j < s$) given for each couple (θ, H) by

$$(\widehat{\sigma}_{s_j, s}^{m_0, \mathcal{S}}[\theta, H])^2 = \frac{1}{M_1} \sum_{m_1=1}^{M_1} \left[\widetilde{\theta}_{s, \bar{s}}^{m_0, \mathcal{S}} + {}^t \mathcal{T}_{s_j, s, M_1}^{m_0}(X_{s_j, s}^{m_0, m_1}) H_{s_j, s}^{m_0, \mathcal{S}} - \sum_{m_1=1}^{M_1} \frac{\bar{\theta}_{s_j, s}^{m_0, \mathcal{S}}(X_{s_j, s}^{m_0, m_1})}{M_1} \right]^2. \quad (3.9)$$

$\sum_{m_1=1}^{M_1} \frac{\bar{\theta}_{s_j, s}^{m_0, \mathcal{S}}(X_{s_j, s}^{m_0, m_1})}{M_1}$ in (3.9) can be replaced by $\sum_{m_1=1}^{M_1} \frac{\widetilde{\theta}_{s, \bar{s}}^{m_0, \mathcal{S}} + {}^t \mathcal{T}_{s_j, s, M_1}^{m_0}(X_{s_j, s}^{m_0, m_1}) H_{s_j, s}^{m_0, \mathcal{S}}}{M_1}$ without changing the value of $(\widehat{\sigma}_{s_j, s}^{m_0, \mathcal{S}})^2$. We introduce also the conditional variance ($s_j < s$)

$$(\sigma_{s_j, s}^{m_0, \mathcal{S}}[\theta])^2 = \frac{1}{M_1} \sum_{m_1=1}^{M_1} \left[\bar{\theta}_{s_j, s}^{m_0, \mathcal{S}}(X_{s_j, s}^{m_0, m_1}) - \frac{1}{M_1} \sum_{m_1=1}^{M_1} \bar{\theta}_{s_j, s}^{m_0, \mathcal{S}}(X_{s_j, s}^{m_0, m_1}) \right]^2 \quad (3.10)$$

and the average conditional variance ($s_j < s$)

$$(\sigma_{0, s_j, s}^{\mathcal{S}}[\theta])^2 = \frac{1}{M_0} \sum_{m_0=1}^{M_0} \left[\widetilde{\theta}_{s, \bar{s}}^{m_0, \mathcal{S}} - \frac{1}{M_1} \sum_{m_1=1}^{M_1} \bar{\theta}_{s_j, s}^{m_0, \mathcal{S}}(X_{s_j, s}^{m_0, m_1}) \right]^2. \quad (3.11)$$

Because expressions (3.10) and (3.11) are the estimators of $\mathbb{E}_{s_j} \left([\Theta_s - \mathbb{E}_{s_j}(\Theta_s)]^2 \right)$ and of $\mathbb{E} \left([\Theta_s - \mathbb{E}_{s_j}(\Theta_s)]^2 \right)$ respectively, it is natural to have asymptotically (as $M_1 \rightarrow \infty$ and $M_0 \rightarrow \infty$) the equality

$$(\sigma_{0, s_j, s}^{\mathcal{S}}[\theta])^2 = \frac{1}{M_0} \sum_{m_0=1}^{M_0} (\sigma_{s_j, s}^{m_0, \mathcal{S}}[\theta])^2 \quad (3.12)$$

for both $s = s_k, \delta(s_k)$.

For $\theta = u, w$, the estimators $\frac{1}{M_1} \sum_{m_1=1}^{M_1} \bar{\theta}_{s_j, s}^{m_0, \mathcal{S}}(X_{s_j, s}^{m_0, m_1})$ and $\widetilde{\theta}_{s, \bar{s}}^{m_0, \mathcal{S}}$ have negligible bias. Thus, we can reasonably assume inductively that (3.12) is true for $s = \delta(s_k)$. Afterwards, we choose the appropriate value of $\gamma_{s_j, s_k}^{m_0, \mathcal{S}}$ that makes $\sigma_{s_j, s_k}^{m_0, \mathcal{S}}$ satisfy (3.12) for $s = s_k$. We have then the following definition.

Definition 3.3. For a discretization set $\mathcal{S} \subset \mathbb{S}$, $s_j \in \mathcal{S}$, $s_k \in \mathcal{S} \cap]s_j, \bar{s}_j]$, $\theta = u, w$ and $H^{m_0, \mathcal{S}} = B^{m_0, \mathcal{S}}, C^{m_0, \mathcal{S}}$ assuming by backward induction that $\gamma_{s_j, s}^{m_0, \mathcal{S}}$ is known for all $s = \bar{s}_j, \dots, \delta(s_k)$ and given a positive tuning value $\epsilon < 1/3$, we set then

$$\gamma_{s_j, s_k}^{m_0, \mathcal{S}} = \frac{\sigma_{s_j, \delta(s_k)}^{m_0, \mathcal{S}}[\theta]}{\widehat{\sigma}_{s_j, s_k}^{m_0, \mathcal{S}}[\theta, H]} \left(\sqrt{\frac{s_k - s_j}{\delta(s_k) - s_j}} 1_{\delta(s_k) - s_j < \epsilon} + \frac{\sigma_{0, s_j, s_k}^{\mathcal{S}}[\theta]}{\sigma_{0, s_j, \delta(s_k)}^{\mathcal{S}}[\theta]} 1_{\delta(s_k) - s_j \geq \epsilon} \right). \quad (3.13)$$

According to (3.13), when $\delta(s_k) - s_j$ is small and a fortiori $s_k - s_j$ is small then the conditional variance $(\sigma_{s_j, s_k}^{m_0, \mathcal{S}}[\theta])^2$ has to be linear with respect to time increment $s_k - s_j$. This fact can be justified for diffusions using first order Taylor expansion of $\phi(t, \beta_t)$ around $\phi(t, 0)$, for a standard Brownian motion β and a smooth function ϕ . Also according to (3.13), when $\delta(s_k) - s_j$ becomes sufficiently big, the conditional variance $(\sigma_{s_j, s_k}^{m_0, \mathcal{S}}[\theta])^2$ has the same average decreasing ratio $\left(\frac{\sigma_{0, s_j, s_k}^{\mathcal{S}}[\theta]}{\sigma_{0, s_j, \delta(s_k)}^{\mathcal{S}}[\theta]} \right)^2$ with respect to $(\sigma_{s_j, \delta(s_k)}^{m_0, \mathcal{S}}[\theta])^2$.

Although Definition 3.3 works well in our simulations, it can be turned into a more local variance adjustment. In fact, similar to what was proposed for the bias control in (3.6) and (3.7), for $s_j < s$ with $s = s_k, \delta(s_k)$, the equality

$$\mathbb{E}\left(\text{Var}_{s_j}(\Theta_s)1_{\{\text{Var}_{s_j}(\Theta_{\delta(s_k)}) \in [a,b]\}}\right) = \mathbb{E}\left(1_{\{\text{Var}_{s_j}(\Theta_{\delta(s_k)}) \in [a,b]\}}[\Theta_s - \mathbb{E}_{s_j}(\Theta_s)]^2\right)$$

is true for any localizing interval $[a, b]$. When M_0 is sufficiently large, one can sort $\{(\sigma_{s_j, \delta(s_k)}^{m_0, \mathcal{S}}[\theta])^2\}_{1 \leq m_0 \leq M_0}$ to define a family of localizing intervals $\{[a_q, a_{q+1}]\}_{q \geq 1}$ and set

$$(\sigma_{0, s_j, s}^{\mathcal{S}, q}[\theta])^2 = \frac{1}{M_0} \sum_{m_0=1}^{M_0} 1_{\{(\sigma_{s_j, \delta(s_k)}^{m_0, \mathcal{S}})^2 \in [a_q, a_{q+1}]\}} \left[\tilde{\theta}_{s, \bar{s}}^{m_0, \mathcal{S}} - \frac{1}{M_1} \sum_{m_1=1}^{M_1} \bar{\theta}_{s_j, s}^{m_0, \mathcal{S}}(X_{s_j, s}^{m_0, m_1}) \right]^2.$$

Condition (3.12) can be then replaced by its localized version

$$(\sigma_{0, s_j, s}^{\mathcal{S}, q}[\theta])^2 = \frac{1}{M_0} \sum_{m_0=1}^{M_0} 1_{\{(\sigma_{s_j, \delta(s_k)}^{m_0, \mathcal{S}}[\theta])^2 \in [a_q, a_{q+1}]\}} (\sigma_{s_j, s}^{m_0, \mathcal{S}}[\theta])^2. \quad (3.14)$$

If $\sigma_{s_j, \delta(s_k)}^{m_0, \mathcal{S}}[\theta] \in [a_{q_0}, a_{q_0+1}]$, for some q_0 , then it makes sense to replace (3.13) by

$$\gamma_{s_j, s_k}^{m_0, \mathcal{S}} = \frac{\sigma_{s_j, \delta(s_k)}^{m_0, \mathcal{S}}[\theta]}{\widehat{\sigma}_{s_j, s_k}^{m_0, \mathcal{S}}[\theta, H]} \left(\sqrt{\frac{s_k - s_j}{\delta(s_k) - s_j}} 1_{\delta(s_k) - s_j < \epsilon} + \frac{\sigma_{0, s_j, s_k}^{\mathcal{S}, q_0}[\theta]}{\sigma_{0, s_j, \delta(s_k)}^{\mathcal{S}, q_0}[\theta]} 1_{\delta(s_k) - s_j \geq \epsilon} \right). \quad (3.15)$$

The local variance adjustment (3.15) was not necessary in our simulations. Nevertheless, it is quite remarkable to point out the high flexibility of the multilayer setting induced by 1NMC. Thus when M_0 and M_1 are sufficiently large, one sees that this double layer Monte Carlo makes possible a very fine tracking of both the bias of the first layer fine estimators (U^{m_0}, V^{m_0}) and the variance of the second layer coarse estimator $(U^{m_0, m_1}, V^{m_0, m_1})$.

