Pricing path-dependent Bermudan options using Wiener chaos expansion: an embarrassingly parallel approach

Jérôme Lelong

Pricing path-dependent Bermudan options using Wiener chaos expansion: an embarrassingly parallel approach*

Jérôme Lelong †

July 7, 2020

Abstract

In this work, we propose a new policy iteration algorithm for pricing Bermudan options when the payoff process cannot be written as a function of a lifted Markov process. Our approach is based on a modification of the well-known Longstaff Schwartz algorithm, in which we basically replace the standard least square regression by a Wiener chaos expansion. Not only does it allow us to deal with a non Markovian setting, but it also breaks the bottleneck induced by the least square regression as the coefficients of the chaos expansion are given by scalar products on the $L^2(\Omega)$ space and can therefore be approximated by independent Monte Carlo computations. This key feature enables us to propose an embarrassingly parallel algorithm to efficiently handle non Markovian payoff.
Key words: path-dependent Bermudan options, optimal stopping, regression methods, high performance computing, Wiener chaos expansion.

AMS subject classification: 62L20, 62L15, 91G60, 65Y05, 60H07

1 Introduction

We fix some finite time horizon $T > 0$ and a filtered probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{0\leq t\leq T}, \mathbb{P})$, where $(\mathcal{F}_t)_{0\leq t\leq T}$ is supposed to be the natural augmented filtration of a $d-$dimensional Brownian motion B. On this space, we consider an adapted process $(S_t)_{0\leq t\leq T}$ with values in $\mathbb{R}^{d'}$ modeling a $d'-$dimensional underlying asset, with $d' \leq d$. The number of assets d' can be strictly smaller than the dimension d of the Brownian motion to encompass the case of stochastic volatility

*The High Performance Computations presented in this paper were performed using the Froggy platform of the CIMENT infrastructure (https://ciment.ujf-grenoble.fr), which is supported by the Rhône-Alpes region (GRANT CPER07_13 CIRA) and the Equip@Meso project (reference ANR-10-EQPX-29-01) of the programme Investissements d'Avenir supervised by the Agence Nationale pour la Recherche.

†Univ. Grenoble Alpes, CNRS, Grenoble INP, LJK, 38000 Grenoble, France.
email: jerome.lelong@univ-grenoble-alpes.fr

models or stochastic interest rates. We assume that $\mathbb{P}$ is a risk neutral measure. We consider a Bermudan option with exercising dates $0 = t_0 \leq T_1 < T_2 < \cdots < T_N = T$ and discounted payoff Z_{T_k} if exercised at time T_k. We assume that the discrete time payoff process $(Z_{T_k})_{0\leq k\leq N}$ is adapted to the filtration $(\mathcal{F}_{T_k})_{0\leq k\leq N}$ and satisfies $\max_{0\leq k\leq N} |Z_{T_k}| \in L^2$. This framework naturally encompasses the case of path-dependent options, ie. when the payoff process writes $\tilde{Z}_{T_k} = \phi_k((S_u; 0 \leq u \leq T_k))$ for any $0 \leq k \leq N$.

Standard arbitrage pricing theory defines the discounted value of the Bermudan option at times $(T_k)_{0\leq k\mathbb{N}}$ by

$$\begin{cases} U_{T_N} &= Z_{T_N} \\ U_{T_k} &= \max\left(Z_{T_k}, \mathbb{E}[U_{T_{k+1}}|\mathcal{F}_{T_k}]\right) \end{cases} \tag{1}$$

Solving this backward recursion known as the dynamic programming principle has been a challenging problem for years and various approaches have been proposed to approximate its solution. The real difficulty lies in the computation of the conditional expectation $\mathbb{E}[U_{T_{k+1}}|\mathcal{F}_{T_k}]$ at each time step of the recursion. If we were to classify the different approaches, we could say that there are regression based approaches (see Carriere [1996], Tsitsiklis and Roy [2001] and quantization approaches (see Bally and Pages [2003], Bronstein et al. [2013]). We refer to Bouchard and Warin [2012] and Pagès [2018] for a survey of the different techniques to price Bermudan options.

Among all the available algorithms to compute U using the dynamic programming principle, the one proposed by Longstaff and Schwartz [2001] has the favour of practitioners. Their approach is based on iteratively selecting the optimal policy. Let τ_k be the smallest optimal policy after time T_k, then

$$\begin{cases} \tau_N = T_N \\ \tau_k = T_k \mathbf{1}_{\left\{Z_{T_k} \geq \mathbb{E}[Z_{\tau_{k+1}}|\mathcal{F}_{T_k}]\right\}} + \tau_{k+1}\mathbf{1}_{\left\{Z_{T_k} < \mathbb{E}[Z_{\tau_{k+1}}|\mathcal{F}_{T_k}]\right\}}, \text{ for } 1 \leq k \leq N-1 \end{cases} \tag{2}$$

All these methods based on the dynamic programming principle either as value iteration (1) or policy iteration (2) require a Markovian setting to be implemented such that the conditional expectation knowing the whole past can be replaced by the conditional expectation knowing only the value of a Markov process at the current date. The theory of the Snell envelope states that the sequence U also satisfies

$$U_{T_k} = \sup_{\tau\in\mathcal{T}_{T_k,T}} \mathbb{E}[Z_\tau|\mathcal{F}_{T_k}]. \tag{3}$$

When the discounted payoff process writes $Z_{T_k} = \phi_k(X_{T_k})$, for any $0 \leq k \leq N$, where $(X_t)_{0\leq t\leq T}$ is an adapted Markov process, the conditional expectation involved in (2) simplifies into

$$\mathbb{E}[Z_{\tau_{k+1}}|\mathcal{F}_{T_k}] = \mathbb{E}[Z_{\tau_{k+1}}|X_{T_k}] = \psi_k(X_{T_k}) \tag{4}$$

where ψ_k solves the following minimization problem

$$\inf_{\psi\in L^2(\mathcal{L}(X_{T_k}))} \mathbb{E}\left[\left|Z_{\tau_{k+1}} - \psi(X_{T_k})\right|^2\right]$$

with $L^2(\mathcal{L}(X_{T_k}))$ being the set of all measurable functions f such that $\mathbb{E}[f(X_{T_k})^2] < \infty$. The real challenge comes from properly approximating the space $L^2(\mathcal{L}(X_{T_k}))$ by a finite dimensional vector space: one typically uses polynomials or local bases. In both cases, to ensure a decent accuracy, the dimension of the approximation of $L^2(\mathcal{L}(X_{T_k}))$ increases exponentially fast with the dimension of X. When X is a high dimensional process, high performance computing can help but it is well known that solving the least square problem does not scale well and then deteriorates the efficiency of the parallel implementation, see for instance Pagès and Wilbertz [2011], Pagès et al. [2016].

In this work, we target *truly* path dependent options, i.e. options for which the payoff cannot be written as a function of a Markov process X with reasonable size. In this case, (4) does not hold anymore and computing the conditional expectation knowing $\mathcal{F}_{t_k}$ becomes really challenging. The new idea proposed in this work consists in computing an approximation of $Z_{T_{k+1}}$ for which the conditional expectation knowing $\mathcal{F}_{T_k}$ is known in a closed form. This will be achieved by using Wiener chaos expansion. Then, we rely on the orthogonality of the chaos expansion to introduce a high degree of parallelism in the algorithm.

In Section 2, we briefly recall the general ideas sustaining Wiener chaos expansion and how it can be used to approximate conditional expectations. Then, we present our algorithm in Section 3 and explain how to efficiently implement it in parallel. Section 4 is devoted to the study of the convergence of the algorithm. We conclude with some numerical experiments in Section 5, which emphasize the impressive scalability of the the parallel implementation and the efficiency of the algorithm for some complex path dependent options.

Notation

In this section, we gather some extensively used notation in the paper

- For $\alpha \in \mathbb{N}^d$, $|\alpha|_1 = \sum_{i=1}^d \alpha_i$. Similarly, for $\alpha \in (\mathbb{N}^n)^d$, $|\alpha|_1 = \sum_{j=1}^d \sum_{i=1}^n \alpha_i^j$.
- For $\alpha \in \mathbb{N}^d$, $\alpha! = \prod_{i=1}^d \alpha_i!$. Similarly, for $\alpha \in (\mathbb{N}^n)^d$, $\alpha! = \prod_{j=1}^d \prod_{i=1}^n \alpha_i^j!$.
- For $d, n, p \in \mathbb{N}$, we define the set of multi-indices with total degree smaller than p by
$$A_{p,n}^{\otimes d} = \left\{\alpha \in (\mathbb{N}^n)^d \ : |\alpha|_1 \leq p\right\}$$
- For $d, n, p \in \mathbb{N}$, and $k \leq n$ we define the set of multi-indices with total degree smaller than p and no degree after k by
$$A_{p,n|k}^{\otimes d} = \left\{\alpha \in A_{p,n}^{\otimes d} \ : \ \forall j \in \{1, \dots, d\}, \forall i > k, \ \alpha_i^j = 0\right\}.$$
- For $i \in \mathbb{N}$, H_i denote the $i-th$ Hermite polynomial.
- For $\alpha \in (\mathbb{N}^n)^d$, $x_1, \dots, x_n \in \mathbb{R}^d$, the multi-variate Hermite polynomials write
$$H_\alpha^{\otimes d}(x_1, \dots, x_n) = \prod_{j=1}^d \prod_{i=1}^n H_{\alpha_i^j}(x_i^j).$$

2 Wiener chaos expansion

2.1 General framework

In this section, we briefly recall the principles of Wiener chaos expansion and its basic properties. We refer to Nualart [1998] for theoretical details.

Let H_i be the $i-th$ Hermite polynomial defined by

$$H_0(x) = 1; \qquad H_i(x) = (-1)^i \, \mathrm{e}^{x^2/2} \, \frac{d^i}{dx^i}(\mathrm{e}^{-x^2/2}), \text{ for } i \geq 1.$$

They satisfy for all integer i, $H_i' = H_{i-1}$ with the convention $H_{-1} = 0$. We recall that if (X, Y) is a standard random normal vector in $\mathbb{R}^2$, $\mathbb{E}[H_i(X)H_j(Y)] = i! \, (\mathbb{E}[XY])^i \; \mathbf{1}_{\{i=j\}}$.

