Conditional Monte Carlo Learning for diffusions

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

October 17, 2019

Abstract

We present a new algorithm based on a One-layered Nested Monte Carlo (1NMC) to simulate functionals U of a Markov process X. The main originality of the proposed method comes from the fact that it provides a recipe to simulate $U_{t\geq s}$ conditionally on X_s. This recipe can be used for a large number of situations including: Backward Stochastic Differential Equations (BSDEs), Reflected BSDEs (RBSDEs), risk measures and beyond. In contrast to previous works, our contribution is based on a judicious combination between regression and 1NMC used for localization purpose. The generality, the stability and the iterative nature of this algorithm, even in high dimension, make its strength. It is of course heavier than a straight Monte Carlo (MC), however it is far more accurate to simulate quantities that are almost impossible to simulate with MC. Indeed, using the double layer of trajectories, we explain how to estimate and control the bias propagation. With this double layer structure, it is also possible to adjust the variance for a better description of tail events. Moreover, the parallel suitability of 1NMC makes it feasible in a reasonable computing time. This paper explains this algorithm and details error estimates. We also provide various numerical examples with a dimension equal to 100 that are executed in few minutes on one Graphics Processing Unit (GPU).

*Email: lokmane.abbas_turki@sorbonne-universite.fr. LPSM (UMR 8001), Sorbonne Université, 4, Place Jussieu 75005 Paris.

†Email: babacar.diallo@ca-cib.com. Quantitative Research GMD/GMT Crédit Agricole CIB, 92160 Montrouge, France; LaMME, Univ. Evry, CNRS, Université Paris-Saclay, 91037, Evry, France; LPSM, Sorbonne Université, 4, Place Jussieu 75005 Paris.

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

Contents

1 Introduction

Numerous contributions in numerical methods based on Monte Carlo reached recently their limits in dealing with the curse of dimensionality [5]. In contrast to previous works, our method is based on a judicious combination between 1NMC and the use of regression. This paper is a natural progress of an increasing interest in NMC started in [19, 21, 22] and used with regression in [1, 7]. Considering a filtered probability space $(\Omega, \mathcal{F}, \mathcal{F}_{0\leq t\leq T}, P)$, an $\mathcal{F}_t$-Markov process $(X_t)_{t\in[0,T]}$ taking its values on $\mathbb{R}^{d_1}$ and the time discretization $\{t_0, ..., t_{2^L}\} = \left\{0, T/2^L, ..., T\right\}$, let U_s be a functional of X defined for $s \in \{t_0, ..., t_{2^L}\}$ by

$$(f) \qquad U_s = \mathbb{E}_s\left(\sum_{t_k\geq s}^{t_{2^L}} f(t_k, X_{t_k}, X_{t_{k+1}})\right) = \mathbb{E}\left(\sum_{t_k\geq s}^{T} f(t_k, X_{t_k}, X_{t_{k+1}})\middle|\mathcal{F}_s\right),$$

where $\mathbb{E}_s(\cdot) = \mathbb{E}\left(\cdot\middle|\mathcal{F}_s\right)$, the expectation is always considered under 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 $\mathbb{E}(f^2(t_k, X_{t_k}, X_{t_{k+1}})) < \infty$ with convention $f(t_{2^L}, X_{t_{2^L}}, X_{t_{2^L+1}}) = f(t_{2^L}, X_{t_{2^L}})$. The simulation of U is generic to all BSDEs and RBSDEs examples presented in this paper. As nested simulations involve heavy notations, it is easier to present the whole algorithm for the simulation of U then apply it on specific examples.

When previous contributions target estimations of U_{t_k} for $k = 0, ..., 2^L$ knowing some realization of $\{X_{t_j}\}_{0\leq j\leq k}$ $(m_0 = 1, ..., M_0)$, our purpose is to simulate approximations $\{U^{m_0,m_1}_{t_k,s}\}_{s\geq t_{k+1}}$, with $(m_0 = 1, ..., M_0)$ and $(m_1 = 1, ..., M_1)$, of $\{U_s\}_{s\geq t_{k+1}}$ conditionally on the realization $\{X^{m_0}_{t_j}\}_{0\leq j\leq k}$. This task requires the simulation of a first layer $(X^{m_0})_{m_0=1,...,M_0}$ of trajectories that are kept in the machine's memory, then a second unstored layer $(X^{m_0,m_1})_{m_1=1,...,M_1}$ of trajectories, on the top of the first layer, only used to learn how should we perform approximations $\{U^{m_0,m_1}_{t_k,s}\}_{s\geq t_{k+1}}$.Although more complex, this procedure provides much more information on the process U. In particular, we use $\displaystyle\frac{1}{M_1}\sum_{m_1=1}^{M_1}\left(f(t_k, X^{m_0}_{t_k}, X^{m_0,m_1}_{t_{k+1}}) + U^{m_0,m_1}_{t_k,t_{k+1}}\right)$ to have the first layer approximation $U^{m_0}_{t_k}$ of U_{t_k}. Knowing the second layer approximation U^{m_0,m_1}, we can compute quantiles on U or, even more remarkable, can simulate another process $\widetilde{U}$ that satisfies equation $(\widetilde{f})$ (Replace f by $\widetilde{f}$ in equation (f)) with an $\widetilde{f}$ that can be a function of U like for instance $\widetilde{f}(t_k, x, y) = f(t_k, U_{t_k}(x), U_{t_{k+1}}(y))$. Consequently, when sufficient assumptions are satisfied, we can learn how to compute functionals of functionals of X with the same 1NMC. This latter fact makes possible a straightforward simulation of Valuation Adjustments [1] as long as one can write them as a composition of functionals then start simulating by the innermost functional till the most outer composition.

Although we are not the first to propose a learning procedure for BSDEs [12], we are the first to do it using nested Monte Carlo instead of a neural network. To the best of our knowledge, we are also first to provide a comprehensive presentation of an iterative algorithm where the accuracy of the estimator $\{U^{m_0}_{t_k}\}_{k=0,...,2^L}$ improves by adding more regression steps and thus by increasing the learning depth. Thanks to our method, one can easily balance between complexity and accuracy. Moreover, it is possible to improve the accuracy in a parareal fashion [24] which increases further the parallel scalability of the algorithm. In addition to that, we use equality

$$\mathbb{E}(U_s) = \mathbb{E}\left(U_{s'} + \sum_{t_{l+1}>s}^{s'} f(t_l, X_{t_l}, X_{t_{l+1}}) \right)$$

true for $s' > s$, and its localized version for each interval $[a, b]$

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

to present a nonparametric method to effectively estimate and control the bias. In the same fashion, we detail a variance adjustment procedure based on the equality

$$\mathbb{E}\left(\mathrm{Var}_s(U_{s'})\right) = \mathbb{E}\left(\mathbb{E}_s\left([U_{s'} - \mathbb{E}_s(U_{s'})]^2\right)\right) = \mathbb{E}\left([U_{s'} - \mathbb{E}_s(U_{s'})]^2\right).$$

true for $s' > s$, and its localized version for each interval $[a, b]$

$$\mathbb{E}\left(\mathrm{Var}_s(U_{s'}) 1_{\{\mathrm{Var}_s(U_{s''})\in[a,b]\}}\right) = \mathbb{E}\left(1_{\{\mathrm{Var}_s(U_{s''})\in[a,b]\}}[U_{s'} - \mathbb{E}_s(U_{s'})]^2\right),$$

true for $s' > s$ and $s'' > s$. The proposed variance adjustment strategy makes possible the nested simulation of distribution tails without requiring an importance sampling technique [20]. The good representation of tail events, via variance adjustment, becomes necessary for some nonlinear problems especially RBSDEs. Both bias control and variance adjustment shows that: 1NMC makes possible a very fine tracking of the bias of the first layer fine estimator U^{m_0} and the variance of the second layer coarse estimator U^{m_0,m_1}.

Focusing on the simulation of U given in (f), Section 2 introduces the method as well as notations. Section 2 also presents the iterative procedure, the bias control and the variance adjustment strategy on the approximation of U. Section 3 illustrates the presented method on some standard problems involving BSDEs, RBSDEs and risk measures. These examples show how the algorithm should be adapted to different situations, in particular how to set: iterations, bias control and variance adjustment for BSDEs and optimal stopping problems. Section 4 details the required assumptions in a general diffusion setting. It also provides different error estimates associated to our method and gives a sense to the overall approximation procedure. Section 5 shows the robustness of our method on highly dimensional problems beyond what is known to be possible in previous papers.

2 Conditional learning procedure: Notations and method

In Section 2.1, we present the algorithm steps and what should be done to stabilize it. As needed for any learning method, the initialization is also explained in Section 2.1. This will set the stage to express, in Section 2.2, the regression based approximations as an output of an iterative procedure. Details on the computation of the regression are provided in Section 2.3 that also includes a bias control and a variance adjustment necessary when targeting the tail events.

2.1 Iterative procedure, regression initialization and stabilization

Using a sufficiently fine discretization $\{t_0, ..., t_{2^L}\} = \{0, \Delta_t, 2\Delta_t, ..., T\}$ with $\Delta_t = T/2^L$, one simulates M_0 realizations $(X^{m_0}_{t_k})^{m_0=1,...,M_0}_{k=1,...,2^L}$ of the Markov process X starting at a deterministic point $X_0 = x_0 \in \mathbb{R}^{d^1}$ with the following induction

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

where $(\xi^{m_0}_{t_k})^{m_0=1,...,M_0}_{k=1,...,2^L}$ are independent realizations of an $\mathbb{R}^{d_2}$ random vector ξ and $(\mathcal{E}_{t_k})_{k=0,...,2^L-1} : \mathbb{R}^{d_1+d_2} \to \mathbb{R}^{d_1}$ are Borel-measurable functions. We use $X^{m_0,1}_{t_k}$, ..., $X^{m_0,d_1}_{t_k}$ to denote the d_1 components of the vector $X^{m_0}_{t_k}$. The sample $(X^{m_0}_{t_k})^{m_0=1,...,M_0}_{k=1,...,2^L}$ stays on the machine memory and is supposed to approximate accurately $(X_t)_{t\in[0,T]}$ in a sense explained in Section 4.

For a decreasing sequence $(s_j)_{j=0,...,2^L}$ that takes its values in the time discretization set $\{t_0, ..., t_{2^L}\}$, an extra simulation conditional to the starting $X^{m_0}_{s_j}$ is needed for the learning procedure. Introducing independent realizations $(\xi^{m_0,m_1}_{t_j,t_k})^{(m_0,m_1)\in\{1,...,M_0\}\times\{1,...,M_1+M_1'\}}_{k\in\{j,...,2^L\},j\in\{1,...,2^L\}}$ of the random vector ξ that are also independent from $(\xi^{m_0}_{t_k})^{m_0=1,...,M_0}_{k=1,...,2^L}$, we set for $t_{k-1} \geq s_j$

$$X^{m_0,m_1}_{s_j,t_k} = \mathcal{E}_{t_{k-1}}(X^{m_0,m_1}_{s_j,t_{k-1}}, \xi^{m_0,m_1}_{s_j,t_k}) \text{ and } X^{m_0,m_1}_{s_j,s_j}\Big|_{m_1=1,...,M_1+M_1'} = X^{m_0}_{s_j}. \tag{2.2}$$

We use $X^{m_0,m_1,1}_{s_j,t_k}$, ..., $X^{m_0,m_1,d_1}_{s_j,t_k}$ to denote the d_1 components of the vector $X^{m_0,m_1}_{s_j,t_k}$. For $s_j \leq s_l \leq s_k$, we also introduce the notation $X^{m_0,m_1}_{s_j,s_l:s_k}$ and $\xi^{m_0,m_1}_{s_j,s_l:s_k}$ for respectively $(X^{m_0,m_1}_{s_j,s_l}, X^{m_0,m_1}_{s_j,s_l+\Delta_t}, ..., X^{m_0,m_1}_{s_j,s_k-\Delta_t}, X^{m_0,m_1}_{s_j,s_k})$ and $(\xi^{m_0,m_1}_{s_j,s_l}, \xi^{m_0,m_1}_{s_j,s_l+\Delta_t}, ..., \xi^{m_0,m_1}_{s_j,s_k-\Delta}, \xi^{m_0,m_1}_{s_j,s_k})$.

For a positive integer $L' \in]L/2, L]$, the value of each term of the sequence $(s_j)_{j=0,...,2^L}$ is given by its corresponding term in $(T - s_j^i)_{j=0,...,2^L}$ which is defined iteratively for $i = 0, ..., L-L'$ starting with a homogeneously distributed sequence where each term is repeated $2^{L-L'}$ times as follows

$$(s_j^0)_{j=1,...,2^L} = \left\{ \frac{T}{2^{L'}}, ..., \frac{T}{2^{L'}}, ..., \frac{(2^{L'}-1)T}{2^{L'}}, ..., \frac{(2^{L'}-1)T}{2^{L'}}, T, ..., T \right\}, \; s_0^0 = 0. \tag{2.3}$$

We denote $\mathcal{S}^i$ the set of values taken by $(T - s_j^i)_{j=0,...,2^L}$, for example $\mathcal{S}^0 = \{T, (2^{L'}-1)T/2^{L'}, ..., T/2^{L'}, 0\}$.

The goal of iterations is to reduce an error term $(e_{T-s_j^i})_{j=1,...,2^L}^{i=0,...,L-L'}$ to make it smaller than some threshold error ε. The expression of the $\mathbb{R}$-valued random processes e and ε will be given in definitions 2.1, 3.1 and 3.2.

We set $j_0^* = s_0^i\Big|_{i=1,...,L-L'} = \max(\emptyset) = 0$, for each step $i = 1, ..., L-L'$ we define $Q_i = 2^{L-L'-i}$ and $(\hat{s}_j^{i-1})_{j=0,...,2^L}$

$$\begin{cases} \text{When } j \leq j_{i-1}^* \text{ define } \hat{s}_j^{i-1} = s_j^{i-1} \\ \text{Otherwise, for } j' > j_{i-1}^*/Q_i \text{ set } \hat{s}_j^i\Big|_{j=Q_i(j'-1)+1}^{Q_i j'} = \dfrac{s_{Q_i j'}^{i-1} + s_{Q_i(j'-1)}^{i-1}}{2}, \end{cases} \tag{2.4}$$

and we denote $\widehat{\mathcal{S}}^{i-1}$ the set of values taken by $(T - \hat{s}_j^{i-1})_{j=0,...,2^L}$. Then, we consider the following actualization strategy:

1. Compute $\left(e_{T-s_j^{i-1}}^{\mathcal{S}^{i-1}}\right)_{j>j_{i-1}^*}$

2. Use $j_i^* = j_{i-1}^* \vee \max\left(\left\{ j > j_{i-1}^*; \; e_{T-s_k^{i-1}}^{\mathcal{S}^{i-1}} < \varepsilon_{T-s_k^{i-1}}^{\mathcal{S}^{i-1}} \text{ for } k \leq j \right\}\right)$ with $x \vee y = \max(x, y)$ to define

$$s_j^i = s_j^{i-1} 1_{j \leq j_i^*} + \hat{s}_j^{i-1} 1_{j > j_i^*}. \tag{2.5}$$

The notation $s_j^i\Big|_{j=Q_i(j'-1)+1}^{Q_i j'}$ is used for $s_{Q_i(j'-1)+1}^i, ..., s_{Q_i j'}^i$. In Figure 1, we illustrate how this discretization strategy is implemented, in particular we chose $L' > L/2$.

Remark 2.1. *Expression (2.5) ensures that $s_{2^L}^i$ is always equal to T. Thus $s_{2^L} = 0$ which will be involved in definitions 2.1, 3.1 and 3.2 to introduce both a simulated value and an average on learned values at time 0.*

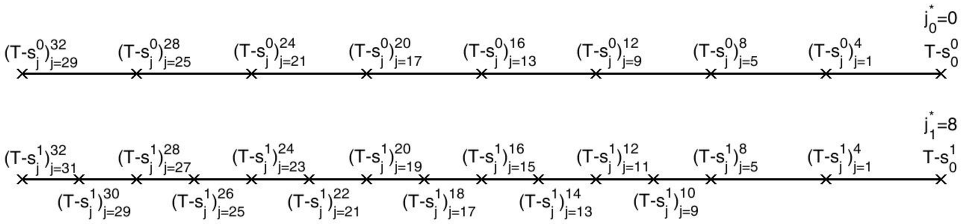

Figure 1: An example for (2.5) when $i = 0, 1$, $L = 5$ and $L' = 3$.

In (2.2), we simulate $M_1 + M_1'$ conditional realizations of X in order to keep those indexed from $m_1 = M_1 + 1$ to $m_1 = M_1 + M_1'$ for the approximation of regression matrices. Consequently, we made explicit the independence between trajectories used for the estimation of regression matrices and those used in the backward induction. To reduce the complexity of the algorithm and memory occupation, trajectories used for regression matrices can be simulated offline then erased from the memory. Given m_0, if an inner trajectory from $\{X^{m_0,m_1}\}_{m_1=1,\dots,M_1}$ is needed α times in the backward induction, we simulate α independent copies of it and use each copy once. This reduces further memory occupation as well as any superfluous dependence structure.

For each ordered couple $(j < k)$ of indices that take their values in $\{1, \dots, 2^L\}$, we introduce a stabilization operator

$$\mathcal{T}^{m_0}_{t_j,t_k,M_1'} : \mathbb{R}^{d_1} \ni x \mapsto {}^t\widetilde{\Gamma}^{m_0}_{t_j,t_k,M_1'} \left(x - X^{m_0}_{t_k}\right) \in \mathbb{R}^{d_1'} \ (d_1' \leq d_1), \tag{2.6}$$

that performs a linear combination of the components of $\left(x - X^{m_0}_{t_k}\right)$ using $\widetilde{\Gamma}^{m_0}_{t_j,t_k,M_1'}$ that contains some eigenvectors from $\Gamma^{m_0}_{t_j,t_k,M_1'}$ obtained with the eigenvalue decomposition

$$\Gamma^{m_0}_{t_j,t_k,M_1'} \Lambda^{m_0}_{t_j,t_k,M_1'} {}^t\Gamma^{m_0}_{t_j,t_k,M_1'} \tag{2.7}$$

of the regression matrix

$$\frac{1}{M_1'} \sum_{m_1=M_1+1}^{M_1+M_1'} \left(X^{m_0,m_1}_{t_j,t_k} - X^{m_0}_{t_k}\right) {}^t\left(X^{m_0,m_1}_{t_j,t_k} - X^{m_0}_{t_k}\right) \tag{2.8}$$

where ^{t} is the transposition operation.

Once factorization (2.7)=(2.8) is performed, we obtain the diagonal matrix $\Lambda^{m_0}_{t_j,t_k,M_1'} = \mathrm{diag}\left(\left\{\lambda^{m_0,l}_{t_j,t_k,M_1'}\right\}_{l=1,\dots,d_1}\right)$ of decreasing positive eigenvalues.

Then, we define $\widetilde{\Lambda}^{m_0}_{t_j,t_k,M'_1} = \mathrm{diag}\left(\left\{\lambda^{m_0,l}_{t_j,t_k,M'_1}\right\}_{l=1,...,d'_1}\right)$ as the truncation of $\Lambda^{m_0}_{t_j,t_k,M'_1}$ with d'_1 defined by

$$d'_1 = \min\left\{k \in \{1,..,d''_1\},\ \sum_{l=1}^{k} \lambda^{m_0,l}_{t_j,t_k,M'_1} \geq 95\% \sum_{l=1}^{d''_1} \lambda^{m_0,l}_{t_j,t_k,M'_1}\right\}, \quad (2.9)$$

where d''_1 keeps only eigenvalues that make the regression problem well-conditioned i.e. The ratio $\left.\dfrac{\lambda^{m_0,l}_{t_j,t_k,M'_1}}{\lambda^{m_0,1}_{t_j,t_k,M'_1}}\right|_{l=1,...,d''_1}$ has to be bigger than 10^{-6} in single precision or bigger than 10^{-15} in double precision floating representation [30]. In addition to ensuring a well-conditioned regression problem, equality (2.9) also performs a principal component analysis [30]. At the same time that we set the components of $\widetilde{\Lambda}^{m_0}_{t_j,t_k,M'_1}$, we define the matrix $\widetilde{\Gamma}^{m_0}_{t_j,t_k,M'_1}$ that contains only the eigenvectors in $\Gamma^{m_0}_{t_j,t_k,M'_1}$ that are associated to $\widetilde{\Lambda}^{m_0}_{t_j,t_k,M'_1}$.

Regressing with respect to ${}^t\widetilde{\Gamma}^{m_0}_{t_j,t_k,M'_1}\left(X^{m_0,m_1}_{t_j,t_k} - X^{m_0}_{t_k}\right) \in \mathbb{R}^{d'_1}$ ($d'_1 \leq d_1$), instead of $\left(X^{m_0,m_1}_{t_j,t_k} - X^{m_0}_{t_k}\right) \in \mathbb{R}^{d_1}$, involves the inversion of the diagonal matrix $\widetilde{\Lambda}^{m_0}_{t_j,t_k,M'_1}$ which replaces the whole regression matrix (2.8). Since $\widetilde{\Lambda}$ is bounded below away from zero, its inverse is bounded and the same for the regression procedure. This stabilizes the computation of the regression estimator whose expression is detailed in Section 2.3. Besides, since we have a large number of regression matrices, we can batch compute these inversions like explained in [2].

For $t_0 \leq s_j < s_k < T$ and conditionally to $X^{m_0}_{s_j}$, we want to keep only first/low order regression terms around $X^{m_0}_{s_k}$. We also want to reduce the bias induced by successive regressions as explained in Section 2.3. A natural way to do this is to make sure that the time distance $s_k - s_j$ is sufficiently small to neglect the higher order terms as well as to reduce bias propagation between s_k and s_j. For this purpose, we appropriately initialize the value L' ($> L/2$) as well as the couple $(\underline{s_j}, \overline{s_j})$ then at each iteration i we actualize the value taken by this couple according to (2.25) and (2.26).

When $i = 0$, for each $j = 0, ..., 2^L$, we define $\overline{s_j^{\mathcal{S}^0}}(s_j^0)$ as

$$\overline{s_j^{\mathcal{S}^0}}(s_j^0) = \max\left\{\begin{matrix} u \in \mathcal{S}^0 \cap]T - s_j^0, \overline{\delta(T - s_j^0)}];\ \text{(Bias Control) satisfied at} \\ T - s_j^0 \text{ and (2.11) fulfilled for all } s \in \mathcal{S}^0 \cap]T - s_j^0, u] \end{matrix}\right\} \quad (2.10)$$

with δ defined in (2.14), to simplify the understanding we can start assuming $\overline{\delta(T - s_j^0)} = T$. The inteval $]T - s_j^0, \overline{\delta(T - s_j^0)}]$ will be better specified at each definition 2.1, 3.1 and 3.2.

Then $J_j^0 = \mathcal{S}^0 \cap \left] s_j, \overline{s_j^{\mathcal{S}^0}}(s_j^0) \right]$ is a set of strictly decreasing time increments with the control (Bias Control), specified in definitions 2.1, 3.1 and 3.2, that also satisfy

$$\frac{1}{M_0} \sum_{m_0=1}^{M_0} \left(\sum_{l=1}^{d_1} \frac{1}{M_1'} \sum_{m_1=M_1+1}^{M_1+M_1'} \left(X_{s_j,s}^{m_0,m_1,l} - X_s^{m_0,l} \right)^2 \right) < \epsilon_{1,s_j} \tag{2.11}$$

for some tuning positive parameter ϵ_{1,s_j}. We consequently initialize

$$\overline{s_j^{\mathcal{S}^0}}(s_j^0) = \max(J_j^0),\ \underline{s_j^{\mathcal{S}^0}}(s_j^0) = \min(J_j^0) \text{ and } J_j^0 = \mathcal{S}^0 \cap \left] s_j, \overline{s_j^{\mathcal{S}^0}}(s_j^0) \right] \tag{2.12}$$

In what follows, if iteration index i is set and there is no confusion on the chosen set $\mathcal{S}^i$, we simplify notations and use $\overline{s_j}$ and $\underline{s_j}$ instead of $\overline{s_j^{\mathcal{S}^i}}(s_j^i)$ and $\underline{s_j^{\mathcal{S}^i}}(s_j^i)$.