4 Error estimates for regression with different starting points

After expressing error estimates for both coarse and fine approximations in CMCLDI (Section 4), we showed in Section 5.4 the importance of using the couple $(\underline{s}, \bar{s})_{s \in \mathcal{S}}$ to cut the bias propagation. Indeed, it is usual that successive regressions associated to a large learning depth (the cardinal $|\llbracket \underline{s}, \bar{s} \rrbracket \cap \mathcal{S}|$ is large for $s \in \mathcal{S}$) induce the accumulation of significant errors, especially for nonlinear problems. When the standard compromise to reduce these errors is to choose a coarser set \mathcal{S} , our trick shortens the distance $|\bar{s} - \underline{s}|$ to cut the propagation of these errors without decreasing the precision on \mathcal{S} . This trick could then introduce a new stage of approximation by making $\bar{s} < T$ for some $s \in \mathcal{S}$.

In this section, we present a control on this new stage of approximation used in the generic example of [2](Section 3.1) extended in Section 2.1 which is adapted to BSDEs in [2](Section 3.2) and to RBSDEs in Section 2.2. Focusing on the generic example, equation (2.7) defines $\tilde{u}_{s, \bar{s}_k}^{m_0, \mathcal{S}}$ for any $s \in \{s_k, s_k + \Delta_t, \dots, \delta_{s_j}(s_k) - \Delta_t\}$ using $\bar{u}_{s_k, \delta_{s_j}(s_k)}^{m_0, \mathcal{S}}(\cdot)$ which is deduced from a regression on $X_{s_k, \delta_{s_j}(s_k)}^{m_0, m_1}$ instead of a regression on $X_{s, \delta_{s_j}(s_k)}^{m_0, m_1}$. Thus, provided that s is sufficiently close to s_k we replaced a regressed function obtained from

inner trajectories starting at s by a regressed function obtained from inner trajectories starting at s_k on the same outer trajectory m_0 . We performed a similar approximation in (2.8) when $\bar{s}_j < T$ as we defined $\bar{u}_{s_j, \bar{s}_j}^{m_0, \mathcal{S}}$ to be equal to $\bar{u}_{\delta_{s_j}(s_j), \bar{s}_j}^{m_0, \mathcal{S}}$ i.e. we replaced a regression on $X_{s_j, \bar{s}_j}^{m_0, m_1}$ by a regression on $X_{\delta_{s_j}(s_j), \bar{s}_j}^{m_0, m_1}$. The adaptation of (2.7) yields similar approximations in (2.16) and in [2](3.12) when the adaptation of (2.8) yields similar approximations in (2.20) and in [2](3.14).

For $s_j < s < s_k$, we consequently approximate the regressed function $\bar{u}_{s_j, s_k}^{m_0, \mathcal{S}}(\cdot)$ resulting from the projection of $\sum_{t_{l+1} > s_k}^T f(t_l, X_{s_j, t_l}^{m_0, m_1}, X_{s_j, t_{l+1}}^{m_0, m_1})$ on $X_{s_j, s_k}^{m_0, m_1}$ by the regressed function $\bar{u}_{s_j, s}^{m_0, \mathcal{S}}(\cdot)$ resulting from the projection of $\sum_{t_{l+1} > s_k}^T f(t_l, X_{s, t_l}^{m_0, m_1}, X_{s, t_{l+1}}^{m_0, m_1})$ on $X_{s, s_k}^{m_0, m_1}$ and vice versa. These two approximations are not absurd since one can straightforwardly see, from the Markov property, that

$$\begin{aligned} u_{s_k}(x) &= \mathbb{E} \left(\sum_{t_l \geq s_k}^T f(t_l, X_{s_j, t_l}^{m_0, m_1}, X_{s_j, t_{l+1}}^{m_0, m_1}) \middle| X_{s_j, s_k}^{m_0, m_1} = x \right) \\ &= \mathbb{E} \left(\sum_{t_l \geq s_k}^T f(t_l, X_{s, t_l}^{m_0, m_1}, X_{s, t_{l+1}}^{m_0, m_1}) \middle| X_{s, s_k}^{m_0, m_1} = x \right). \end{aligned} \quad (4.1)$$

In the following, we denote $\mathbb{E}_{s_j}^{m_0}$ the conditional expectation knowing the trajectory $(X_t^{m_0})_{t \geq s_j}$. When \mathbb{E}_{s_j} is the usual conditional expectation knowing $X_{s_j}^{m_0}$, $\mathbb{E}_{s_j}^{m_0}$ is needed since the regression basis depends on X^{m_0} . In Proposition 4.1, we establish a control on $\mathbb{E}_{s_j}^{m_0} \left(\left[\bar{u}_{s_j, s_k}^{m_0, \mathcal{S}}(X_{s, s_k}^{m_0, m_1}) - u_{s_k}(X_{s, s_k}^{m_0, m_1}) \right]^2 \right)$ and on $\mathbb{E}_s^{m_0} \left(\left[\bar{u}_{s, s_k}^{m_0, \mathcal{S}}(X_{s_j, s_k}^{m_0, m_1}) - u_{s_k}(X_{s_j, s_k}^{m_0, m_1}) \right]^2 \right)$ using two auxiliary processes \bar{X} and \tilde{X} defined for $t_l \geq s$ by

$$\begin{cases} \bar{X}_{s_j, s, s}^{m_0, m_1} = X_s^{m_0}, \tilde{X}_{s_j, s, s}^{m_0, m_1} = X_{s_j, s}^{m_0, m_1} \text{ and for } t_l = s + \Delta_t, \dots, T \\ \bar{X}_{s_j, s, t_l}^{m_0, m_1} = \mathcal{E}_{t_l-1}(\mathcal{E}_{t_l-2}(\dots \mathcal{E}_s(X_s^{m_0}, \xi_{s_j, s+\Delta_t}^{m_0, m_1}), \dots \xi_{s_j, t_{l-1}}^{m_0, m_1}), \xi_{s_j, t_l}^{m_0, m_1}) \\ \tilde{X}_{s_j, s, t_l}^{m_0, m_1} = \mathcal{E}_{t_l-1}(\mathcal{E}_{t_l-2}(\dots \mathcal{E}_s(X_{s_j, s}^{m_0, m_1}, \xi_{s, s+\Delta_t}^{m_0, m_1}), \dots \xi_{s, t_{l-1}}^{m_0, m_1}), \xi_{s, t_l}^{m_0, m_1}). \end{cases} \quad (4.2)$$

where \mathcal{E} is given in (2.3).

As shown on Figure 5 for $t_l > s_k$, $\bar{X}_{s_j, s, t_l}^{m_0, m_1}$ is defined using $X_{s, s}^{m_0, m_1} = X_s^{m_0}$ and increments from the process $X_{s_j, t_l}^{m_0, m_1}$, in contrast to $\tilde{X}_{s_j, s, t_l}^{m_0, m_1}$ defined using $X_{s_j, s}^{m_0, m_1}$ and increments from the process $X_{s, t_l}^{m_0, m_1}$. Proposition 4.1 provides a strong formulation of a possible compromise between two error terms on the right of each inequality.