It is well-known that every square integrable $\mathcal{F}_T$-measurable random variable F admits the following orthonormal decomposition

$$F = \mathbb{E}[F] + \sum_{\alpha \in (\mathbb{N}^{\mathbb{N}})^d} \lambda_\alpha \prod_{j=1}^{d} \prod_{i \geq 1} H_{\alpha_i^j} \left(\int_0^T \eta_i^j(t) dB_t^j \right)$$

where $\left((\eta_i^j)_{1 \leq j \leq d}\right)_{i \geq 1}$ is an orthonormal basis of $L^2([0,T], \mathbb{R}^d)$. We denote by $L_1^2([0,T], \mathbb{R}^d)$ the set of functions $f = (f_1, \ldots, f_d) \in L^2([0,T], \mathbb{R}^d)$ such that for all $1 \leq i \leq d$, $\int_0^T f_i^2(t)dt = 1$. For all $p \geq 0$, we define the Wiener chaos of order p by

$$\mathcal{H}_p = \overline{\mathrm{span}}^{\mathrm{L}^2(\Omega, \mathcal{F}_\mathrm{T})} \left\{ \prod_{j=1}^{d} H_{p_j} \left(\int_0^T f_t^j dB_t^j \right) \; : \; f \in L_1^2([0,T], \mathbb{R}^d), \; \sum_{j=1}^{d} p_j = p \right\}.$$

We denote the projection of a random variable $F \in L^2(\mathcal{F}_T)$ onto $\bigoplus_{\ell=0}^{p} \mathcal{H}_\ell$ by $C_p(F)$. Note that the spaces $\mathcal{H}_\ell$ are orthogonal to each other thanks to the properties of the Hermite polynomials.

Consider the indicator functions of the grid defined by $0 = t_0 < t_1 < \cdots < t_n = T$ with values in $\mathbb{R}^d$ defined by

$$f_i^j(t) = \frac{\mathbf{1}_{\{]t_{i-1}, t_i]\}}(t)}{\sqrt{t_i - t_{i-1}}} \mathbf{e}_j, \; i = 1, \ldots, n, \; j = 1, \ldots, d$$

where $(\mathbf{e}_1, \ldots, \mathbf{e}_d)$ denotes the canonical basis of $\mathbb{R}^d$. Based on the definition of $\mathcal{H}_p$, we introduce the *truncated* Wiener chaos of order up to p

$$\mathcal{C}_{p,n} = \mathrm{span} \left\{ H_\alpha^{\otimes d}(G_1, \ldots, G_n) \; : \; \alpha \in (\mathbb{N}^n)^d, \; |\alpha|_1 \leq p \right\}$$

where

$$H_\alpha^{\otimes d}(G_1, \ldots, G_n) = \prod_{j=1}^{d} \prod_{i=1}^{n} H_{\alpha_i^j}(G_i^j) \quad \text{with} \quad G_i^j = \frac{B_{t_i}^j - B_{t_{i-1}}^j}{\sqrt{t_i - t_{i-1}}}.$$

From the orthogonality of the Hermite polynomials, we immediately deduce the following result.

Proposition 2.1 *Let F be a real valued random variable in $L^2(\Omega, \mathcal{F}_T, \mathbb{P})$. Its L^2 projection onto $\mathcal{C}_{p,n}$ writes*

$$C_{p,n}(F) = \sum_{\alpha \in A_{p,n}^{\otimes d}} \lambda_\alpha H_\alpha^{\otimes d}(G_1, \ldots, G_n)$$

where

$$A_{p,n}^{\otimes d} = \left\{ \alpha \in (\mathbb{N}^n)^d \ : |\alpha|_1 \leq p \right\}$$

and the coefficients λ_α are obtained as a dot product

$$\lambda_\alpha = \frac{1}{\alpha!} \, \mathbb{E}[F H_\alpha^{\otimes d}(G_1, \ldots, G_n)]. \tag{5}$$

The random variable $C_{p,n}(F)$ is called the truncated chaos expansion of order p of the random variable F. With an obvious abuse of notation, we write, for $\lambda \in \mathbb{R}^{A_{p,n}^{\otimes d}}$,

$$C_{p,n}(\lambda) = \sum_{\alpha \in A_{p,n}^{\otimes d}} \lambda_\alpha H_\alpha^{\otimes d}(G_1, \ldots, G_n). \tag{6}$$

We recall the main result concerning the convergence of the truncated chaos expansion (see Theorem 1.1.1 and Proposition 1.1.1 of Nualart [1998])

Proposition 2.2 *Let F be a real valued random variable in $L^2(\Omega, \mathcal{F}_T, \mathbb{P})$. Then, $C_{p,n}(F)$ converges to F in $L^2(\Omega, \mathcal{F}_T, \mathbb{P})$ when both p and n go to infinity.*

The space of truncated Wiener chaos $\mathcal{C}_{p,n}$ has the key property of being stable by the conditional expectation operator. More precisely, the following result explains how to compute, in a closed form, the conditional expectation of an element of $\mathcal{C}_{p,n}$.

Proposition 2.3 *Let F be a real valued random variable in $L^2(\Omega, \mathcal{F}_T, \mathbb{P})$ and let $k \in \{1, \ldots, n\}$ and $p \geq 0$*

$$\mathbb{E}[C_{p,n}(F)|\mathcal{F}_{t_k}] = \sum_{\alpha \in A_{p,n|k}^{\otimes d}} \lambda_\alpha \; H_\alpha^{\otimes d}(G_1, \ldots, G_n)$$

where $A_{p,n|k}^{\otimes d}$ is the set of multi-indices vanishing after time t_k

$$A_{p,n|k}^{\otimes d} = \left\{ \alpha \in A_{p,n}^{\otimes d} \ : \ \forall j \in \{1, \ldots, d\}, \, \forall i > k, \ \alpha_i^j = 0 \right\}.$$

Proof. Taking the conditional expectation in (6) leads to

$$\mathbb{E}[C_{p,n}(F)|\mathcal{F}_{t_k}] = \sum_{\alpha \in A_{p,n}^{\otimes d}} \lambda_\alpha \left(\prod_{i=1}^{k} \prod_{j=1}^{d} H_{\alpha_i^j}(G_i^j) \right) \mathbb{E}\left[\prod_{i=k+1}^{n} \prod_{j=1}^{d} H_{\alpha_i^j}(G_i^j) \Big| \mathcal{F}_{t_k} \right]. \tag{7}$$

Since the Brownian increments after time t_k are independent of $\mathcal{F}_{t_k}$ and are independent of one another, $\mathbb{E}\left[\prod_{i=k+1}^{n}\prod_{j=1}^{d} H_{\alpha_i^j}(G_i^j)\middle|\mathcal{F}_{t_k}\right] = \prod_{i=k+1}^{n}\prod_{j=1}^{d}\mathbb{E}\left[H_{\alpha_i^j}(G_i^j)\right]$, which is zero as soon as $\sum_{i=k+1}^{n}\sum_{j=1}^{d}\alpha_i^j > 0$. Hence, the sum in (7) reduces to the sum over the set of multi-indices $\alpha \in A_{p,n}^{\otimes d}$ such that $\alpha_i^j = 0$ for all $i > k$ and $1 \leq j \leq d$, which is exactly the definition of the set $A_{p,n|k}^{\otimes d}$. ■

Since the sum appearing in $\mathbb{E}[C_{p,n}(F)|\mathcal{F}_{t_k}]$ is reduced to a sum over the set of multi-indices $\alpha \in A_{p,n|k}^{\otimes d}$, it actually only depends on the first k increments $(G_1, \ldots, G_k)$. One can easily check that $\mathbb{E}[C_{p,n}(F)|\mathcal{F}_{t_k}]$ is actually given by the chaos expansion of F on the first k Brownian increments. Hence, computing a conditional expectation simply boils down to dropping the non measurable terms. While it may look like a naive way to proceed, it is indeed correct in this setting. To denote the chaos expansion on the time grid $(t_0, \ldots, t_n)$ truncated to the first k increments, we introduce the notation

$$C_{p,n|k}(F) = \sum_{\alpha \in A_{p,n|k}^{\otimes d}} \lambda_\alpha \; H_\alpha^{\otimes d}(G_1, \ldots, G_n) = \mathbb{E}[C_{p,n}(F)|\mathcal{F}_{t_k}]. \tag{8}$$

With an obvious abuse of notation, we write for $\lambda \in A_{p,n|k}^{\otimes d}$,

$$C_{p,n|k}(\lambda) = \sum_{\alpha \in A_{p,n|k}^{\otimes d}} \lambda_\alpha \; H_\alpha^{\otimes d}(G_1, \ldots, G_n).$$

2.2 Application to the approximation of conditional expectations

In this section, we explain how to use the truncated Wiener chaos expansion of a random variable $F \in L^2(\Omega, \mathcal{F}_T, \mathbb{P})$, to compute its conditional expectation.

Assume that we need M samples of the conditional expectations. We sample M paths $(B_{t_1}^{(m)}, \ldots, B_{t_n}^{(m)}, F^{(m)})$ of $(B_{t_1}, \ldots, B_{t_n}, F)$ and thanks to Proposition 2.3 we approximate $\mathbb{E}[F|\mathcal{F}_{t_k}]$ on the sample path with index m by

$$C_{p,n|k}^{(m)}(\widehat{\lambda}^M) = \sum_{\alpha \in A_{p,n|k}^{\otimes d}} \widehat{\lambda}_\alpha^M \; H_\alpha^{\otimes d}(G_1^{(m)}, \ldots, G_k^{(m)})$$

where

$$\widehat{\lambda}_\alpha^M = \frac{1}{M\alpha!}\sum_{\ell=1}^{M} F^{(\ell)} H_\alpha^{\otimes d}(G_1^{(\ell)}, \ldots, G_k^{(\ell)}).$$

Using the strong law of large numbers, we clearly have that for every $\alpha \in A_{p,n|k}^{\otimes d}$, $\widehat{\lambda}_\alpha^M$ converges a.s. to λ_α when M goes to infinity. Then, we deduce that for any fixed m, $C_{p,n|k}^{(m)}(\widehat{\lambda}^M)$ converges almost surely to $C_{p,n|k}^{(m)}(\lambda)$ when $M \to \infty$.

Remark 2.4 *Note that we use the same samples to compute the coefficients of the chaos expansion $\widehat{\lambda}_\alpha^M$ and to approximate $C_{p,n|k}^{(m)}$. It could have been possible to use different set of samples for the two parts and would have even simplified the theoretical analysis of the algorithm but the price to pay in terms of computational time is prohibitive. Using independent sets of samples would require to simulate new samples of the whole path at each date T_k.*

3 The algorithm

3.1 Description of the algorithm

We aim at solving the dynamic programming equation (2) to obtain τ_1. Then, the time-0 price of the Bermudan option writes

$$U_0 = \max(Z_0, \mathbb{E}[Z_{\tau_1}]).$$

For all $n \geq N$, consider a time grid $0 < t_0 < t_1 < \cdots < t_n = T$ of $[0, T]$, such that $\{T_1, \ldots, T_N\} \subset \{t_1, \ldots, t_n\}$. We assume that $\lim_{n\to\infty} \sup_{0\leq k\leq n-1} |t_{k+1} - t_k| = 0$. For $k \leq N$, we define $\sigma_k \in \mathbb{N}$ such that

$$t_{\sigma_k} = T_k.$$

Even though, we do not make the dependency on n explicit, it is clear that σ_k is an increasing function of n.