By definition, J_j^0 contains different elements and we use $|J_j^0|$ to denote its cardinal. For any j such that $\overline{s_j} < T$, we choose the right values for L' to ensure that $2^{L-L'} < |J_j^0| \leq 2^{L'}$. Consequently, at the initialization step, one increases progressively L' till the latter condition is fulfilled. When $i > 0$ and $j = 0, ..., 2^L$, we define

$$J_j^i = \mathcal{S}^i \cap]s_j, \overline{s_j^{\mathcal{S}^i}}(s_j^i)]. \tag{2.13}$$

Given that $(s_j)_{j \in \{0,...,2^L\}}$ is a decreasing, and not strictly decreasing, sequence of coarse increments, we need to define on $\mathcal{S}^i$ a new operator $\delta^{\mathcal{S}^i}$ that associates to each $s \in \mathcal{S}^i$ the next increment in $\mathcal{S}^i$. For a fixed index $j \in \{1, ..., 2^L\}$, we define $\delta_{s_j}^{\mathcal{S}^i}(\cdot)$ on $(s_k)_{k \leq j}$, taken its values in $\mathcal{S}^i \cap [s_j, \overline{s_j}]$ $(= \{s_j\} \cup J_j^i)$, by

$$\delta_{s_j}^{\mathcal{S}^i}(s_k) = \min\left(\overline{s_j}, \min\{s \in \mathcal{S}^i; s_k < s \leq \overline{s_j}\}\right) \tag{2.14}$$

with $\min(\emptyset) = \infty$.

When there is no confusion on the chosen set $\mathcal{S}^i$, we use δ_{s_j} notation instead of $\delta_{s_j}^{\mathcal{S}^i}$. When $s_k < \overline{s_j}$, we use $\delta^{\mathcal{S}^i}$ notation instead of $\delta_{s_j}^{\mathcal{S}^i}$. When there is no confusion on the chosen set $\mathcal{S}^i$ and $s_k < \overline{s_j}$, we simplify both indices and use δ instead of $\delta_{s_j}^{\mathcal{S}^i}$.

This time operator will be largely used and for a given set $\mathcal{S}^i$ it has the following properties

- Pr1. (2.26) makes $\underline{s_j} = \delta_{s_j}(s_j) = \delta(s_j)$.
- Pr2. As long as $\max(s_{j_1}, s_{j_2}) \leq s_k < \min(\overline{s_{j_1}}, \overline{s_{j_2}})$, $\delta_{s_{j_1}}(s_k) = \delta_{s_{j_2}}(s_k) = \delta(s_k)$.
- Pr3. For fixed iteration step i, the nth composition of δ_{s_j} denoted $\delta_{s_j}^n(\cdot)$ is equal to $\overline{s_j}$ when $n \geq |J_j^i|$.

2.2 Fine and coarse approximations

Based on what was presented in Section 2.1, we detail here the simulation of approximations of U defined by (f). Considering the discretization sequence $(s_j)_{j=0,...,2^L}$ that takes its values in the set $\mathcal{S} \subset \{t_0, ..., t_{2^L}\}$, we use a learning procedure to associate to each scenario m_0 and each discretization set $\mathcal{S}$ a couple of function families $(\widetilde{h}^{m_0,\mathcal{S}}, \overline{h}^{m_0,\mathcal{S}})$.

Now, for given indices $k < j \in \{1, ..., 2^L\}$ that satisfy $s_j < s_k \leq \overline{s_j}$, for $x \in \mathbb{R}^{d_1}$ and $s \in \{s_k, s_k + \Delta_t, ..., \delta(s_k) - \Delta_t\}$, we define two approximation levels: A coarse approximation around $X^{m_0}_{s_k}$ conditionally on $X^{m_0}_{s_j}$ defined by

$$\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(x) = \ell\left[\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}\right] + {}^t\mathcal{T}^{m_0}_{s_j,s_k,M_1'}(x) H^{m_0,\mathcal{S}}_{s_j,s_k}, \tag{2.15}$$

and a fine approximation at $X^{m_0}_s$ defined by

$$\widetilde{h}^{m_0,\mathcal{S}}_{s,\overline{s_k}} = \frac{1}{M_1}\sum_{m_1=1}^{M_1}\left[\overline{h}^{m_0,\mathcal{S}}_{s_k,\delta(s_k)}(X^{m_0,m_1}_{s,\delta(s_k)}) + \sum_{t_{l+1}>s}^{\delta(s_k)} f(t_l, X^{m_0,m_1}_{s,t_l}, X^{m_0,m_1}_{s,t_{l+1}})\right]. \tag{2.16}$$

To complete this inductive interconnected backward definition of $\overline{h}$ and $\widetilde{h}$, we set the final coarse approximation to

$$\overline{h}^{m_0,\mathcal{S}}_{s_j,\overline{s_j}}(x) = \begin{cases} f(T, x) & \text{if } \overline{s_j} = T, \\ \overline{h}^{m_0,\mathcal{S}}_{\underline{s_j},\overline{s_j}}(x) = \overline{h}^{m_0,\mathcal{S}}_{\delta(s_j),\overline{s_j}}(x) & \text{if } \overline{s_j} < T, \end{cases} \tag{2.17}$$

where $\overline{s_j} > \underline{s_j} > s_j$ are specified during the initialization phase (cf. (2.12)) then actualized at each step (cf. (2.25) and (2.26)). $\overline{s_j}$ and $\underline{s_j}$ are really needed when T is sufficiently big or the variance produced by X is large enough. Otherwise, (2.17) can be replaced by $\overline{h}^{m_0,\mathcal{S}}_{s_j,T}(x) = f(t_{2^L}, x) = f(T, x)$.

$\mathcal{T}$ involved in (2.15) was already defined in (2.6). The value of the regression constant $\ell\left[\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}\right]$ depends on the variance adjustment procedure presented in section 2.3. However, the straight implementation can simply set $\ell\left[\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}\right] = \widetilde{h}^{m_0,\mathcal{S}}_{s_k,\overline{s_k}}$ for any couple (s_j, s_k) satisfying $s_j < s_k \leq \overline{s_j}$. Regarding the regression vector $H^{m_0,\mathcal{S}}_{s_j,s_k}$, its value is obtained from an estimation of the vector $a \in \mathbb{R}^{d_1'}$ that minimizes the quadratic error given by

$$\mathbb{E}\left[\mathbb{H}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k:\delta_{s_j}(s_k)}) - {}^t a\mathcal{T}^{m_0}_{s_j,s_k,M_1'}(X^{m_0,m_1}_{s_j,s_k})\right]^2 \tag{2.18}$$

with $X^{m_0,m_1}_{s_j,s_k:\delta_{s_j}(s_k)} = \left(X^{m_0,m_1}_{s_j,s_k}, X^{m_0,m_1}_{s_j,s_k+\Delta_t}, ..., X^{m_0,m_1}_{s_j,\delta_{s_j}(s_k)-\Delta_t}, X^{m_0,m_1}_{s_j,\delta_{s_j}(s_k)}\right)$ and

$$\mathbb{H}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k}(x) = \overline{h}^{m_0,\mathcal{S}}_{s_j,\delta_{s_j}(s_k)}\left(x_{\frac{\delta_{s_j}(s_k)-s_k}{\Delta_t}}\right) - \widetilde{h}^{m_0,\mathcal{S}}_{s_k,\overline{s_k}} + \sum_{l=1}^{\frac{\delta_{s_j}(s_k)-s_k}{\Delta_t}-1} f(t_{k_{s_k}+l}, x_l, x_{l+1}) \tag{2.19}$$

where $k_{s_k} = s_k/\Delta_t - 1$, $x = (x_1, ..., x_{(\delta_{s_j}(s_k)-s_k)/\Delta_t})$ with each coordinate of x belonging to $\mathbb{R}^{d_1}$.

When $s_k = \delta_{s_j}(s_j)$ and $\ell\left[\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}\right] = \widetilde{h}^{m_0,\mathcal{S}}_{s_k,\overline{s_k}}$, one can check the coherence of the previous definitions aimed to approximate U defined in (f). Indeed, (2.16) would provide for any $s \in \{s_j, s_j + \Delta_t, ..., \delta(s_j) - \Delta_t\} = \{s_j, s_j + \Delta_t, ..., s_k - \Delta_t\}$

$$\widetilde{h}^{m_0,\mathcal{S}}_{s,\overline{s_j}} = \frac{1}{M_1}\sum_{m_1=1}^{M_1}\left[\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s,s_k}) + \sum_{t_{l+1}>s}^{s_k} f(t_l, X^{m_0,m_1}_{s,t_l}, X^{m_0,m_1}_{s,t_{l+1}})\right] \tag{2.20}$$

where the term $\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}$, defined in (2.15), is obtained through the projection of $\displaystyle\sum_{t_{l+1}>s_k}^{\delta(s_k)} f(t_l, X^{m_0,m_1}_{s_j,t_l}, X^{m_0,m_1}_{s_j,t_{l+1}}) + \overline{h}^{m_0,\mathcal{S}}_{s_j,\delta(s_k)}(X^{m_0,m_1}_{s_j,\delta(s_k)})$, involved in (2.30), on $\mathcal{T}^{m_0}_{s_j,s_k,M_1'}(X^{m_0,m_1}_{s_j,s_k})$. In addition, if we had $\delta^2(s_j) = \delta(s_k) = \overline{s_j} = t_{2^L} = T$ then (2.17) would make $\overline{h}^{m_0,\mathcal{S}}_{s_j,\delta(s_k)}(\cdot) = f(t_{2^L}, \cdot)$ and as $s \in \{s_j, s_j+\Delta_t, ..., s_k-\Delta_t\}$ the definition of $\widetilde{h}^{m_0,\mathcal{S}}_{s,\overline{s_j}}$ would involve $\displaystyle\sum_{t_{l+1}>s}^{s_k} f(t_l, X^{m_0,m_1}_{s,t_l}, X^{m_0,m_1}_{s,t_{l+1}})$ plus the projection of $\displaystyle\sum_{t_{l+1}>s_k}^{t_{2^L}} f(t_l, X^{m_0,m_1}_{s_j,t_l}, X^{m_0,m_1}_{s_j,t_{l+1}}) + f(T, X^{m_0,m_1}_{s_j,T})$ as shown on Figure 2. $\widetilde{h}^{m_0,\mathcal{S}}_{s_k,\overline{s_k}}$ is equal to
$\displaystyle\frac{1}{M_1}\sum_{m_1=1}^{M_1}\left[\sum_{t_{l+1}>s_k}^{t_{2^L}} f(t_l, X^{m_0,m_1}_{s_k,t_l}, X^{m_0,m_1}_{s_k,t_{l+1}}) + f(T, X^{m_0,m_1}_{s_k,T})\right]$ because $\delta(s_k) = T$.

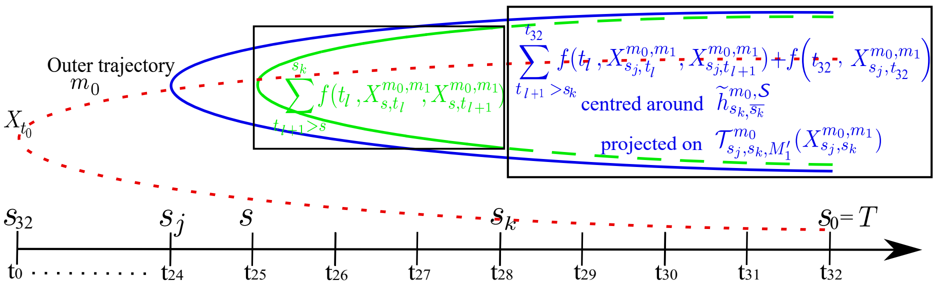

Figure 2: An example for (2.20) when $\delta^2(s_j) = \delta(s_k) = \overline{s_j} = t_{2^L} = T$, $L = 5$ and $L' = 3$.

According to equations (2.15), (2.16), (2.18) and (2.19), the functions $\overline{h}$ and $\widetilde{h}$ are defined backwardly. When $\widetilde{h}$ is a straight Monte Carlo involving $\overline{h}$, the latter is defined using a regression around a point at which we expressed $\widetilde{h}$. Consequently, $\overline{h}$ can be seen as a conditional first order Taylor expansion around the first layer of trajectories $(X_{t_k}^{m_0})_{k=1,...,2^L}^{m_0=1,...,M_0}$. The term of order zero in this expansion is played by $\ell[\cdot]$, where the term ${}^t\mathcal{T}_{s_j,s_k,M_1'}^{m_0}(x)H_{s_j,s_k}^{m_0,\mathcal{S}}$, deduced from the minimization of (2.18), plays the order one.

Remark 2.2. *1. Since we do not want to increase further the algorithm complexity by considering higher order terms, the definition of $\overline{h}$ involves only linear regression around $X_{s_k}^{m_0}$.*

2. *When the dimension d_1 is not too high, it is possible to regress the residual of the first regression on higher order terms. These successive regressions do not increase drastically the complexity when compared to the standard procedure. Nevertheless, as it separates regression with respect to first order terms and regression with respect to higher order terms, it loses orthogonality between first and higher order terms.*

3. *In case X is a martingale, the linearity simplifies further computations since, for instance, (2.16) can be replaced by*

$$\widetilde{h}_{s,\overline{s_k}}^{m_0,\mathcal{S}} = \overline{h}_{s_k,\delta(s_k)}^{m_0,\mathcal{S}}(X_s^{m_0}) + \frac{1}{M_1}\sum_{m_1=1}^{M_1}\sum_{t_{l+1}>s}^{\delta(s_k)} f(t_l, X_{s,t_l}^{m_0,m_1}, X_{s,t_{l+1}}^{m_0,m_1}).$$

Definition 2.1. *For $i^* = \min(\min\{i=1,...,L-L', j_i^* = 2^L\}, L-L')$*

- *For $k < j \in \{1,...,2^L\}$ that satisfy $s_j < s_k \leq \overline{s_j} < t_{2^L} = T$, 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 $\overline{h}_{s_j,s_k}^{m_0,\mathcal{S}^{i^*}}(X_{s_j,s_k}^{m_0,m_1})$ where $\overline{h}$ is given in (2.15) and (2.17).*
- *For $k \in \{1,...,2^L\}$ and $s \in \{s_k, s_k+\Delta_t, ..., \delta(s_k)-\Delta_t\} - \{0\}$, the simulation $U_s^{m_0}$ of U at $X_s^{m_0}$ is set to be equal to $\widetilde{h}_{s,\overline{s_k}}^{m_0,\mathcal{S}^{i^*}}$ with $\widetilde{h}$ expressed in (2.16).*
- *The average U_0^{lear} of learned values on U_0 is equal to*

$$U_0^{lear} = \frac{1}{M_0}\sum_{m_0=1}^{M_0}\widetilde{h}_{0,\overline{0}}^{m_0,\mathcal{S}^{i^*}} \tag{2.21}$$

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

$$U_0^{sim} = \frac{1}{M_0}\sum_{m_0=1}^{M_0}\left[\widetilde{h}_{\delta(0),\overline{\delta(0)}}^{m_0,\mathcal{S}^{i^*}} + \sum_{t_{l+1}>0}^{\delta(0)} f(t_l, X_{t_l}^{m_0}, X_{t_{l+1}}^{m_0})\right] \tag{2.22}$$

with $\widetilde{h}$ expressed in (2.16).

- Introduced in (2.10), (Bias Control) associated to (f) is defined at $s \in \mathcal{S}^0$ for $u \in \mathcal{S}^0 \cap]s, \overline{\delta(s)}]$ by

$$\left| \frac{1}{M_0} \sum_{m_0=1}^{M_0} \left(\widetilde{h}_{s,u}^{m_0,\mathcal{S}^0} - \widetilde{h}_{\delta(s),\overline{\delta(s)}}^{m_0,\mathcal{S}^0} - \sum_{t_{l+1}>s}^{\delta(s)} f(t_l, X_{t_l}^{m_0}, X_{t_{l+1}}^{m_0}) \right) \right| < \epsilon_{2,s}^{\mathcal{S}^0}$$

where for each set $\mathcal{S}$, $\{\epsilon_{2,s}^{\mathcal{S}}\}_{s\in\mathcal{S}}$ is a family of positive bias tuning parameters.

- For $k \in \{j_i^* + 1, ..., 2^L\}$, setting $s_k = T - s_k^i$ and noticing that $\delta^{\mathcal{S}^i}(s_k) = \delta^{\widehat{\mathcal{S}}^i}(\delta^{\widehat{\mathcal{S}}^i}(s_k))$, $e_{s_k}^{\mathcal{S}^i}$ and $\varepsilon_{s_k}^{\mathcal{S}^i}$ are given by

$$e_{s_k}^{\mathcal{S}^i} = \frac{1}{M_0 M_1} \sum_{m_0=1}^{M_0} \sum_{m_1=1}^{M_1} \left[\overline{h}_{\delta^{\widehat{\mathcal{S}}^i}(s_k),\delta^{\mathcal{S}^i}(s_k)}^{m_0,\widehat{\mathcal{S}}^i} (X_{\delta^{\widehat{\mathcal{S}}^i}(s_k),\delta^{\mathcal{S}^i}(s_k)}^{m_0,m_1}) - \overline{h}_{s_k,\delta^{\mathcal{S}^i}(s_k)}^{m_0,\mathcal{S}^i} (X_{\delta^{\widehat{\mathcal{S}}^i}(s_k),\delta^{\mathcal{S}^i}(s_k)}^{m_0,m_1}) \right],$$

$$\varepsilon_{s_k}^{\mathcal{S}^i} = \sum_{s\in\mathcal{S}^i, s>s_k} \epsilon_{2,s}^{\mathcal{S}^i}.$$

Remark 2.3. *1.* U^{m_0,m_1} can be seen as the inner or second layer approximation of U and U^{m_0} can be seen as the outer or first layer approximation of U.

2. When U^{m_0,m_1} is only defined on $\mathcal{S}^{i^*}$, it is remarkable to see that U^{m_0} is defined on the whole fine discretization set $\{t_0, ..., t_{2^L}\}$.

3. For any $\mathcal{S}$, it is natural to have $\epsilon_{2,s}^{\mathcal{S}}$ proportional to the value of the estimation $\frac{1}{M_0}\sum_{m_0=1}^{M_0} \widetilde{h}_{s,\overline{s}}^{m_0,\mathcal{S}}$. Used to control the bias, the choice of $\epsilon_{2,s}^{\mathcal{S}}$ has also to take into account the confidence interval of the estimator of the left side of inequality (Bias Control).

4. Although (Bias Control) is quite sufficient to have almost unbiased estimates, Section 2.3 introduces a more stringent local bias control.

5. $e^{\mathcal{S}^i}$ is defined as the average difference between the estimation $\overline{h}^{m_0,\mathcal{S}^i}$ that involves the discretization set $\mathcal{S}^i$ and the estimation $\overline{h}^{m_0,\widehat{\mathcal{S}}^i}$ that involves a finer discretization set $\widehat{\mathcal{S}}^i$ defined below (2.4). With actualization (2.5), we are basically saying that the discretization set should be finer only when the difference between approximations is superior to the sum of possible accumulation of bias $\varepsilon^{\mathcal{S}^i}$.

2.3 Regression computations: Bias control and variance adjustment

As a continuation to Section 2.1, we explain the (Bias Control) expression and how the value of $(\underline{s_j}, \overline{s_j})$ should be actualized. Then, as a continuation to Section 2.2, for each couple (scenario/discretization set) $= (m_0, \mathcal{S})$ we provide possible values of the couple $\left(\ell\left[\overline{h}^{m_0,\mathcal{S}}\right], H^{m_0,\mathcal{S}}\right)$ including a variance adjustment procedure. We remind that both procedures, explained in this section, are only feasible because of the nested nature of our simulation and they would not be possible otherwise.

In Section 2.1 equation (2.12), we defined $(\underline{s}, \overline{s})$ on the discretization set $\mathcal{S}^0$. In order to reduce the backward bias propagation, this definition used the double layer Monte Carlo to control the average bias. Indeed, as $\frac{1}{M_0}\sum_{m_0=1}^{M_0} \widetilde{h}^{m_0,\mathcal{S}^0}_{s_j,u}$ and $\frac{1}{M_0}\sum_{m_0=1}^{M_0}\left(\widetilde{h}^{m_0,\mathcal{S}^0}_{\delta(s_j),\overline{\delta(s_j)}} + \sum_{t_{l+1}>s_j}^{\delta(s_j)} f(t_l, X^{m_0}_{t_l}, X^{m_0}_{t_{l+1}})\right)$ are both approximations of $\mathbb{E}(U_{s_j}) = \mathbb{E}\left(U_{\delta(s_j)} + \sum_{t_{l+1}>s_j}^{\delta(s_j)} f(t_l, X^{m_0}_{t_l}, X^{m_0}_{t_{l+1}})\right)$, it is natural to have them almost equal. For large values of M_0, the difference between these approximations is due to bias. As explained at the end of Section 4.1 and the beginning of Section 4.2, a judicious method to reduce this bias propagation is to adjust the number of successive regressions through the appropriate choice of u.

The choice of u in (Bias Control) ought to decrease the global average value of the bias. More local approach can be developed using equality

$$\mathbb{E}\Big(U_{s_j} 1_{\{U_{s_j}\in[a,b]\}}\Big) = \mathbb{E}\left(1_{\{U_{s_j}\in[a,b]\}}\left[U_{\delta(s_j)} + \sum_{t_{l+1}>s_j}^{\delta(s_j)} f(t_l, X^{m_0}_{t_l}, X^{m_0}_{t_{l+1}})\right]\right) \tag{2.23}$$

which is true for any localizing interval $[a,b]$. When M_0 is sufficiently large, one can sort $\{\widetilde{h}^{m_0,\mathcal{S}^0}_{s_j,\overline{s_j}}\}_{m_0\le M_0}$ and define a subdivision of localizing intervals $\{[a_q, a_{q+1}]\}_{q\ge 1}$ then choose $\overline{s_j}$ that does not induce a large difference between $\frac{1}{M_0}\sum_{m_0=1}^{M_0}\left(1_{\left\{\widetilde{h}^{m_0,\mathcal{S}^0}_{s_j,\overline{s_j}}\in[a_q,a_{q+1}]\right\}}\left[\widetilde{h}^{m_0,\mathcal{S}^0}_{\delta(s_j),\overline{\delta(s_j)}} + \sum_{t_{l+1}>s_j}^{\delta(s_j)} f(t_l, X^{m_0}_{t_l}, X^{m_0}_{t_{l+1}})\right]\right)$ and $\frac{1}{M_0}\sum_{m_0=1}^{M_0}\left(\widetilde{h}^{m_0,\mathcal{S}^0}_{s_j,\overline{s_j}} 1_{\left\{\widetilde{h}^{m_0,\mathcal{S}^0}_{s_j,\overline{s_j}}\in[a_q,a_{q+1}]\right\}}\right)$ for any q. This local increase of bias can be even tracked for any $s \in \mathcal{S}_0 \cap [\delta(s_j), \overline{s_j}[$ using the difference

$$\frac{1}{M_0}\sum_{m_0=1}^{M_0} 1_{\left\{\widetilde{h}^{m_0,\mathcal{S}^0}_{s_j,\overline{s_j}}\in[a_q,a_{q+1}]\right\}}\left(\widetilde{h}^{m_0,\mathcal{S}^0}_{s_j,\overline{s_j}} - \widetilde{h}^{m_0,\mathcal{S}^0}_{s,\overline{s}} - \sum_{t_{l+1}>s_j}^{s} f(t_l, X^{m_0}_{t_l}, X^{m_0}_{t_{l+1}})\right). \tag{2.24}$$

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.

For $j = 0, ..., 2^L$ and $s_j = T - s_j^i \in \mathcal{S}^i$, the actualization of $(\underline{s_j}, \overline{s_j})$ is given by

$$\overline{s_j^{\mathcal{S}^i}}(s_j^i) = \overline{s_j^{\mathcal{S}^{i-1}}}(s_j^{i-1}) 1_{\overline{I_{i,j}}} + \max\left(\mathcal{S}^i \cap]s_j, \overline{s_j^{\mathcal{S}^{i-1}}}(s_j^{i-1})[\right) 1_{\overline{I_{i,j}}^c}, \quad (2.25)$$

$$\underline{s_j^{\mathcal{S}^i}}(s_j^i) = \underline{s_j^{\mathcal{S}^{i-1}}}(s_j^{i-1}) 1_{\underline{I_{i,j}}} + \min\left(\mathcal{S}^i \cap]s_j, \overline{s_j^{\mathcal{S}^{i-1}}}(s_j^{i-1})[\right) 1_{\underline{I_{i,j}}^c} \quad (2.26)$$

where the sets of indices $\underline{I_{i,j}} = \{j \leq j_i^*\} \cup \{s_j^i \neq s_j^{i-1}\}$ and $\overline{I_{i,j}} = \underline{I_{i,j}} \cup \{\overline{s_j^{\mathcal{S}^{i-1}}}(s_j^{i-1}) = T\}$. In Figure 3, we illustrate what happens when $j > j_1^*$ with either $s_j^1 \neq s_j^0$ $(j = 25, 26)$ or $s_j^1 = s_j^0$ $(j = 27, 28)$. Except when $\{\overline{s_j^{\mathcal{S}^{i-1}}}(s_j^{i-1}) = T\}$, the actualization strategy given by equations (2.25) and (2.26) aims at ensuring $\overline{s_{j_1}^{\mathcal{S}^i}}(s_{j_1}^i) \neq \overline{s_{j_2}^{\mathcal{S}^i}}(s_{j_2}^i)$ and $\underline{s_{j_1}^{\mathcal{S}^i}}(s_{j_1}^i) \neq \underline{s_{j_2}^{\mathcal{S}^i}}(s_{j_2}^i)$ as long as $s_{j_1}^i \neq s_{j_2}^i$. As mentioned before, if iteration index i is set and there is no confusion on the chosen set $\mathcal{S}^i$, we simplify notations and use $\overline{s_j}$ and $\underline{s_j}$ instead of $\overline{s_j^{\mathcal{S}^i}}(s_j^i)$ and $\underline{s_j^{\mathcal{S}^i}}(s_j^i)$.