Proposition 4.1. *For any $t \in \mathbb{S}$, we assume u_t is $[u_t]_{Lip}$ -Lipschitz. For $s_j < s < s_k$ taking their values in the discretization set \mathcal{S} , we define $\mathcal{K}_{2, s_j, s_k}^{m_0} = [u_{s_k}]_{Lip} + \sqrt{\mathbb{E}_{s_j}^{m_0}(|B_{s_j, s_k}^{m_0, \mathcal{S}}|_{d_1}^2)}$ and $\mathcal{K}_{2, s, s_k}^{m_0} = [u_{s_k}]_{Lip} + \sqrt{\mathbb{E}_s^{m_0}(|B_{s, s_k}^{m_0, \mathcal{S}}|_{d_1}^2)}$ where $|\cdot|_{d_1}$ is the Euclidean norm on \mathbb{R}^{d_1} , then*

$$\begin{aligned} \sqrt{\mathbb{E}_{s_j}^{m_0} \left(\left[\bar{u}_{s_j, s_k}^{m_0, \mathcal{S}}(X_{s_j, s_k}^{m_0, m_1}) - u_{s_k}(X_{s_j, s_k}^{m_0, m_1}) \right]^2 \right)} &\leq \sqrt{\mathbb{E}_s^{m_0} \left(\left[\bar{u}_{s, s_k}^{m_0, \mathcal{S}}(X_{s_j, s_k}^{m_0, m_1}) - u_{s_k}(X_{s, s_k}^{m_0, m_1}) \right]^2 \right)} \\ &\quad + \mathcal{K}_{2, s, s_k}^{m_0} \sqrt{\mathbb{E}_s^{m_0} \left(\left| \tilde{X}_{s_j, s, s_k}^{m_0, m_1} - X_{s, s_k}^{m_0, m_1} \right|_{d_1}^2 \right)} \end{aligned} \quad (4.3)$$

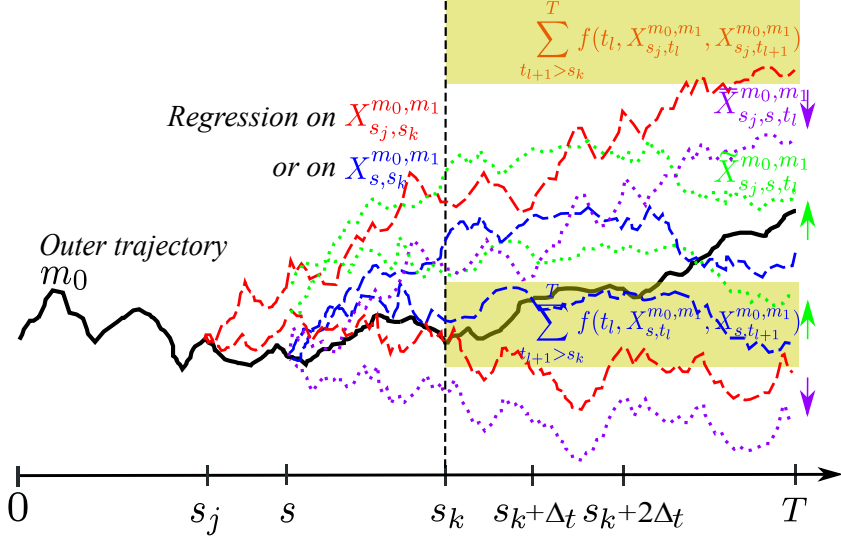


Figure 5: Regression of $\sum_{t_{l+1} > s_k}^T f(t_l, X_{s_j, t_l}^{m_0, m_1}, X_{s_j, t_{l+1}}^{m_0, m_1})$ on $X_{s_j, s_k}^{m_0, m_1}$ compared to regression of

$$\sum_{t_{l+1} > s_k}^T f(t_l, X_{s, t_l}^{m_0, m_1}, X_{s, t_{l+1}}^{m_0, m_1}) \text{ on } X_{s, s_k}^{m_0, m_1} \text{ with } s \in \{s_j + \Delta t, \dots, s_k - \Delta t\}.$$

and

$$\begin{aligned} \sqrt{\mathbb{E}_{s_j}^{m_0} \left(\left[\bar{u}_{s_j, s_k}^{m_0, S}(X_{s, s_k}^{m_0, m_1}) - u_{s_k}(X_{s, s_k}^{m_0, m_1}) \right]^2 \right)} &\leq \sqrt{\mathbb{E}_{s_j}^{m_0} \left(\left[\bar{u}_{s_j, s_k}^{m_0, S}(X_{s_j, s_k}^{m_0, m_1}) - u_{s_k}(X_{s_j, s_k}^{m_0, m_1}) \right]^2 \right)} \\ &\quad + \mathcal{K}_{2, s_j, s_k}^{m_0} \sqrt{\mathbb{E}_{s_j}^{m_0} \left(\left| \bar{X}_{s_j, s_k}^{m_0, m_1} - X_{s_j, s_k}^{m_0, m_1} \right|_{d'_1}^2 \right)}. \end{aligned} \quad (4.4)$$

Proof. As we simulate several independent copies of X^{m_0, m_1} (cf. few lines before (2.4)), we make sure that the approximations \bar{u} are independent from X^{m_0, m_1} , from \bar{X}^{m_0, m_1} and from \tilde{X}^{m_0, m_1} conditionally on X^{m_0} . Moreover, from definition (4.2), $(\bar{X}_{s_j, s_k}^{m_0, m_1})_{t_l \geq s}$ has the same law as $(X_{s, t_l}^{m_0, m_1})_{t_l \geq s}$ and $(\tilde{X}_{s_j, s_k}^{m_0, m_1})_{t_l \geq s}$ has the same law as $(X_{s_j, t_l}^{m_0, m_1})_{t_l \geq s}$. Then one can write the following

$$\begin{aligned} \sqrt{\mathbb{E}_s^{m_0} \left(\left[\bar{u}_{s, s_k}^{m_0, S}(X_{s_j, s_k}^{m_0, m_1}) - u_{s_k}(X_{s_j, s_k}^{m_0, m_1}) \right]^2 \right)} &= \sqrt{\mathbb{E}_s^{m_0} \left(\left[\bar{u}_{s, s_k}^{m_0, S}(\tilde{X}_{s_j, s_k}^{m_0, m_1}) - u_{s_k}(\tilde{X}_{s_j, s_k}^{m_0, m_1}) \right]^2 \right)} \\ &\leq \sqrt{\mathbb{E}_s^{m_0} \left(\left[\bar{u}_{s, s_k}^{m_0, S}(\tilde{X}_{s_j, s_k}^{m_0, m_1}) - \bar{u}_{s, s_k}^{m_0, S}(X_{s, s_k}^{m_0, m_1}) \right]^2 \right)} \\ &\quad + \sqrt{\mathbb{E}_s^{m_0} \left(\left[\bar{u}_{s, s_k}^{m_0, S}(X_{s, s_k}^{m_0, m_1}) - u_{s_k}(X_{s, s_k}^{m_0, m_1}) \right]^2 \right)} \\ &\quad + \sqrt{\mathbb{E}_s^{m_0} \left(\left[u_{s_k}(X_{s, s_k}^{m_0, m_1}) - u_{s_k}(\tilde{X}_{s_j, s_k}^{m_0, m_1}) \right]^2 \right)} \end{aligned} \quad (4.5)$$

as well as

$$\begin{aligned}
\sqrt{\mathbb{E}_{s_j}^{m_0} \left(\left[\bar{u}_{s_j, s_k}^{m_0, \mathcal{S}}(X_{s_j, s_k}^{m_0, m_1}) - u_{s_k}(X_{s_j, s_k}^{m_0, m_1}) \right]^2 \right)} &= \sqrt{\mathbb{E}_{s_j}^{m_0} \left(\left[\bar{u}_{s_j, s_k}^{m_0, \mathcal{S}}(\bar{X}_{s_j, s_k}^{m_0, m_1}) - u_{s_k}(\bar{X}_{s_j, s_k}^{m_0, m_1}) \right]^2 \right)} \\
&\leq \sqrt{\mathbb{E}_{s_j}^{m_0} \left(\left[\bar{u}_{s_j, s_k}^{m_0, \mathcal{S}}(\bar{X}_{s_j, s_k}^{m_0, m_1}) - \bar{u}_{s_j, s_k}^{m_0, \mathcal{S}}(X_{s_j, s_k}^{m_0, m_1}) \right]^2 \right)} \\
&\quad + \sqrt{\mathbb{E}_{s_j}^{m_0} \left(\left[\bar{u}_{s_j, s_k}^{m_0, \mathcal{S}}(X_{s_j, s_k}^{m_0, m_1}) - u_{s_k}(X_{s_j, s_k}^{m_0, m_1}) \right]^2 \right)} \\
&\quad + \sqrt{\mathbb{E}_{s_j}^{m_0} \left(\left[u_{s_k}(X_{s_j, s_k}^{m_0, m_1}) - u_{s_k}(\bar{X}_{s_j, s_k}^{m_0, m_1}) \right]^2 \right)}
\end{aligned} \tag{4.6}$$

which yield (4.3) and (4.4). \square

Remark 4.1. • We introduced notation $\mathcal{K}_{2, \cdot}^{m_0}$, since $\mathcal{K}_{1, \cdot}^{m_0}$ was already used in [2] to control the bias term \mathcal{W} . Indeed, Proposition 4.2 in [2] establishes the following upper bound on the bias at time s_k involving the mean squared error at time $\delta(s_k)$

$$\mathbb{E}_{s_j}^{m_0} \left([\mathcal{W}_{s_j, s_k}^{m_0, \mathcal{S}}(X_{s_j, s_k}^{m_0, m_1})]^2 \right) \leq \mathcal{K}_{1, s_j, s_k}^{m_0} \mathbb{E}_{s_j}^{m_0} \left([\bar{u}_{s_j, \delta(s_k)}^{m_0, \mathcal{S}}(X_{s_j, \delta(s_k)}^{m_0, m_1}) - u_{\delta(s_k)}(X_{s_j, \delta(s_k)}^{m_0, m_1})]^2 \right).$$