Now, we introduce some successive approximations of (2). First, we replace the true conditional expectation $E[Z_{\tau_{k+1}}|\mathcal{F}_{T_k}]$ by the conditional expectation of the truncated Wiener chaos expansion of $Z_{\tau_{k+1}}$

$$\begin{cases} \tau_N^{p,n} = T_N \\ \tau_k^{p,n} = T_k \mathbf{1}_{\left\{Z_{T_k} \geq C_{p,n|\sigma_k}(\lambda_k)\right\}} + \tau_{k+1}^{p,n} \mathbf{1}_{\left\{Z_{T_k} < C_{p,n|\sigma_k}(\lambda_k)\right\}}, \text{ for } 1 \leq k \leq N-1 \end{cases}$$

where the λ_k's are the coefficients of the truncated expansion of $Z_{\tau_{k+1}^{p,n}}$

$$\lambda_{k,\alpha} = \frac{1}{\alpha!}\ \mathbb{E}[Z_{\tau_{k+1}^{p,n}} H_\alpha^{\otimes d}(G_1, \ldots, G_{\sigma_k})] \quad \text{for } \alpha \in A_{p,n|\sigma_k}^{\otimes d}$$

The standard approach is to sample a bunch of paths of the model $S_{T_0}^{(m)}, S_{T_1}^{(m)}, \ldots, S_{T_N}^{(m)}$ along with the corresponding payoff paths $Z_{T_0}^{(m)}, Z_{T_1}^{(m)}, \ldots, Z_{T_N}^{(m)}$, for $m = 1, \ldots, M$. We denote by $B^{(m)}$ the Brownian path used to sample $S_{T_0}^{(m)}, S_{T_1}^{(m)}, \ldots, S_{T_N}^{(m)}$. Note that B is sampled on the finer grid $t_0, \ldots, t_n$, which enables us to deal with model discretization issues. The vector $G_1^{(m)}, \ldots, G_n^{(m)}$ corresponds to the increments of the Brownian motion B on the finer time grid. To compute the τ_k's on each path, one needs to compute the conditional expectations $\mathbb{E}[Z_{\tau_{k+1}}|\mathcal{F}_{T_k}]$ for $k =$

$1, \ldots, N-1$. Then, we introduce the final approximation of the backward iteration policy, in which the truncated chaos expansion is computed using a Monte Carlo approximation

$$\begin{cases} \widehat{\tau}_N^{p,n,(m)} = T_N \\ \widehat{\tau}_k^{p,n,(m)} = T_k \mathbf{1}_{\left\{Z_{T_k}^{(m)} \geq C_{p,n|\sigma_k}^{(m)}(\widehat{\lambda}_k^M)\right\}} + \widehat{\tau}_{k+1}^{p,n,(m)} \mathbf{1}_{\left\{Z_{T_k}^{(m)} < C_{p,n|\sigma_k}^{(m)}(\widehat{\lambda}_k^M)\right\}}, \text{ for } 1 \leq k \leq N-1 \end{cases}$$

where the $\widehat{\lambda}_k^M$ are computed as described in Section 2.2. For $k = 1, \ldots, N-1$, the vector $\widehat{\lambda}_k^M$ is an element of $\mathbb{R}^{A_{p,n}^{\otimes d,\sigma_k}}$ and for every $\alpha \in A_{p,n}^{\otimes d,\sigma_k}$,

$$\widehat{\lambda}_{k,\alpha}^M = \frac{1}{M\alpha!} \sum_{\ell=1}^{M} Z_{\widehat{\tau}_{k+1}^{p,n,(\ell)}}^{(\ell)} H_\alpha^{\otimes d}(G^{(\ell)}). \tag{9}$$

Then, we finally approximate the time-0 price of the option by

$$U_0^{p,n,M} = \max\left(Z_0, \frac{1}{M} \sum_{m=1}^{M} Z_{\widehat{\tau}_1^{p,n,(m)}}^{(m)}\right). \tag{10}$$

The pseudo code of our approach corresponds to Algorithm 3.1.

Remark 3.1 *From a practical point of view, we advise to consider in the money paths in the chaos expansion as it was already noticed in Longstaff and Schwartz [2001]. Hence, the set* $\{Z_{T_k}^{(m)} \geq C_{p,n|\sigma_k}^{(m)}(\widehat{\lambda}_k^M)\}$ *is replaced by* $\{Z_{T_k}^{(m)} > 0\} \cup \{Z_{T_k}^{(m)} \geq C_{p,n|\sigma_k}^{(m)}(\widehat{\lambda}_k^M)\}$ *and the coefficients of the chaos expansion are given by*

$$\widehat{\lambda}_{k,\alpha}^M = \frac{1}{M\alpha!} \sum_{\ell=1}^{M} Z_{\widehat{\tau}_{k+1}^{p,n,(\ell)}}^{(\ell)} \mathbf{1}_{\left\{Z_{T_k}^{(\ell)} > 0\right\}} H_\alpha^{\otimes d}(G^{(\ell)}).$$

This modification does not change the theoretical analysis of the algorithm but improves its numerical behavior.

Our algorithm is designed as a black box taking as inputs simulations of the Brownian motion and the corresponding payoff process. From a practical point of view, you can design the implementation in such as way that pricing a new product simply amounts to implementing the discretization of the model and the computation of the payoff.

3.2 Comments on the algorithm

The obvious and generic way to deal with truly path-dependent options or non Markovian model using the standard Longstaff Schwartz algorithm would be to consider the whole path as a regressor. It is very much unlikely that one can easily build a set of basis functions which are orthogonal for the law of the discretized path process. Hence, the regression problem would grow exponentially fast and as explained in Benguigui and Baude [2012], parallelism would not help much. Going beyond the Markovian setting requires an orthogonality property, which turns

Generate $(G^{(1)}, Z^{(1)}), \dots, (G^{(M)}, Z^{(M)})$ M i.i.d. samples following the law of $(Z_{t_i}, G_{t_i})_{1 \le i \le N}$

$\widehat{\tau}_N^{p,n,(m)} \leftarrow T$ for all $m = 1, \dots, M$

for $k = N-1, \dots, 1$ **do**

  for $\alpha \in A_{p,n|\sigma_k}^{\otimes d}$ **do**

$$\widehat{\lambda}_{k,\alpha}^{M} = \frac{1}{M\alpha!} \sum_{\ell=1}^{M} Z^{(\ell)}_{\widehat{\tau}_{k+1}^{p,n,(\ell)}} H_\alpha^{\otimes d}(G^{(\ell)})$$

  end

  for $m = 1, \dots, M$ **do**

$$\widehat{\tau}_k^{p,n,(m)} = T_k \mathbf{1}_{\left\{ Z^{(m)}_{T_k} \ge C^{(m)}_{p,n|\sigma_k}(\widehat{\lambda}_k^M) \right\}} + \widehat{\tau}_{k+1}^{p,n,(m)} \mathbf{1}_{\left\{ Z^{(m)}_{T_k} < C^{(m)}_{p,n|\sigma_k}(\widehat{\lambda}_k^M) \right\}}$$

  end

end

$$U_0^{p,n,M} = \max\left(Z_0, \frac{1}{M} \sum_{m=1}^{M} Z^{(m)}_{\widehat{\tau}_1^{p,n,(m)}} \right)$$

Algorithm 3.1: Dynamic programming principle using Wiener chaos expansion

the regression problem into a series of independent inner-products. Of course, it is always possible to pretend everything is Markovian, but then you have no guaranty on the error you are making.

Our algorithm may be related to a *regress later* method as investigated by Glasserman and Yu [2004a], Balata and Palczewski [2018]. At time T_k, a regress later approach is typically composed of two steps: first $Z_{\tau_{k+1}}$ is decomposed on a set of $\mathcal{F}_{T_{k+1}}$ measurable basis functions, which looks like a least squares approximation of the conditional expectation with respect to $\mathcal{F}_{T_{k+1}}$. Then, the conditional expectation of each basis function is computed analytically to obtain an approximation of $\mathbb{E}[Z_{\tau_{k+1}} | \mathcal{F}_{T_k}]$. Our algorithm can also be seen as a two stage method: first we compute the chaos expansion of $Z_{\tau_{k+1}}$ and then we compute its conditional expectation. Although this way of formulating the algorithm is mathematically correct, it would be totally inefficient to implement it this way. As a matter of fact, taking the conditional expectation of a Wiener chaos expansion simply amounts to dropping non measurable terms and because every coefficient is computed on its own using an inner product, we can directly compute the conditional expectation of the chaos expansion by actually computing a chaos expansion with respect to the Brownian increments up to time T_k only. This more pragmatic way of understanding our

algorithm makes it actually closer to a *regress now* approach.

Closely looking at Algorithm 3.1, it is clear that the central part of the algorithm is the computation of the chaos expansion. Conveniently implementing this step plays a major role in the efficiency of the algorithm. In our C++ implementation, the chaos expansion is performed using the generic multivariate polynomial toolbox from Lelong [2007-2017].

3.3 Complexity analysis

Most of the computational time is spent computing the coefficients of the chaos expansions. Remember that the cardinality of $A_{p,n|k}^{\otimes d}$ is given by $\binom{\sigma_k d+p}{n\sigma_k} = \frac{(\sigma_k d+p)\cdots(\sigma_k d+1)}{p!}$. As the optimal policy is only updated on the in-the-money paths at each time step (see Remark 3.1), the complexity of iteration k of the loop on line 3 of Algorithm 3.1 is proportional to

$$\sharp\{\text{in-the-money paths at time } T_k\} \times \binom{\sigma_k d + p}{n\sigma_k}.$$

It is worth noting that the complexity decreases when time decreases. The order p of the expansion plays a major role in the computational time of the algorithm. So, when the order of the expansion increases from p to $p+1$, the computational time is multiplied by $\frac{\sigma_k d+p+1}{p+1}$.

3.4 The parallel implementation

The key computational trick of our algorithm is that the chaos coefficients λ are written as independent expectations and can therefore be parallelized both across α and the number M of Monte Carlo samples. Simply put, Algorithm 3.1 can be reduced to computing several independent Monte Carlo averages and is therefore very well suited for parallel programming. For a fixed time T_k, there are two ways of introducing parallelism.

(i) The coefficients of the truncated Wiener chaos expansion can be computed in parallel. For two multi-indices $\alpha, \beta \in A_{p,n|\sigma_k}^{\otimes d}$, the computations of $\widehat{\lambda}_{k,\alpha}^{M}$ and $\widehat{\lambda}_{k,\beta}^{M}$ are independent and can therefore be carried out simultaneously. The update of all the $\widehat{\tau}_k^{p,n,(m)}$ can also be performed in parallel. This approach looks very promising provided that the cardinality of $A_{p,n|\sigma_k}^{\otimes d}$ is large enough, at least larger than the number of available computing resources. Note that

$$\#A_{p,n|\sigma_k}^{\otimes d} = \binom{\sigma_k\, d \;+\; p}{\sigma_k\, d}$$

where we recall that $\sigma_k \to 0$ when $k \to 0$. This approach will be efficient for large enough k but will inevitably fail to scale when k decreases, ie for smaller dates.