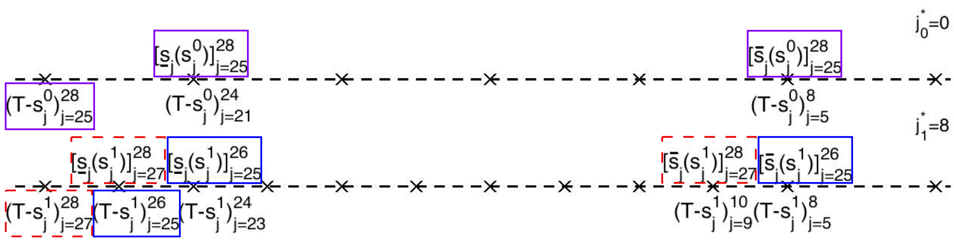

Figure 3: An example for (2.12), (2.25) and (2.26) based on the example of Figure 1.

Given two indices $k < j \in \{1, ..., 2^L\}$ that satisfy $s_j < s_k \leq \overline{s_j}$, the expression of $(\ell\left[\overline{h}_{s_j,s_k}^{m_0,\mathcal{S}}\right], H_{s_j,s_k}^{m_0,\mathcal{S}})$ involves the use of an intermediary variable $\gamma_{s_j,s_k}^{m_0,\mathcal{S}}$ and an intermediary vector $\widehat{H}_{s_j,s_k}^{m_0,\mathcal{S}}$. Given the value of the couple $(\gamma_{s_j,s_k}^{m_0,\mathcal{S}}, \widehat{H}_{s_j,s_k}^{m_0,\mathcal{S}})$ specified in (2.29) and (2.32), we define

$$H_{s_j,s_k}^{m_0,\mathcal{S}} = \gamma_{s_j,s_k}^{m_0,\mathcal{S}} \widehat{H}_{s_j,s_k}^{m_0,\mathcal{S}} \quad (2.27)$$

and

$$\ell\left[\overline{h}_{s_j,s_k}^{m_0,\mathcal{S}}\right] = \widetilde{h}_{s_k,\overline{s_k}}^{m_0,\mathcal{S}} + \frac{(1 - \gamma_{s_j,s_k}^{m_0,\mathcal{S}})}{M_1} \sum_{m_1=1}^{M_1} {}^t\mathcal{T}_{s_j,s_k,M_1'}^{m_0}(X_{s_j,s_k}^{m_0,m_1}) \widehat{H}_{s_j,s_k}^{m_0,\mathcal{S}}. \quad (2.28)$$

Then, $\gamma^{m_0,\mathcal{S}}_{s_j,s_k}$ is used to adjust the variance of $\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}$ defined in (2.15) without changing its average value. Indeed, the expression of $\ell\Big[\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}\Big]$ makes $\sum_{m_1=1}^{M_1}\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k})/M_1$ invariable with respect to $\gamma^{m_0,\mathcal{S}}_{s_j,s_k}$.

The value of $\widehat{H}^{m_0,\mathcal{S}}_{s_j,s_k}$ is given by

$$\widehat{H}^{m_0,\mathcal{S}}_{s_j,s_k} = (\widetilde{\Lambda}^{m_0}_{s_j,s_k,M_1'})^{-1}\frac{1}{M_1}\sum_{m_1=1}^{M_1}\mathcal{H}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k,M_1'}(X^{m_0,m_1}_{s_j,s_k:\delta_{s_j}(s_k)}) \tag{2.29}$$

where $X^{m_0,m_1}_{s_j,s_k:\delta_{s_j}(s_k)} = \left(X^{m_0,m_1}_{s_j,s_k}, X^{m_0,m_1}_{s_j,s_k+\Delta_t}, ..., X^{m_0,m_1}_{s_j,\delta_{s_j}(s_k)-\Delta_t}, X^{m_0,m_1}_{s_j,\delta_{s_j}(s_k)}\right)$ and the function $\mathcal{H}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k,M_1'} : \Omega\times\mathbb{R}^{d_1(\delta_{s_j}(s_k)-s_k)/\Delta_t} \ni (\omega, x_1, ..., x_{(\delta_{s_j}(s_k)-s_k)/\Delta_t}) \to \Omega\times\mathbb{R}^{d_1'}$ is $\mathcal{F}_{\delta_{s_j}(s_k)}\otimes\mathcal{B}(\mathbb{R}^{d_1(\delta_{s_j}(s_k)-s_k)/\Delta_t})$-measurable and defined by

$$\mathcal{H}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k,M_1'}(x) = \mathcal{T}^{m_0}_{s_j,s_k,M_1'}(x_1)\underbrace{\begin{bmatrix}\overline{h}^{m_0,\mathcal{S}}_{s_j,\delta_{s_j}(s_k)}\left(x_{\frac{\delta_{s_j}(s_k)-s_k}{\Delta_t}}\right) - \widetilde{h}^{m_0,\mathcal{S}}_{s_k,\delta_{s_k}(s_k)} \\ + \displaystyle\sum_{l=1}^{\frac{\delta_{s_j}(s_k)-s_k}{\Delta_t}-1} f(t_{k_{s_k}+l}, x_l, x_{l+1})\end{bmatrix}}_{\mathbb{H}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k}(x)} \tag{2.30}$$

where $k_{s_k} = s_k/\Delta_t - 1$ and $x = (x_1, ..., x_{(\delta_{s_j}(s_k)-s_k)/\Delta_t})$.

Regarding $\gamma^{m_0,\mathcal{S}}_{s_j,s_k}$, various values can be considered. The straight choice is to take $\gamma^{m_0,\mathcal{S}}_{s_j,s_k} = 1$ which reduces the procedure to a standard regression. 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 with respect to χ_2. Because generally regression preserves the mean value, it is reasonable to assume $\mathbb{E}(\chi_3) = \mathbb{E}(\chi_1)$. However, regressions decreases the second moment i.e. $\mathbb{E}(\chi_3^2) < \mathbb{E}(\chi_1^2)$ and thus $Var(\chi_3) < Var(\chi_1)$. The latter fact becomes a real problem for tail distribution when $Var(\chi_3) << Var(\mathbb{E}(\chi_1|\chi_2))$. Some contributions tackle rare event simulation using a change of probability trick [8, 14] and more recent contribution [3] implements reversible shaking transformations.

In (Bias Control), we established strong constraints to make $\widetilde{h}^{m_0,\mathcal{S}}$ an almost unbiased estimator of U. 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

$$\mathbb{E}\left(\mathrm{Var}_{s_j}(U_s)\right) = \mathbb{E}\left(\mathbb{E}_{s_j}\left(\left[U_s - \mathbb{E}_{s_j}(U_s)\right]^2\right)\right) = \mathbb{E}\left(\left[U_s - \mathbb{E}_{s_j}(U_s)\right]^2\right).$$

Defining $(\sigma^{\mathcal{S}}_{0,s_j,s})^2 = \frac{1}{M_0}\sum_{m_0=1}^{M_0}\left[\widetilde{h}^{m_0,\mathcal{S}}_{s,\overline{s}} - \frac{1}{M_1}\sum_{m_1=1}^{M_1}\overline{h}^{m_0,\mathcal{S}}_{s_j,s}(X^{m_0,m_1}_{s_j,s})\right]^2$ and $(\sigma^{m_0,\mathcal{S}}_{s_j,s})^2 = \frac{1}{M_1}\sum_{m_1=1}^{M_1}\left[\overline{h}^{m_0,\mathcal{S}}_{s_j,s}(X^{m_0,m_1}_{s_j,s}) - \frac{1}{M_1}\sum_{m_1=1}^{M_1}\overline{h}^{m_0,\mathcal{S}}_{s_j,s}(X^{m_0,m_1}_{s_j,s})\right]^2$ as the estimators of $\mathbb{E}_{s_j}\left(\left[U_s - \mathbb{E}_{s_j}(U_s)\right]^2\right)$ and $\mathbb{E}\left(\left[U_s - \mathbb{E}_{s_j}(U_s)\right]^2\right)$ respectively, it is then natural to have for $s = s_k, \delta(s_k)$ as M_1 and $M_0 \to \infty$

$$(\sigma^{\mathcal{S}}_{0,s_j,s})^2 = \frac{1}{M_0}\sum_{m_0=1}^{M_0}(\sigma^{m_0,\mathcal{S}}_{s_j,s})^2. \tag{2.31}$$

Because of (Bias Control), the estimators $\frac{1}{M_1}\sum_{m_1=1}^{M_1}\overline{h}^{m_0,\mathcal{S}}_{s_j,s}(X^{m_0,m_1}_{s_j,s})$ and $\widetilde{h}^{m_0,\mathcal{S}}_{s,\overline{s}}$ have negligible bias. Starting from $\delta(s_k) = \overline{s_k}$, we can reasonably assume inductively that (2.31) is true for $s = \delta(s_k)$. Afterwards, we choose the appropriate value of $\gamma^{m_0,\mathcal{S}}_{s_j,s_k}$, subsequently the value of $\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k})$, that makes $\sigma^{m_0,\mathcal{S}}_{s_j,s_k}$ satisfy (2.31) for $s = s_k$. For this task, we introduce an intermediary non-adjusted conditional variance $(\widehat{\sigma}^{m_0,\mathcal{S}}_{s_j,s})^2$ defined by

$$(\widehat{\sigma}^{m_0,\mathcal{S}}_{s_j,s_k})^2 = \frac{1}{M_1}\sum_{m_1=1}^{M_1}\left[\widetilde{h}^{m_0,\mathcal{S}}_{s_k,\overline{s_k}} + {}^t\mathcal{T}^{m_0}_{s_j,s_k,M_1'}(X^{m_0,m_1}_{s_j,s_k})\widehat{H}^{m_0,\mathcal{S}}_{s_j,s_k} - \sum_{m_1=1}^{M_1}\frac{\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k})}{M_1}\right]^2.$$

$\sum_{m_1=1}^{M_1}\frac{\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k})}{M_1}$ can be replaced by $\sum_{m_1=1}^{M_1}\frac{\widetilde{h}^{m_0,\mathcal{S}}_{s_k,\overline{s_k}} + {}^t\mathcal{T}^{m_0}_{s_j,s_k,M_1'}(X^{m_0,m_1}_{s_j,s_k})\widehat{H}^{m_0,\mathcal{S}}_{s_j,s_k}}{M_1}$ without changing the value of $(\widehat{\sigma}^{m_0,\mathcal{S}}_{s_j,s})^2$. For positive tuning value $\epsilon_3 < 1/3$, we set then

$$\gamma^{m_0,\mathcal{S}}_{s_j,s_k} = \frac{\sigma^{m_0,\mathcal{S}}_{s_j,\delta(s_k)}}{\widehat{\sigma}^{m_0,\mathcal{S}}_{s_j,s_k}}\left(\sqrt{\frac{s_k - s_j}{\delta(s_k) - s_j}}1_{\delta(s_k)-s_j<\epsilon_3} + \frac{\sigma^{\mathcal{S}}_{0,s_j,s_k}}{\sigma^{\mathcal{S}}_{0,s_j,\delta(s_k)}}1_{\delta(s_k)-s_j\geq\epsilon_3}\right). \tag{2.32}$$

According to (2.32), when $\delta(s_k) - s_j$ is small and a fortiori $s_k - s_j$ is small then the conditional variance $(\sigma^{m_0,\mathcal{S}}_{s_j,s_k})^2$ is linear with respect to time increment $s_k - s_j$. This fact can be justified for diffusions using first order Taylor expansion of $E(\phi(t, W_t))$ around $\phi(t, 0)$, where W is a Brownian motion. Also according to (2.32), when $\delta(s_k) - s_j$ becomes sufficiently big, the conditional variance $(\sigma^{m_0,\mathcal{S}}_{s_j,s_k})^2$ has the same unconditional decreasing ratio $\left(\frac{\sigma^{\mathcal{S}}_{0,s_j,s_k}}{\sigma^{\mathcal{S}}_{0,s_j,\delta(s_k)}}\right)^2$ with respect to $(\sigma^{m_0,\mathcal{S}}_{s_j,\delta(s_k)})^2$. Although this adjustment works well in our simulations, it can be turn into a more local approach. In fact, similar to what was proposed for the bias control in (2.23), for $s_j < s$ with $s = s_k, \delta(s_k)$, the equality

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

is true for any localizing interval $[a, b]$. When M_0 is sufficiently large, one can sort $\{(\sigma^{m_0,\mathcal{S}}_{s_j,\delta(s_k)})^2\}_{m_0 \le M_0}$ and define a subdivision of localizing intervals $\{[a_q, a_{q+1}]\}_{q\ge 1}$ and define

$$(\sigma^{\mathcal{S},q}_{0,s_j,s})^2 = \frac{1}{M_0}\sum_{m_0=1}^{M_0} 1_{\{(\sigma^{m_0,\mathcal{S}}_{s_j,\delta(s_k)})^2 \in [a_q,a_{q+1}]\}}\left[\widetilde{h}^{m_0,\mathcal{S}}_{s,\overline{s}} - \frac{1}{M_1}\sum_{m_1=1}^{M_1}\overline{h}^{m_0,\mathcal{S}}_{s_j,s}(X^{m_0,m_1}_{s_j,s})\right]^2.$$

Condition (2.31) can be then replaced by its localized version

$$(\sigma^{\mathcal{S},q}_{0,s_j,s})^2 = \frac{1}{M_0}\sum_{m_0=1}^{M_0} 1_{\{(\sigma^{m_0,\mathcal{S}}_{s_j,\delta(s_k)})^2 \in [a_q,a_{q+1}]\}}(\sigma^{m_0,\mathcal{S}}_{s_j,s})^2. \tag{2.33}$$

If $\sigma^{m_0,\mathcal{S}}_{s_j,\delta(s_k)} \in [a_{q_0}, a_{q_0+1}]$ then it makes sense to replace (2.32) by

$$\gamma^{m_0,\mathcal{S}}_{s_j,s_k} = \frac{\sigma^{m_0,\mathcal{S}}_{s_j,\delta(s_k)}}{\widehat{\sigma}^{m_0,\mathcal{S}}_{s_j,s_k}}\left(\sqrt{\frac{s_k - s_j}{\delta(s_k) - s_j}}1_{\delta(s_k)-s_j<\epsilon_3} + \frac{\sigma^{\mathcal{S},q_0}_{0,s_j,s_k}}{\sigma^{\mathcal{S},q_0}_{0,s_j,\delta(s_k)}}1_{\delta(s_k)-s_j\ge\epsilon_3}\right). \tag{2.34}$$

Although the local variance adjustment (2.34) was not necessary in our simulations, 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 the bias of the first layer fine estimator U^{m_0} and the variance of the second layer coarse estimator U^{m_0,m_1}.

3 Some applications: Risk measures, BSDEs and RBSDEs

The simulation procedure presented in the previous section is supposed to be used for any functional approximated by or solution of (f). In this section, we show the use of this procedure on standard problems that inspired this work. We first clarify the method on the approximation of a conditional expectation of some $\mathcal{F}_T$-measurable random variable and how to compute a risk measure. We also illustrate the adaptation to BSDEs then to RBSDEs.

3.1 Conditional expectation and risk measures

We consider here the following process

$$U_t = E\left(f(X_T)\middle| X_t\right),$$

with a deterministic function f. Thus, we assume that there is no path dependence through the sum on the realizations of X as done in (f). In this path-independent situation for the fixed time set (2.3), it is clear that one can simulate $\{U_{T-s_j^0}\}_{j=0,...,2^L}$ using 1NMC without any need of regression and thus without using our method. However, we choose to illustrate our method on this simple case and we will see at the end of this section what are the benefits. To simplify further the presentation, we set the variance adjustment parameter γ, introduced in Section 2.3, to 1.

For known values $s_{j'} < s_j \in \{\Delta_t, ..., T\}$ and for a fixed outer trajectory $(X_{t_k}^{m_0})_{k=0,...,2^L}$, let us assume that we want to simulate $U_{s_{j'}}$ and U_{s_j}. A straight way to do it is to draw inner trajectories of X, as in Figure 4, then average on the realizations of $f(X_T)$. If s_j and $s_{j'}$ are close to each other in some sense[1], we are able to simulate U_{t_k} for any $t_k \in [s_{j'}, s_j)$ using

$$U_{t_k}^{m_0} = \widetilde{h}_{t_k,T}^{m_0,\mathcal{S}^{i^*}} = \frac{1}{M_1}\sum_{m_1=1}^{M_1} \overline{h}_{s_{j'},s_j}^{m_0,\mathcal{S}^{i^*}}(X_{t_k,s_j}^{m_0,m_1}).$$

We point out that $\overline{h}_{s_{j'},s_j}^{m_0,\mathcal{S}^{i^*}}(x)$ and $X_{t_k,s_j}^{m_0,m_1}$ replace respectively $f(x)$ and $X_{t_k,T}^{m_0,m_1}$ involved in standard Nested simulation. Below, we establish how $\overline{h}_{s_{j'},s_j}$ should be computed.

First of all, since $s_{j'}$ and s_j are assumed to be "close enough", the initialization phase presented in the end of Section 2 and the actualization of $\overline{s_{j'}}$, $\underline{s_{j'}}$, $\overline{s_j}$ and $\underline{s_j}$ are not necessary. Thus, in the light of (2.11), one has to take $\overline{s_j} = \overline{s_{j'}} = \overline{t_{2^L}} = T$ and consequently

$$(\overline{h}_{s_{j'}T}) \; \& \; (\overline{h}_{s_jT}) \qquad \overline{h}_{s_{j'},T}^{m_0,\mathcal{S}^{i^*}}(x) = \overline{h}_{s_j,T}^{m_0,\mathcal{S}^{i^*}}(x) = f(x)$$

that sets

$$(\widetilde{h}_{s_j}) \qquad U_{s_j}^{m_0} = \widetilde{h}_{s_j,T}^{m_0,\mathcal{S}^{i^*}} = \frac{1}{M_1}\sum_{m_1=1}^{M_1} \overline{h}_{s_j,T}^{m_0,\mathcal{S}^{i^*}}(X_{s_j,T}^{m_0,m_1}).$$

As in (2.15), we define

$$(\overline{h}_{s_{j'}s_j}) \qquad \overline{h}_{s_{j'},s_j}^{m_0,\mathcal{S}^{i^*}}(x) = \widetilde{h}_{s_j,T}^{m_0,\mathcal{S}^{i^*}} + {}^t\mathcal{T}_{s_{j'},s_j,M_1'}^{m_0}(x) A_{s_{j'},s_j}^{m_0,\mathcal{S}^{i^*}},$$

where the adaptation of (2.27) and (2.30) makes

$$(A_{s_{j'}s_j}^T) \; A_{s_{j'},s_j}^{m_0,\mathcal{S}^{i^*}} = \frac{\left(\widetilde{\Lambda}_{s_{j'},s_j,M_1'}^{m_0}\right)^{-1}}{M_1}\sum_{m_1=1}^{M_1} \mathcal{T}_{s_{j'},s_j,M_1'}^{m_0}(X_{s_{j'},s_j}^{m_0,m_1}) \begin{bmatrix} \overline{h}_{s_{j'},T}^{m_0,\mathcal{S}^{i^*}}\left(X_{s_{j'},T}^{m_0,m_1}\right) \\ -\widetilde{h}_{s_j,T}^{m_0,\mathcal{S}^{i^*}} \end{bmatrix}.$$

[1]Not necessary an Euclidean distance.

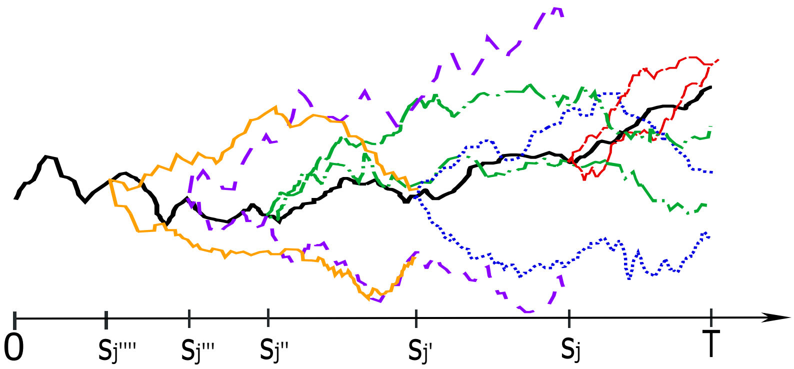

Figure 4: Given the realization of one outer trajectory (bold), we simulate inner trajectories to approximate U_{s_j}, $U_{s_{j'}}$, $U_{s_{j''}}$, $U_{s_{j'''}}$ and $U_{s_{j''''}}$.

If we add a third increment $s_{j''}$ (cf. Figure 4) such that $s_{j''}$, $s_{j'}$ and s_j are close enough, for any $t_k \in [s_{j''}, s_{j'})$ one can set $U_{t_k}^{m_0} = \widetilde{h}_{t_k,T}^{m_0,\mathcal{S}^{i^*}}$. The latter equality requires the definition of $\overline{h}_{s_{j''},s_{j'}}^{m_0,\mathcal{S}^{i^*}}$ which can be obtained from $(\overline{h}_{s_{j''}s_{j'}})$ involving $\widetilde{h}_{s_{j'},T}^{m_0,\mathcal{S}^{i^*}}$ and $A_{s_{j''},s_{j'}}^{m_0,\mathcal{S}^{i^*}}$ that can be computed using $(A_{s_{j''}s_{j'}}^{s_j})$. The calculations in $(A_{s_{j''}s_{j'}}^{s_j})$ use $\widetilde{h}_{s_{j'},T}^{m_0,\mathcal{S}^{i^*}}$ and $\overline{h}_{s_{j''},s_j}^{m_0,\mathcal{S}^{i^*}}$ whose expression depends on $\widetilde{h}_{s_j,T}^{m_0,\mathcal{S}^{i^*}}$ and $A_{s_{j''},s_j}^{m_0,\mathcal{S}^{i^*}}$. Finally, $A_{s_{j''},s_j}^{m_0,\mathcal{S}^{i^*}}$ is the regression vector of $\overline{h}_{s_{j''},T}^{m_0,\mathcal{S}^{i^*}}$ around $\widetilde{h}_{s_j,T}^{m_0,\mathcal{S}^{i^*}}$. Subsequently, the computations of $\widetilde{h}_{s_{j''},T}^{m_0,\mathcal{S}^{i^*}}$, $\widetilde{h}_{s_{j'},T}^{m_0,\mathcal{S}^{i^*}}$ and $\widetilde{h}_{s_j,T}^{m_0,\mathcal{S}^{i^*}}$ involve the dependence structure given in (3.1).

$$\begin{tikzcd}
\widetilde{h}_{s_{j''},T}^{m_0,\mathcal{S}^{i^*}} \arrow[r] & \overline{h}_{s_{j''},s_{j'}}^{m_0,\mathcal{S}^{i^*}} \arrow[r] \arrow[dr] & \widetilde{h}_{s_{j'},T}^{m_0,\mathcal{S}^{i^*}} \arrow[r] & \overline{h}_{s_{j'},s_j}^{m_0,\mathcal{S}^{i^*}} \arrow[r] \arrow[dr] & \widetilde{h}_{s_j,T}^{m_0,\mathcal{S}^{i^*}} \arrow[r] & \overline{h}_{s_j,T}^{m_0,\mathcal{S}^{i^*}} & {= f} \\
 & & A_{s_{j''},s_{j'}}^{m_0,\mathcal{S}^{i^*}} \arrow[u] \arrow[dr] & & A_{s_{j'},s_j}^{m_0,\mathcal{S}^{i^*}} \arrow[u] \arrow[r] & \overline{h}_{s_{j'},T}^{m_0,\mathcal{S}^{i^*}} & {= f} \\
 & & & \overline{h}_{s_{j''},s_j}^{m_0,\mathcal{S}^{i^*}} \arrow[r] \arrow[dr] & \widetilde{h}_{s_j,T}^{m_0,\mathcal{S}^{i^*}} \arrow[r] & \overline{h}_{s_j,T}^{m_0,\mathcal{S}^{i^*}} & {= f} \\
 & & & & A_{s_{j''},s_j}^{m_0,\mathcal{S}^{i^*}} \arrow[u] \arrow[r] & \overline{h}_{s_{j''},T}^{m_0,\mathcal{S}^{i^*}} & {= f}
\end{tikzcd} \tag{3.1}$$

By adding other increments $s_{j'''}$ and $s_{j''''}$ (cf. Figure 4), it can happen that s_j can no longer be considered close enough. In this situation, a linear regression around $X_{s_j}^{m_0}$ would not be considered sufficient for inner trajectories that start at $X^{m_0}_{s_{j'''}}$ or $X^{m_0}_{s_{j''''}}$. To deal with this situation, one should introduce $(\overline{s_{j'''}}, \underline{s_{j'''}})$ and $(\overline{s_{j''''}}, \underline{s_{j''''}})$ defined in the end of Section 2. For instance if $\overline{s_{j'''}} = s_j$ and $\underline{s_{j'''}} = s_{j''}$, one starts the backward induction associated to the increment $s_{j'''}$ by the final condition $\overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j'''},s_j}(x) = \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''},s_j}(x)$ instead of $\overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j'''},T}(x) = f(x)$ and (3.1) becomes

$$
\begin{tikzcd}[column sep=small]
\widetilde{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j'''},s_j} \arrow[r] & \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j'''},s_{j''}} \arrow[r] \arrow[dr] & \widetilde{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''},T} \arrow[r] & \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''},s_{j'}} \arrow[r] \arrow[dr] & \widetilde{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j'},T} \arrow[r] & \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j'},s_j} \arrow[r] \arrow[dr] & \widetilde{h}^{m_0,\mathcal{S}^{i^*}}_{s_j,T}\ldots f \\
& & A^{m_0,\mathcal{S}^{i^*}}_{s_{j'''},s_{j''}} \arrow[u] \arrow[dr] & & A^{m_0,\mathcal{S}^{i^*}}_{s_{j''},s_{j'}} \arrow[u] \arrow[dr] & & A^{m_0,\mathcal{S}^{i^*}}_{s_{j'},s_j}\ldots f \arrow[u] \\
& & & \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j'''},s_{j'}} \arrow[r] \arrow[dr] & \widetilde{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''},T} \arrow[r] & \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''},s_j} \arrow[r] \arrow[dr] & \widetilde{h}^{m_0,\mathcal{S}^{i^*}}_{s_j,T}\ldots f \\
& & & & A^{m_0,\mathcal{S}^{i^*}}_{s_{j'''},s_{j'}} \arrow[u] \arrow[dr] & & A^{m_0,\mathcal{S}^{i^*}}_{s_{j''},s_j}\ldots f \arrow[u] \\
& & & & & \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j'''},s_j} = \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''},s_j} &
\end{tikzcd}
\tag{3.2}
$$

In Figure 4, we also set $\overline{s_{j''''}} = s_{j'}$ as well as $\underline{s_{j''''}} = s_{j'''}$ and the tree (3.2) can be further changed to include the dependency structure induced by $s_{j''''}$. Indeed, we urge the reader to check that (3.2) can be as easily completed as done for (3.1) to include the dependency structure induced by $s_{j''''}$.