- In addition to the bias term \mathcal{W} , Proposition 4.2 in [2] introduced also a variance term \mathcal{V} and a regression error term \mathcal{R} that can be used to decompose each mean squared error involved in the right side of inequality (4.3) and of inequality (4.4). For instance, using Proposition 4.2 of [2] we have

$$\begin{aligned}
\mathbb{E}_{s_j}^{m_0} \left([\bar{u}_{s_j, s_k}^{m_0, \mathcal{S}}(X_{s_j, s_k}^{m_0, m_1}) - u_{s_k}(X_{s_j, s_k}^{m_0, m_1})]^2 \right) &= \mathbb{E}_{s_j}^{m_0} (\mathcal{V}_{s_j, s_k}^{m_0, \mathcal{S}}(X_{s_j, s_k}^{m_0, m_1})) \\
&\quad + \mathbb{E}_{s_j}^{m_0} \left([\mathcal{R}_{s_j, s_k}^{m_0, \mathcal{S}}(X_{s_j, s_k}^{m_0, m_1}) + \mathcal{W}_{s_j, s_k}^{m_0, \mathcal{S}}(X_{s_j, s_k}^{m_0, m_1})]^2 \right)
\end{aligned}$$

which leads to a compromise between (regression error \mathcal{R} + variance \mathcal{V} + bias \mathcal{W}) and $\mathcal{K}_{2, s_j, s_k}^{m_0} \mathbb{E}_{s_j}^{m_0} \left(\left| \bar{X}_{s_j, s_k}^{m_0, m_1} - X_{s_j, s_k}^{m_0, m_1} \right|_{d_1'}^2 \right)$ in (4.4). As the latter term can be made very small with a continuity argument and fine discretization, we can thus stop the bias backpropagation by accepting an error that is virtually equal to zero!

Proposition 4.1 requires Lipschitz property of U that is fulfilled if f is Lipschitz. If f and g are Lipschitz, one can also show Lipschitz property for V solution of (SnI). It is then possible to replace U by V , u by v and B by C in Proposition 4.1. In contrast to U and V , it is harder to establish Lipschitz property of the solution of a discretized BSDE with a Z term. Using similar steps to the one presented in [29], Lemma 4.1 shows this Lipschitz property for the One step forward Dynamic Programming (ODP) scheme

$$\begin{cases} \hat{Y}_{s_k} &= \mathbb{E}_{s_k} \left(\hat{Y}_{\delta(s_k)} + \Delta_{s_k} f_{s_k}(X_{s_k}, \hat{Y}_{\delta(s_k)}, \hat{Z}_{s_k}) \right) \\ \hat{Z}_{s_k} &= \frac{1}{\Delta_{s_k}} \mathbb{E}_{s_k} \left(\hat{Y}_{\delta(s_k)} (W_{\delta(s_k)} - W_{s_k}) \right) = \frac{1}{\sqrt{\Delta_{s_k}}} \mathbb{E}_{s_k} \left(\hat{Y}_{\delta(s_k)} \zeta_{\delta(s_k)} \right) \end{cases}$$

where f uses the X component and the vector $\zeta_{\delta(s_k)}$ has a normal distribution $\mathcal{N}(0, I_{d_1})$. Replacing s_k by k , Δ_{s_k} by Δ_k and using Markov property (cf [18]) with

$$\begin{cases} y_k(x) &= \mathbb{E} \left(y_{k+1}(\mathcal{E}_k(x, \zeta_{k+1})) + \Delta_k f_k(x, y_{k+1}(\mathcal{E}_k(x, \zeta_{k+1})), z_k(x)) \right) \\ z_k(x) &= \frac{1}{\sqrt{\Delta_k}} \mathbb{E} \left(y_{k+1}(\mathcal{E}_k(x, \zeta_{k+1})) \zeta_{k+1} \right), \end{cases}$$

then $\hat{Y}_{s_k} = y_k(X_{s_k})$ and $\hat{Z}_{s_k} = z_k(X_{s_k})$.

Lemma 4.1. Assume that $f(t, x, y, z)$ is $[f]_{Lip}$ -Lipschitz continuous with respect to x, y and z uniformly in $t \in [0, T]$, for the particular case $f(T, x)$ we denote by $[f_T]_{Lip}$ the Lipschitz coefficient. The coefficients $b(t, x)$ and $\sigma(t, x)$ of the Markov process (2.3) are also assumed Lipschitz continuous in x uniformly with respect to $t \in [0, T]$ with Lipschitz coefficients denoted $[b]_{Lip}$ and $[\sigma]_{Lip}$. For a fixed value $n_0 \geq 1$, we assume that all increments Δ_k are smaller than T/n_0 and we define $C = [f]_{Lip} \left(1 + \left[\sqrt{\frac{d_1 [f]_{Lip}}{2}} + \sqrt{\frac{T [f]_{Lip}}{2n_0}} \right]^2 \right) + C_{b,\sigma}$ with $C_{b,\sigma} = [b]_{Lip} + \frac{1}{2} [\sigma]_{Lip}^2$, then for every time index k , y_k is $[y_k]_{Lip}$ -Lipschitz continuous with

$$[y_k]_{Lip} \leq \left([y_{k+1}]_{Lip} e^{\Delta_k C} + \Delta_k [f]_{Lip} \right).$$

Moreover the functions z_k are $[z_k]_{Lip}$ -Lipschitz continuous with

$$[z_k]_{Lip} \leq \frac{1}{\sqrt{\Delta_k}} [y_{k+1}]_{Lip} e^{\Delta_k C_{b,\sigma,T}} \sqrt{d_1}.$$

If $\Delta_k = h$ is homogeneous with respect to k , we have

$$[y_k]_{Lip} \leq \left([f_T]_{Lip} + \frac{[f]_{Lip}}{C} \right) e^{C(T-s_k)}$$

and

$$[z_k]_{Lip} \leq \sqrt{d_1} \sqrt{\frac{n_0}{T}} \left([f_T]_{Lip} + \frac{[f]_{Lip}}{C} \right) e^{C(T-s_k)} e^{\frac{T}{n_0} C_{b,\sigma,T}}.$$

Proof. Assume by backward induction that y_{k+1} is $[y_{k+1}]_{Lip}$ -Lipschitz continuous. For every $x, x' \in \mathbb{R}^{d_1}$, we have

$$\begin{aligned} y_k(x) - y_k(x') &= \mathbb{E} [y_{k+1}(\mathcal{E}_k(x, \zeta_{k+1})) - y_{k+1}(\mathcal{E}_k(x', \zeta_{k+1}))] \\ &+ \Delta_k \mathbb{E} [f_k(x, y_{k+1}(\mathcal{E}_k(x, \zeta_{k+1})), z_k(x)) - f_k(x', y_{k+1}(\mathcal{E}_k(x, \zeta_{k+1})), z_k(x))] \\ &+ \Delta_k \mathbb{E} [f_k(x', y_{k+1}(\mathcal{E}_k(x, \zeta_{k+1})), z_k(x)) - f_k(x', y_{k+1}(\mathcal{E}_k(x', \zeta_{k+1})), z_k(x))] \\ &+ \Delta_k \mathbb{E} [f_k(x', y_{k+1}(\mathcal{E}_k(x', \zeta_{k+1})), z_k(x)) - f_k(x', y_{k+1}(\mathcal{E}_k(x', \zeta_{k+1})), z_k(x'))] \end{aligned}$$

and, for any random vector $\zeta \sim \mathcal{N}(0, I_{d_1})$

$$z_k(x) - z_k(x') = \frac{1}{\sqrt{\Delta_k}} \mathbb{E} ([y_{k+1}(\mathcal{E}_k(x, \zeta)) - y_{k+1}(\mathcal{E}_k(x', \zeta))] \zeta). \quad (4.7)$$