(ii) Alternatively, we can use the number of Monte Carlo samples as the leverage for parallelism. Since the number of samples remains fixed during the whole algorithm, the parallelism will be as efficient for large k as for small ones. Assume we have R computing resources at our disposal, then each resource handles $M_R = M/R$ sample paths and runs

the sequential algorithm 3.1 on these paths except that at each time step, a reduction followed by a broadcast are done right before updating the $\widehat{\tau}_k^{p,n,(m)}$, $m = 1, \dots, M$. In this way, the chaos expansions are computed using the M paths. We precisely describe this parallel algorithm in Algorithm 3.2.

We have followed the approach (ii) for our parallel implementation to make sure all the resources are always fully busy, which is the least requirement to ensure a decent scalability. The comparison of Algorithms 3.1 and 3.2 shows that the sequential and parallel algorithms differ very little. We even managed to merge the sequential and parallel implementations into a single code, which is hardly ever feasible especially when using MPI. Each computing resource samples a bunch of paths, on which it updates the optimal stopping policy and contributes to the computation of the $\widehat{\lambda}_k^M$'s. At each time step, we compute an average (a *reduction*) to get the value of the $\widehat{\lambda}_k^M$'s and then we send (*broadcast*) the coefficients to every resources. In practice, we actually use the *AllReduce*[1] method from MPI.

It was noted in Benguigui and Baude [2012], Pagès et al. [2016], that using mini-batches to introduce parallelism in least-square Monte Carlo was not convincingly efficient, mainly because the backward induction is essentially sequential. This is due to the regression step itself, which cannot be solved efficiently when Monte Carlo paths are allocated by blocks on each processor. In any parallel implementation of least-squares Monte Carlo, the regression step eventually becomes the bottleneck because of its bad scalability. To circumvent this main issue with least-square Monte Carlo, Pagès and Wilbertz [2011] replaced the regression step by a quantization approach, which allows for natural parallelism. Similarly, Gobet et al. [2016] used stratification to introduce conditional independence between the samples used in each strata. It may look as a mini-batch approach while they have to use new stratified sampling at each time step because they work in a backward stochastic differential setting. Hence, interpreting their approach in terms of mini-batches is not straightforward. In our approach, we rely on the orthogonality of the chaos expansion to replace the regression step by inner products computed by Monte Carlo. Therefore, our approach naturally fits into the mini-batch paradigm with no extra cost.

4 Convergence of the algorithm

In this section, we basically follow the lines of the methodology introduced in Clément et al. [2002]. The statements of the convergence results are quite similar even if some assumptions had to be modified to match our framework, but the proofs differ to adapt to the new formulation of the regression step.

There are two independent parts in this section. In Section 4.2, we study the convergence of the algorithm with respect to the chaos expansion when all expectations are assumed to be computed exactly (no Monte Carlo approximation). In Section 4.3, we fix the order and the discretization used in the chaos expansion and we study the convergence with respect to the number

[1] See `https://mpitutorial.com/tutorials/mpi-reduce-and-allreduce/` for an explanation of how reduce and broadcast can be efficiently coupled.

$M_R \leftarrow M/R$

In parallel do

Generate $(G^{(1)}, Z^{(1)}), \dots, (G^{(M_R)}, Z^{(M_R)})$ M_R i.i.d. samples following the law of $(Z_{t_i}, G_{t_i})_{1 \leq i \leq N}$

$\widehat{\tau}_N^{p,n,(m)} \leftarrow T$ for all $m = 1, \dots, M_R$

for $k = N-1, \dots, 1$ **do**

for $\alpha \in A_{p,n|\sigma_k}^{\otimes d}$ **do**

$$\widehat{\lambda}_{k,\alpha}^{M_R} = \frac{1}{M_R \alpha!} \sum_{\ell=1}^{M_R} Z^{(\ell)}_{\widehat{\tau}_{k+1}^{p,n,(m)}} H_\alpha^{\otimes d}(G^{(\ell)})$$

end

Reduce the $\widehat{\lambda}_{k,\alpha}^{M_R}$ to obtain $\widehat{\lambda}_{k,\alpha}^{M}$

Broadcast $\widehat{\lambda}_{k,\alpha}^{M}$ for $\alpha \in A_{p,n|\sigma_k}^{\otimes d}$

for $m = 1, \dots, M_R$ **do**

$$\widehat{\tau}_k^{p,n,(m)} = T_k \mathbf{1}_{\left\{Z_{T_k}^{(m)} \geq C_{p,n|\sigma_k}^{(m)}(\widehat{\lambda}_k^M)\right\}} + \widehat{\tau}_{k+1}^{p,n,(m)} \mathbf{1}_{\left\{Z_{T_k}^{(m)} < C_{p,n|\sigma_k}^{(m)}(\widehat{\lambda}_k^M)\right\}}$$

end

end

$$U_1^{p,n,M_R} = \frac{1}{M_R} \sum_{m=1}^{M_R} Z^{(m)}_{\widehat{\tau}_1^{p,n,(m)}}$$

end

Reduce the U_1^{p,n,M_R}

$U_0^{p,n,M} = \max\left(Z_0, U_1^{p,n}\right)$

Algorithm 3.2: Parallel algorithm for solving the dynamic programming principle using Wiener chaos expansion

of Monte Carlo samples. This is achieved by first proving that the Monte Carlo approximations of the chaos expansion at each time step converge to the true coefficients.

4.1 Notation

To avoid over expanding notation, we simply write G instead of $(G_1, \dots, G_n)$ in the chaos expansions. At some points, it may be important to make precise which Brownian increments are used in the chaos expansion. To do so, we introduce the notation

$$C_{p,n}(\lambda; G) = \sum_{\alpha \in A_{p,n}^{\otimes d}} \lambda_\alpha H_\alpha^{\otimes d}(G).$$

First, it is important to note that the paths $\tau_1^{p,n,(m)}, \dots, \tau_N^{p,n,(m)}$ for $m = 1, \dots, M$ are identically distributed but not independent since the Monte Carlo computation of the chaos expansion coefficients $\widehat{\lambda}_k^M$ mixes all the paths. We define the vector Λ of the coefficients of the successive expansions $\Lambda = (\lambda_1, \dots, \lambda_{N-1})$ and its Monte Carlo approximation $\widehat{\Lambda}^M = (\widehat{\lambda}_1^M, \dots, \widehat{\lambda}_{N-1}^M)$.

Now, we recall the notation used by Clément et al. [2002] to study the convergence of the original Longstaff Schwartz approach.

Given a parameter $\ell = (\ell_1, \dots, \ell_{N-1})$ in $\mathbb{R}^{A_{p,n|\sigma_1}^{\otimes d}} \times \cdots \times \mathbb{R}^{A_{p,n|\sigma_{N-1}}^{\otimes d}}$ and vectors $z = z_1, \dots, z_N$ in $\mathbb{R}^N$ and $g = (g_1, \dots, g_n)$ in $(\mathbb{R}^d)^n$, we define the vector field $F = F_1, \dots, F_N$ by

$$\begin{cases} F_N(\ell, z, g) & = z_N \\ F_k(\ell, z, g) & = z_k \mathbf{1}_{\left\{z_k \geq C_{p,n|\sigma_k}(\ell; g)\right\}} + F_{k+1}(\ell, z, g) \mathbf{1}_{\left\{z_k < C_{p,n|\sigma_k}(\ell; g)\right\}}, \text{ for } 1 \leq k \leq N-1. \end{cases}$$

Note that $F_k(\ell, z, x)$ does not depend on the first $k-1$ components of ℓ, ie $\ell_1, \dots, \ell_{k-1}$. Moreover,

$$\begin{aligned} F_k(\Lambda, Z, G) &= Z_{\tau_k^{p,n}}, \\ F_k(\widehat{\Lambda}^M, Z^{(m)}, G^{(m)}) &= Z^{(m)}_{\widehat{\tau}_k^{p,n,(m)}}. \end{aligned}$$

For $k = 1, \dots, N$, we also define the functions $\phi_k : \mathbb{R}^{A_{p,n|\sigma_1}^{\otimes d}} \times \cdots \times \mathbb{R}^{A_{p,n|\sigma_{N-1}}^{\otimes d}} \to \mathbb{R}$ and $\psi_k : \mathbb{R}^{A_{p,n|\sigma_1}^{\otimes d}} \times \cdots \times \mathbb{R}^{A_{p,n|\sigma_{N-1}}^{\otimes d}} \to \mathbb{R}^{A_{p,n|\sigma_k}^{\otimes d}}$ by

$$\phi_k(\ell) = \mathbb{E}[F_k(\ell, Z, G)] \quad \text{and} \quad \psi_k(\ell) = \left(\mathbb{E}[F_k(\ell, Z, G) H_\alpha^{\otimes d}(G)]\right)_{\alpha \in A_{p,n|\sigma_k}^{\otimes d}}.$$

Note that ϕ_k and ψ_k actually only depends on $\ell_k, \dots, \ell_{N-1}$ but not on the first $k-1$ components of ℓ.