Even with the simple example presented in this subsection, one can show the benefit of this method. Indeed, in addition to a fine simulation of U using $\widetilde{h}$, this method defines a set of functions $\overline{h}$ that can be considered as coarse conditional approximation of U. These conditional approximations can be used as forward components of another functional. For instance, given the example presented above and illustrated in Figure 4, the simulation of an m_0 realization of $V_{s_{j''}} = \mathbb{E}\left((U_{s_j} - U_{s_{j'}})_+ \middle| X_{s_{j''}}\right)$ can be done with

$$\widetilde{V}^{m_0}_{s_{j''}} = \frac{1}{M_1}\sum_{m_1=1}^{M_1}\left(\left[\overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''},s_j}(X^{m_0,m_1}_{s_{j''},s_j}) - \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''},s_{j'}}(X^{m_0,m_1}_{s_{j''},s_{j'}})\right]_+\right).$$

These functions $\overline{h}$ can be also used for risk measures. For example, the conditional value at risk $\mathbb{V}a\mathbb{R}^{\alpha\%}\left[U_{s_j} - U_{s_{j'}} \middle| X_{s_{j''}}\right]$ of level $\alpha\%$ can be computed after sorting $\left(\overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''},s_j}(X^{m_0,m_1}_{s_{j''},s_j}) - \overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''},s_{j'}}(X^{m_0,m_1}_{s_{j''},s_{j'}})\right)_{1\leq m_1\leq M_1}$.

Remark 3.1. *Referring to Figure 4, for any g, when $\mathbb{E}\left(g(U_{s_{j'''}})\middle| X_{s_{j''''}}\right)$,*

$\mathbb{E}\left(g(U_{s_{j''}})\Big|X_{s_{j''''}}\right)$ *and* $\mathbb{E}\left(g(U_{s_{j'}})\Big|X_{s_{j''''}}\right)$ *are well defined their simulation can be directly performed using* $\overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''''},s_{j'''}}$*,* $\overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''''},s_{j''}}$ *or* $\overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''''},s_{j'}}$*. This is not the case for* $\mathbb{E}\left(g(U_{s_j})\Big|X_{s_{j''''}}\right)$ *since* $\overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''''},s_j}$ *were not computed because* $\overline{s_{j''''}} = s_{j'} < s_j$*. If* $\mathbb{E}\left(g(U_{s_j})\Big|X_{s_{j''''}}\right)$ *is needed, one should be either less conservative for the choice of* $\epsilon_{1,s_{j''''}}$ *and* $\epsilon_{2,s_{j''''}}$ *(cf. (2.11) and (Bias Control)) that makes, or use higher order terms for the regression as presented in Remark 2.2.*

The other benefit of our method is the possibility to have a parareal alike implementation [24] and thus make the algorithm parallel in time in addition to have it parallel in paths. Indeed, refering to Figure 4, if we associate the final conditions $\overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j''},s_{j'}}$ and $\overline{h}^{m_0,\mathcal{S}^{i^*}}_{s_{j'},s_j}$ respectively to each subinterval $[s_{j''}, s_{j'})$ and $[s_{j'}, s_j)$, we can perform concurrent calculations on these intervals.

3.2 BSDEs with a Markov forward process

In the previous subsection 3.1, we saw the implementation of our method on a simple problem and we showed its benefits when one has to simulate functionals of functionals of a Markov process. BSDEs and RBSDEs are specific functionals of functionals of a forward process assumed Markov in various situations. After [29], BSDEs became very widely studied, especially in the quantitative finance community starting with [13]. Here we consider the One step forward Dynamic Programming (ODP) scheme for discrete BSDEs

$$(ODP) \quad Y_T = \zeta \text{ and for } k < 2^L \left\{ \begin{array}{c} Y_{t_k} = \mathbb{E}_{t_k}[Y_{t_{k+1}} + \Delta_t f(t_k, Y_{t_{k+1}}, Z_{t_k})], \\ Z_{t_k} = \mathbb{E}_{t_k}[Y_{t_{k+1}}(W_{t_{k+1}} - W_{t_k})/\Delta_t]. \end{array} \right.$$

(ODP) was studied for instance in [15, 23]. Here we consider $\zeta = f(t_{2^L}, X_T)$ to be some square integrable random variable that depends on X_T. Given a discretization sequence $(s_j)_{j=0,...,2^L} \in \mathcal{S}$ and referring to (2.1) and (2.2), the simulation of X involves the increments of an $\mathbb{R}^{d_2}$-Brownian motion W with $\xi^{m_0}_{t_k} = W^{m_0}_{t_k} - W^{m_0}_{t_{k-1}}$ and $\xi^{m_0,m_1}_{s_j,t_k} = W^{m_0,m_1}_{s_j,t_k} - W^{m_0,m_1}_{s_j,t_{k-1}}$ where $W^1, ..., W^{M_0}$ are independent realizations of W with

$$W^{m_0,m_1}_{s_j,t_k} = W^{m_0,m_1}_{s_j,t_{k-1}} + \Delta W^{m_0,m_1}_{s_j,t_k} \text{ and } \quad W^{m_0,m_1}_{s_j,s_j}\Big|_{m_1=1,...,M_1+M_1'} = W^{m_0}_{s_j},$$

$(\Delta W^{m_0,m_1}_{s_j,t_k})^{(m_0,m_1)\in\{1,...,M_0\}\times\{1,...,M_1+M_1'\}}_{k\in\{j,...,2^L\},j\in\{1,...,2^L\}}$ are independent Brownian motion increments independent from $W^1, ..., W^{M_0}$ with $\mathbb{E}([\Delta W^{m_0,m_1}_{s_j,t_k}]^2) = \Delta_t$. As pointed out below Remark 2.1, if an inner trajectory $\{X^{m_0,m_1}\}$ is needed several times in the backward induction, we simulate independent copies of it and thus independent copies of ξ^{m_0,m_1} and use each copy once.

For given indices $k < j \in \{1, ..., 2^L\}$ that satisfy $s_j < s_k \leq \overline{s_j}$ and using $\delta_{s_j}(s_k)$ defined in (2.14), we also set $\Delta W^{m_0,m_1}_{s_j,s_k,\delta_{s_j}(s_k)} = W^{m_0,m_1}_{s_j,\delta_{s_j}(s_k)} - W^{m_0,m_1}_{s_j,s_k}$. For each k, the Borel $\mathcal{B}(\mathbb{R}) \otimes \mathcal{B}(\mathbb{R}^{d_2})$-measurable driver $f(t_k, \cdot, \cdot)$ is assumed to satisfy Lipschitz condition of Section 4.

Given the discretization set $\mathcal{S}$, one can define two coarse approximations around $X^{m_0}_{s_k}$ conditionally on $X^{m_0}_{s_j}$ given by

$$\overline{y}^{m_0,\mathcal{S}}_{s_j,s_k}(x) = \ell\Big[\overline{y}^{m_0,\mathcal{S}}_{s_j,s_k}\Big] + {}^t\mathcal{T}^{m_0}_{s_j,s_k,M_1'}(x) C^{m_0,\mathcal{S}}_{s_j,s_k}, \tag{3.3}$$

$${}^t\overline{z}^{m_0,\mathcal{S}}_{s_j,s_k}(x) = {}^t\widetilde{z}^{m_0,\mathcal{S}}_{s_k,\overline{s_k}} + {}^t\mathcal{T}^{m_0}_{s_j,s_k,M_1'}(x) D^{m_0,\mathcal{S}}_{s_j,s_k}, \tag{3.4}$$

as well as two fine approximations at $X^{m_0}_s$, for $s \in \{s_k, s_k+\Delta_t, ..., \delta_{s_j}(s_k)-\Delta_t\}$ with $\Delta_s = \delta_{s_j}(s_k) - s$ and $\Delta_{s_k} = \delta_{s_j}(s_k) - s_k$, given by

$$\widetilde{y}^{m_0,\mathcal{S}}_{s,\overline{s_k}} = \frac{1}{M_1}\sum_{m_1=1}^{M_1}\left[\begin{array}{c}\Delta_s f(s_k, \overline{y}^{m_0,\mathcal{S}}_{s_k,\delta_{s_j}(s_k)}(X^{m_0,m_1}_{s,\delta_{s_j}(s_k)}), \widetilde{z}^{m_0,\mathcal{S}}_{s,\overline{s_k}}) \\ +\overline{y}^{m_0,\mathcal{S}}_{s_k,\delta_{s_j}(s_k)}(X^{m_0,m_1}_{s,\delta_{s_j}(s_k)})\end{array}\right], \tag{3.5}$$

$$\widetilde{z}^{m_0,\mathcal{S}}_{s,\overline{s_k}} = \frac{1}{M_1\Delta_s}\sum_{m_1=1}^{M_1}\overline{y}^{m_0,\mathcal{S}}_{s_k,\delta_{s_j}(s_k)}(X^{m_0,m_1}_{s,\delta_{s_j}(s_k)})\left(W^{m_0,m_1}_{s,\delta_{s_j}(s_k)} - W^{m_0}_s\right) \tag{3.6}$$

and we set the final coarse approximation to

$$\overline{y}^{m_0,\mathcal{S}}_{s_j,\overline{s_j}} = \begin{cases} f(t_{2^L}, X^{m_0,m_1}_{s_j,t_{2^L}}) & \text{if } \overline{s_j} = t_{2^L}, \\ \overline{y}^{m_0,\mathcal{S}}_{\underline{s_j},\overline{s_j}}(X^{m_0,m_1}_{s_j,\overline{s_j}}) = \overline{y}^{m_0,\mathcal{S}}_{\delta_{s_j}(s_j),\overline{s_j}}(X^{m_0,m_1}_{s_j,\overline{s_j}}) & \text{if } \overline{s_j} < t_{2^L}, \end{cases} \tag{3.7}$$

$\overline{s_j} > \underline{s_j} > s_j$ are specified during the initialization phase (cf. (2.12)) then actualized at each step (cf. (2.25) and (2.26)) where (Bias Control), ε and e are expressed in Definition 3.1.

Since $\mathcal{T}$ was already expressed in (2.6), to complete this inductive interconnected definition of $(\overline{y}, \widetilde{y}, \overline{z}, \widetilde{z})$, we set the vector $C^{m_0,\mathcal{S}}_{s_j,s_k} = \gamma^{m_0,\mathcal{S}}_{s_j,s_k}\widehat{C}^{m_0,\mathcal{S}}_{s_j,s_k}$ and the matrix $D^{m_0,\mathcal{S}}_{s_j,s_k}$ to be equal to

$$\widehat{C}^{m_0,\mathcal{S}}_{s_j,s_k} = \frac{(\widetilde{\Lambda}^{m_0}_{s_j,s_k,M_1'})^{-1}}{M_1}\sum_{m_1=1}^{M_1}\mathcal{Y}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k,M_1'}(X^{m_0,m_1}_{s_j,s_k}, X^{m_0,m_1}_{s_j,\delta_{s_j}(s_k)}), \tag{3.8}$$

$$D^{m_0,\mathcal{S}}_{s_j,s_k} = \frac{(\widetilde{\Lambda}^{m_0}_{s_j,s_k,M_1'})^{-1}}{M_1} \sum_{m_1=1}^{M_1} \mathcal{Z}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k,M_1'}(X^{m_0,m_1}_{s_j,s_k}, \Delta W^{m_0,m_1}_{s_j,s_k,\delta_{s_j}(s_k)}, X^{m_0,m_1}_{s_j,\delta_{s_j}(s_k)}), \quad (3.9)$$

with $\mathcal{Y}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k,M_1'}(x',x) = \mathcal{T}^{m_0}_{s_j,s_k,M_1'}(x')\mathbb{Y}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k}(x',x)$ is $\mathcal{F}_{\delta_{s_j}(s_k)} \otimes \mathcal{B}(\mathbb{R}^{2d_1})-$measurable and $\mathcal{Z}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k,M_1'}(x',w,x) = \mathcal{T}^{m_0}_{s_j,s_k,M_1'}(x')^t\mathbb{Z}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k}(w,x)$ is a vector function measurable with respect to $\mathcal{F}_{\delta_{s_j}(s_k)} \otimes \mathcal{B}(\mathbb{R}^{2d_1+d_2})$, where

$$\mathbb{Y}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k}(x',x) = \begin{bmatrix} \Delta_{s_k} f(s_k, \overline{y}^{m_0,\mathcal{S}}_{s_j,\delta_{s_j}(s_k)}(x), \overline{z}^{m_0,\mathcal{S}}_{s_j,s_k}(x')) \\ +\overline{y}^{m_0,\mathcal{S}}_{s_j,\delta_{s_j}(s_k)}(x) - \widetilde{y}^{m_0,\mathcal{S}}_{s_k,\overline{s_k}} \end{bmatrix}, \quad (3.10)$$

and

$$\mathbb{Z}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k}(w,x) = \overline{y}^{m_0,\mathcal{S}}_{s_j,\delta_{s_j}(s_k)}(x)\frac{w}{\Delta_{s_k}} - \widetilde{z}^{m_0,\mathcal{S}}_{s_k,\overline{s_k}}. \quad (3.11)$$

Finally, applying similar variance adjustment procedure as the one presented in Section 2.3, we set the value of $\gamma^{m_0,\mathcal{S}}_{s_j,s_k}$ and we define

$$\ell\Big[\overline{y}^{m_0,\mathcal{S}}_{s_j,s_k}\Big] = \widetilde{y}^{m_0,\mathcal{S}}_{s_k,\overline{s_k}} + \frac{(1-\gamma^{m_0,\mathcal{S}}_{s_j,s_k})}{M_1}\sum_{m_1=1}^{M_1} {}^t\mathcal{T}^{m_0}_{s_j,s_k,M_1'}(X^{m_0,m_1}_{s_j,s_k})\widehat{C}^{m_0,\mathcal{S}}_{s_j,s_k}. \quad (3.12)$$

From equations above, one can associate quadratic minimization problems to $C^{m_0,i}_{s_j,s_k}$ and to $D^{m_0,i}_{s_j,s_k}$, as done for $H^{m_0,i}_{s_j,s_k}$ in (2.18). In the same fashion as in Definition 2.1, we define the double layer approximations (Y^{m_0}, Z^{m_0}) and $(Y^{m_0,m_1}, Z^{m_0,m_1})$ of functionals Y and Z.

Definition 3.1. *For $i^* = \min(\min\{i = 1, ..., L-L', j^*_i = 2^L\}, L-L')$*

- *For $k < j \in \{1, ..., 2^L\}$ that satisfy $s_j < s_k \leq \overline{s_j} < t_{2^L} = T$, the simulation $Y^{m_0,m_1}_{s_j,s_k}$ and $Z^{m_0,m_1}_{s_j,s_k}$ of Y and Z respectively around $X^{m_0}_{s_k}$ conditionally on $X^{m_0}_{s_j}$ are set to be equal to $\overline{y}^{m_0,\mathcal{S}i^*}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k})$ and $\overline{z}^{m_0,\mathcal{S}i^*}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k})$ where $\overline{y}$ and $\overline{z}$ are given in (3.3), (3.4) and (3.7).*

- *For $k \in \{1, ..., 2^L\}$ and $s \in \{s_k, s_k+\Delta_t, ..., \delta_{s_k}(s_k)-\Delta_t\} - \{0\}$, the simulation $Y^{m_0}_s$ and $Z^{m_0}_s$ of Y and Z respectively at $X^{m_0}_s$ are set to be equal to $\widetilde{y}^{m_0,\mathcal{S}^{i^*}}_{s,\overline{s_k}}$ and to $\widetilde{z}^{m_0,\mathcal{S}^{i^*}}_{s,\delta_{s_k}(s_k)}$ with $\widetilde{y}$ and $\widetilde{z}$ expressed in (3.5) and (3.6).*

- *The average Y^{lear}_0 and Z^{lear}_0 of learned values on Y_0 and Z_0 are respectively equal to*

$$Y^{lear}_0 = \frac{1}{M_0}\sum_{m_0=1}^{M_0} \widetilde{y}^{m_0,\mathcal{S}^{i^*}}_{0,\overline{0}}, \quad Z^{lear}_0 = \frac{1}{M_0}\sum_{m_0=1}^{M_0} \widetilde{z}^{m_0,\mathcal{S}^{i^*}}_{0,\overline{0}} \quad (3.13)$$

and the simulated values Y_0^{sim} and Z_0^{sim} of Y_0 and Z_0 are respectively equal to

$$\begin{aligned} Y_0^{sim} &= \frac{1}{M_0}\sum_{m_0=1}^{M_0}\left[\delta(0)f\left(\delta(0),\widetilde{y}^{m_0,\mathcal{S}^{i^*}}_{\delta(0),\overline{\delta(0)}},Z_0^{sim}\right)+\widetilde{y}^{m_0,\mathcal{S}^{i^*}}_{\delta(0),\overline{\delta(0)}}\right],\\ Z_0^{sim} &= \sum_{m_0=1}^{M_0}\widetilde{y}^{m_0,\mathcal{S}^{i^*}}_{\delta(0),\overline{\delta(0)}}\frac{W^{m_0}_{\delta(0)}}{\delta(0)M_0}. \end{aligned} \tag{3.14}$$

- *Introduced in* (2.10)*, (Bias Control) associated to (ODP) is defined at $s\in\mathcal{S}^0$ for $u\in\mathcal{S}^0\cap]s,\overline{\delta(s)}]$ by*

$$\left|\frac{1}{M_0}\sum_{m_0=1}^{M_0}\left(\widetilde{y}^{m_0,\mathcal{S}^0}_{s,u}-\widetilde{y}^{m_0,\mathcal{S}^0}_{\delta(s),\overline{\delta(s)}}-(\delta(s)-s)f(s,\widetilde{y}^{m_0,\mathcal{S}^0}_{\delta(s),\overline{\delta(s)}},\widetilde{z}^{m_0,\mathcal{S}^0}_{s,\overline{s}})\right)\right|<\epsilon^{\mathcal{S}^0}_{2,s}$$

where for each set $\mathcal{S}$, $\{\epsilon^{\mathcal{S}}_{2,s}\}_{s\in\mathcal{S}}$ is a family of positive bias tuning parameters.

- *For $k\in\{j_i^*+1,...,2^L\}$, setting $s_k=T-s_k^i$ and noticing that $\delta^{\mathcal{S}^i}(s_k)=\delta^{\widehat{\mathcal{S}}^i}(\delta^{\widehat{\mathcal{S}}^i}(s_k))$, $e^{\mathcal{S}^i}_{s_k}$ and $\varepsilon^{\mathcal{S}^i}_{s_k}$ are given by*

$$e^{\mathcal{S}^i}_{s_k}=\frac{1}{M_0M_1}\sum_{m_0=1}^{M_0}\sum_{m_1=1}^{M_1}\left[\overline{y}^{m_0,\widehat{\mathcal{S}}^i}_{\delta^{\widehat{\mathcal{S}}^i}(s_k),\delta^{\mathcal{S}^i}(s_k)}(X^{m_0,m_1}_{\delta^{\widehat{\mathcal{S}}^i}(s_k),\delta^{\mathcal{S}^i}(s_k)})-\overline{y}^{m_0,\mathcal{S}^i}_{s_k,\delta^{\mathcal{S}^i}(s_k)}(X^{m_0,m_1}_{\delta^{\widehat{\mathcal{S}}^i}(s_k),\delta^{\mathcal{S}^i}(s_k)})\right],$$

$$\varepsilon^{\mathcal{S}^i}_{s_k}=\sum_{s\in\mathcal{S}^i,s>s_k}\epsilon^{\mathcal{S}^i}_{2,s}.$$

Remark 3.2. *0. The different points of Remark* 2.3 *can be highlighted here.*

1. *Given a discretization set $\mathcal{S}$ and $s_k\in\mathcal{S}$, the choice of $\underline{s_k}$ and on $\overline{s_k}$ is completely known in Definition* 3.1 *through the value of e, ε and inequality (Bias Control).*
2. *The value of e, ε and inequality (Bias Control) involve mainly the approximation of Y since using criteria on the approximation of Z would involve very large number of trajectories, making it impracticable.*
3. *Although possible, we did not judge necessary to implement a variance adjustment method on the Z component.*

4. *As a future work, we would like to apply variance reduction methods with 1NMC and provide very accurate double layer estimations of the Z term.*

5. *With BSDEs, it is possible to use other (Bias Control) inequalities. Indeed, using rather an MDP scheme (cf. [17]), (Bias Control) of Definition 3.1 can be replaced by*

$$\left|\frac{1}{M_0}\sum_{m_0=1}^{M_0}\left(\widetilde{y}^{m_0,\mathcal{S}^0}_{s,u} - \widetilde{y}^{m_0,\mathcal{S}^0}_{\delta(r),\overline{\delta(r)}} - \sum_{\theta\in\mathcal{S}^0,\theta=s}^{r}(\delta(\theta)-\theta)f(s,\widetilde{y}^{m_0,\mathcal{S}^0}_{\delta(\theta),\overline{\delta(\theta)}},\widetilde{z}^{m_0,\mathcal{S}^0}_{\theta,\overline{\theta}})\right)\right| < \epsilon^{\mathcal{S}^0}_{2,s},$$

for any $r \in \mathcal{S}^0 \cap [s,u[$.