We denote, when the denominator of these expressions are not equal to zero

$$\mathcal{J}_{x,x'} = \frac{[f_k(x, y_{k+1}(\mathcal{E}_k(x, \zeta_{k+1})), z_k(x)) - f_k(x', y_{k+1}(\mathcal{E}_k(x, \zeta_{k+1})), z_k(x))]_t}{|x - x'|_{d_1}^2} (x - x')$$

$$\mathcal{K}_{x,x'} = \frac{[f_k(x', y_{k+1}(\mathcal{E}_k(x, \zeta_{k+1})), z_k(x)) - f_k(x', y_{k+1}(\mathcal{E}_k(x', \zeta_{k+1})), z_k(x))]_t}{y_{k+1}(\mathcal{E}_k(x, \zeta_{k+1})) - y_{k+1}(\mathcal{E}_k(x', \zeta_{k+1}))}$$

$$\mathcal{L}_{x,x'} = \frac{[f_k(x', y_{k+1}(\mathcal{E}_k(x', \zeta_{k+1})), z_k(x)) - f_k(x', y_{k+1}(\mathcal{E}_k(x', \zeta_{k+1})), z_k(x'))]_t}{|z_k(x) - z_k(x')|_{d_1}^2} (z_k(x) - z_k(x'))$$

where, for any $x, x' \in \mathbb{R}^{d_1}$, $|x|_{d_1} = \sqrt{\langle x, x \rangle}$ and $\langle x, x' \rangle = {}^t x x'$. When the denominator is zero we set these quantities to 0 as well. Then $\mathcal{J}_{x,x}$, $\mathcal{K}_{x,x}$ and $\mathcal{L}_{x,x}$ are $\sigma(\zeta_{k+1})$ -measurable random vectors bounded by $[f]_{Lip}$ and

$$y_k(x) - y_k(x') = \mathbb{E}[(y_{k+1}(\mathcal{E}_k(x, \zeta_{k+1})) - y_{k+1}(\mathcal{E}_k(x', \zeta_{k+1}))) (1 + \Delta_k \mathcal{K}_{x,x'})] \\ + \Delta_k \langle \mathbb{E}(\mathcal{L}_{x,x'}), z_k(x) - z_k(x') \rangle + \Delta_k \mathbb{E}[\mathcal{J}_{x,x'}](x - x').$$

Note that

$$\langle \mathbb{E}(\mathcal{L}_{x,x'}), z_k(x) - z_k(x') \rangle = \mathbb{E} \left[y_{k+1}(\mathcal{E}_k(x, \zeta_{k+1})) - y_{k+1}(\mathcal{E}_k(x', \zeta_{k+1})) \langle \mathbb{E}(\mathcal{L}_{x,x'}), \zeta_{k+1} / \sqrt{\Delta_k} \rangle \right]$$

so that

$$y_k(x) - y_k(x') = \mathbb{E}[(y_{k+1}(\mathcal{E}_k(x, \zeta_{k+1})) - y_{k+1}(\mathcal{E}_k(x', \zeta_{k+1}))) (1 + \Delta_k \mathcal{K}_{x,x'} + \sqrt{\Delta_k} \langle \mathbb{E}(\mathcal{L}_{x,x'}), \zeta_{k+1} \rangle)] \\ + \Delta_k \mathbb{E}[\mathcal{J}_{x,x'}](x - x').$$

Consequently, using Cauchy-Schwartz inequality for the first term on the right hand side of the above decomposition and $|\mathcal{J}_{x,x'}| \leq [f]_{Lip}$ for the second,

$$|y_k(x) - y_k(x')| \leq \|y_{k+1}(\mathcal{E}_k(x, \zeta_{k+1})) - y_{k+1}(\mathcal{E}_k(x', \zeta_{k+1}))\|_2 \|1 + \Delta_k \mathcal{K}_{x,x'} + \sqrt{\Delta_k} \langle \mathbb{E}(\mathcal{L}_{x,x'}), \zeta_{k+1} \rangle\|_2 \\ + \Delta_k [f]_{Lip} |x - x'|_{d_1}.$$

By elementary computations already carried out in [5, 29], one classically has for the Euler scheme with step Δ_k ,

$$\|y_{k+1}(\mathcal{E}_k(x, \zeta_{k+1})) - y_{k+1}(\mathcal{E}_k(x', \zeta_{k+1}))\|_2 \leq [y_{k+1}]_{Lip} \|\mathcal{E}_k(x, \zeta_{k+1}) - \mathcal{E}_k(x', \zeta_{k+1})\|_2 \\ \leq [y_{k+1}]_{Lip} |x - x'|_{d_1} (1 + C_{b,\sigma} \Delta_k)$$

with $C_{b,\sigma} = [b]_{Lip} + \frac{1}{2}[\sigma]_{Lip}^2$. On the other hand

$$\mathbb{E}|1 + \Delta_k \mathcal{K}_{x,x'} + \sqrt{\Delta_k} \langle \mathbb{E}(\mathcal{L}_{x,x'}), \zeta_{k+1} \rangle|^2 = 1 + \Delta_k^2 \mathbb{E} \mathcal{K}_{x,x'}^2 + \Delta_k \mathbb{E} |\mathcal{L}_{x,x'}|^2 |\zeta_{k+1}|^2 + 2\Delta_k \mathbb{E}(\mathcal{K}_{x,x'}) \\ + 2\sqrt{\Delta_k} \underbrace{\langle \mathbb{E}(\mathcal{L}_{x,x'}), \mathbb{E} \zeta_{k+1} \rangle}_{=0} + 2\Delta_k^{3/2} \mathbb{E}[\mathcal{K}_{x,x'} \langle \mathbb{E}(\mathcal{L}_{x,x'}), \zeta_{k+1} \rangle] \\ \leq 1 + \Delta_k [f]_{Lip}^2 d_1 + \Delta_k^2 [f]_{Lip}^2 + 2\Delta_k^{3/2} [f]_{Lip}^2 \sqrt{d_1} + 2\Delta_k [f]_{Lip} \\ \leq 1 + 2\Delta_k [f]_{Lip} C_{f,d_1,T}$$

where we used $\mathbb{E}|\zeta_1|^2 = d_1$ in the first inequality and for fixed $n_0 > 1$ given $\Delta_k \leq T/n_0$ we set $C_{f,d_1,T} = \left(1 + \left[\sqrt{\frac{d_1 [f]_{Lip}}{2}} + \sqrt{\frac{T [f]_{Lip}}{2n_0}}\right]^2\right)$. Thus, we obtain

$$\|1 + \Delta_k \mathcal{K}_{x,x'} + \sqrt{\Delta_k} \langle \mathbb{E}(\mathcal{L}_{x,x'}), \zeta_{k+1} \rangle\|_2 \leq 1 + [f]_{Lip} C_{f,d_1,T} \Delta_k.$$

Plugging these in the above upper bound yields

$$|y_k(x) - y_k(x')| \leq [y_{k+1}]_{Lip} \left((1 + [f]_{Lip} C_{f,d_1,T} \Delta_k) (1 + C_{b,\sigma} \Delta_k) + \Delta_k [f]_{Lip} \right) |x - x'|_{d_1}$$

i.e.

$$\begin{aligned} [y_k]_{Lip} &\leq [y_{k+1}]_{Lip} \left((1 + [f]_{Lip} C_{f,d_1,T} \Delta_k) (1 + C_{b,\sigma} \Delta_k) \right) + \Delta_k [f]_{Lip} \\ &\leq \left([y_{k+1}]_{Lip} e^{\Delta_k C} + \Delta_k [f]_{Lip} \right) \end{aligned}$$

with

$$C = [f]_{Lip} C_{f,d_1,T} + C_{b,\sigma}.$$

As for the z term, using Lipschitz property of y_{k+1} and Cauchy-Schwarz's inequality then $\mathbb{E}|\zeta_1|^2 = d_1$ and $\|\mathcal{E}_k(x, \zeta_{k+1}) - \mathcal{E}_k(x', \zeta_{k+1})\|_2 \leq |x - x'|_{d_1} (1 + C_{b,\sigma} \Delta_k)$ we get

$$\begin{aligned} |z_k(x) - z_k(x')|_{d_1} &\leq \frac{1}{\sqrt{\Delta_k}} [y_{k+1}]_{Lip} \sqrt{\sum_{i=1}^{d_1} \mathbb{E}[(\mathcal{E}_k(x, \zeta_{k+1}) - \mathcal{E}_k(x', \zeta_{k+1}))^2] \mathbb{E}[\zeta_{k+1}^i]^2} \\ &\leq \frac{1}{\sqrt{\Delta_k}} [y_{k+1}]_{Lip} \sqrt{d_1} e^{\Delta_k C_{b,\sigma}} |x - x'|_{d_1}. \end{aligned}$$

Thus, z_k is Lipschitz continuous with coefficient $[z_k]_{Lip}$ satisfying

$$[z_k]_{Lip} \leq \frac{1}{\sqrt{\Delta_k}} [y_{k+1}]_{Lip} e^{\Delta_k C_{b,\sigma,T}} \sqrt{d_1}.$$

Assuming homogeneous time increment $\Delta_k = h = T/n$ with $n \leq 2^L$, then for $k \leq n$ we have

$$e^{Ckh} [y_k]_{Lip} \leq [y_{k+1}]_{Lip} e^{C(k+1)h} + e^{Ckh} [f]_{Lip} h.$$

which yields

$$\begin{aligned} e^{Ckh} [y_k]_{Lip} &\leq [f_T]_{Lip} e^{Cnh} + [f]_{Lip} h \sum_{l=k}^{n-1} e^{Clh} \\ &\leq [f_T]_{Lip} e^{CT} + [f]_{Lip} h \frac{e^{CT} - e^{Ckh}}{e^{Ch} - 1} \\ &\leq [f_T]_{Lip} e^{CT} + [f]_{Lip} h \frac{e^{CT}}{e^{Ch} - 1} \\ &\leq [f_T]_{Lip} e^{CT} + [f]_{Lip} e^{CT}/C. \end{aligned}$$

Finally we have

$$[y_k]_{Lip} \leq [f_T]_{Lip} e^{C(T-s_k)} + [f]_{Lip} e^{C(T-s_k)}/C \quad (4.8)$$

and

$$[z_k]_{Lip} \leq \frac{1}{\sqrt{h}} ([f_T]_{Lip} + \frac{[f]_{Lip}}{C}) e^{C(T-s_k-h)} e^{hC_{b,\sigma}} \sqrt{d_1}.$$

□

5 Some numerical results

In this section we test the presented conditional MC learning procedure on American option and risk measure. All simulations are run on a laptop that has an Intel i7-7700HQ CPU and a single GeForce GTX 1060 GPU programmed with the CUDA/C application programming interface. We refer the reader to [31] for an introduction to CUDA programming. Like in [2], we used the code of the contribution [3] in order to perform the batch parallel regressions induced by the proposed 1NMC simulation combined with local regressions. The very competitive execution times that we present hereafter are possible not only because of the GPU use but also due to the adaptation of parallelization strategies presented in [3].