4.2 Chaos approximation of conditional expectations

Proposition 4.1 *For all $k = 1, \dots, N$, $\lim_{p,n \to \infty} \mathbb{E}[Z_{\tau_k^{p,n}} | \mathcal{F}_{T_k}] = \mathbb{E}[Z_{\tau_k} | \mathcal{F}_{T_k}]$ in $L^2(\Omega)$.*

Proof. We proceed by induction. The result is true for $k = N$ as $\tau_N = \tau_k^{p,n} = T$. Assume it holds for $k+1$ ($k \leq N-1$), we will prove it is true for k.

$$\begin{aligned}
&\mathbb{E}[Z_{\tau_k^{p,n}} - Z_{\tau_k}|\mathcal{F}_{T_k}] \\
&= Z_{T_k}\left(\mathbf{1}_{\left\{Z_{T_k} \geq C_{p,n|\sigma_k}(\lambda_k)\right\}} - \mathbf{1}_{\left\{Z_{T_k} \geq \mathbb{E}[Z_{\tau_{k+1}}|\mathcal{F}_{T_k}]\right\}}\right) \\
&\quad + \mathbb{E}\left[Z_{\tau_{k+1}^{p,n}}\mathbf{1}_{\left\{Z_{T_k} < C_{p,n|\sigma_k}(\lambda_k)\right\}} - Z_{\tau_{k+1}}\mathbf{1}_{\left\{Z_{T_k} < \mathbb{E}[Z_{\tau_{k+1}}|\mathcal{F}_{T_k}]\right\}}|\mathcal{F}_{T_k}\right] \\
&= \left(Z_{T_k} - \mathbb{E}[Z_{\tau_{k+1}}|\mathcal{F}_{T_k}]\right)\left(\mathbf{1}_{\left\{Z_{T_k} \geq C_{p,n|\sigma_k}(\lambda_k)\right\}} - \mathbf{1}_{\left\{Z_{T_k} \geq \mathbb{E}[Z_{\tau_{k+1}}|\mathcal{F}_{T_k}]\right\}}\right) \\
&\quad + \mathbb{E}\left[Z_{\tau_{k+1}^{p,n}} - Z_{\tau_{k+1}}|\mathcal{F}_{T_k}\right]\mathbf{1}_{\left\{Z_{T_k} < C_{p,n|\sigma_k}(\lambda_k)\right\}}.
\end{aligned}$$

By the induction assumption, the term $\mathbb{E}\left[Z_{\tau_{k+1}^{p,n}} - Z_{\tau_{k+1}}|\mathcal{F}_{T_k}\right]$ goes to zero in $L^2(\Omega)$ as p, n both go to infinity. So, we just have to prove that

$$A_k = \left(Z_{T_k} - \mathbb{E}[Z_{\tau_{k+1}}|\mathcal{F}_{T_k}]\right)\left(\mathbf{1}_{\left\{Z_{T_k} \geq C_{p,n|\sigma_k}(\lambda_k)\right\}} - \mathbf{1}_{\left\{Z_{T_k} \geq \mathbb{E}[Z_{\tau_{k+1}}|\mathcal{F}_{T_k}]\right\}}\right)$$

converges to zero in $L^2(\Omega)$.

$$\begin{aligned}
|A_k| &\leq \left|Z_{T_k} - \mathbb{E}[Z_{\tau_{k+1}}|\mathcal{F}_{T_k}]\right|\left|\mathbf{1}_{\left\{Z_{T_k} \geq C_{p,n|\sigma_k}(\lambda_k)\right\}} - \mathbf{1}_{\left\{Z_{T_k} \geq \mathbb{E}[Z_{\tau_{k+1}}|\mathcal{F}_{T_k}]\right\}}\right| \\
&\leq \left|Z_{T_k} - \mathbb{E}[Z_{\tau_{k+1}}|\mathcal{F}_{T_k}]\right|\left|\mathbf{1}_{\left\{\mathbb{E}[Z_{\tau_{k+1}}|\mathcal{F}_{T_k}] > Z_{T_k} \geq C_{p,n|\sigma_k}(\lambda_k)\right\}} - \mathbf{1}_{\left\{C_{p,n|\sigma_k}(\lambda_k) > Z_{T_k} \geq \mathbb{E}[Z_{\tau_{k+1}}|\mathcal{F}_{T_k}]\right\}}\right| \\
&\leq \left|Z_{T_k} - \mathbb{E}[Z_{\tau_{k+1}}|\mathcal{F}_{T_k}]\right|\mathbf{1}_{\left\{\left|Z_{T_k} - \mathbb{E}[Z_{\tau_{k+1}}|\mathcal{F}_{T_k}]\right| \leq \left|C_{p,n|\sigma_k}(\lambda_k) - \mathbb{E}[Z_{\tau_{k+1}}|\mathcal{F}_{T_k}]\right|\right\}} \\
&\leq \left|C_{p,n|\sigma_k}(\lambda_k) - \mathbb{E}[Z_{\tau_{k+1}}|\mathcal{F}_{T_k}]\right| \\
&\leq \left|C_{p,n|\sigma_k}(\lambda_k) - C_{p,n|\sigma_k}(\mathbb{E}[Z_{\tau_{k+1}}|\mathcal{F}_{T_k}])\right| + \left|C_{p,n|\sigma_k}(\mathbb{E}[Z_{\tau_{k+1}}|\mathcal{F}_{T_k}]) - \mathbb{E}[Z_{\tau_{k+1}}|\mathcal{F}_{T_k}]\right|. \quad (11)
\end{aligned}$$

Note that $C_{p,n|\sigma_k}(\lambda_k) = C_{p,n|\sigma_k}(\mathbb{E}[Z^{p,n}_{\tau_{k+1}}|\mathcal{F}_{T_k}])$. The truncated chaos expansion $C_{p,n|\sigma_k}$ being an orthogonal projection on the space of random variables measurable with respect to the Brownian increments $G_1, \ldots, G_k$, we clearly have that

$$\begin{aligned}
&\mathbb{E}\left[\left|C_{p,n|\sigma_k}(\lambda_k) - C_{p,n|\sigma_k}(\mathbb{E}[Z_{\tau_{k+1}}|\mathcal{F}_{T_k}])\right|^2\right] \\
&\leq \mathbb{E}\left[\left|\mathbb{E}[Z_{\tau_{k+1}^{p,n}}|\mathcal{F}_{T_k}] - \mathbb{E}[Z_{\tau_{k+1}}|\mathcal{F}_{T_k}]\right|^2\right] \\
&\leq \mathbb{E}\left[\left|\mathbb{E}[Z_{\tau_{k+1}^{p,n}}|\mathcal{F}_{T_{k+1}}] - \mathbb{E}[Z_{\tau_{k+1}}|\mathcal{F}_{T_{k+1}}]\right|^2\right]
\end{aligned}$$

where the last inequality comes from the orthogonal projection feature of the conditional expectation. Then, the induction assumption for $k+1$ yields that $C_{p,n|\sigma_k}(\lambda_k) - C_{p,n|\sigma_k}(\mathbb{E}[Z_{\tau_{k+1}}|\mathcal{F}_{T_k}])$ goes to zero in $L^2(\Omega)$ as p, n go to infinity. So, the first term on the r.h.s of (11) goes to zero.

As $C_{p,n|\sigma_k}(\mathbb{E}[Z_{\tau_{k+1}}|\mathcal{F}_{T_k}]) = C_{p,n}(\mathbb{E}[Z_{\tau_{k+1}}|\mathcal{F}_{T_k}])$, the second term on the r.h.s of (11) goes to zero in $L^2(\Omega)$ thanks to Proposition 2.2. Combining these two results yields the convergence statement of the proposition. ■

Remark 4.2 *When the discrete time payoff process $(Z_{T_k})_{0\leq k\leq N}$ is measurable for the filtration generated by the discrete time Brownian increments $(\mathcal{G}_k)_{0\leq k\leq N} = (\sigma(B_{T_{i+1}} - B_{T_i}, i \leq k))_{0\leq k\leq N}$, the result of Proposition 4.1 simplifies to $\lim_{p\to\infty} \mathbb{E}[Z_{\tau_k^{p,N}}|\mathcal{F}_{T_k}] = \mathbb{E}[Z_{\tau_k}|\mathcal{F}_{T_k}]$ in L^2. There is no need to let n go to infinity, it is sufficient to take $n = N$. From a practical point of view, one should choose n in order to monitor the discretization error between the true payoff process Z and its implementable discretization Z^n on a time grid with n steps. Then, the parameter n has to be considered as being fixed and we actually compute the price of the Bermudan option with payoff process Z^n instead of Z. Therefore, when the model can be exactly sampled, one should choose $n = N$.*

4.3 Convergence of the Monte Carlo approximation

In the following, we assume that p and n are fixed and we study the convergence with respect to the number of samples M.

4.3.1 Strong law of large numbers

To start, we prove the convergence of the coefficients of the chaos expansions.

Proposition 4.3 *Assume that for every $k = 1, \dots, N$, $\mathbb{P}(Z_{T_k} \in \mathcal{C}_{p,n|\sigma_k}) = 0$. Then, for every $k = 1, \dots, N$, $\widehat{\Lambda}_k^M$ converges to Λ_k a.s. as $M \to \infty$.*

The proof of Proposition 4.3 based on the following key lemma from Clément et al. [2002]. The assumption $\mathbb{P}(Z_{T_k} \in \mathcal{C}_{p,n}) = 0$ may look surprising but a very similar assumption was already required in [Clément et al., 2002, Lemma 3.2]. This assumption combined with the following lemma proves that the vector field $F(a, Z, G)$ is a.s. continuous w.r.t the expansion coefficients a.

Lemma 4.4 *For every $k = 1, \dots, N-1$,*

$$|F_k(a, Z, G) - F_k(b, Z, G)| \leq \left(\sum_{i=k}^{N} |Z_{T_i}|\right) \left(\sum_{i=k}^{N-1} \mathbf{1}_{\left\{|Z_{T_i} - C_{p,n|\sigma_i}(b_i)| \leq |a_i - b_i| \, \|C_{p,n|\sigma_i}\|\right\}}\right)$$

where

$$\|C_{p,n}\| = \sup_{|\lambda|=1} |C_{p,n}(\lambda)| \,.$$

Proof (Proof of Proposition 4.3). We proceed by induction. For $k = N-1$, the result directly follows from the standard strong law of large numbers. Choose $k \leq N-2$ and assume the result holds for $k+1, \dots, N-1$, we aim at proving this is true for k.

$$\widehat{\lambda}_{k,\alpha}^M = \frac{1}{M\alpha!} \sum_{m=1}^{M} F_{k+1}(\widehat{\Lambda}_{k+1}^M, Z^{(m)}, G^{(m)}) H_\alpha^{\otimes d}(G^{(m)}).$$

By the standard strong law of large number, $\frac{1}{M\alpha!}\sum_{m=1}^{M} F_{k+1}(\widehat{\Lambda}_{k+1}, Z^{(m)}, G^{(m)}) H_\alpha^{\otimes d}(G^{(m)})$ converges a.s. to $\frac{1}{\alpha!}\mathbb{E}[F_{k+1}(\widehat{\Lambda}_{k+1}, Z, G) H_\alpha^{\otimes d}(G)] = \lambda_{k,\alpha}$. Then, it is sufficient to prove that

$$\Psi_M = \frac{1}{M}\sum_{m=1}^{M}\left(F_{k+1}(\widehat{\Lambda}_{k+1}^M, Z^{(m)}, G^{(m)}) - F_{k+1}(\widehat{\Lambda}_{k+1}, Z^{(m)}, G^{(m)})\right) H_\alpha^{\otimes d}(G^{(m)}) \to 0 \quad a.s.$$

Then, using Lemma 4.4, we have

$$\begin{aligned}
|\Psi_M| &\le \frac{1}{M}\sum_{m=1}^{M}\left|F_{k+1}(\widehat{\Lambda}_{k+1}^M, Z^{(m)}, G^{(m)}) - F_{k+1}(\widehat{\Lambda}_{k+1}, Z^{(m)}, G^{(m)})\right| \left|H_\alpha^{\otimes d}(G^{(m)})\right| \\
&\le \frac{1}{M}\sum_{m=1}^{M}\sum_{i=k+1}^{N}\left|Z_{T_{i+1}}^{(m)}\right|\left(\sum_{i=k+1}^{N-1}\mathbf{1}_{\left\{\left|Z_{T_i}^{(m)} - C_{p,n|\sigma_i}^{(m)}(\Lambda_i)\right| \le \left|\widehat{\Lambda}_i^M - \Lambda_i\right| \left\|C_{p,n|\sigma_i}\right\|\right\}}\right)\left|H_\alpha^{\otimes d}(G^{(m)})\right|
\end{aligned}$$