3.3 RBSDEs with a Markov forward process

The generally studied RBSDEs are functionals of a Markov process. Here, we consider an application to RBSDEs as the one presented in [6] with X simulated like in Section 3.3 and functions $g(\cdot)$ and driver $\{f(t_k,\cdot)\}_{k=0}^{2^L-1}$ assumed to satisfy Lipschitz condition of Section 4. We want to propose a double layer approximation V^{m_0} and V^{m_0,m_1} of the Snell envelope V, solution to

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

that can be done using straightforwardly the recipe of Section 2 combined with a maximization by g. In fact, given a discretization set $\mathcal{S}$ and indices $k<j\in\{1,...,2^L\}$ that satisfy $s_j < s_k \le \overline{s_j}$ and using $\delta_{s_j}(s_k)$ defined in (2.14), we set the coarse approximation $\overline{v}_{s_j,s_k}$ around $X^{m_0}_{s_k}$ conditionally on $X^{m_0}_{s_j}$ to

$$\overline{v}^{m_0,\mathcal{S}}_{s_j,s_k}(x) = \overline{w}^{m_0,\mathcal{S}}_{s_j,s_k}(x) \vee g(x), \tag{3.15}$$

and the fine approximation $\widetilde{v}_{s,\overline{s_k}}$ at $X^{m_0}_s$, $s\in\{s_k, s_k+\Delta_t, ..., \delta_{s_j}(s_k)-\Delta_t\}$, to

$$\widetilde{v}^{m_0,\mathcal{S}}_{s,\overline{s_k}} = \widetilde{w}^{m_0,\mathcal{S}}_{s,\overline{s_k}} \vee g(X^{m_0}_s). \tag{3.16}$$

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

$$\widetilde{w}^{m_0,\mathcal{S}}_{s,\overline{s_k}} = \frac{1}{M_1}\sum_{m_1=1}^{M_1}\Big(\Delta_s f(s_k, \overline{v}^{m_0,\mathcal{S}}_{s_k,\delta_{s_j}(s_k)}(X^{m_0,m_1}_{s,\delta_{s_j}(s_k)})) + \overline{v}^{m_0,\mathcal{S}}_{s_k,\delta_{s_j}(s_k)}(X^{m_0,m_1}_{s,\delta_{s_j}(s_k)})\Big), \tag{3.17}$$

$$\overline{w}^{m_0,\mathcal{S}}_{s_j,s_k}(x) = \ell\Big[\overline{w}^{m_0,\mathcal{S}}_{s_j,s_k}\Big] + {}^t\mathcal{T}^{m_0}_{s_j,s_k,M_1'}(x) B^{m_0,\mathcal{S}}_{s_j,s_k}, \tag{3.18}$$

where

$$\ell\Big[\overline{w}^{m_0,\mathcal{S}}_{s_j,s_k}\Big] = \widetilde{w}^{m_0,\mathcal{S}}_{s_k,\overline{s_k}} + \frac{(1-\gamma^{m_0,\mathcal{S}}_{s_j,s_k})}{M_1}\sum_{m_1=1}^{M_1} {}^t\mathcal{T}^{m_0}_{s_j,s_k,M_1'}(X^{m_0,m_1}_{s_j,s_k})\widehat{B}^{m_0,\mathcal{S}}_{s_j,s_k}, \tag{3.19}$$

$B^{m_0,\mathcal{S}}_{s_j,s_k} = \gamma^{m_0,\mathcal{S}}_{s_j,s_k}\widehat{B}^{m_0,\mathcal{S}}_{s_j,s_k}$ with

$$\widehat{B}^{m_0,\mathcal{S}}_{s_j,s_k} = \frac{(\widetilde{\Lambda}^{m_0}_{s_j,s_k,M_1'})^{-1}}{M_1}\sum_{m_1=1}^{M_1}\mathcal{B}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k,M_1'}(X^{m_0,m_1}_{s_j,s_k}, X^{m_0,m_1}_{s_j,\delta_{s_j}(s_k)})$$

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

$$\mathbb{B}^{m_0,\mathcal{S},\delta_{s_j}(s_k)}_{s_j,s_k}(x) = \begin{bmatrix} \Delta_{s_k} f(s_k, \overline{v}^{m_0,\mathcal{S}}_{s_j,\delta_{s_j}(s_k)}(x)) \\ +\,\overline{v}^{m_0,\mathcal{S}}_{s_j,\delta_{s_j}(s_k)}(x) - \widetilde{w}^{m_0,\mathcal{S}}_{s_k,\overline{s_k}} \end{bmatrix}, \tag{3.20}$$

with a final coarse approximation given by

$$\overline{v}^{m_0,\mathcal{S}}_{s_j,\overline{s_j}}(x) = \begin{cases} g(x) & \text{if } \overline{s_j} = t_{2^L}, \\ \overline{v}^{m_0,\mathcal{S}}_{\underline{s_j},\overline{s_j}}(x) = \overline{v}^{m_0,\mathcal{S}}_{\delta_{s_j}(s_j),\overline{s_j}}(x) & \text{if } \overline{s_j} < t_{2^L}, \end{cases} \tag{3.21}$$

where $\overline{s_j} > \underline{s_j} > s_j$ are specified during the initialization phase (cf. (2.12)) then actualized at each step (cf. (2.25) and (2.26)).

Definition 3.2. *For* $i^* = \min(\min\{i = 1, ..., L-L', j_i^* = 2^L\}, L-L')$

- *For* $k < j \in \{1, ..., 2^L\}$ *that satisfy* $s_j < s_k \leq \overline{s_j} < t_{2^L} = T$, *the simulation* $V^{m_0,m_1}_{s_j,s_k}$ *of* V *around* $X^{m_0}_{s_k}$ *conditionally on* $X^{m_0}_{s_j}$ *is set to be equal to* $\overline{v}^{m_0,\mathcal{S}^{i^*}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k})$ *where* $\overline{v}$ *is given in* (3.15), (3.18) *and* (3.21).
- *For* $k \in \{1, ..., 2^L\}$ *and* $s \in \{s_k, s_k + \Delta_t, ..., \delta_{s_k}(s_k) - \Delta_t\} - \{0\}$, *the simulation* $V^{m_0}_s$ *of* V *at* $X^{m_0}_s$ *is set to be equal to* $\widetilde{v}^{m_0,\mathcal{S}^{i^*}}_{s,\overline{s_k}}$ *with* $\widetilde{v}$ *expressed in* (3.16) *and* (3.17).
- *The average* V_0^{lear} *of the learned values on* V_0 *is equal to*

$$V_0^{lear} = \frac{1}{M_0}\sum_{m_0=1}^{M_0}\widetilde{v}^{m_0,\mathcal{S}^{i^*}}_{0,\overline{0}}, \tag{3.22}$$

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

$$V_0^{sim} = g(x_0) \vee \frac{1}{M_0}\sum_{m_0=1}^{M_0}\left[\delta(0) f\left(\delta(0), \widetilde{v}^{m_0,\mathcal{S}^{i^*}}_{\delta(0),\overline{\delta(0)}}\right) + \widetilde{v}^{m_0,\mathcal{S}^{i^*}}_{\delta(0),\overline{\delta(0)}}\right] \tag{3.23}$$

- *Introduced in (2.10), $\underline{\text{(Bias}}$ Control) associated to (Snl) is defined at $s \in \mathcal{S}^0$ for $u \in \mathcal{S}^0 \cap]s, \overline{\delta(s)}]$ by*

$$\left| \frac{1}{M_0} \sum_{m_0=1}^{M_0} \left(\widetilde{w}_{s,u}^{m_0,\mathcal{S}^0} - \widetilde{v}_{\delta(s),\overline{\delta(s)}}^{m_0,\mathcal{S}^0} - (\delta(s) - s) f(s, \widetilde{v}_{\delta(s),\overline{\delta(s)}}^{m_0,\mathcal{S}^0}) \right) \right| < \epsilon_{2,s}^{\mathcal{S}^0}$$

where for each set $\mathcal{S}$, $\{\epsilon_{2,s}^{\mathcal{S}}\}_{s \in \mathcal{S}}$ is a family of positive bias tuning parameters.

- *For $k \in \{j_i^* + 1, ..., 2^L\}$, setting $s_k = T - s_k^i$ and noticing that $\delta^{\mathcal{S}^i}(s_k) = \delta^{\widehat{\mathcal{S}}^i}(\delta^{\widehat{\mathcal{S}}^i}(s_k))$, $e_{s_k}^{\mathcal{S}^i}$ and $\varepsilon_{s_k}^{\mathcal{S}^i}$ are given by*

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

$$\varepsilon_{s_k}^{\mathcal{S}^i} = \sum_{s \in \mathcal{S}^i, s > s_k} \epsilon_{2,s}^{\mathcal{S}^i}.$$

Remark 3.3. *0. The different points of Remark 2.3 can be highlighted here.*

1. *Given a discretization set $\mathcal{S}$ and $s_k \in \mathcal{S}$, the choice of $\underline{s_k}$ and on $\overline{s_k}$ is completely known in Definition 3.2 through the value of e, ε and inequality (Bias Control).*

2. *Unlike BSDEs, it is not possible to use an MDP scheme for (Bias Control) as explained in Remark 3.2.5.*

3. *Although using an optimal stopping formulation [11] of the dynamic programming is known to provide better numerical results [25], we prefared here to use NMC on the top of the original algorithm [32] since its error estimates remains similar to the one presented in Section 4 for BSDEs.*

4. *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.*

4 Error estimates and cutting bias propagation

After expressing error estimates for both coarse and fine approximations in Section 4.1, we show how to cut bias propagation using our new judicious trick presented in Section 4.2.

4.1 Regression-based NMC and increasing the learning depth

Before presenting the main elements, we point out that we have intentionally considered only discrete functionals of a Markov process. The approximation due to discretization of the continuous version of BSDEs is not studied and we refer to [15, 23] among others that quantify well the resulting error. Moreover, we also consider the discretized version of the Markov process introduced in (2.1) and (2.2) where

$$\mathcal{E}_{t_k}(x,\xi) = x + \Delta_t b(t_k,x) + \sigma(t_k,x)\xi \tag{4.1}$$

with the usual (cf. [27]) Lipschitz continuity condition on the coefficients $b(t,x)$ and $\sigma(t,x)$ uniformly with respect to $t \in [0,T]$. Similar to what was considered in Section 3.2, the noise ξ is given by increments of a vector of independent Brownian motions i.e. $\xi^{m_0}_{t_k} = W^{m_0}_{t_k} - W^{m_0}_{t_{k-1}}$ and $\xi^{m_0,m_1}_{s_j,t_k} = W^{m_0,m_1}_{s_j,t_k} - W^{m_0,m_1}_{s_j,t_{k-1}}$. (4.1) can be read as an Euler scheme of a stochastic differential equation that admits a strong solution. In this paper, when the discretization is needed, we assume that L is sufficiently large to neglect the discretization error of the forward process X.

Given two arbitrary square integrable random variables χ_1 and χ_2, consider $\{\overline{\chi_3}^{m_1}\}_{m_1=1}^{M_1}$ to be the empirical regression of χ_1 with respect to χ_2, the authors of [7] established an upper bound error of the regression-based NMC estimator $\dfrac{1}{M_1}\displaystyle\sum_{m_1=1}^{M_1}\phi(\overline{\chi_3}^{m_1})$ of $\mathbb{E}(\phi(\mathbb{E}(\chi_1|\chi_2)))$ once we know the representation error $\kappa = \mathbb{E}(\chi_1|\chi_2) - {}^t R\mathcal{B}(\chi_2)$ induced by the projection of $\mathbb{E}(\chi_1|\chi_2)$ on the basis $\mathcal{B}(\chi_2)$. The fine approximations $\widetilde{h}$, $\widetilde{y}$ and $\widetilde{w}$ presented earlier were computed by averaging on the empirical regressions $\overline{h}$, $\overline{y}$ and $\overline{v}$. It is then interesting to see how to control the error of the fine approximations through the representation error like in [7].

First, for $s_j < s_k < \overline{s_j}$ and Borel measurable Θ function of $(X^{m_0,m_1}_{s_j,s_k:\delta(s_k)})$ with $\Theta(X^{m_0,m_1}_{s_j,s_k:\delta(s_k)})$ integrable, we denote $\mathbb{E}^{m_0,x}_{s_j,s_k}$, $\overline{\mathbb{E}}^{m_0,x}_{s_j,s_k}$ and $\widehat{\mathbb{E}}^{m_0,x}_{s_j,s_k}$ the operators defined by

$$\mathbb{E}^{m_0,x}_{s_j,s_k}(\Theta(X^{m_0,m_1}_{s_j,s_k:\delta(s_k)})) = \mathbb{E}^{m_0}_{s_j}\left(\Theta(X^{m_0,m_1}_{s_j,s_k:\delta(s_k)})|X^{m_0,m_1}_{s_j,s_k} = x\right),$$

$$\begin{aligned}&\overline{\mathbb{E}}^{m_0,x}_{s_j,s_k}(\Theta(X^{m_0,m_1}_{s_j,s_k:\delta(s_k)}))\\ &= {}^t\mathcal{T}^{m_0}_{s_j,s_k}(x - X^{m_0}_{s_k})\frac{\left(\overline{\Lambda}^{m_0}_{s_j,s_k}\right)^{-1}}{M_1}\sum_{m_1=1}^{M_1}\left[\mathcal{T}^{m_0}_{s_j,s_k}\left(X^{m_0,m_1}_{s_j,s_k} - X^{m_0}_{s_k}\right)\Theta(X^{m_0,m_1}_{s_j,s_k:\delta(s_k)})\right]\end{aligned}$$

and

$$\widehat{\mathbb{E}}^{m_0,x}_{s_j,s_k}(\Theta(X^{m_0,m_1}_{s_j,s_k:\delta(s_k)})) = {}^t\mathcal{T}^{m_0}_{s_j,s_k}(x - X^{m_0}_{s_k})R^{m_0}_{s_j,s_k}\left[\Theta(X^{m_0,m_1}_{s_j,s_k:\delta(s_k)})\right]$$

with

$$R^{m_0}_{s_j,s_k}\left[\Theta(X^{m_0,m_1}_{s_j,s_k:\delta(s_k)})\right] \in \underset{r\in\mathbb{R}^{d_1}}{\operatorname{argmin}}\,\mathbb{E}^{m_0}_{s_j}\left(\begin{bmatrix}\mathbb{E}^{m_0}_{s_j}\left(\Theta(X^{m_0,m_1}_{s_j,s_k:\delta(s_k)})|X^{m_0,m_1}_{s_j,s_k}\right)\\ -{}^t\mathcal{T}^{m_0}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k} - X^{m_0}_{s_k})r\end{bmatrix}^2\right)$$

When $\mathbb{E}_{s_j}$ is the conditional expectation knowing $X^{m_0}_{s_j}$, $\mathbb{E}^{m_0}_{s_j}$ is the conditional expectation knowing the trajectory of X^{m_0} starting from $X^{m_0}_{s_j}$. $\mathbb{E}^{m_0}_{s_j}$ is used as the regression basis depends on X^{m_0}. For a given $s_j \in \mathcal{S}$, in contrast to expressions presented in sections 2.2, 3.2 and 3.3, we simplify the presentation here and we omit to center the regressions around $\widetilde{h}^{m_0,\mathcal{S}}_{\delta(s_j),\overline{\delta(s_j)}}$, $\widetilde{y}^{m_0,\mathcal{S}}_{\delta(s_j),\overline{\delta(s_j)}}$ or $\widetilde{w}^{m_0,\mathcal{S}}_{\delta(s_j),\overline{\delta(s_j)}}$. Consequently, the value of $\widetilde{h}^{m_0,\mathcal{S}}_{s_j,\overline{s_j}}$, $\widetilde{y}^{m_0,\mathcal{S}}_{s_j,\overline{s_j}}$ and $\widetilde{w}^{m_0,\mathcal{S}}_{s_j,\overline{s_j}}$ are obtained through respectively averaging on $\overline{h}^{m_0,\mathcal{S}}_{s_j,\delta(s_j)}(x) = \overline{\mathbb{E}}^{m_0,x}_{s_j,\delta(s_j)}(\Theta^h(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)}))$, $\overline{y}^{m_0,\mathcal{S}}_{s_j,\delta(s_j)}(x) = \overline{\mathbb{E}}^{m_0,x}_{s_j,\delta(s_j)}(\Theta^y(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)}))$ and on
$\overline{v}^{m_0,\mathcal{S}}_{s_j,\delta(s_j)}(x) = g(x)\vee\overline{\mathbb{E}}^{m_0,x}_{s_j,\delta(s_j)}(\Theta^v(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)}))$, where

$$\Theta^h(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)}) = \overline{h}^{m_0,\mathcal{S}}_{s_j,\delta^2(s_j)}(X^{m_0,m_1}_{s_j,\delta^2(s_j)}) + \sum_{t_l\geq\delta(s_j)}^{\delta^2(s_j)} f(t_l, X^{m_0,m_1}_{s_j,t_l}, X^{m_0,m_1}_{s_j,t_{l+1}}),$$

$$\begin{aligned}\Theta^y(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)}) &= \overline{y}^{m_0,\mathcal{S}}_{s_j,\delta^2(s_j)}(X^{m_0,m_1}_{s_j,\delta^2(s_j)})\\ &\quad+\Delta_{\delta(s_j)}f(\delta(s_j), \overline{y}^{m_0,\mathcal{S}}_{s_j,\delta^2(s_j)}(X^{m_0,m_1}_{s_j,\delta^2(s_j)}), \overline{z}^{m_0,\mathcal{S}}_{s_j,\delta(s_j)}(X^{m_0,m_1}_{s_j,\delta(s_j)})),\end{aligned}$$

$$\begin{aligned}\Theta^v(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)}) &= \overline{v}^{m_0,\mathcal{S}}_{s_j,\delta^2(s_j)}(X^{m_0,m_1}_{s_j,\delta^2(s_j)})\\ &\quad+\Delta_{\delta(s_j)}f(\delta(s_j), \overline{v}^{m_0,\mathcal{S}}_{s_j,\delta^2(s_j)}(X^{m_0,m_1}_{s_j,\delta^2(s_j)})).\end{aligned}$$

We assume Lipschitz condition uniformly in time of the driver f involved in (ODP) and (Snl) with respect to its Y and Z coordinates or with respect to its V coordinate. Although this conditions are not necessary to obtain good numerical results in Section 5, they are required to apply Theorem 2 of [7] (cf. Assumption F2 in [7]) that yeild the following asymptotical result.

Proposition 4.1. *Given that assumptions A1, A2 and A3 of [7]) are fulfilled and that both drivers involved in (ODP) and in (Snl) are $[f]_{Lip}$-Lipschitz we have the following asymptotical inequality*

$$(\widetilde{\rho} - \rho)^2 \leq [\rho]_{Lip} \mathbb{E}^{m_0}_{s_j}(\kappa^2(X^{m_0,m_1}_{s_j,\delta(s_j)})) + O_p(1/M_1) \tag{4.2}$$

as $M_1 \longrightarrow \infty$ where $(\widetilde{\rho}, \rho, [\rho]_{Lip}, \kappa)$ is either equal to $(\widetilde{\rho}^h, \rho^h, [\rho]^h_{Lip}, \kappa^h)$ for (f), $(\widetilde{\rho}^y, \rho^y, [\rho]^y_{Lip}, \kappa^y)$ for (ODP) or equal to $(\widetilde{\rho}^v, \rho^v, [\rho]^v_{Lip}, \kappa^v)$ for (Snl) with $\widetilde{\rho}^h = \widetilde{h}^{m_0,\mathcal{S}}_{s_j,\overline{s_j}}$, $\widetilde{\rho}^y = \widetilde{y}^{m_0,\mathcal{S}}_{s_j,\overline{s_j}}$, $\widetilde{\rho}^v = \widetilde{w}^{m_0,\mathcal{S}}_{s_j,\overline{s_j}}$,

$$\rho^h = \mathbb{E}^{m_0}_{s_j}\left(\mathbb{E}^{m_0,X^{m_0,m_1}_{s_j,\delta(s_j)}}_{s_j,\delta(s_j)}\left[\Theta^h(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)})\right] + \sum_{t_l \geq s_j}^{\delta(s_j)} f(t_l, X^{m_0,m_1}_{s_j,t_l}, X^{m_0,m_1}_{s_j,t_{l+1}})\right),$$

$$\rho^y = \mathbb{E}^{m_0}_{s_j}\left(\mathbb{E}^{m_0,X^{m_0,m_1}_{s_j,\delta(s_j)}}_{s_j,\delta(s_j)}\left[\Theta^y(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)})\right]\right.$$
$$\left. + \Delta_{s_j} f\left(s_j, \mathbb{E}^{m_0,X^{m_0,m_1}_{s_j,\delta(s_j)}}_{s_j,\delta(s_j)}\left[\Theta^y(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)})\right], \widetilde{z}^{m_0,\mathcal{S}}_{s_j,\overline{s_j}}\right)\right),$$

$$\rho^v = \mathbb{E}^{m_0}_{s_j}\left(g(X^{m_0,m_1}_{s_j,\delta(s_j)}) \vee \mathbb{E}^{m_0,X^{m_0,m_1}_{s_j,\delta(s_j)}}_{s_j,\delta(s_j)}\left[\Theta^v(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)})\right]\right.$$
$$\left. + \Delta_{s_j} f\left(s_j, g(X^{m_0,m_1}_{s_j,\delta(s_j)}) \vee \mathbb{E}^{m_0,X^{m_0,m_1}_{s_j,\delta(s_j)}}_{s_j,\delta(s_j)}\left[\Theta^v(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)})\right]\right)\right),$$

$[\rho]^h_{Lip} = 1$, $[\rho]^y_{Lip} = 1 + [f]_{Lip}$, $[\rho]^v_{Lip} = 1 + [f]_{Lip}$ *and*

$$\kappa^h(x) = \mathbb{E}^{m_0,x}_{s_j,\delta(s_j)}\left[\Theta^h(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)})\right] - \widehat{\mathbb{E}}^{m_0,x}_{s_j,\delta(s_j)}\left[\Theta^h(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)})\right],$$

$$\kappa^y(x) = \mathbb{E}^{m_0,x}_{s_j,\delta(s_j)}\left[\Theta^y(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)})\right] - \widehat{\mathbb{E}}^{m_0,x}_{s_j,\delta(s_j)}\left[\Theta^y(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)})\right],$$

$$\kappa^v(x) = \mathbb{E}^{m_0,x}_{s_j,\delta(s_j)}\left[\Theta^v(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)})\right] - \widehat{\mathbb{E}}^{m_0,x}_{s_j,\delta(s_j)}\left[\Theta^v(X^{m_0,m_1}_{s_j,\delta(s_j):\delta^2(s_j)})\right].$$

Proposition 4.1 results from Theorem 2 and Remark 2 of [7]; we expressed $[\rho]_{Lip}$ associated to each problem and we replaced $\mathbb{E}$ by $\mathbb{E}^{m_0}_{s_j}$ as the regression basis depends on X^{m_0}. Assumptions A1, A2 and A3 of [7] are standard assumptions for regressions (cf. [33]). Considering the regression basis presented in Section 2.1 with $\mathbb{E}(|X_t|^2) < \infty$ for any $t \in [0,T]$, these assumptions are fulfilled if: i) the conditional variance of each regressed quantity is integrable and bounded from below by $v_0 > 0$, ii) the regression value is unbiased and iii) each component of the regression basis as well as κ (denoted M in [7]) admit a finite fourth moment. When the latter moment assumption iii is needed to establish error control and can be modified using truncation (cf. [16]), the further i&ii are sufficient to ensure the existence and uniqueness of the regressed representation.

In Proposition 4.1, we provided a control on fine approximations $\widetilde{h}$, $\widetilde{y}$ and $\widetilde{v}$. In Proposition 4.2, we rather focus on coarse approximations and decompose the conditional mean square error $\mathbb{E}^{m_0}_{s_j}\Big([\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k}) - U_{s_k}(X^{m_0,m_1}_{s_j,s_k})]^2\Big)$ into a bias term $\mathcal{W}$, a variance term $\mathcal{V}$ and a regression error term $\mathcal{R}$.