5.1 American geometric put option

Given the (Snl) setting of Section 2.2 with a driver $f = 0$, we consider an American geometric put option with constant interest rate r and a payoff

$$g(x) = \left[K - \prod_{i=1}^{d_1} (x_i)^{1/d_1} \right]^+ \quad (5.1)$$

with an asset X given by $X_t^i = X_s^i \exp\left(\left(r - \frac{\sigma^2}{2}\right)(t - s) + \sigma(W_t^i - W_s^i)\right)$, $t > s$, $1 \leq i \leq d_1$, $r = \log(1.1)$, $\sigma = 0.4$, $K = X_0^i = 100$ and $d_1 = 20$.

Table 1: Numerical simulations for American option (5.1) simulated formula, [BC] bias control [VA] variance adjustment: $d_1 = 20$, $M_0 = 2^{11}$, $M_1 = 2^{12}$.

	L = 2 (T = 0.5)	L = 3 (T = 1)	L = 4 (T = 2)
[VA] ($\bar{s} = T$)	2.561 (± 0.035)	4.236 (± 0.042)	6.363 (± 0.054)
[BC] ($\bar{s} = (s + 1/4) \wedge T$)	2.493 (± 0.041)	3.734 (± 0.061)	5.130 (± 0.089)
[VA] + [BC] ($\bar{s} = (s + 1/4) \wedge T$)	2.291 (± 0.035)	2.890 (± 0.037)	3.961 (± 0.055)
Real Price	2.153	2.871	3.754

We approximate the price V_0 associated to payoff (5.1). We choose the dimension $d_1 = 20$ to make sure that the variance of $g(X)$ is sufficiently large. We point out however that it works well for $d_1 = 100$. In Table 1 we show the price of an American geometric put option, calculated by simulated expression V_0^{sim} , for different maturities. Indeed, V_0^{learn} provides almost the same values. From top to bottom we have: a variance adjustment [VA], a bias control [BC] and a combination of [BC] and [VA]. We show that the simulated expression with a combination of [BC] and [VA] gives a good approximation of the price even for long maturity $T = 2$. One can strengthen the bias control on a finer discretization time grid \mathcal{S} but this will increase further the complexity. In Table 1 [VA] + [BC], we choose a standard time discretization for the approximation of American options with $\bar{s} = (s + 1/4) \wedge T$ and we show that the variance adjustment is needed for better approximation of the conditional expectation of scenarios close to the exercise frontier.

Figure 6 shows the difference between two approximations $\frac{1}{M_0} \sum_{m_0=1}^{M_0} e^{-r\Delta_s} \tilde{v}_{\delta(s), \bar{\delta}(s)}^{m_0, \mathcal{S}}$ and $\frac{1}{M_0} \sum_{m_0=1}^{M_0} (\tilde{w}_{s, \bar{s}}^{m_0, \mathcal{S}})$ of the same quantity with respect to the time discretization. For $s \in \mathcal{S} = \{0, \frac{1}{8}, \frac{2}{8}, \frac{3}{8}, \frac{4}{8}, \frac{5}{8}, \frac{6}{8}, \frac{7}{8}, 1\}$, we implement on the left the conditional MC procedure using $\bar{s} = T$. On the right, we implement the conditional MC procedure controlling the bias with $\bar{s} = (s + \frac{1}{4}) \wedge T$. We show that the control allows to reduce the bias propagation.

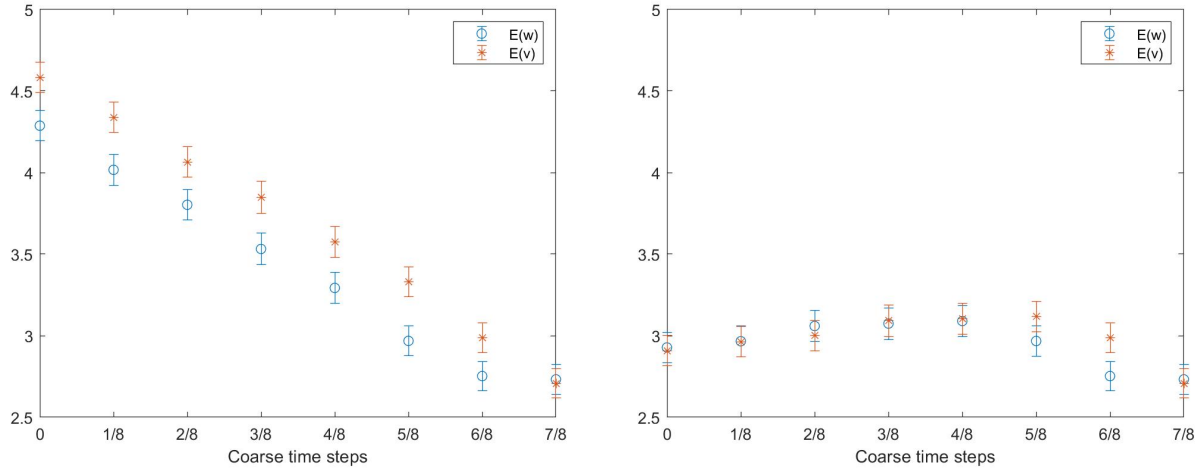


Figure 6: $\frac{1}{M_0} \sum_{m_0=1}^{M_0} e^{-r(\delta(s)-s)} \tilde{v}_{\delta(s), \bar{\delta}(s)}^{m_0, \mathcal{S}}$ vs. $\frac{1}{M_0} \sum_{m_0=1}^{M_0} (\tilde{w}_{s, \bar{s}}^{m_0, \mathcal{S}})$; [Left] $\bar{s} = T$ [Right] $\bar{s} = (s + \frac{1}{4}) \wedge T$; $d_1 = 20$, $M_0 = 2^{11}$, $M_1 = 2^{12}$, $T = 1$ and $L = 3$.

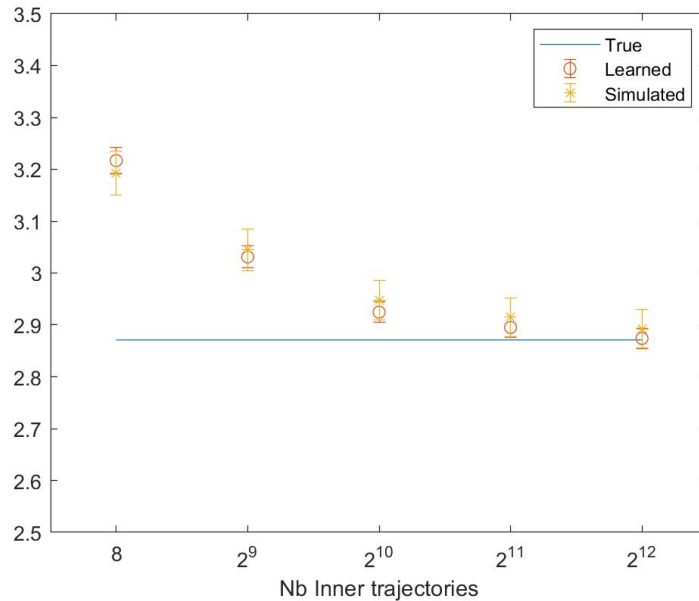


Figure 7: Numerical approximation of the price V_0 : $d_1 = 20$, $M_0 = 2^9$, $T = 1$, $L = 3$.

Figure 7 shows the approximation of the American geometric put option, calculated by learned and simulated expression, with respect to the number of inner trajectories for different maturities. Both expressions converge to the benchmark value for 2^9 outer trajectories and 2^{12} inner trajectories in 3 seconds.