From the induction assumption for $k+1, \ldots, N-1$, we have that for $i = k+1, \ldots, N-1$, $\widehat{\Lambda}_i^M \to \Lambda_i$. Then, for any $\varepsilon > 0$, we have

$$\begin{aligned}
&\limsup_M |\Psi_M| \\
&\le \limsup_M \frac{1}{M}\sum_{m=1}^{M}\sum_{i=k+1}^{N}\left|Z_{T_{i+1}}^{(m)}\right|\left(\sum_{i=k+1}^{N-1}\mathbf{1}_{\left\{\left|Z_{T_i}^{(m)} - C_{p,n|\sigma_i}^{(m)}(\Lambda_i)\right| \le \varepsilon \left\|C_{p,n|\sigma_i}\right\|\right\}}\right)\left|H_\alpha^{\otimes d}(G^{(m)})\right| \\
&\le E\left[\sum_{i=k+1}^{N}\left|Z_{T_{i+1}}\right|\left(\sum_{i=k+1}^{N-1}\mathbf{1}_{\left\{\left|Z_{T_i} - C_{p,n|\sigma_i}(\Lambda_i)\right| \le \varepsilon \left\|C_{p,n|\sigma_i}\right\|\right\}}\right)\left|H_\alpha^{\otimes d}(G)\right|\right]
\end{aligned}$$

where the last equality follows from the strong law of large numbers. As $\mathbb{P}(Z_{T_k} \in \mathcal{C}_{p,n|\sigma_k}) = 0$ for all k, we can let ε go to 0 to obtain that $\limsup_M |\Psi_M| = 0$ a.s. ■

Once the convergence of the expansion is established, we can now study the convergence of $U_0^{p,n,M}$ to $U_0^{p,n}$ when $M \to \infty$.

Theorem 4.5 *Assume that for every $k = 1, \ldots, N$, $\mathbb{P}(Z_{T_k} \in \mathcal{C}_{p,n}) = 0$. Then, for $q = 1, 2$ and all $k = 1, \ldots, N$,*

$$\lim_{M\to\infty}\frac{1}{M}\sum_{m=1}^{M}\left(Z_{\widehat{\tau}_k^{p,n,(m)}}^{(m)}\right)^q = \mathbb{E}\left[\left(Z_{\tau_k^{p,n}}\right)^q\right] \quad a.s.$$

Proof. Note that $\mathbb{E}[(Z_{\tau_k^{p,n}})^q] = \mathbb{E}[F_k(\widehat{\Lambda}, Z, G)^q]$ and by the strong law of large numbers

$$\lim_{M\to\infty}\frac{1}{M}\sum_{m=1}^{M}F_k(\widehat{\Lambda}, Z^{(m)}, G^{(m)})^q = \mathbb{E}[F_k(\widehat{\Lambda}, Z, G)^q] \quad a.s.$$

Hence, we have to prove that

$$\Delta F_M = \frac{1}{M}\sum_{m=1}^{M}\left(F_k(\widehat{\Lambda}^M, Z^{(m)}, G^{(m)})^q - F_k(\widehat{\Lambda}, Z^{(m)}, G^{(m)})^q\right) \xrightarrow[M\to\infty]{a.s} 0.$$

For any $x, y \in \mathbb{R}$, and $q = 1, 2$, $|x^q - y^q| = |x - y|\,|x^{q-1} + y^{q-1}|$. Using Lemma 4.4 and that $|F_k(\gamma, z, g)| \leq \max_{k \leq j \leq N} |z_j|$, we have

$$
\begin{aligned}
|\Delta F_M| &\leq \frac{1}{M} \sum_{m=1}^{M} \left| F_k(\widehat{\Lambda}_k^M, Z^{(m)}, G^{(m)})^q - F_k(\widehat{\Lambda}_k, Z^{(m)}, G^{(m)})^q \right| \\
&\leq 2\,\frac{1}{M} \sum_{m=1}^{M} \sum_{i=k}^{N} \max_{k \leq j \leq N} \left| Z_{T_j}^{(m)} \right| \left| Z_{T_{i+1}}^{(m)} \right| \left(\sum_{i=k}^{N-1} \mathbf{1}_{\left\{ \left| Z_{T_i}^{(m)} - C_{p,n|\sigma_i}^{(m)}(\Lambda_i) \right| \leq \left| \widehat{\Lambda}_i^M - \Lambda_i \right| \left\| C_{p,n|\sigma_i} \right\| \right\}} \right)
\end{aligned}
$$

Using Proposition 4.3, $\widehat{\Lambda}_i^M \to \Lambda_i$ for all $i = 1, \ldots, N-1$. Then for any $\varepsilon > 0$,

$$
\begin{aligned}
&\limsup_M |\Delta F_M| \\
&\leq 2 \limsup_M \frac{1}{M} \sum_{m=1}^{M} \sum_{i=k}^{N} \max_{k \leq j \leq N} \left| Z_{T_j}^{(m)} \right| \left| Z_{T_{i+1}}^{(m)} \right| \left(\sum_{i=k}^{N-1} \mathbf{1}_{\left\{ \left| Z_{T_i}^{(m)} - C_{p,n|\sigma_i}^{(m)}(\Lambda_i) \right| \leq \varepsilon \left\| C_{p,n|\sigma_i} \right\| \right\}} \right) \\
&\leq 2\mathbb{E} \left[\sum_{i=k}^{N} \max_{k \leq j \leq N} \left| Z_{T_j} \right| \left| Z_{T_{i+1}} \right| \left(\sum_{i=k}^{N-1} \mathbf{1}_{\left\{ \left| Z_{T_i} - C_{p,n|\sigma_i}(\Lambda_i) \right| \leq \varepsilon \left\| C_{p,n|\sigma_i} \right\| \right\}} \right) \right]
\end{aligned}
$$

where the last inequality follows from the strong law of large numbers as $\mathbb{E}[\max_{k \leq j \leq N} \left| Z_{T_j} \right|^2] < \infty$. We conclude that $\limsup_M |\Delta F_M| = 0$ by letting ε go to 0 and by using that for every $k = 1, \ldots, N$, $\mathbb{P}(Z_{T_k} \in \mathcal{C}_{p,n}) = 0$. ■

The case $q = 1$ proves the strong law of large numbers for the algorithm. Considering that all the paths are actually mixed through the chaos expansion, it is unlikely that the estimators $\frac{1}{M} \sum_{m=1}^{M} Z_{\widehat{\tau}_k^{p,n,(m)}}^{(m)}$ for $k = 1, \ldots, N$ are unbiased. We recall that $U_k^{p,n,M} = \frac{1}{M} \sum_{m=1}^{M} F_k(\widehat{\Lambda}^M, Z^{(m)}, G^{(m)})$ and $Z_{\tau_k^{p,n}} = F_k(\Lambda, Z, G)$. Then,

$$
\begin{aligned}
&\mathbb{E}\left[U_k^{p,n,M} \right] - \mathbb{E}\left[Z_{\tau_k^{p,n}} \right] = \mathbb{E}\left[\frac{1}{M} \sum_{m=1}^{M} \left(F_k(\widehat{\Lambda}^M, Z^{(m)}, G^{(m)}) - F_k(\Lambda, Z^{(m)}, G^{(m)}) \right) \right] \\
&= \mathbb{E}\left[F_k(\widehat{\Lambda}^M, Z^{(1)}, G^{(1)}) - F_k(\Lambda, Z^{(1)}, G^{(1)}) \right]
\end{aligned}
$$

where we have used that all the random variables have the same distribution. Hence, the bias of our estimator is directly linked to the gap between $\widehat{\Lambda}^M$ and the true value Λ. Let $p < p'$, then for any $\alpha \in A_{p,n}^{\otimes d}$, $\alpha \in A_{p',n}^{\otimes d}$ and the corresponding value $\widehat{\lambda}_{k,\alpha}^M$ is the same for p and p'. This means that when p increases, the length of $\widehat{\Lambda}^M$ increases with the first components remaining unchanged. Therefore, $\left| \widehat{\Lambda}^M - \Lambda \right|$ increases with p, which suggests that, for a fixed M, the bias also increases with p. Moreover, it was already noted in Glasserman and Yu [2004b] that for a fixed number of samples M, the mean square error on the coefficients of the regression explodes with the number of regressors. In our framework, this means that, for a fixed M, $\mathbb{E}\left[\left| \widehat{\Lambda}^M - \Lambda \right|^2 \right]$ will increase with p.

4.3.2 Discussion on the rate of convergence

From Theorem 4.5, we deduce that the standard empirical variance estimator applied to our algorithm converges. For every $k = 1, \dots, N$,

$$\lim_{M\to\infty} \quad \frac{1}{M}\sum_{m=1}^{M}\left(Z^{(m)}_{\widehat{\tau}_k^{p,n,(m)}}\right)^2 - \left(\frac{1}{M}\sum_{m=1}^{M} Z^{(m)}_{\widehat{\tau}_k^{p,n,(m)}}\right)^2 = \mathrm{Var}(Z_{\tau_k^{p,n}}) \quad a.s. \tag{12}$$

The convergence rate analysis carried out in Clément et al. [2002] applies steadily to our approach. Then, under suitable assumptions, the vector

$$\left(\sqrt{M}\left(\frac{1}{M}\sum_{m=1}^{M} Z^{(m)}_{\widehat{\tau}_k^{p,n,(m)}} - \mathbb{E}[Z_{\tau_k^{p,n}}]\right)\right)_{k=1,\dots,N} \tag{13}$$

converges in law to a normal distribution with mean zero. As noted in Clément et al. [2002], determining the asymptotic variance directly from the data generated by a single run of the algorithm is almost impossible. From the proof of the central limit theorem for their algorithm, we have, when M goes to infinity, in the L^2 sense

$$\begin{aligned}\sqrt{M}\left(\frac{1}{M}\sum_{m=1}^{M} Z^{(m)}_{\widehat{\tau}_k^{p,n,(m)}} - \mathbb{E}[Z_{\tau_k^{p,n}}]\right) \\ = \sqrt{M}\left(\frac{1}{M}\sum_{m=1}^{M} Z^{(m)}_{\tau_k^{p,n,(m)}} - \phi_k(\Lambda)\right) + \sqrt{M}(\phi_k(\widehat{\Lambda}^M) - \phi_k(\Lambda)).\end{aligned} \tag{14}$$

Remember that $Z^{(m)}_{\tau_k^{p,n,(m)}} = F_k(\Lambda, Z^{(m)}, G^{(m)})$. By the standard central limit theorem, $\sqrt{M}\left(\frac{1}{M}\sum_{m=1}^{M} Z^{(m)}_{\tau_k^{p,n,(m)}} - \phi_k(\Lambda)\right)$ converges in law to a normal distribution with variance $\mathrm{Var}(Z_{\tau_k^{p,n}})$. Then, using the empirical variance of the estimator as a measurement of the algorithm converge actually misses part of the variance since from (12), we know that the empirical variance only takes into account the first term on the r.h.s of (14).