Proposition 4.2. *Assuming i and iii introduced above, for $s_j < s < s_k$ taking their values in the discretization set $\mathcal{S}$, we define*

$$\mathcal{W}^{m_0,\mathcal{S}}_{s_j,s_k}(x) = \mathbb{E}^{m_0}_{s_j}\Big(\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(x) - \mathcal{U}^{m_0,\mathcal{S}}_{s_j,s_k}(x)\Big),$$

$$\mathcal{R}^{m_0,\mathcal{S}}_{s_j,s_k}(x) = \mathbb{E}^{m_0}_{s_j}\Big(\mathcal{U}^{m_0,\mathcal{S}}_{s_j,s_k}(x) - U_{s_k}(x)\Big),$$

$$\mathcal{V}^{m_0,\mathcal{S}}_{s_j,s_k}(x) = Var^{m_0}_{s_j}\Big(\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(x)\Big),$$

with

$$\mathcal{U}^{m_0,\mathcal{S}}_{s_j,s_k}(x) = \overline{\mathbb{E}}^{m_0,x}_{s_j,s_k}\left(U_{\delta(s_k)}(X^{m_0,m_1}_{s_j,\delta(s_k)}) + \sum_{t_{l+1}>s_k}^{\delta(s_k)} f(t_l, X^{m_0,m_1}_{s_j,t_l}, X^{m_0,m_1}_{s_j,t_{l+1}})\right)$$

then

$$\mathbb{E}^{m_0}_{s_j}\Big([\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k}) - U_{s_k}(X^{m_0,m_1}_{s_j,s_k})]^2\Big) = \mathbb{E}^{m_0}_{s_j}(\mathcal{V}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k})) + \mathbb{E}^{m_0}_{s_j}\Big([\mathcal{R}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k}) + \mathcal{W}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k})]^2\Big)$$

and there exists a positive constant $\mathcal{K}^{m_0}_{1,s_j,s_k}$ depending on the regression basis such that

$$\mathbb{E}^{m_0}_{s_j}\Big([\mathcal{W}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k})]^2\Big) \le \mathcal{K}^{m_0}_{1,s_j,s_k}\mathbb{E}^{m_0}_{s_j}\Big([\overline{h}^{m_0,\mathcal{S}}_{s_j,\delta(s_k)}(X^{m_0,m_1}_{s_j,\delta(s_k)}) - U_{\delta(s_k)}(X^{m_0,m_1}_{s_j,\delta(s_k)})]^2\Big).$$

Proof. As we simulate several independent copies of X^{m_0,m_1} (cf. the paragraph under Remark 2.1), we make sure that the approximations $\overline{h}$ are independent from X^{m_0,m_1} conditionally on X^{m_0}. Then, the expansion of the conditional mean square error $\mathbb{E}^{m_0}_{s_j}\Big([\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k}) - U_{s_k}(X^{m_0,m_1}_{s_j,s_k})]^2\Big)$ can be get when we notice that

$$U_{s_k}(X^{m_0,m_1}_{s_j,s_k}) = \mathbb{E}^{m_0,X^{m_0,m_1}_{s_j,s_k}}_{s_j,s_k}\left(U_{\delta(s_k)}(X^{m_0,m_1}_{s_j,\delta(s_k)}) + \sum_{t_{l+1}>s_k}^{\delta(s_k)} f(t_l, X^{m_0,m_1}_{s_j,t_l}, X^{m_0,m_1}_{s_j,t_{l+1}})\right).$$

An expression for the constant $\mathcal{K}^{m_0}_{1,s_j,s_k}$ can be obtained after expanding $\mathbb{E}^{m_0}_{s_j}\Big([\mathcal{W}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k})]^2\Big)$ using

$$\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(x) = \overline{\mathbb{E}}^{m_0,x}_{s_j,s_k}\left(\overline{h}^{m_0,\mathcal{S}}_{s_j,\delta(s_k)}(X^{m_0,m_1}_{s_j,\delta(s_k)}) + \sum_{t_{l+1}>s_k}^{\delta(s_k)} f(t_l, X^{m_0,m_1}_{s_j,t_l}, X^{m_0,m_1}_{s_j,t_{l+1}})\right).$$

□

Finally, we should point out that one could establish a similar result for (ODP) and (Snl). Indeed, for instance, using the following coarse discretization to approximate (ODP)

$$\begin{cases} \hat{Y}_{s_k} = \mathbb{E}_{s_k}\left(\widetilde{f}_{s_k}(\hat{Y}_{\delta(s_k)}, \hat{Z}_{s_k})\right), \\ \hat{Z}_{s_k} = \frac{1}{\Delta_{s_k}}\mathbb{E}_{s_k}\left(\hat{Y}_{\delta(s_k)}(W_{\delta(s_k)} - W_{s_k})\right), \end{cases} \tag{4.3}$$

with $\widetilde{f}_{s_k}(y,z) = y + \Delta_{s_k} f_{s_k}(y,z)$, the bias is then controlled as follows

$$\mathbb{E}^{m_0}_{s_j}\Big([\widetilde{\mathcal{W}}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k})]^2\Big) \leq \widetilde{\mathcal{K}}^{m_0}_{1,s_j,s_k}\mathbb{E}^{m_0}_{s_j}\left(\begin{bmatrix} \widetilde{f}_{s_k}(\widehat{Y}_{s_j,\delta(s_k)}(X^{m_0,m_1}_{s_j,\delta(s_k)}), \widehat{Z}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k})) \\ -\widetilde{f}_{s_k}(\overline{y}^{m_0,\mathcal{S}}_{s_j,\delta(s_k)}(X^{m_0,m_1}_{s_j,\delta(s_k)}), \overline{z}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k})) \end{bmatrix}^2\right),$$

for some positive constant $\widetilde{\mathcal{K}}^{m_0}_{1,s_j,s_k}$ depending on the regression basis. Therefore, the bias upper bound depends heavily on the driver choice. In the case of (Snl), g also plays an important role on the nonlinearity and subsequently on bias.

4.2 Regression with different starting points

As shown in Proposition 4.2, the bias $\mathcal{W}$ at time step s_k is controlled by the mean square error at time step $\delta(s_k)$ decomposed into a variance term

$\mathcal{V}$, a regression error term $\mathcal{R}$ and a bias term at time step $\delta(s_k)$. Thus, increasing the number of time steps weaken the bias control as it involves more and more terms. In some situations, this accumulation of errors is a source of a significant bias back propagation. In this paper, we proposed a new approximation trick to cut this bias back propagation.

In this section, we present a control on this new approximation that is used twice in the generic presentation of our method in Section 2.2. This same approximation was also adapted in Section 3.2 to BSDEs and in Section 3.3 to RBSDEs. In the generic situation, equations (2.16) defines $\widetilde{h}^{m_0,\mathcal{S}}_{s,\overline{s_k}}$ for any $s \in \{s_k, s_k+\Delta_t, ..., \delta_{s_j}(s_k)-\Delta_t\}$ using $\overline{h}^{m_0,\mathcal{S}}_{s_k,\delta_{s_j}(s_k)}(\cdot)$ which is deduced from a regression on $X^{m_0,m_1}_{s_k,\delta_{s_j}(s_k)}$ instead of a regression on $X^{m_0,m_1}_{s,\delta_{s_j}(s_k)}$. Said differently, 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 did more or less the same thing in (2.17) when $\overline{s_j} < T$ as we defined $\overline{h}^{m_0,\mathcal{S}}_{s_j,\overline{s_j}}$ to be equal to $\overline{h}^{m_0,\mathcal{S}}_{\delta_{s_j}(s_j),\overline{s_j}}$ i.e. we replaced a regression on $X^{m_0,m_1}_{s_j,\overline{s_j}}$ by a regression on $X^{m_0,m_1}_{\delta_{s_j}(s_j),\overline{s_j}}$. The adaptations of (2.16) yield similar approximations in (3.5), (3.6) and (3.17). In the same fashion, the adaptations of (2.17) yield similar approximations in (3.7) and (3.21).

For $s_j < s < s_k$, we summarize both situations saying that the regressed function $\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(\cdot)$ resulting form the projection of $\sum_{t_{l+1}>s_k}^{T} f(t_l, X^{m_0,m_1}_{s_j,t_l}, X^{m_0,m_1}_{s_j,t_{l+1}})$ on $X^{m_0,m_1}_{s_j,s_k}$ is approximated by the regressed function $\overline{h}^{m_0,\mathcal{S}}_{s_j,s}(\cdot)$ resulting form the projection of $\sum_{t_{l+1}>s_k}^{T} f(t_l, X^{m_0,m_1}_{s,t_l}, X^{m_0,m_1}_{s,t_{l+1}})$ on X^{m_0,m_1}_{s,s_k} and vice versa. This approximation is not absurd since one can straightforwardly see, from the Markov property, that

$$
\begin{aligned}
U_{s_k}(x) &= \mathbb{E}\left(\sum_{t_l\geq s_k}^{T} f(t_l, X^{m_0,m_1}_{s_j,t_l}, X^{m_0,m_1}_{s_j,t_{l+1}})\middle| X^{m_0,m_1}_{s_j,s_k}=x\right) \\
&= \mathbb{E}\left(\sum_{t_l\geq s_k}^{T} f(t_l, X^{m_0,m_1}_{s,t_l}, X^{m_0,m_1}_{s,t_{l+1}})\middle| X^{m_0,m_1}_{s,s_k}=x\right).
\end{aligned} \tag{4.4}
$$

To establish a control on $\mathbb{E}^{m_0}_{s_j}\left(\left[\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s,s_k}) - U_{s_k}(X^{m_0,m_1}_{s,s_k})\right]^2\right)$ and on $\mathbb{E}^{m_0}_{s}\left(\left[\overline{h}^{m_0,\mathcal{S}}_{s,s_k}(X^{m_0,m_1}_{s_j,s_k}) - U_{s_k}(X^{m_0,m_1}_{s_j,s_k})\right]^2\right)$, we define for $t_l \geq s$ two auxilary

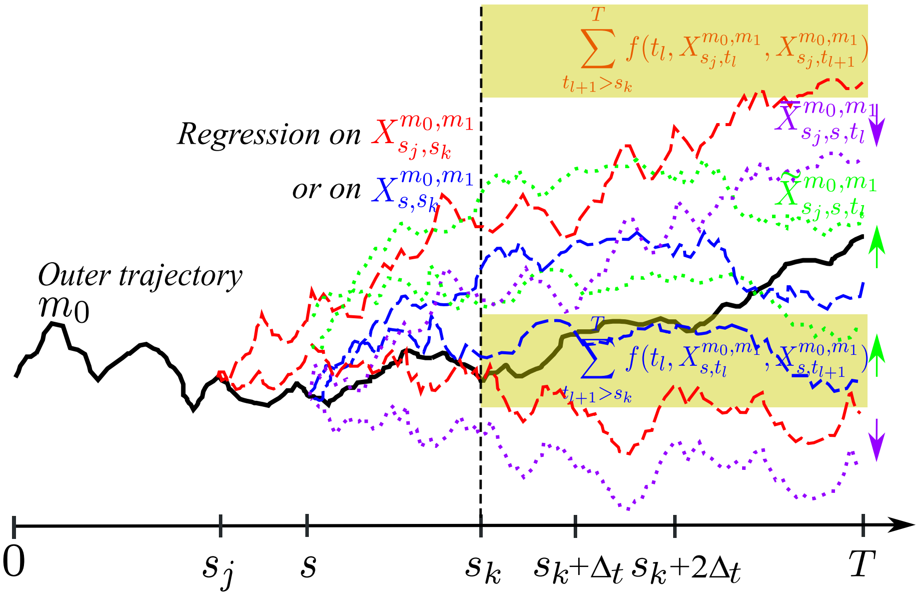

Figure 5: Comparing regression of $\displaystyle\sum_{t_{l+1}>s_k}^{T} f(t_l, X^{m_0,m_1}_{s_j,t_l}, X^{m_0,m_1}_{s_j,t_{l+1}})$ on $X^{m_0,m_1}_{s_j,s_k}$ and of $\displaystyle\sum_{t_{l+1}>s_k}^{T} f(t_l, X^{m_0,m_1}_{s,t_l}, X^{m_0,m_1}_{s,t_{l+1}})$ on X^{m_0,m_1}_{s,s_k} with $s \in \{s_j + \Delta_t, ..., s_k - \Delta_t\}$.

processes $\overline{X}$ and $\widetilde{X}$ as

$$\begin{cases} \overline{X}^{m_0,m_1}_{s_j,s,s} = X^{m_0}_s,\ \widetilde{X}^{m_0,m_1}_{s_j,s,s} = X^{m_0,m_1}_{s_j,s} \text{ and for } t_l = s+\Delta_t, ..., T \\ \overline{X}^{m_0,m_1}_{s_j,s,t_l} = \mathcal{E}_{t_{l-1}}(\mathcal{E}_{t_{l-2}}(...\mathcal{E}_s(X^{m_0}_s, \xi^{m_0,m_1}_{s_j,s+\Delta_t}), ...\xi^{m_0,m_1}_{s_j,t_{l-1}}), \xi^{m_0,m_1}_{s_j,t_l}) \\ \widetilde{X}^{m_0,m_1}_{s_j,s,t_l} = \mathcal{E}_{t_{l-1}}(\mathcal{E}_{t_{l-2}}(...\mathcal{E}_s(X^{m_0,m_1}_{s_j,s}, \xi^{m_0,m_1}_{s,s+\Delta_t}), ...\xi^{m_0,m_1}_{s,t_{l-1}}), \xi^{m_0,m_1}_{s,t_l}). \end{cases} \tag{4.5}$$

where $\mathcal{E}$ is given in (4.1). We remind that $E^{m_0}_{s_j}$ and $\mathbb{E}^{m_0}_s$ are the conditional expectations knowing the trajectory of X^{m_0} starting respectively from $X^{m_0}_{s_j}$ and from $X^{m_0}_s$.

As shown on Figure 5 for $t_l > s_k$, $\overline{X}^{m_0,m_1}_{s_j,s,t_l}$ is defined using $X^{m_0,m_1}_{s,s} = X^{m_0}_s$ and increments from the process $X^{m_0,m_1}_{s_j,t_l}$, in contrast to $\widetilde{X}^{m_0,m_1}_{s_j,s,t_l}$ defined using $X^{m_0,m_1}_{s_j,s}$ and increments from the process X^{m_0,m_1}_{s,t_l}. Proposition 4.3 provides a strong formulation of a possible compromise between two error terms on the right of each inequality (4.6) and (4.7).

Proposition 4.3. *For any $t \in \{0, \frac{T}{2^L}, ..., T\}$, 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*

$\mathcal{K}^{m_0}_{2,s_j,s_k} = [U_{s_k}]^2_{Lip} + \mathbb{E}^{m_0}_{s_j}(|H^{m_0,\mathcal{S}}_{s_j,s_k}|^2_{d'_1})$ *and* $\mathcal{K}^{m_0}_{2,s,s_k} = [U_{s_k}]^2_{Lip} + \mathbb{E}^{m_0}_{s}(|H^{m_0,\mathcal{S}}_{s,s_k}|^2_{d'_1})$ *where* $|\cdot|_{d'_1}$ *is the Euclidean norm on* $\mathbb{R}^{d'_1}$*, then*

$$\begin{aligned}\mathbb{E}^{m_0}_s\Big(\Big[\overline{h}^{m_0,\mathcal{S}}_{s,s_k}(X^{m_0,m_1}_{s_j,s_k}) - U_{s_k}(X^{m_0,m_1}_{s_j,s_k})\Big]^2\Big) \le{}& \mathbb{E}^{m_0}_s\Big(\Big[\overline{h}^{m_0,\mathcal{S}}_{s,s_k}(X^{m_0,m_1}_{s,s_k}) - U_{s_k}(X^{m_0,m_1}_{s,s_k})\Big]^2\Big)\\ &+\mathcal{K}^{m_0}_{2,s,s_k}\mathbb{E}^{m_0}_s\Big(\Big|\widetilde{X}^{m_0,m_1}_{s_j,s,s_k} - X^{m_0,m_1}_{s,s_k}\Big|^2_{d'_1}\Big)\end{aligned} \tag{4.6}$$

and

$$\begin{aligned}\mathbb{E}^{m_0}_{s_j}\Big(\Big[\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s,s_k}) - U_{s_k}(X^{m_0,m_1}_{s,s_k})\Big]^2\Big) \le{}& \mathbb{E}^{m_0}_{s_j}\Big(\Big[\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k}) - U_{s_k}(X^{m_0,m_1}_{s_j,s_k})\Big]^2\Big)\\ &+\mathcal{K}^{m_0}_{2,s_j,s_k}\mathbb{E}^{m_0}_{s_j}\Big(\Big|\overline{X}^{m_0,m_1}_{s_j,s,s_k} - X^{m_0,m_1}_{s_j,s_k}\Big|^2_{d'_1}\Big).\end{aligned} \tag{4.7}$$

Proof. As we simulate several independent copies of X^{m_0,m_1} (cf. the paragraph under Remark 2.1), we make sure that the approximations $\overline{h}$ are independent from X^{m_0,m_1}, from $\overline{X}^{m_0,m_1}$ and from $\widetilde{X}^{m_0,m_1}$ conditionally on X^{m_0}. Moreover, from definition (4.5), $(\overline{X}^{m_0,m_1}_{s_j,s,t_l})_{t_l\ge s}$ has the same law as $(X^{m_0,m_1}_{s,t_l})_{t_l\ge s}$ and $(\widetilde{X}^{m_0,m_1}_{s_j,s,t_l})_{t_l\ge s}$ has the same law as $(X^{m_0,m_1}_{s_j,t_l})_{t_l\ge s}$. Then one can write the following

$$\begin{aligned}\mathbb{E}^{m_0}_s\Big(\Big[\overline{h}^{m_0,\mathcal{S}}_{s,s_k}(X^{m_0,m_1}_{s_j,s_k}) - U_{s_k}(X^{m_0,m_1}_{s_j,s_k})\Big]^2\Big) ={}& \mathbb{E}^{m_0}_s\Big(\Big[\overline{h}^{m_0,\mathcal{S}}_{s,s_k}(\widetilde{X}^{m_0,m_1}_{s_j,s,s_k}) - U_{s_k}(\widetilde{X}^{m_0,m_1}_{s_j,s,s_k})\Big]^2\Big)\\ \le{}& \mathbb{E}^{m_0}_s\Big(\Big[\overline{h}^{m_0,\mathcal{S}}_{s,s_k}(\widetilde{X}^{m_0,m_1}_{s_j,s,s_k}) - \overline{h}^{m_0,\mathcal{S}}_{s,s_k}(X^{m_0,m_1}_{s,s_k})\Big]^2\Big)\\ &+\mathbb{E}^{m_0}_s\Big(\Big[\overline{h}^{m_0,\mathcal{S}}_{s,s_k}(X^{m_0,m_1}_{s,s_k}) - U_{s_k}(X^{m_0,m_1}_{s,s_k})\Big]^2\Big)\\ &+\mathbb{E}^{m_0}_s\Big(\Big[U_{s_k}(X^{m_0,m_1}_{s,s_k}) - U_{s_k}(\widetilde{X}^{m_0,m_1}_{s_j,s,s_k})\Big]^2\Big)\end{aligned} \tag{4.8}$$

as well as

$$\begin{aligned}\mathbb{E}^{m_0}_{s_j}\Big(\Big[\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s,s_k}) - U_{s_k}(X^{m_0,m_1}_{s,s_k})\Big]^2\Big) ={}& \mathbb{E}^{m_0}_{s_j}\Big(\Big[\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(\overline{X}^{m_0,m_1}_{s_j,s,s_k}) - U_{s_k}(\overline{X}^{m_0,m_1}_{s_j,s,s_k})\Big]^2\Big)\\ \le{}& \mathbb{E}^{m_0}_{s_j}\Big(\Big[\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(\overline{X}^{m_0,m_1}_{s_j,s,s_k}) - \overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k})\Big]^2\Big)\\ &+\mathbb{E}^{m_0}_{s_j}\Big(\Big[\overline{h}^{m_0,\mathcal{S}}_{s_j,s_k}(X^{m_0,m_1}_{s_j,s_k}) - U_{s_k}(X^{m_0,m_1}_{s_j,s_k})\Big]^2\Big)\\ &+\mathbb{E}^{m_0}_{s_j}\Big(\Big[U_{s_k}(X^{m_0,m_1}_{s_j,s_k}) - U_{s_k}(\overline{X}^{m_0,m_1}_{s_j,s,s_k})\Big]^2\Big)\end{aligned} \tag{4.9}$$

which yield (4.6) and (4.7). □

Proposition 4.3 requires Lipschitz property of U which is fulfilled if f is Lipschitz. Using similar steps to the one presented in [28], we show in Lemma 4.1 this Lipschitz property for (ODP). Using similar arguments, one can also show this property for (Snl) if g is also Lipschitz. We point out that another option is the one based on differentiability assumptions as in [26].

Consider the following extension of (4.3) with a driver f that depends also on X

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

where $\theta_{\delta(s_k)} \sim \mathcal{N}(0, I_{d_1})$. Replacing s_k by k and using Markov property, $\hat{Y}_{s_k} = y_k(X_{s_k})$ and $\hat{Z}_{s_k} = z_k(X_{s_k})$ (cf [17]) with

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

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 4.1 are also assumed Lipschitz continuous in x uniformly with respect to $t \in [0,T]$ with Lipschitz coefficients denoted $[b]_{Lip}$ and $[\sigma]_{Lip}$. Assume that $n \geqslant n_0$ (in order to provide sharper constants depending on $n_0 \geqslant 1$).*

Then for every $k \in \{0, ..., n-1\}$, y_k is $[y_k]_{Lip}$-Lipschitz continuous with

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

where $C' = [b]_{Lip} + \frac{1}{2}\left([\sigma]^2_{Lip} + \frac{T}{n_0}[b]^2_{Lip}\right) + [f]_{Lip}\left(1 + \sqrt{d_1}\sqrt{\frac{n_0}{T}}\right)$.

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

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

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

$$[y_k]_{Lip} \leqslant \left([f_T]_{Lip} + [f]_{Lip}\frac{T}{n_0}C'\right)e^{C'(T-t_k)}$$

and

$$[z_k]_{Lip} \leqslant \sqrt{d_1}\sqrt{\frac{n_0}{T}}\left([f_T]_{Lip} - C'\,[f]_{Lip}\right)e^{C'(T-t_k)}e^{\frac{T}{n_0}C_{b,\sigma,T}}.$$

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

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

and

$$z_k(x) - z_k(x') = \frac{1}{\sqrt{\Delta_k}}\mathbb{E}\left((y_{k+1}(\mathcal{E}_k(x,\theta_{k+1})) - y_{k+1}(\mathcal{E}_k(x',\theta_{k+1})))\theta_{k+1}\right).$$

We denote

$$\begin{aligned}
A_{x,x'} &= \frac{\left[f_k(x, y_{k+1}(\mathcal{E}_k(x,\theta_{k+1})), z_k(x)) - f_k(x', y_{k+1}(\mathcal{E}_k(x,\theta_{k+1}), z_k(x))\right]1_{\{\mathcal{A}_{x,x'}\neq 0\}}}{\mid x - x' \mid_{d_1}}\\
B_{x,x'} &= \frac{\left[f_k(x', y_{k+1}(\mathcal{E}_k(x,\theta_{k+1}), z_k(x)) - f_k(x', y_{k+1}(\mathcal{E}_k(x',\theta_{k+1}), z_k(x))\right]1_{\{\mathcal{B}_{x,x'}\neq 0\}}}{y_{k+1}(\mathcal{E}_k(x,\theta_{k+1})) - y_{k+1}(\mathcal{E}_k(x',\theta_{k+1}))}\\
C_{x,x'} &= \frac{\left[f_k(x, y_{k+1}(\mathcal{E}_k(x,\theta_{k+1}), z_k(x)) - f_k(x', y_{k+1}(\mathcal{E}_k(x,\theta_{k+1}), z_k(x))\right]1_{\{\mathcal{C}_{x,x'}\neq 0\}}}{\mid z_k(x) - z_k(x') \mid_{d_1}}
\end{aligned}$$

where $\mid x \mid_{d_1} = \sqrt{x_1^2 + ... + x_{d_1}^2}$, $\mathcal{A}_{x,x} = \mid x - x' \mid_{d_1}$, $\mathcal{B}_{x,x} = y_{k+1}(\mathcal{E}_k(x,\theta_{k+1})) - y_{k+1}(\mathcal{E}_k(x',\theta_{k+1}))$ and $\mathcal{C}_{x,x} = \mid z_k(x) - z_k(x') \mid_{d_1}$.