5.2 Initial Margin

We consider a portfolio of one hundred put options, the price U_{s_k} of the portfolio at time step s_k is given by

$$U_{s_k} = \sum_{i=0}^{d_1} e^{-(T-s_k)r} \mathbb{E}_{s_k} \left([K - X_T^i]^+ \right) \quad (5.2)$$

with an asset X given by $X_t^i = X_s^i \exp \left(\left(r - \frac{\sigma^2}{2} \right) (t - s) + \sigma (W_t^i - W_s^i) \right)$, $t > s$, $1 \leq i \leq d_1$, with r the interest rate, K the strike and T the maturity.

We calculate the Initial Margin (IM) of this portfolio that is an amount posted by the counterparty (or the bank) to overcome the loss of the portfolio during the liquidation period after a default. IM is formalized here as follows

$$\text{IM}_{s_k} = \mathbb{E}S_{s_k}^a (L_{s_k, \delta(s_k)}) \quad (5.3)$$

where the loss of the portfolio at time s_k over a period Δ_{s_k} is denoted $L_{s_k, \delta(s_k)}$ and is defined here by

$$L_{s_k, \delta(s_k)} = U_{\delta(s_k)} - U_{s_k},$$

and the expected shortfall $\mathbb{E}S$ of level a is defined at time s_k for any random variable Y by

$$\mathbb{E}S_{s_k}^a (Y) = \frac{1}{(1-a)} \int_a^1 \text{VaR}_{s_k}^\alpha (Y) d\alpha.$$

The value-at-risk $\text{VaR}^\alpha(Y)$ of level α of some random variable Y conditionally to \mathcal{F}_{s_k} is defined by

$$\text{VaR}_{s_k}^\alpha (Y) = \inf \{ x \in \mathbb{R} : \mathbb{P}(Y \leq x \mid \mathcal{F}_{s_k}) \geq \alpha \}.$$

We considered the following parameters: $T = 1$, $d_1 = 100$, $K = X_0^i = 100$, $r = 0.01$, $a = 99\%$ and $\Delta_{s_k} = \frac{1}{32}$. For each $s_k \in \mathcal{S} = \left\{ \frac{l}{32}; l = 0, \dots, 32 \right\}$, we simulate U_{s_k} using $\tilde{u}_{s_k, T}^{m_0, \mathcal{S}}$ and we simulate $U_{\delta(s_k)}$ using $\bar{u}_{s_k, s_k + \Delta_{s_k}}^{m_0, \mathcal{S}} (X_{s_k, s_k + \Delta_{s_k}}^{m_0, m_1})$. A benchmark approximation of the IM is obtained using Black & Scholes formula for put options.

Figure 8 shows few distributions of the loss process. From top to bottom we show different time steps $s_k \in \left\{ \frac{29}{32}, \frac{19}{32}, \frac{9}{32} \right\}$. On the left, we perform the procedure without variance adjustment and on the right we perform the variance adjustment. We show that the variance adjustment is necessary to fit the benchmark distribution of the loss process.

Figure 9 shows the initial margin distribution. From top to bottom we show different time steps $s_k \in \left\{ \frac{29}{32}, \frac{19}{32}, \frac{9}{32} \right\}$. The tail distribution of the loss process is very well approximated and provides a fairly good representation of the distribution of IM.

Figure 10 shows the mean of IM with respect to the time horizon of the portfolio at the top and the L2 relative error at the bottom. The relative error is sufficiently small as it is generally less than 8% and does not exceed 11%.

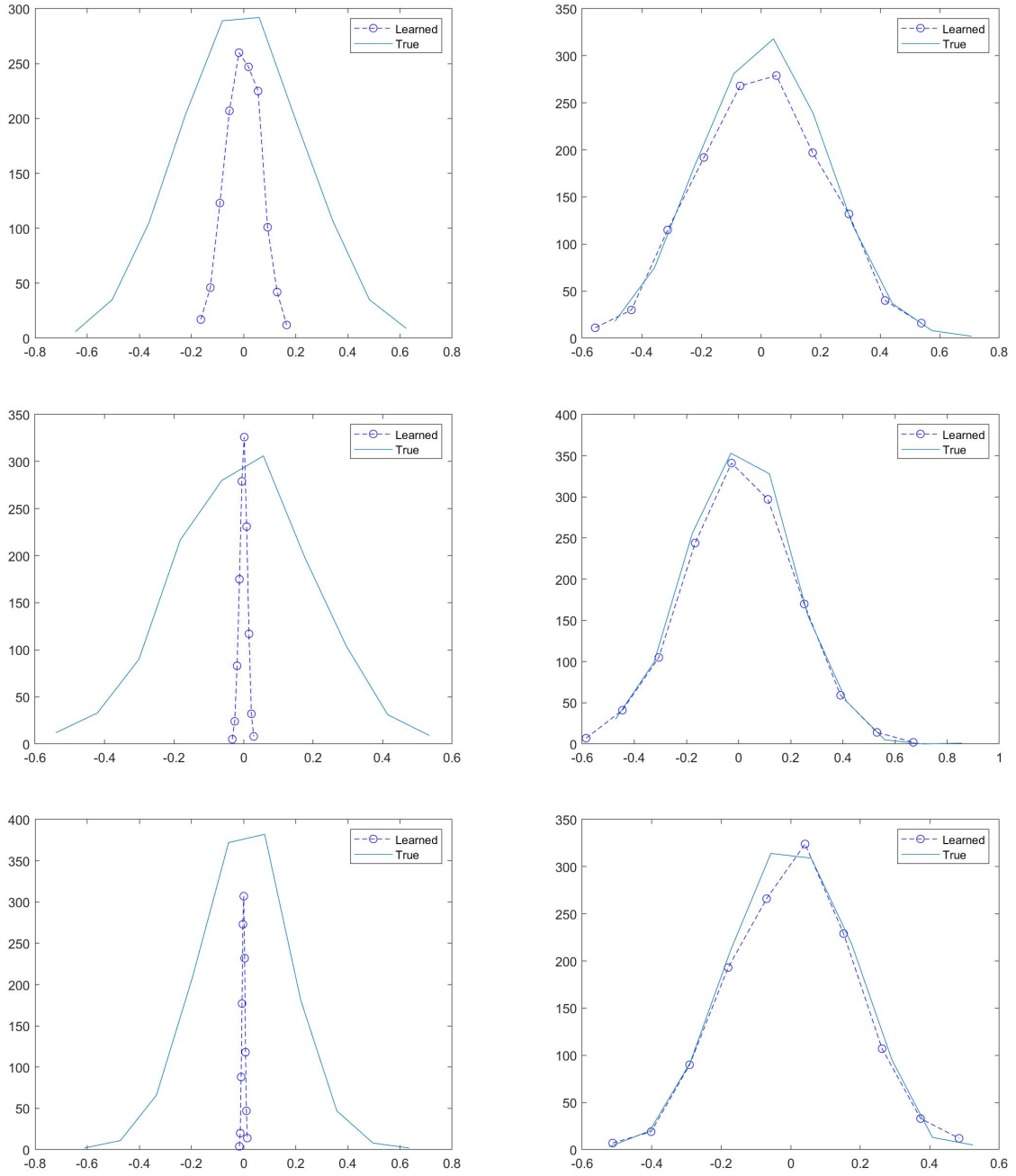


Figure 8: Numerical approximation of loss $L_{s,s+\frac{1}{32}}$ distribution [Left] Without variance adjustment, [Right] With variance adjustment; [top to bottom] $s_k \in \{\frac{29}{32}, \frac{19}{32}, \frac{9}{32}\}$; $M_0 = 2^8$, $M_1 = 2^8 \times 5$.

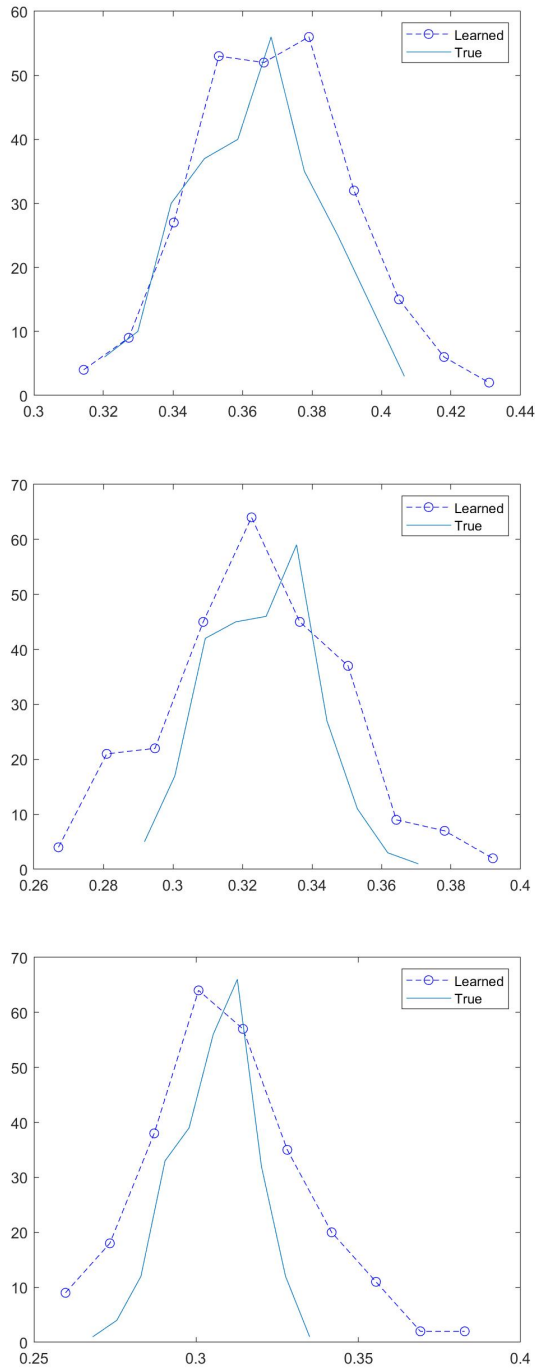


Figure 9: Numerical approximation of IM_s distribution: [top to bottom] $s \in \{\frac{29}{32}, \frac{19}{32}, \frac{9}{32}\}$; $M_0 = 2^8$, $M_1 = 2^8 \times 5$.