5 Numerical experiments

In this section, we carry out several numerical experiments using our algorithm. In the different tables, the "Price" column corresponds to the value of $U_0^{p,n,M}$ averaged over 25 independent runs of the algorithm and the "Variance" column is the variance of $U_0^{p,n,M}$ computed on these 25 independent runs. The first two experiments, which deal with put options, enable us to compare the accuracy of our method with the standard Longstaff Schwartz algorithm using only the in-the-money paths at each time step, whose price is reported in the "LS" column. Then, we consider more sophisticated truly path dependent options for which the use of the standard Longstaff Schwartz algorithm becomes prohibitive because of the well-known curse of dimensionality. In all the examples, we use $N = n$, ie we do not subdiscretize the grid given by the exercising dates to compute the chaos expansions.

5.1 Examples in the Black Scholes model

The $d-$dimensional Black Scholes model writes for $j \in \{1, \dots, d\}$

$$dS_t^j = S_t^j(r_t dt + \sigma^j L_j dB_t)$$

where B is a Brownian motion with values in $\mathbb{R}^d$, $\sigma = (\sigma^1, \dots, \sigma^d)$ is the vector of volatilities, assumed to be deterministic and positive at all times and L_j is the j-th row of the matrix L defined as a square root of the correlation matrix Γ, given by

$$\Gamma = \begin{pmatrix} 1 & \rho & \dots & \rho \\ \rho & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \rho \\ \rho & \dots & \rho & 1 \end{pmatrix}$$

where $\rho \in]-1/(d-1), 1]$ to ensure that Γ is positive definite.

5.1.1 Assessing the method on the one-dimensional put option

Before investigating more elaborate numerical example, we want to test our method on the Bermudan put option. As standard as this example might be, getting a trustworthy reference price is not an easy task. We rely on prices computed by a convolution method in Lord et al. [2008] and later used as reference prices in Fang and Oosterlee [2009]. We report in Table 1 our values compared to the reference prices for two different volatilities. Our prices are already very close the *true* prices even with a second order expansion $p = 2$. On these examples, we are within 0.2% of the reference prices.

σ	p	M	Price	Variance	Reference price
0.2	2	1E5	10.48	7E-4	10.4795
0.2	2	1E6	10.47	7E-5	
0.2	3	1E5	10.48	6E-4	
0.2	3	1E6	10.47	6E-5	
0.25	2	1E5	11.96	1E-3	11.987
0.25	2	1E6	11.94	2E-4	
0.25	3	1E5	11.96	9E-4	
0.25	3	1E6	11.96	1E-4	

Table 1: Put option with $r = 0.1$, $T = 1$, $K = 110$, $S_0 = 100$ and $N = 10$.

5.1.2 A put basket option

We consider a put basket option with payoff

$$\left(K - \sum_{i=1}^{d} \omega_i S_T^i\right)_+,$$

which can be priced using the classical Longstaff Schwartz algorithm and therefore enables us to test the accuracy of our approach in a multidimensional setting. We test our algorithm in dimension 5 and report the results in Table 2 for different numbers of samples M and different orders p of chaos expansion. The values reported in the "LS" column correspond to the prices computed with the Longstaff Schwartz algorithm with 10^6 samples and using as regression functions the set of polynomials of total order 3 completed with the payoff function.

We notice that an expansion of order $p = 2$ already gives a price fairly close to the "LS" one for a quite reasonable computational time. Increasing p to 3 improves the accuracy only when the number of samples M is also increased. Indeed, we can see that the prices obtained for $K = 90$ and $p = 3$ for small values of M ($M = 5E4$ or $M = 1E5$) are above the dual price. This clearly happens because p is too large compared to M, which induces a bias. We refer the reader to the discussion following Theorem 4.5 for more information on this point. Hence, in a brand new setting, we advise to start with $p = 2$ and to monitor the variance to fix how many Monte Carlo samples are required M. Then, if need be, one can try $p = 3$ with keeping in mind that M should be increased at the same time. In our example, we basically add an order of magnitude to M, when going from $p = 2$ to $p = 3$.

T	K	N	p	M	Price	Variance	LS	Dual price
3	100	20	2	5E4	4.01793	0.00039	4.07	4.3
3	100	20	2	1E5	4.00769	0.00028		
3	100	20	2	1E6	3.99801	2.15E-05		
3	100	20	3	5E4	4.2544	0.00041		
3	100	20	3	1E5	4.1965	0.00024		
3	100	20	3	1E6	4.06587	2.19E-05		
3	90	20	2	5E4	1.29423	0.00013	1.32	1.47
3	90	20	2	1E5	1.27274	0.00011		
3	90	20	2	1E6	1.25166	2.242E-05		
3	90	20	3	5E4	1.52426	8.84E-05		
3	90	20	3	1E5	1.49847	0.00010		
3	90	20	3	1E6	1.31845	2.72E-05		

Table 2: Basket option with $r = 0.05$, $d = 5$, $\sigma^i = 0.2$, $\omega^i = 1/d$, $S_0^i = 100$ and $\rho = 0.2$.

5.1.3 Asian option

For this example, we consider a one dimensional Black Scholes model, $d = 1$. We consider an Asian with payoff $Z_t = (K - X_t)_+$ with $X_0 = S_0$ and for $t > 0$

$$X_t = \frac{1}{t}\int_0^t S_u du.$$

We approximate the continuous time integral by an arithmetic average and we compare our results with the one reported by Hull and White [1993] (in the "HW" column in Table 3), which,

despite being quite old, is still considered as a benchmark by many papers investigating American Asian options.

T	K	N	p	M	Price	Variance	HW
1	45	20	2	1E6	8.55	1E-4	8.55
1	45	20	3	1E6	8.47	1E-4	
1	45	20	3	1E7	8.61	3E-6	
1	50	20	2	1E6	4.81	1E-4	4.89
1	50	20	3	1E6	4.7	1E-4	
1	50	20	3	1E7	4.79	4E-6	
2	45	20	2	1E6	10.63	2E-4	10.62
2	45	20	3	1E6	10.46	2E-4	
2	45	20	3	1E7	10.66	6E-6	
2	50	20	2	1E6	7.28	2E-4	7.33
2	50	20	3	1E6	7.24	2E-4	
2	50	20	3	1E7	7.29	7E-6	

Table 3: Asian option with $r = 0.1$, $d = 1$, $\sigma = 0.3$, $S_0 = 50$ and $N = 40$ (resp. 80) for $T = 1$ (resp. $T = 2$).

It is known that although the payoff does not seem to be Markovian in dimension 1, if we augment the state space and consider the pair (S, X), then the option becomes Markovian again. Hence, Asian options can serve as a good example to assess the efficiency of our algorithm by considering the non Markovian representation of the Asian option in our method. As in the previous example, we notice that a second order expansion $p = 2$ already gives very accurate price, within 1% of the benchmark price computed by Hull and White [1993] using a tree method. Increasing p to 3 does not significantly improve the accuracy of the process but does require to increase the number of Monte Carlo samples.

5.1.4 Moving average option

For this example, we consider a one dimensional Black Scholes model, $d = 1$. We consider a moving average option with payoff $Z_t = (S_t - X_t)_+$ for $t \geq \delta + \ell$ with

$$X_t = \frac{1}{\delta} \int_{t-\delta-\ell}^{t-\ell} S_u du$$

where $\delta > 0$ is the length of the averaging window and ℓ is a delay.

We approximate the continuous time integral by an arithmetic average and compare our results with the benchmark prices computed by Bernhart et al. [2011]. Let $N_\delta = \frac{\delta}{T} N$ and $N_\ell = \frac{\ell}{T} N$. For every $T_i \geq \delta + \ell$, we approximate X_{t_i} by

$$X_{T_i}^N = \frac{1}{N_\delta} \sum_{j=i-N_\delta-N_\ell+1}^{i-N_\ell} S_{T_j}.$$

The benchmark prices reported in the "LS" column come from Bernhart et al. [2011] and were computed using the standard Longstaff Schwartz algorithm with regression factors at time T_i given by

$$\left(S_{T_{i-N_\delta-N_\ell+1}}, S_{T_{i-N_\delta-N_\ell+2}}, \ldots, S_{T_{i-N_\ell}}\right).$$

This leads to a regression problem with N_δ variables, which makes it very CPU demanding. While our approach may also look like a multi variate regression, the main difference lies in the choice of an orthogonal basis function which turns the computation of the coefficients of the regression from a linear system into a bunch of independent Monte Carlo computations. Although this seems a minor change, it is indeed a huge improvement as it breaks the bottleneck of the standard Longstaff Schwartz algorithm and makes it easy to parallelize.

We run two series of tests on the moving average option, which is a typical example of a true path-dependent option in the sense that the size of the underlying Markov process X (see (4)) is basically the number of exercising dates. We report in Table 4 the results for the non delayed option, ie $N_\ell = 0$ and in Table 5 the results for the option with delay. When there is no delay (Table 4), we are able to recover the prices computed with the Longstaff Schwartz method using the full list of regressors. Our results are already very accurate for a chaos expansion of order $p = 2$. To really benefit from a more accurate chaos expansion of order $p = 3$, one also needs to increase the number of samples M to cut down the bias. Note the price > 4.268 in the "LS-price" column for $w = 0.04$. In Bernhart et al. [2011], they did not succeed in computing the price of this option using the Longstaff Schwartz method using the full list of regressors, so they only provided a non Markovian approximation 4.268, which is always below the true price. Hence, the value 4.30329 obtained for $p = 3$ and $M = 10^6$ does make sense. We also report in the column "Dual price" of Table 4 the upper bound obtained from Lelong [2018]. A small gap remains between the lower and upper bounds, but it can be considered as more than acceptable considering the numerical challenges represented by the highly path-dependent products. Since the work by Bernhart et al. [2011], new methods have been developed to handle high dimensional regressions mostly by using machine learning techniques. For instance, we can cite the recent works Becker et al. [2019a,b]. It would be very interesting to use these algorithms to build a price comparator for the non-Markovian settings.