We have

$$\begin{aligned}
y_k(x) - y_k(x') &= \mathbb{E}\left[(y_{k+1}(\mathcal{E}_k(x,\theta_{k+1})) - y_{k+1}(\mathcal{E}_k(x',\theta_{k+1})))\,(1 + B_{x,x'}\Delta_k)\right]\\
&+\Delta_k\mathbb{E}\left[\frac{1}{\sqrt{\Delta_k}}C_{x,x'}\mathbb{E}\left(\mid (y_{k+1}(\mathcal{E}_k(x,\theta_{k+1})) - y_{k+1}(\mathcal{E}_k(x',\theta_{k+1}))).\theta_{k+1} \mid_{d_1}\right)\right]\\
&+\Delta_k\mathbb{E}\left[A_{x,x'} \mid x - x' \mid_{d_1}\right].
\end{aligned}$$

Using the Lipschitz property of f_k and y_{k+1} we have

$$\begin{aligned}
\mid y_k(x) - y_k(x') \mid &\leqslant [y_{k+1}]_{Lip}(1 + [f]_{Lip}\Delta_k)\mathbb{E}\left[\mid \mathcal{E}_k(x,\theta_{k+1}) - \mathcal{E}_k(x',\theta_{k+1}) \mid\right]\\
&+\Delta_k[f]_{Lip}\frac{1}{\sqrt{\Delta_k}}[y_{k+1}]_{Lip}\sqrt{\sum_{i=1}^{d_1}\mathbb{E}\left[(\mathcal{E}_k(x,\theta_{k+1}) - \mathcal{E}_k(x,\theta_{k+1}))\theta_{k+1}^i\right]^2}\\
&+\Delta_k\,[f]_{Lip} \mid x - x' \mid_{d_1}.
\end{aligned}$$

By applying Cauchy-Schwarz's inequality and knowing that $\mathbb{E}((\theta^i_{k+1})^2) = 1$, for $i \in \{1, ..., d_1\}$, we have

$$\begin{aligned}
\mid y_k(x) - y_k(x') \mid &\leqslant [y_{k+1}]_{Lip} \left(1 + \Delta_k [f]_{Lip} + \Delta_k [f]_{Lip} \tfrac{1}{\sqrt{\Delta_k}} \sqrt{d_1}\right) \\
&\times \sqrt{\underbrace{\mathbb{E}\left[(\mathcal{E}_k(x, \theta_{k+1}) - \mathcal{E}_k(x', \theta_{k+1}))^2\right]}_{D_{x,x'}}} \\
&+ \Delta_k \, [f]_{Lip} \mid x - x' \mid_{d_1} .
\end{aligned}$$

As $b_k(.)$ and $\sigma_k(.)$ are Lipschitz, by elementary computations already carried out in [4, 28], we have

$$\begin{aligned}
D_{x,x'} &= \mathbb{E}\left([\mathcal{E}_k(x, \theta_{k+1}) - \mathcal{E}_k(x', \theta_{k+1})]^2\right) \\
&= \mathbb{E}\left(\left[x - x' + \Delta_k \, [b_k(x) - b_k(x')] + \sqrt{\Delta_k} \theta_{k+1} \, [\sigma_k(x) - \sigma_k(x')]\right]^2\right) \\
&\leqslant \left(1 + \Delta_k (2 \, [b]_{Lip} + [\sigma]^2_{Lip} + \Delta_k \, [b]^2_{Lip})\right) \mid x - x' \mid^2_{d_1} \\
&\leqslant (1 + \Delta_k C_{b,\sigma,T})^2 \mid x - x' \mid^2_{d_1} \\
&\leqslant e^{2 \Delta_k C_{b,\sigma,T}} \mid x - x' \mid^2_{d_1}
\end{aligned}$$

where $C_{b,\sigma,T}$ can be taken equal to $[b]_{Lip} + \frac{1}{2}\left([\sigma]^2_{Lip} + \frac{T}{n_0} \, [b]^2_{Lip}\right)$.
This brings us to,

$$\begin{aligned}
\mid y_k(x) - y_k(x') \mid &\leqslant \left(\Delta_k \, [f]_{Lip} + [y_{k+1}]_{Lip} \left(1 + \Delta_k [f]_{Lip} \left(1 + \sqrt{\frac{d_1}{\Delta_k}}\right)\right) e^{\Delta_k C_{b,\sigma,T}}\right) \mid x - x' \mid_{d_1} \\
&\leqslant \left(\Delta_k \, [f]_{Lip} + [y_{k+1}]_{Lip} e^{\Delta_k (C_{b,\sigma,T} + C_{f,d_1,T})}\right) \mid x - x' \mid_{d_1}
\end{aligned}$$

where $C_{f,d_1,T}$ is taken equal to $[f]_{Lip} \left(1 + \sqrt{d_1} \sqrt{\frac{n_0}{T}}\right)$.

We conclude that y_k is Lipschitz continuous with coefficient $[y_k]_{Lip}$ satisfying

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

where $C' = C_{b,\sigma,T} + C_{f,d_1,T}$.
Moreover, using that $\theta_{k+1} \sim \mathcal{N}(0, I_{d_1})$, combined with Cauchy-Schwarz's

inequality and Lipschitz property we get

$$
\begin{aligned}
\mid z_k(x) - z_k(x') \mid_{d_1} &\leqslant \frac{1}{\sqrt{\Delta_k}} [y_{k+1}]_{Lip} \sqrt{\sum_{i=1}^{d_1} \mathbb{E}\left[(\mathcal{E}_k(x, \theta_{k+1}) - \mathcal{E}_k(x, \theta_{k+1}))\theta_{k+1}^i\right]^2} \\
&\leqslant \frac{1}{\sqrt{\Delta_k}} \left[y_{k+1}\right]_{Lip} \sqrt{\sum_{i=1}^{d_1} \mathbb{E}\left[(\mathcal{E}_k(x, \theta_{k+1}) - \mathcal{E}_k(x, \theta_{k+1}))\right]^2 \mathbb{E}\left[\theta_{k+1}^i\right]^2} \\
&\leqslant \frac{1}{\sqrt{\Delta_k}} \left[y_{k+1}\right]_{Lip} \sqrt{d_1} \sqrt{D_{x,x'}} \\
&\leqslant \frac{1}{\sqrt{\Delta_k}} \left[y_{k+1}\right]_{Lip} \sqrt{d_1} e^{\Delta_k C_{b,\sigma,T}} \mid x - x' \mid_{d_1} .
\end{aligned}
$$

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

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

Assuming homogeneous time increment $\Delta_k = h$, we have

$$
e^{C'kh} \left[y_k\right]_{Lip} \leqslant \left[y_{k+1}\right]_{Lip} e^{C'(k+1)h} + e^{C'kh} \left[f\right]_{Lip} h.
$$

which yields

$$
\begin{aligned}
e^{C'kh} \left[y_k\right]_{Lip} &\leqslant \left[f_T\right]_{Lip} e^{C'nh} + \left[f\right]_{Lip} h \sum_{l=k}^{n-1} e^{C'lh} \\
&\leqslant \left[f_T\right]_{Lip} e^{C'T} + \left[f\right]_{Lip} h \frac{e^{C'T} - e^{C'kh}}{e^{C'T} - 1} \\
&\leqslant \left[f_T\right]_{Lip} e^{C'T} + \left[f\right]_{Lip} h \frac{e^{C'T}}{e^{C'T} - 1} \\
&\leqslant \left[f_T\right]_{Lip} e^{C'T} + \left[f\right]_{Lip} hC' e^{C'T}.
\end{aligned}
$$

Finally we have

$$
\left[y_k\right]_{Lip} \leqslant \left[f_T\right]_{Lip} e^{C'(T-s_k)} + \left[f\right]_{Lip} hC' e^{C'(T-s_k)}. \tag{4.10}
$$

and

$$
\left[z_k\right]_{Lip} \leqslant \frac{1}{\sqrt{h}} (\left[f_T\right]_{Lip} - C' \left[f\right]_{Lip}) e^{C'(T-t_k)} e^{hC_{b,\sigma,T}} \sqrt{d_1}.
$$

□

5 Some numerical results

In this section we test the above conditional MC learning procedure on various examples including BSDE, American option and risk measure. The fact that the driver f depends also on X is not a burden to the use of our method. 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 [34] for an introduction to CUDA programming.

5.1 Allen-Cahn equation

We consider (ODP) simulation as presented in Section 3.2, we use the following functions

$$f(t,x,y,z) = y - y^3,$$
$$f(T,x) = \left[2 + \frac{2}{5}|x|^2_{d_1}\right]$$

and

$$\mathcal{E}_{t_k}(x,w) = x + \sqrt{2}w, \quad X_{t_0} = 0.$$

We would like to approximate the solution $u(t,x)$ of the Allen-Cahn PDE defined as follows, $u(T,x) = f(T,x)$,

$$\frac{\partial u}{\partial t}(t,x) + u(t,x) - [u(t,x)]^3 + (\Delta_x u)(t,x) = 0. \tag{5.1}$$

A benchmark approximation $u_b(0,x)$ for the solution $u(0,x)$ of the PDE (5.1) is given in [Section 4.2; [12]].

Table 1 shows the solution $u(0,0)$ of equation (5.1), calculated by learned and simulated expression, with respect to the number of inner trajectories M_1. The benchmark solution $u_b(0,0)$ is equal to 0.0528 for $T = 0.3$ and $d_1 = 100$. The standard deviation of each expression and the runtime in seconds are also given. We reduce the bias by increasing the number of inner trajectories. Table 1 shows that a relative small number of outer and inner trajectories is sufficient to observe a small variance and bias for both options. In fact, we show that the standard deviation is already acceptable for $M_0 = 2^4$ outer trajectories and the bias is acceptable for $M_1 = 2^6$ inner trajectories with an execution time of 56 millisecond.

Table 1: Numerical simulations for PDE (5.1): $T = 0.3$, $M_0 = 2^4$, $d_1 = 100$, $L = 4$; [Benchmarck solution] $u_b(0,0) = 0.0528$.

M_1	Learned		Simulated		Runtime in sec. (10^{-3})
	Y_0^{learn}	std	Y_0^{sim}	std	
2^4	0.0454	($\pm$ 0.0093)	0.0455	($\pm$ 0.0073)	13
2^5	0.0513	($\pm$ 0.0011)	0.0517	($\pm$ 0.0008)	23
2^6	0.0523	($\pm$ 0.0004)	0.0518	($\pm$ 0.0006)	56
2^7	0.0526	($\pm$ 0.0003)	0.0515	($\pm$ 0.0001)	119
2^8	0.0525	($\pm$ 0.0002)	0.0517	($\pm$ 0.0002)	227
2^9	0.0527	($\pm$ 0.0002)	0.0515	($\pm$ 0.0002)	414

Table 2 shows the solution $u(0,0)$ of equation (5.1), calculated by learned and simulated expression, with respect to the number of inner trajectories M_1, for a long time horizon ($T = 1$). The benchmark solution is equal to 0.0338 for $T = 1$, $d_1 = 100$. To achieve a similar level of variance and bias we need more outer and inner trajectories than in Table 1. In fact for $M_0 = 2^5$ of outer trajectories and $M_1 = 2^6$ of inner trajectories we obtained an acceptable bias and standard deviation in 4 seconds.

Table 2: Numerical simulations for PDE (5.1): $T = 1$, $d_1 = 100$, $M_0 = 2^5$, $L = 6$; [Benchmarck solution] $u_b(0,0) = 0.0338$.

M_1	Learned		Simulated		Runtime in sec.
	Y_0^{learn}	std	Y_0^{sim}	std	
2^5	0.0345	($\pm$ 0.0008)	0.0350	($\pm$, 0.0021)	2
2^6	0.0333	($\pm$ 0.0003)	0.0326	($\pm$ 0.0004)	4
2^7	0.0334	($\pm$ 0.0002)	0.0330	($\pm$ 0.0003)	7
2^8	0.0336	($\pm$ 0.0002)	0.0332	($\pm$ 0.0002)	12
2^9	0.0336	($\pm$ 0.0001)	0.0331	($\pm$ 0.0001)	27

5.2 Multidimensional Burgers-type PDEs with explicit solution

We assume the (ODP) setting presented in Section 3.2, we use the following functions

$$f(t,x,y,z) = \left(y - \frac{2+d_1}{2d_1}\right)\left(\sum_{i=1}^{d_1} z_i\right),$$

$$f(T,x) = \frac{\exp\left(T + \frac{1}{d_1}\sum_{i=1}^{d_1} x_i\right)}{1+\exp\left(T + \frac{1}{d_1}\sum_{i=1}^{d_1} x_i\right)}$$

and

$$\mathcal{E}_{t_k}(x, w) = x + \frac{d_1}{\sqrt{2}} w, \quad X_{t_0} = 0.$$

We simulate the solution $u(t, x)$ of the multidimensional Burgers-type PDE (cf [9], Example 4.6) defined as follows, $u(T, x) = f(T, x)$,

$$\frac{\partial u}{\partial t}(t, x) + \frac{{d_1}^2}{2} (\Delta_x u)(t, x) + \left(u(t, x) - \frac{2 + d_1}{2 d_1} \right) \left(d_1 \sum_{i=1}^{d_1} \frac{\partial u}{\partial x_i}(t, x) \right) = 0. \tag{5.2}$$

PDE (5.2) admits an explicit solution, we refer the reader to [Lemma 4.3, [12]] for more details. The value of the solution $u(0, 0)$ is 0.5000 for $T = 0.2$ and $d_1 = 100$.

Table 3 shows the solution $u(0, 0)$ of the equation (5.2), calculated by learned and simulated expression, with respect to the number of inner trajectories M_1. The approximation of the standard deviation of each expression and the runtime in seconds are also given. We show that the standard deviation of both results should be reduced by increasing the number of outer trajectories.

Table 3: Numerical simulations for PDE (5.2): $T = 0.2$, $d_1 = 100$, $M_0 = 2^6$, $L = 5$; [Explicit solution] $u(0, 0) = 0.5000$.

M_1	Learned		Simulated		Runtime in sec.
	Y_0^{learn}	std	Y_0^{sim}	std	
2^8	0.4785	(± 0.0428)	0.0517	(± 0.0431)	7
2^9	0.5113	(± 0.0450)	0.5108	(± 0.0450)	16
2^{10}	0.4966	(± 0.0448)	0.4912	(± 0.0447)	27
2^{11}	0.5022	(± 0.0421)	0.5012	(± 0.0435)	49

In Table 4 we show the computed solution of the equation (5.2), calculated by learned and simulated expression with respect to the number of outer trajectories M_0. The standard deviation of each expression and the runtime in seconds are also given. We reduce the standard deviation by increasing the number of outer trajectories.

Table 4: Numerical simulations for PDE (5.2): $T = 0.2$, $d_1 = 100$, $M_1 = 2^{11}$, $L = 5$; [Explicit solution] $u(0, 0) = 0.5000$.

M_0	Learned		Simulated		Runtime in sec.
	Y_0^{learn}	std	Y_0^{sim}	std	
2^5	0.4953	($\pm$ 0.0618)	0.4941	($\pm$ 0.0615)	24
2^6	0.5022	($\pm$ 0.0424)	0.501284	($\pm$ 0.0435)	49
2^7	0.5079	($\pm$ 0.0346)	0.5066	($\pm$ 0.0342)	103
2^8	0.5158	($\pm$ 0.0221)	0.5151	($\pm$ 0.0221)	194
2^9	0.5023	($\pm$ 0.0164)	0.5029	($\pm$ 0.0164)	408

5.3 Time-dependent reaction-diffusion-type example PDEs with oscillating explicit solutions

Let $\kappa = 0.6$, $\lambda = \frac{1}{\sqrt{d_1}}$, we use the following functions

$$f(t,x,y,z) = \min\left\{1, \left[y - \kappa - 1 - \sin\left(\lambda \sum_{i=1}^{d_1} x_i\right)\right]\right\},$$

$$f(T,x) = 1 + \kappa + \sin\left(\lambda \sum_{i=1}^{d_1} x_i\right)$$

and

$$\mathcal{E}_{t_k}(x,w) = x + w, \quad X_{t_0} = 0.$$

We simulate the solution $u(t,x)$ of the time dependent reaction-diffusion-type PDE (cf [18], Section 6.1) defined as follows, $u(T,x) = f(T,x)$,

$$\frac{\partial u}{\partial t}(t,x) + \min\left\{1, \left[y - \kappa - 1 - \sin\left(\lambda \sum_{i=1}^{d_1} x_i\right)\right]\right\} + \frac{1}{2}(\Delta_x u)(t,x) = 0. \quad (5.3)$$

The explicit solution of the PDE (5.3) is given in [Lemma 4.4; [12]].
Table 5 shows the approximated solution of the equation (5.3), calculated by learned and simulated expression, with respect to the number of inner trajectories M_1. The standard deviation of each expression, and the runtime in seconds are also given. The benchmark solution is equal to 1.6000 for $T = 1$, $d_1 = 100$.

Table 5: Numerical simulations for PDE (5.3): $T = 0.5$, $d_1 = 100$, $M_0 = 2^{10}$, $L = 3$; [Benchmark solution] $u_b(0,0) = 1.6000$.

M_1	Learned		Simulated		Runtime in sec. (10^{-3})
	Y_0^{learn}	std	Y_0^{sim}	std	
2^5	1.8197	$(\pm\ 0.0386)$	1.7587	$(\pm\ 0.0287)$	244
2^6	1.7125	$(\pm\ 0.0104)$	1.6799	$(\pm\ 0.0116)$	311
2^7	1.6605	$(\pm\ 0.0037)$	1.6376	$(\pm\ 0.0091)$	466
2^8	1.6458	$(\pm\ 0.0023)$	1.6290	$(\pm\ 0.0089)$	817
2^9	1.6439	$(\pm\ 0.0019)$	1.6283	$(\pm\ 0.0061)$	1526

5.4 A Hamilton-Jacobi-Bellman (HJB) equation

We assume here the driver to be equal to

$$f(t,x,y,z) = -|z|_{d_1}^2,$$

$$f(T,x) = ln\left(\left[1+|x|_{d_1}^2\right]\right)$$

and

$$\mathcal{E}_{t_k}(x,w) = x + \sqrt{2}w, \quad X_{t_0} = 0.$$

We calculate the solution $u(t,x)$ of the HJB equation (cf [10] Section 4.2) defined by $u(T,x) = f(T,x)$,

$$\frac{\partial u}{\partial t}(t,x) + (\Delta_x u)(t,x) - |(\nabla_x u)(t,x)|_{d_1}^2 = 0. \tag{5.4}$$

PDE (5.4) admits a benchmark solution. We refer the reader to [Lemma 4.2; [12]] for more details.

In Figure 6 we show the difference between $Y_{s_k} = \frac{1}{M_0}\sum_{m_0=1}^{M_0} \widetilde{y}^{m_0,\mathcal{S}^{i^*}}_{s_k,\overline{s_k}}$ and $\frac{1}{M_0}\sum_{m_0=1}^{M_0}\left(\widetilde{y}^{m_0,\mathcal{S}^{i^*}}_{\delta(s_k),\overline{\delta(s_k)}} + (\delta(s_k)-s_k)f(s_k,\widetilde{y}^{m_0,\mathcal{S}^{i^*}}_{\delta(s_k),\overline{\delta(s_k)}},\widetilde{z}^{m_0,\mathcal{S}^{i^*}}_{s_k,\overline{s_k}})\right)$ with respect to the discretization time steps. On the left, we perform the conditional MC procedure taking $\overline{s_k} = T$. On the right, we perform the procedure with the bias control presented in Section 2.3 , taking $\overline{s_k} = \left(s_k + \frac{3}{8}\right)\wedge T$ with $s_k \in \mathcal{S}^{i^*} = \{0, \frac{1}{8}, \frac{2}{8}, \frac{3}{8}, \frac{4}{8}, \frac{5}{8}, \frac{6}{8}, \frac{7}{8}, 1\}$. We show that the control allows to reduce the bias propagation.

Figure 7 shows the convergence of the learned and simulated expression to the benchmark value with respect to the number of inner trajectories. In particular, we observe that the both expressions converge to the benchmark solution with a small variance when $M_1 = 2^{17}$.

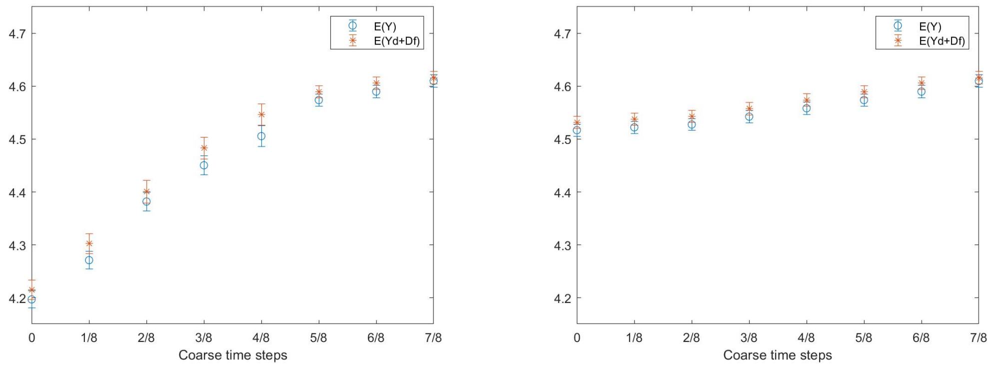

Figure 6: Y_{s_k} vs. $\dfrac{1}{M_0}\displaystyle\sum_{m_0=1}^{M_0}\Big(\widetilde{y}^{m_0,\mathcal{S}^{i^*}}_{\delta(s_k),\overline{\delta(s_k)}} + (\delta(s_k)-s_k)f(s_k,\widetilde{y}^{m_0,\mathcal{S}^{i^*}}_{\delta(s_k),\overline{\delta(s_k)}},\widetilde{z}^{m_0,\mathcal{S}^{i^*}}_{s_k,\overline{s_k}})\Big)$ [Left] $\overline{s_k}=T$ without bias control, [Right] $\overline{s_k}=\big(s_k+\frac{3}{8}\big)\wedge T$ with bias control: $T=1$, $d_1=100$, $M_0=2^7$, $M_1=2^{15}$, $L=3$.

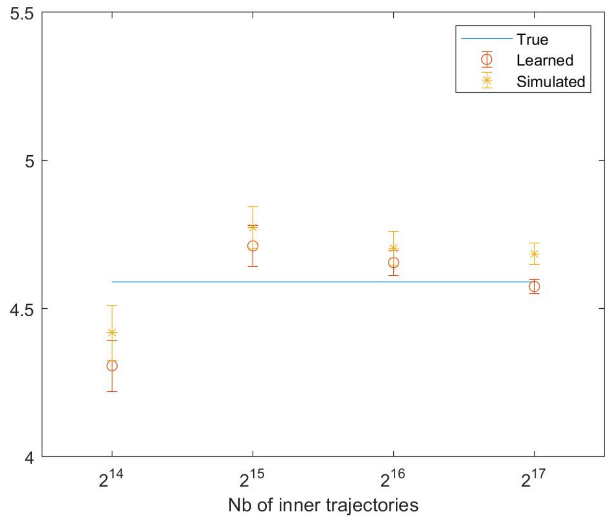

Figure 7: Numerical solution of PDE (5.4) calculated by learned and simulated expression: $T=1$, $d_1=100$, $M_0=2^7$, $L=3$.

5.5 Pricing of European financial derivatives with different interest rates for borrowing and lending

Assuming $\mu = 0.06$, $\sigma = 0.2$, $R^l = 0.04$ and $R^b = 0.06$, we introduce the following functions

$$f(t,x,y,z) = -R^l y - \frac{(\mu - R^l)}{\sigma} \sum_{i=1}^{d_1} z_i + (R^b - R^l) \max\{0, \frac{1}{\sigma}\sum_{i=1}^{d_1} z_i - y\},$$

$$f(T,x) = \max\{\max_{1\le i\le d_1} x_i - 120, 0\} - 2\max\{\max_{1\le i\le d_1} x_i - 150, 0\}$$

and

$$\mathcal{E}_{t_k}(x,w) = x\exp\left((\mu - \frac{\sigma^2}{2})\Delta_t + \sigma w\right), \quad X_{t_0} = 100.$$

Let u defined as the solution of the following PDE, $u(T,x) = f(T,x)$,

$$\begin{aligned} &\frac{\partial u}{\partial t}(t,x) + \frac{\sigma}{2}\sum_{i=1}^{d_1}|x_i|^2\frac{\partial^2 u}{\partial x_i^2}(t,x) \\ &+\max\{R^b\left(\sum_{i=1}^{d_1} x_i\left(\frac{\partial u}{\partial x_i}(t,x)\right) - u(t,x)\right), R^l\left(\sum_{i=1}^{d_1} x_i\left(\frac{\partial u}{\partial x_i}(t,x)\right) - u(t,x)\right)\} = 0. \end{aligned} \tag{5.5}$$

PDE (5.5) has a benchmark solution given in [Section 4.4; [12]]. This benchmark solution is equal to 21.299 for $T = 0.5$ and $d_1 = 100$.

Figure 8 shows the approximation of the solution of PDE (5.5), calculated by learned and simulated expression, with respect to the number of inner trajectories. We show that 2^7 outer trajectories and 2^{11} inner trajectories are sufficient to get an accurate approximation of the solution as the obtained values are in the corridor of the standard deviation of the benchmark solution. No bias cut is needed here. The runtime with 2^7 outer trajectories and 2^{11} inner trajectories is 53 seconds.

5.6 A PDE example with quadratically growing derivatives and an explicit solution

Assuming the (ODP) setting presented in Section 3.2, let $\alpha = 0.4$ and $\psi(t,x) = \sin\left(\left[T - t + |x|_{d_1}^2\right]^\alpha\right)$, we introduce the following functions,

$$f(t,x,y,z) = |z|_{d_1}^2 - |\nabla_x\psi(t,x)|_{d_1}^2 - \frac{\partial\psi}{\partial t}(t,x) - \frac{1}{2}(\Delta_x\psi)(t,x),$$

$$f(T,x) = \sin\left(|x|_{d_1}^{2\alpha}\right)$$

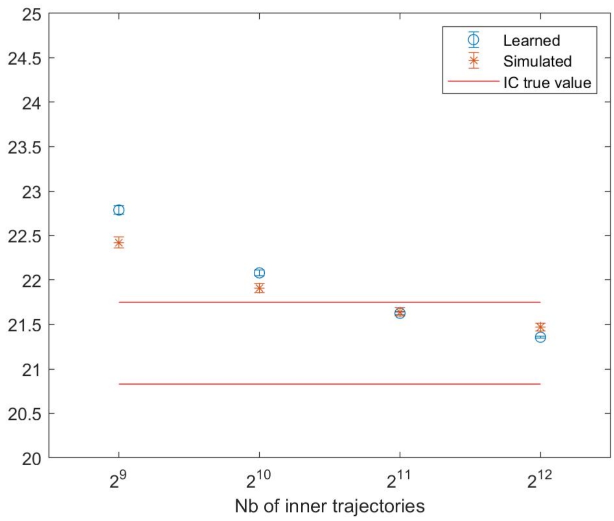

Figure 8: Numerical solution of PDE (5.5) calculated by learned and simulated expression: $T = 0.5$, $d_1 = 100$, $M_0 = 2^7$, $L = 2$.

and

$$\mathcal{E}_{t_k}(x, w) = x + w, \quad X_{t_0} = 0.$$

Let u defined as the solution of the following PDE, $u(T, x) = f(T, x)$,

$$\begin{aligned} \tfrac{\partial u}{\partial t}(t,x) + |\nabla_x u(t,x)|^2_{d_1} + \tfrac{1}{2}(\Delta_x u)(t,x) &= \tfrac{\partial \psi}{\partial t}(t,x) \\ &+ |\nabla_x \psi(t,x)|^2_{d_1} + \tfrac{1}{2}(\Delta_x \psi)(t,x). \end{aligned} \tag{5.6}$$

Straight use of Itô's Lemma shows that PDE (5.6) has an explicit solution $u(t, x) = \psi(t, x)$, we refer the reader to [Section 6.1; [18]] for more details.