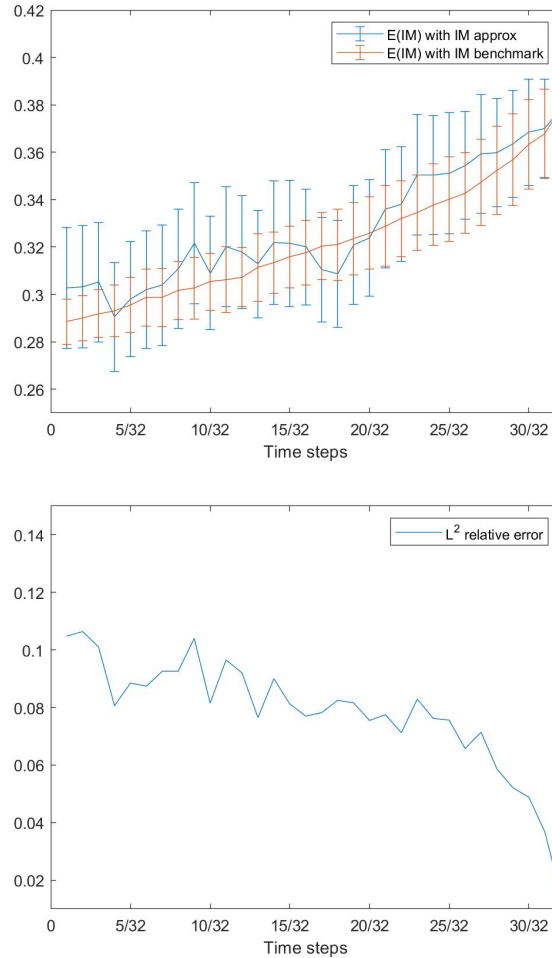


Figure 10: Initial Margin: [Top] mean of IM_s ; [Bottom] L^2 relative error.

References

- [1] ABBAS-TURKI, L. A., CRÉPEY, S. and DIALLO, B. (2018). XVA principles, nested Monte Carlo strategies, and GPU optimizations. *International Journal of Theoretical and Applied Finance*. **21(06)**.
- [2] ABBAS-TURKI, L. A., DIALLO, B., PAGÈS, G. (2020) Conditional Monte Carlo Learning for Diffusions I: main methodology and application to backward stochastic differential equations. *Preprint on HAL*.
- [3] ABBAS-TURKI, L. A. and GRAILLAT, S. (2017). Resolving small random symmetric linear systems on graphics processing units. *The Journal of Supercomputing*. **73(4)**, 1360–1386.
- [4] AGARWAL, A., DE MARCO, S., GOBET, E. and LIU, G. (2018). Study of new rare event simulation schemes and their application to extreme scenario generation. *Mathematics and Computers in Simulation*. **143** 89–98.
- [5] BALLY, A. and PAGÈS, G. (2003). A quantization algorithm for solving discrete time multidimensional optimal stopping problems. *Bernoulli*. **9(6)** 1003–1049.

- [6] BELLMAN, R. (2010). *Dynamic programming*, Princeton Landmarks in Mathematics. Princeton University Press, Princeton, NJ. Reprint of the 1957 edition, With a new introduction by Stuart Dreyfus.
- [7] BOUCHARD, B. and TOUZI, N. (2004). Discrete time approximation and Monte Carlo simulation of backward stochastic differential equations. *Stochastic Processes and their Applications*. **111(2)** 175–206.
- [8] BROADIE, M., DU, Y. and MOALLEMI, C. C. (2015). Risk estimation via regression. *Operations Research*. **63(5)** 1077–1097.
- [9] BUCKLEW, J. (2004). *Introduction to rare event simulation*, Springer Series in Statistics.
- [10] BUEHLER, H., GONON, L., TEICHMANN, J. and WOOD, B. (2019). Deep hedging. *Quantitative Finance*. **19(8)** 1271–1291.
- [11] CLÉMENT, E., LAMBERTON, D. and PROTTER, P. (2002). An analysis of a least squares regression algorithm for American option pricing. *Finance and Stochastics*. **17** 448–471.
- [12] E, W., HAN, J. and JENTZEN, A. (2018). Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Communications in Mathematics and Statistics*. **5(4)** 349–380.
- [13] EL KAROUI, N., PENG, S. and QUENEZ, M. C. (1997). Backward stochastic differential equations in Finance. *Mathematical Finance*. **7(1)** 349–380.
- [14] GLASSERMAN, P. (2003). *Monte Carlo methods in financial engineering*, Stochastic Modelling and Applied Probability, Springer-Verlag New York Inc.
- [15] GIORGI, D., LEMAIRE, V. and PAGÈS, G. (2020). Weak Error for Nested Multilevel Monte Carlo. *Methodology and Computing in Applied Probability*. **22** 1325–1348.
- [16] GOBET, E. and LABART, C. (2007). Error expansion for the discretization of backward stochastic differential equations. *Stochastic Processes and their Applications*. **117(7)** 803–829.
- [17] GOBET, E., LEMOR, J. P. and WARIN, X. (2005). A regression-based Monte Carlo method to solve backward stochastic differential equations *The Annals of Applied Probability*. **15(3)** 2172–2202.
- [18] GOBET, E. and TURKEDJIEV, P. (2016). Linear regression MDP scheme for discrete backward stochastic differential equations under general conditions. *Mathematics of Computation*. **85** 1359–1391.
- [19] GOBET, E. and TURKEDJIEV, P. (2017). Adaptive importance sampling in least-squares Monte Carlo algorithms for backward stochastic differential equations. *Stochastic Process. Appl.* **127(4)** 1171–1203.
- [20] GORDY, M. B. and JUNEJA, S. (2010). Nested Simulation in Portfolio Risk Measurement. *Management Science*. **56(10)** 1833–1848.

- [21] GOUDENÈGE, L., MOLENT, A. and ZANETTE, A. (2019). Variance Reduction Applied to Machine Learning for Pricing Bermudan/American Options in High Dimension. *Preprint*. <https://arxiv.org/abs/1903.11275>
- [22] JOURDAIN, B. and LELONG, J. (2009). Robust adaptive importance sampling for normal random vectors. *The Annals of Applied Probability*. **19(5)** 1687–1718.
- [23] LEE, S.-H. (1998). *Monte Carlo Computation of Conditional Expectation Quantiles*, Ph.D. thesis, Stanford University.
- [24] LEE, S.-H. and GLYNN, P. W. (2003). Computing the distribution function of a conditional expectation via Monte Carlo: Discrete conditioning spaces. *ACM Transactions on Modeling and Computer Simulation*. **13(3)** 238–258.
- [25] LEMOR, J. P., GOBET, E. and WARIN, X. (2006). Rate of convergence of an empirical regression method for solving generalized backward stochastic differential equations. *Bernoulli*. **12(5)** 889–916.
- [26] LONGSTAFF, F. A. and SCHWARTZ, E. S. (2001). Valuing American options by simulation: A simple least-squares approach. *The Review of Financial Studies*. **14(1)** 113–147.
- [27] NEWEY, W. K. (1997). Convergence rates and asymptotic normality for series estimators. *Journal of Econometrics*. **79** 147–168.
- [28] PAGÈS, G. (2018). *Numerical probability: an introduction with applications to finance*, Universitext, Springer.
- [29] PAGÈS, G. and SAGNA, A. (2018). Improved error bounds for quantization based numerical schemes for BSDE and nonlinear filtering. *Stochastic Processes and their Applications*. **128(3)** 847–883.
- [30] TSITSIKLIS, J. N. and VAN ROY, B. (2001). Regression methods for pricing complex American-style options. *IEEE Transactions on Neural Networks*. **12(4)** 694–703.
- [31] NVIDIA (2017). *Cuda C PROGRAMMING GUIDE*.