δ	p	M	Price	Variance	LS-Price	Dual price
0.02	2	1E5	3.53118	8.97E-06	3.531	3.76
0.02	2	1E6	3.53863	9.7E-07		
0.02	3	1E5	3.45177	7.05E-06		
0.02	3	1E6	3.52758	7.12E-07		
0.04	2	1E5	4.30318	1.7E-04	> 4.268	4.52
0.04	2	1E6	4.31781	8.82E-07		
0.04	3	1E5	4.18467	1.31E-04		
0.04	3	1E6	4.30239	1.10E-06		

Table 4: Moving average option with $S_0 = 100$, $\sigma = 0.3$, $r = 0.05$, $T = 0.2$, $N = n = 50$ and $\ell = 0$ (no delay).

p	M	Price	Variance
2	5E4	6.62011	7.5E-4
2	1E5	6.67733	2.5E-4
2	1E6	6.74565	2.00E-05
3	5E4	6.28484	4.2E-4
3	1E5	6.36383	3.1E-4
3	1E6	6.65446	8.02E-06

Table 5: Moving average option with $S_0 = 100$, $\sigma = 0.3$, $r = 0.05$, $T = 0.2$, $N = n = 50$, $\ell = 0.08$ ($N_\ell = 20$) and $\delta = 0.02$ ($N_\delta = 5$).

5.2 A put option in the Heston model

We start with a put option in the Heston model to assess the accuracy of our algorithm. We recall the definition of the Heston model

$$dS_t = S_t(r_t dt + \sqrt{\sigma_t}(\rho dW_t^1 + \sqrt{1-\rho^2} dW_t^2))$$
$$d\sigma_t = \kappa(\theta - \sigma_t)dt + \xi\sqrt{\sigma_t} dW_t^1.$$

d	p	M	Price	Variance
1	2	1E5	1.71756	4.68E-05
1	2	1E6	1.69802	7.68E-06
1	2	1E7	1.69699	4.37E-07
1	3	1E5	1.73389	8.43E-05
1	3	1E6	1.72354	6.63E-06
1	3	1E7	1.72274	8.53E-07

Table 6: Put option in the Heston model with $S_0 = K = 100$, $T = 1$, $\sigma_0 = 0.01$, $\xi = 0.2$, $\theta = 0.01$, $\kappa = 2$, $\rho = -0.3$, $r = 0.1$, $N = 20$

For the put option used in the numerical experiments of Table 6, the Longstaff Schwartz algorithm gives 1.74 using degree 3 polynomials for the regression and 10^6 samples. Note that as we only consider in the money paths for the regression step, the payoff function is actually a linear function of the underlying asset — a degree one polynomial. So there is no need to add the payoff function to the regression basis as for more sophisticated options. Obviously, we consider both the asset price and the volatility process as regression factors.

Clearly, we see in the figures of Table 6 that going from $1E6$ to $1E7$ samples does not make any difference on the prices. On the contrary, the prices obtained with $M = 1E5$ are always a little higher, which may look surprising. This is a actually related to the bias phenomenon described at the end of Section 4.3.1. The variance of the coefficients of the chaos expansion is responsible for introducing a bias into the price. To avoid this, one needs to use sufficiently many

Monte Carlo samples to compute the chaos expansions. Anyway, the prices obtained with $p = 3$ and $M = 1E6$ or $M = 1E7$ are within 1% of the standard Longstaff Schwartz price.

5.3 Scalability of the parallel implementation

The scalability tests were run on a BullX DLC supercomputer containing 3204 cores. The code is written in C++ using the OpenMPI library to handle the communication and the PNL library Lelong [2007-2017] to compute the chaos expansions in a generic way for any order p. We report in Table 7 the evolution of the efficiency with respect to the number of resources used. We recall that the efficiency is defined as the ratio between the sequential running time and the product of the parallel running time times the number of resources. Clearly, the efficiency takes values between 0 and 1 and the closer to one, the better. In the example used for the scalability study, we managed to cut down the computational time from an hour and a half to 14 seconds while maintaining the efficiency at almost 0.7, which represents an astonishing improvement in terms of scalability. For a fixed size problem, it is well-known that the efficiency eventually decreases to zero when the number of processors go to infinity as every algorithm has a purely sequential part which becomes predominant in the end. Hence, the efficiency value of 0.7 has to be considered together with the absolute computational time. We refer to Dung Doan et al. [2008], Dung Doan et al. [2010] for experiments on the scalability of different parallel approaches for Bermudan options. Although their framework is a bit different from ours, we can assert that our 0.7 efficiency proves a very good scalability.

#Procs	Time (sec.)	Efficiency
1	4768	1
2	2402	0.99
4	1234	0.97
16	353	0.84
32	173	0.86
64	89	0.84
128	47	0.79
256	24	0.76
512	14	0.68

Table 7: Scalability of the parallel algorithm on the moving average option with delay used of Table 5 with $M = 10^6$ and $p = 3$.

6 Conclusion

In this work, we have presented a new algorithm to price Bermudan option in non Markovian settings: the non Markovian feature can either come from the truly path dependent feature of the option or from the use of rough volatility models for instance. Our algorithm makes it easy to

design a generic American option pricer, actually not more difficult than for a European option pricer. Although this may sound a bit ambitious, our algorithm is designed as a black box taking as inputs sample paths of the underlying multi-dimensional Brownian motion and the associated samples of the payoff process, which is basically the same as for European options. The smart design of our algorithm combined with orthogonality feature of the Wiener chaos expansion leads to an embarrassingly parallel algorithm, in which each node samples a bunch of paths, on which it updates the optimal stopping policy. Each node contributes to the computation of the $\widehat{\lambda}_k^M$'s and at each time step, we make a reduction to get the value of the $\widehat{\lambda}_k^M$'s and then a broadcast makes the coefficients available to everyone. The parallel implementation requires very few communications and therefore shows an impressive efficiency.

The methodology developed in this work in a Brownian setting could be adapted to Lévy processes by adding Charlier polynomials to the Hermite polynomials. We refer to Geiss and Labart [2017] for the use of chaos expansions with jumps.

References

A. Balata and J. Palczewski. Regress-later Monte-Carlo for optimal inventory control with applications in energy. *arXiv e-prints*, page arXiv:1703.06461, Mar 2018.

V. Bally and G. Pages. A quantization algorithm for solving multidimensional discrete-time optimal stopping problems. *Bernoulli*, 9(6):1003–1049, 2003.

S. Becker, P. Cheridito, and A. Jentzen. Deep optimal stopping. *Journal of Machine Learning Research*, 20(74):1–25, 2019a.

S. Becker, P. Cheridito, A. Jentzen, and T. Welti. Solving high-dimensional optimal stopping problems using deep learning, 2019b.

M. Benguigui and F. Baude. Towards parallel and distributed computing on GPU for American basket option pricing. In *4th IEEE International Conference on Cloud Computing Technology and Science Proceedings*, pages 723–728. IEEE, 2012.

M. Bernhart, P. Tankov, and X. Warin. A finite-dimensional approximation for pricing moving average options. *SIAM J. Financial Math.*, 2(1):989–1013, 2011.

B. Bouchard and X. Warin. Monte-carlo valuation of American options: facts and new algorithms to improve existing methods. In R. A. Carmona, P. Del Moral, P. Hu, and N. Oudjane, editors, *Numerical Methods in Finance*, volume 12 of *Springer Proceedings in Mathematics*, pages 215–255. Springer Berlin Heidelberg, 2012.

A. L. Bronstein, G. Pagès, and J. Portès. Multi-asset American options and parallel quantization. *Methodology and Computing in Applied Probability*, 15(3):547–561, 2013.

J. F. Carriere. Valuation of the early-exercise price for options using simulations and nonparametric regression. *Insurance: mathematics and Economics*, 19(1):19–30, 1996.

E. Clément, D. Lamberton, and P. Protter. An analysis of a least squares regression method for American option pricing. *Finance and Stochastics*, 6(4):449–471, 2002.

V. Dung Doan, A. Gaikwad, F. Baude, and M. Bossy. "Gridifying" classification Monte-Carlo algorithm for pricing high-dimensional Bermudan-American options. In *Workshop on high performance computational finance, WHPCF Austin, TX - November 16th, 2008*, pages 1–8, Austin, United States, Nov. 2008.

V. Dung Doan, A. Gaiwad, M. Bossy, F. Baude, and I. Stokes-Rees. Parallel pricing algorithms for multimensional Bermudan/American options using Monte Carlo methods. *Mathematics and Computers in Simulation*, 81(3):568–577, 2010.

F. Fang and C. W. Oosterlee. Pricing early-exercise and discrete barrier options by fourier-cosine series expansions. *Numerische Mathematik*, 114(1):27, 2009.

C. Geiss and C. Labart. Simulation of BSDEs with jumps by Wiener chaos expansion. *Stochastic Processes and their Applications*, 127(3), 2017.

P. Glasserman and B. Yu. Simulation for American options: regression now or regression later? In H. Niederreiter, editor, *Monte Carlo and Quasi-Monte Carlo Methods 2002*, pages 213–226, Berlin, Heidelberg, 2004a. Springer Berlin Heidelberg.

P. Glasserman and B. Yu. Number of paths versus number of basis functions in American option pricing. *The Annals of Applied Probability*, 14(4):2090–2119, 2004b.

E. Gobet, J. G. López-Salas, P. Turkedjiev, and C. Vázquez. Stratified regression Monte-Carlo scheme for semilinear PDEs and BSDEs with large scale parallelization on GPUs. *SIAM Journal on Scientific Computing*, 38(6):C652–C677, 2016.

J. C. Hull and A. D. White. Efficient procedures for valuing European and American path-dependent options. *The Journal of Derivatives*, 1(1):21–31, 1993.

J. Lelong. Pnl : a free scientific library. `https://pnlnum.github.io/pnl`, 2007-2017.

J. Lelong. Dual pricing of American options by Wiener chaos expansion. *SIAM J. Finan. Math.*, 9(2), 2018. doi: 10.1137/16M1102161.

F. Longstaff and R. Schwartz. Valuing American options by simulation : A simple least-square approach. *Review of Financial Studies*, 14:113–147, 2001.

R. Lord, F. Fang, F. Bervoets, and C. W. Oosterlee. A fast and accurate FFT-based method for pricing early-exercise options under Lévy processes. *SIAM Journal on Scientific Computing*, 30(4):1678–1705, 2008.

D. Nualart. Analysis on Wiener space and anticipating stochastic calculus. In B. Springer-Verlag, editor, *Lectures on Probability Theory and Statistics (Saint- Flour, 1995)*, pages 123–227. 1998.

G. Pagès. *Numerical Probability: An Introduction with Applications to Finance*. Springer, 2018. doi: 10.1007/978-3-319-90276-0.

G. Pagès and B. Wilbertz. GPGPUs in computational finance: massive parallel computing for American style options. *Concurrency and Computation: Practice and Experience*, Special Issue:12 p., 2011.

G. Pagès, O. Pironneau, and G. Sall. The parareal algorithm for American options. *Comptes Rendus Mathematique*, 354(11):1132 – 1138, 2016.

J. Tsitsiklis and B. V. Roy. Regression methods for pricing complex American-style options. *IEEE Trans. Neural Netw.*, 12(4):694–703, 2001.