Figure 9 is related to Propostion 4.3 that controls the error of regressions with different starting points. Here we prefered to show the distributions rather than the quadratic error which is small. On the left of Figure 9 we have the "Trained" value $Y^{m_0}_{\frac{249}{256}} = \widetilde{y}^{m_0,\mathcal{S}^{i^*}}_{\frac{249}{256},T}$ and the "Tested" $\displaystyle\frac{1}{M_1}\sum_{m_1=1}^{M_1}\left(\overline{y}^{m_0,\mathcal{S}^{i^*}}_{\frac{248}{256},\frac{250}{256}}\left(X^{m_0,m_1}_{\frac{249}{256},\frac{250}{256}}\right) + \Delta_s f\left(\frac{249}{256}, \overline{y}^{m_0,\mathcal{S}^{i^*}}_{\frac{248}{256},\frac{250}{256}}\left(X^{m_0,m_1}_{\frac{249}{256},\frac{250}{256}}\right), \widetilde{z}^{m_0,\mathcal{S}^{i^*}}_{\frac{249}{256},\frac{250}{256}}\right)\right)$ with $\mathcal{S}^{i^*} \in \{0, \frac{1}{256}, \frac{2}{256}, ..., 1\}$. On the right we show the "Trained" $Y^{m_0}_{\frac{253}{256}} = \widetilde{y}^{m_0,\mathcal{S}^{i^*}}_{\frac{253}{256},T}$ and the "Tested"

$$\frac{1}{M_1}\sum_{m_1=1}^{M_1}\left(\overline{y}^{m_0,\mathcal{S}^{i^*}}_{\frac{252}{256},\frac{254}{256}}\left(X^{m_0,m_1}_{\frac{253}{256},\frac{254}{256}}\right)+\Delta_s f\left(\frac{253}{256},\overline{y}^{m_0,\mathcal{S}^{i^*}}_{\frac{252}{256},\frac{254}{256}}\left(X^{m_0,m_1}_{\frac{253}{256},\frac{254}{256}}\right),\widetilde{z}^{m_0,\mathcal{S}^{i^*}}_{\frac{253}{256},\frac{254}{256}}\right)\right)$$

at a different time step 254/256. Figure 9 shows very similar distributions which strengthen the benefit of our trick.

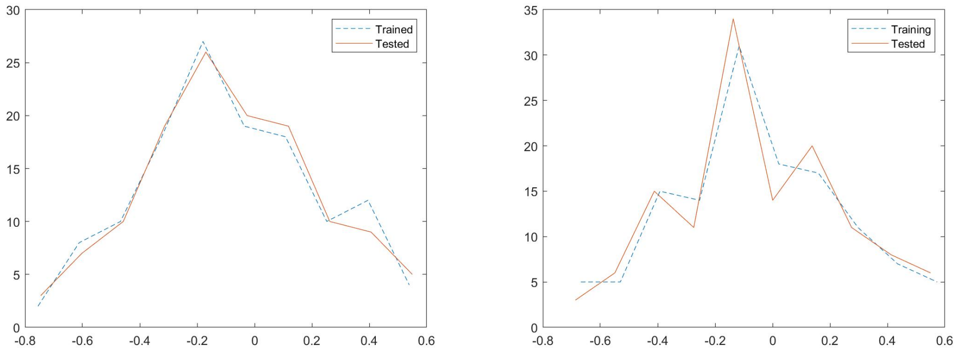

Figure 9: [Left] Distribution of $Y^{m_0}_{\frac{249}{256}}$ called "Trained" vs. its different starting point approximation called "Tested" [Right] $Y^{m_0}_{\frac{253}{256}}$ called "Trained" vs. its different starting point approximation called "Tested": $T = 1$, $d = 100$, $M_0 = 2^7$, $M_1 = 2^{12}$, $L = 8$.

Figure 10 shows the numerical solution of PDE (5.6), calculated by learned and simulated expression, with respect to different number of coarse time step. We show that $L = 8$ is sufficient to discretize the problem when the time horizon T is equal to 1. This convergence is achieved in 620 seconds of runtime.

5.7 American geometric put option

Given the (Snl) setting of Section 3.3 with a driver $f = 0$, we consider an American geometric put option with constant interest rate r and a payoff

$$g(x) = \left[K - \prod_{i=1}^{d_1}(x_i)^{1/d_1}\right]^+ \tag{5.7}$$

with an asset X given by $X^i_t = X^i_s \exp\left((r - \frac{\sigma^2}{2})(t-s) + \sigma(W^i_t - W^i_s)\right)$, $t > s$, $1 \le i \le d_1$, $r = \log(1.1)$, $\sigma = 0.4$, $K = X^i_0 = 100$ and $d_1 = 20$.

We approximate the price V_0 associated to payoff (5.7). We choose

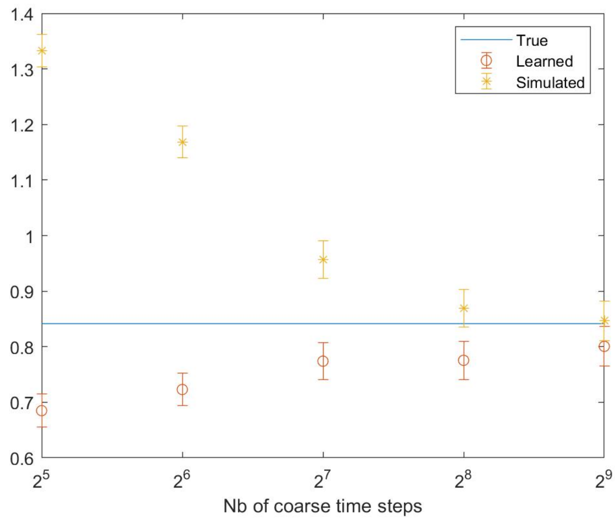

Figure 10: Numerical solution of PDE (5.6) calculated by learned and simulated expression with a bias control: $T = 1$, $d_1 = 100$, $M_0 = 2^7$, $M_1 = 2^7$.

the dimension $d_1 = 20$ to make sure that the variance of the problem is sufficiently large. We point out however that it works well for $d_1 = 100$.

In Table 6 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 $T = 2$. However, one needs to use variance adjustment that is important for events on the exercise frontier as well as bias control to cut the propagation of bias.

Table 6: Numerical simulations for American option (5.7) 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]	2.561 (± 0.035)	4.236 (± 0.042)	6.363 (± 0.054)
[BC]	2.493 (± 0.041)	3.734 (± 0.061)	5.130 (± 0.089)
[VA] + [BC]	2.291 (± 0.035)	2.890 (± 0.037)	3.961 (± 0.055)
Real Price	2.153	2.871	3.754

Figure 11 shows the difference between $\frac{1}{M_0}\sum_{m_0=1}^{M_0} e^{-r(\delta(s_k)-s_k)} V^{m_0}_{\delta(s_k)} = \frac{1}{M_0}\sum_{m_0=1}^{M_0} e^{-r(\delta(s_k)-s_k)} (\widetilde{v}^{m_0,\mathcal{S}^{i^*}}_{\delta(s_k),\overline{\delta(s_k)}})$ and $\frac{1}{M_0}\sum_{m_0=1}^{M_0} (\widetilde{w}^{m_0,\mathcal{S}^{i^*}}_{s_k,\overline{s_k}})$ with respect to the time discretization. On the left, we perform the conditional MC procedure by taking $\overline{s_k} = T$. On the right, we perform the procedure with the bias control presented in section 2.3 by taking $\overline{s_k} = \left(s_k + \frac{3}{8}\right) \wedge T$ with $s_k \in \mathcal{S}^{i^*} = \{0, \frac{1}{8}, \frac{2}{8}, \frac{3}{8}, \frac{4}{8}, \frac{5}{8}, \frac{6}{8}, \frac{7}{8}, 1\}$. We show that the control allows to reduce the bias propagation.

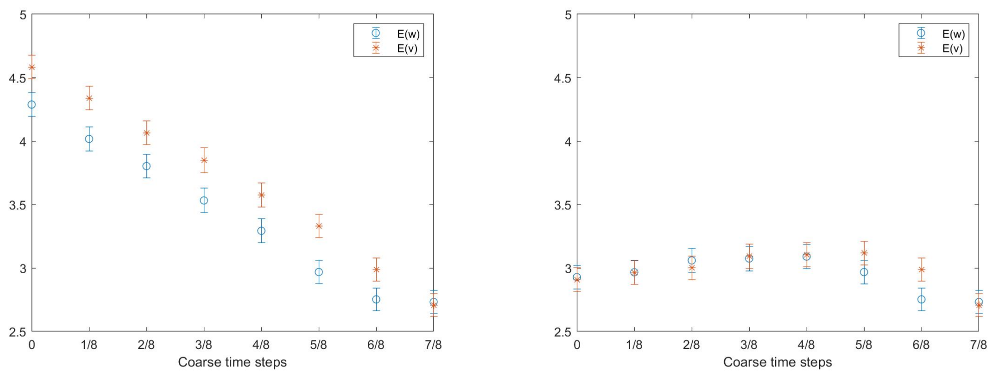

Figure 11: $\frac{1}{M_0}\sum_{m_0=1}^{M_0} e^{-r(\delta(s_k)-s_k)} V^{m_0}_{\delta(s_k)}$ vs. $\frac{1}{M_0}\sum_{m_0=1}^{M_0} (\widetilde{w}^{m_0,\mathcal{S}^{i^*}}_{s_k,\overline{s_k}})$; [Left] $\overline{s_k} = T$ without bias control [Right] $\overline{s_k} = \left(s_k + \frac{3}{8}\right) \wedge T$ with bias control: $d_1 = 20$, $M_0 = 2^{11}$, $M_1 = 2^{12}$, $T = 1$ and $L = 3$.

Figure 12 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.

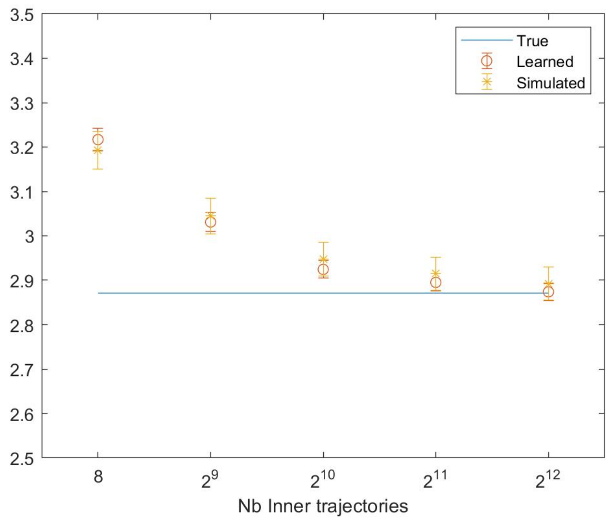

Figure 12: Numerical approximation of price V_0: $d_1 = 20$, $M_0 = 2^9$, $T = 1$, $L = 3$.

5.8 Initial Margin

Assume the setting presented in Section 3.2, we consider a portfolio of one hundred put options, the price V_{s_k} of the portfolio at time step s_k is given by

$$V_{s_k} = \sum_{i=0}^{d_1} e^{-(T-s_k)r} \mathbb{E}_{s_k} \left(\left[K - X_T^i \right]^+ \right) \tag{5.8}$$

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

We are interested to calculate the initial margin (IM) of this portfolio. IM 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

$$\mathrm{IM}_{s_k} = \mathbb{ES}^a_{s_k}(L_{s_k, s_k+\delta}) \tag{5.9}$$

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

$$L_{s_k,s_k+\delta} = V_{s_k+\delta} - V_{s_k},$$

and the expected shortfall $\mathbb{ES}$ is defined by

$$\mathbb{ES}^a_{s_k}(X) = \frac{1}{(1-a)} \int_a^1 \mathbb{V}a\mathbb{R}^\alpha_{s_k}(X) d\alpha.$$

The value-at-risk of some random variable $\mathbb{V}a\mathbb{R}^\alpha(X)$ conditionally to $\mathcal{F}_{s_k}$ is defined by

$$\mathbb{V}a\mathbb{R}^\alpha_{s_k}(X) = \inf\{x \in \mathbb{R} : \mathbb{P}(X \leq x \mid \mathcal{F}_{s_k}) \geq \alpha\}.$$

We considered the following parameters: $T = 1$, $d_1 = 100$, $K = X^i_0 = 100$, $r = 0.01$, $a = 99\%$, $NI = 32$ is the number of time step, $\delta = \frac{1}{32}$. A benchmark approximation of the IM is obtained using Black & Scholes formula for put options.

Figue 13 shows some distributions of the loss process. From top to bottom we show different time steps $s_k \in \{\frac{29}{32}, \frac{19}{32}, \frac{9}{32}\}$. On the left, we perform the procedure without variance adjustment and on the right we perform the variance adjustment introduced in section 2.3. We show that the variance adjustment is necessary to fit the benchmark distribution of the loss process. Figure 14 shows the initial margin distribution. From top to bottom we show different time steps $s_k \in \{\frac{29}{32}, \frac{19}{32}, \frac{9}{32}\}$. Although we are interested in distribution tails of the loss process we have a fairly good representation of the distribution of IM. Figure 15 shows at the top the mean of IM with respect to the time horizon of the portfolio and we show 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%.

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. and Graillat, S. (2017). Resolving small random symmetric linear systems on graphics processing units. *The Journal of Supercomputing.* **73(4)**, 1360–1386.

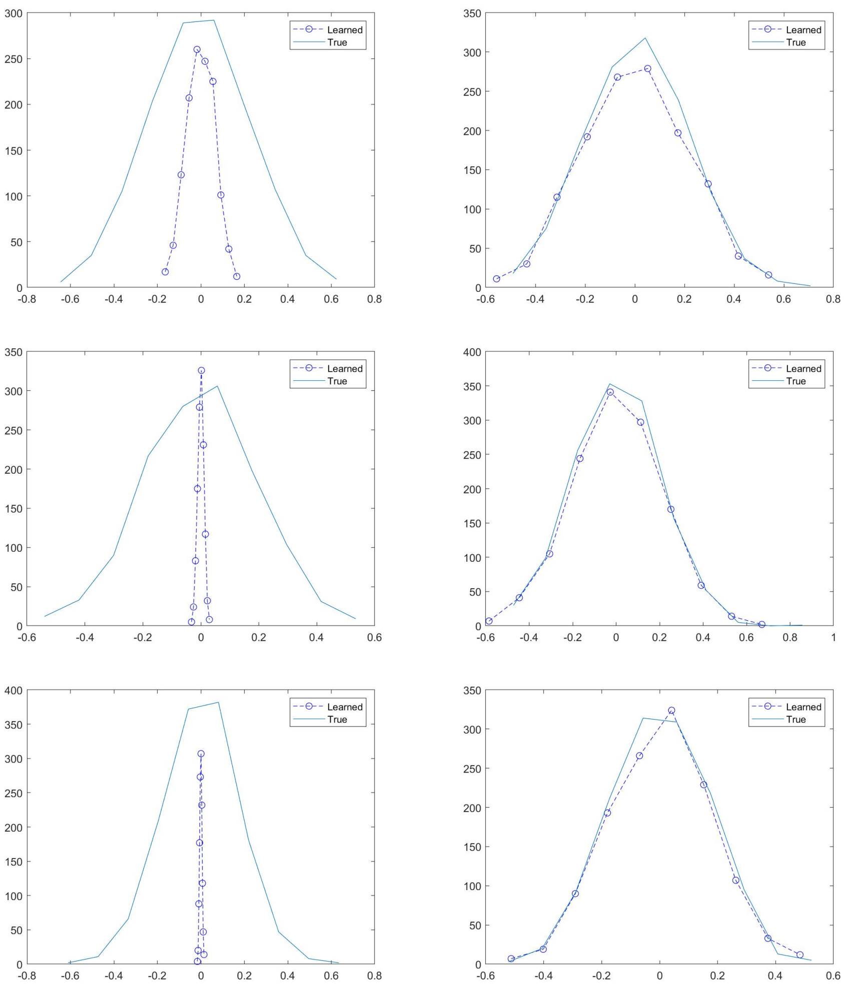

Figure 13: Numerical approximation of the loss distribution [Left] Without variance adjustment, [Right] With variance adjustment; [top to bottom] $s_k \in \{\frac{29}{32}, \frac{19}{32}, \frac{9}{32}\}$; $M_0 = 2^8$, $M_1 = 2^8 * 5$.

[3] 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.

[4] Bally, A. and Pagès, G. (2003). A quantization algorithm for solving discrete time multidimensional optimal stopping problems. *Bernoulli.* **6** 1003–1049.

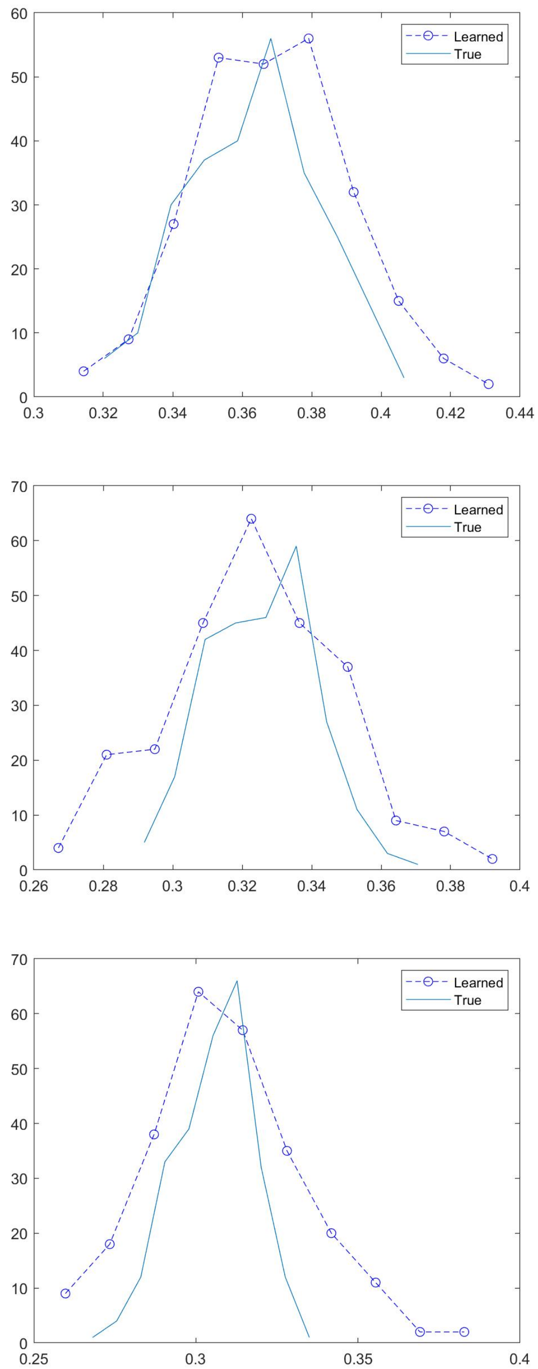

Figure 14: Numerical approximation of the IM distribution: [top to bottom] $s_k \in \{\frac{29}{32}, \frac{19}{32}, \frac{9}{32}\}$; $M_0 = 2^8,\ M_1 = 2^8 * 5$.

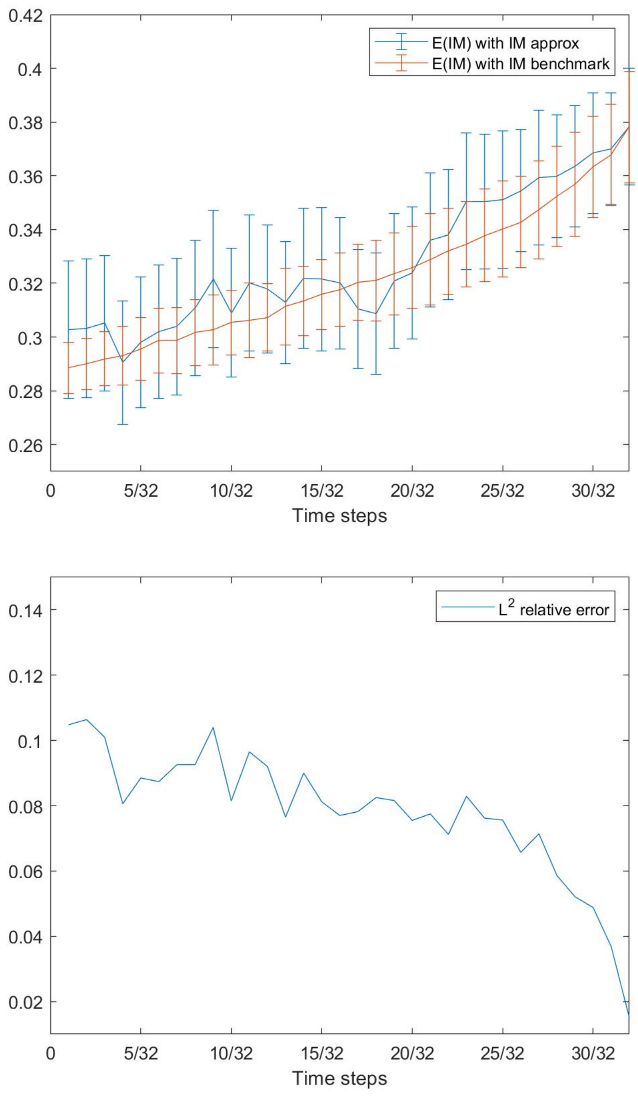

Figure 15: Initial Margin: [Top] mean of IM_{s_k}; [Bottom] L2 relative error.

[5] 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.

[6] Bouchard, B. and Touzi, N. (2004). Discrete time approximation and Monte Carlo simulation of backward stochastic differential equations. *Stochastic Processes and their Applications.* **111** 175–206.

[7] Broadie, M., Du, Y. and Moallemi, C. C. (2015). Risk estimation via regression. *Operations Research.* **63(5)** 979–1244.

[8] Bucklew, J. (2004). *Introduction to rare event simulation*, Springer Series in Statistics.

[9] Chassagneux, J.-F. (2014). Linear multistep schemes for BSDEs. *SIAM J. Numer. Anal.* **52(6)** 2815–2836.

[10] Chassagneux, J.-F., and Richou A. (2016). Numerical simulation of quadratic BSDEs. *Ann. Appl. Probab.* **26(1)** 262–304.

[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] 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.

[16] 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.

[17] 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.

[18] 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.

[19] Gordy, M. B. and Juneja, S. (2010). Nested Simulation in Portfolio Risk Measurement. *Management Science.* **56(10)** 1833–1848.

[20] Jourdain, B. and Lelong, J. (2009). Robust adaptive importance sampling for normal random vectors. *The Annals of Applied Probability.* **19(5)** 1687–1718.

[21] Lee, S.-H. (1998). *Monte Carlo Computation of Conditional Expectation Quantiles*, Ph.D. thesis, Stanford University.

[22] 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.

[23] 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.

[24] Lions, J.-L., Maday, Y. and Turinici, G. (2015). A "parareal" in time discretization of PDE's. *Comptes Rendus de l'Académie des Sciences.* **Série I. 332(7)** 661–668.

[25] 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.

[26] Newey, W. K. (1997). Convergence rates and asymptotic normality for series estimators. *Journal of Econometrics.* **79** 147–168.

[27] Pagès, G. (2002). *Numerical probability: an introduction with applications to finance*, Universitext, Springer.

[28] 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.

[29] PARDOUX, E. and PENG, S. (1990). Adapted solution of a backward stochastic differential equation. *Systems & Control Letters.* **14** 55–61.

[30] PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W. T. and FLANNERY, B. P. (2002). *Numerical Recipes in C++: The Art of Scientific Computing*, Cambridge University Press.

[31] SHAPIRO, A., DENTCHEVA, D. and RUSZCZYNSKI, A. (2009). *Lectures on stochastic programming: modeling and theory*, Society for Industrial and Applied Mathematics.

[32] 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.

[33] WHITE, H. (2001). *Asymptotic theory for econometricians.* Academic Press.

[34] NVIDIA (2017). *Cuda C PROGRAMMING GUIDE.*