

Variance Reduction Applied to Machine Learning for Pricing Bermudan/American Options in High Dimension

LUDOVIC GOUDENÈGE *

ANDREA MOLENT †

ANTONINO ZANETTE ‡§

Abstract

In this paper we propose an efficient method to compute the price of multi-asset American options, based on Machine Learning, Monte Carlo simulations and variance reduction technique. Specifically, the options we consider are written on a basket of assets, each of them following a Black-Scholes dynamics. In the wake of Ludkovski's approach [?], we implement here a backward dynamic programming algorithm which considers a finite number of uniformly distributed exercise dates. On these dates, the option value is computed as the maximum between the exercise value and the continuation value, which is obtained by means of Gaussian process regression technique and Monte Carlo simulations. Such a method performs well for low dimension baskets but it is not accurate for very high dimension baskets. In order to improve the dimension range, we employ the European option price as a control variate, which allows us to treat very large baskets and moreover to reduce the variance of price estimators. Numerical tests show that the proposed algorithm is fast and reliable, and it can handle also American options on very large baskets of assets, overcoming the problem of the curse of dimensionality.

Premia 22

Keywords:

Machine Learning; Gaussian process regression; Control variate; American options; Monte Carlo methods.

*Fédération de Mathématiques de CentraleSupélec - CNRS FR3487, France - ludovic.goudenège@math.cnrs.fr

†Dipartimento di Scienze Economiche e Statistiche, Università degli Studi di Udine, Italy - andrea.molent@uniud.it

‡Dipartimento di Scienze Economiche e Statistiche, Università degli Studi di Udine, Italy - antonino.zanette@uniud.it

§Corresponding author.

1 Introduction

In this paper we consider one of the most compelling problems among the still open issues in the field of computational finance: pricing and hedging American options in high dimension. From a practical point of view, the efficient numerical evaluation of American options which consider as underlying a baskets of d assets is very challenging because of the so-called “curse of dimensionality”, which avoids the direct application of standard numerical schemes such as finite difference or tree methods. Specifically, this curse of dimensionality means that the computational cost and the memory requirement increase exponentially with the dimension of the problem.

Several new ideas have appeared in this research area, which can be divided into five groups. The first type of approach consists in replacing the underlying diffusion with an approximating process. An example of this mode is given by the stochastic mesh method of Broadie and Glasserman [7], the quantization algorithms of Bally, et al. [3], the stochastic grid method of Jain and Oosterlee [16], the adaptive sparse-grid cubature for a smoothed integrand of Bayer et al. [4], the Markovian projection of Bayer et al. [?]. The second idea makes use of regression on a truncated basis of L^2 in order to compute the conditional expectations. This is done in Longstaff and Schwartz [22] and in Tsisiklis and Van Roy [24]. The third concept consists in exploiting the representation formulas for the conditional expectation using Malliavin calculus. This has been done by Lions and Reigner [21], Bouchard and Touzi [6], Bally et al. [2] Caramellino and Zanette [?] and Abbas-Turki and Lapeyre[1]. Another group of ideas relies on duality-based approaches for Bermudan option pricing, which are proposed by Rogers [23], Haugh and Kogan [15], Andersen and Broadie [?], Schoenmakers et al. [?] and Lelong [20], which can be used to construct bounds on the option value. Finally, the last group consists of methods that employ Machine Learning techniques to learn the continuation value or the stopping rules. This has been proposed by Becker et al. [5], Kohler et al. [18], Lapeyre and Lelong [?] and Ludkovski [?].

European prices can be used as control variate while pricing American options, as done, for example, by Bally et al. [2] and by Caramellino and Zanette [?]. Since multi-asset products are considered, efficiently computing European prices is not trivial and many authors developed valid methods in this field. Some of them focused on computing lower and upper bounds, such as Deelstra et al. [11], Carmona and Durrleman [9], Caldana et al. [8]. Other approaches for basket options are based on the approximation of the sum of the log-normal distributions with a simple distribution by matching some moments, as done by Levy [?], Milevsky and Posner [?, ?], Zhou and Wang [?], Korn and Zeytun [19]. Moreover, an approximation approach is also proposed by Li and Wu [?] for options on several mean-reverting assets. Recently, Glau et al. [13] and Glau et al. [?] consider Chebyshev based methods for pricing. Deep Learning techniques are nowadays widely used in solving large differential equations, which is intimately related to option pricing: recent progresses in this field have been achieved by Han et al. [?], E et al. [?] and Beck et al. [?]. Finally, efficient Monte Carlo approaches are developed by Jourdain and Lelong [17] and more recently by Bayer et al. [4].

In this paper, we propose a new method that combines Machine Learning, Monte Carlo simulations and variance reduction control variate technique. In particular, the use of a control variate makes the method more stable and extends its applicability range to high very large baskets. Moreover, the variance of price estimator is significantly reduced.

First of all, we implement a version of the Ludkovski’s algorithm [?]. Such an algorithm proceeds backward over time by computing the price function on a set of prearranged points which represents possible values of the underlying. In particular, at each time step, it uses a set of Monte Carlo simulations together with Gaussian Process Regression (GPR) to approximate the continuation value at these points. The option

price is then obtained as the maximum between the continuation value and the intrinsic value of the option. We term such an algorithm GPR Monte Carlo (GPR-MC). The GPR-MC algorithm works very well for small baskets (in his paper, Ludkovski considers up to 5 dimensional basket), but it does not for large ones.

In this paper, we show that, if one considers the European price as a control variate, the algorithm improves significantly and the variance of the price estimator is reduced. We term GPR Monte Carlo Control Variate (GPR-MC-CV) this new algorithm. Moreover, in order to compute the European prices, we suggest to use a semi-analytical formula, named GPR-EI formula, introduced by Goudenège et al. in [14], which proves to be efficient when many repeated computations of European prices have to be performed, or alternatively, Quasi-Monte Carlo simulations. Finally, we investigate the benefits brought by control variate technique to the GPR-Tree and GPR-EI approaches introduced by Goudenège et al. [14]. The paper is organized as follows. In Section 2 we present American options for the Black-Scholes d -dimensional model. In Section 3 we briefly review Gaussian Process Regression, we present the GPR-EI formula, the GPR-MC method and the GPR-MC-CV method. Furthermore, we also investigate the use of control variate technique for the GPR-Tree and GPR-EI methods. In Section 4 we report some numerical results about pricing and variance reduction. Finally, Section 5 draws the conclusions.

2 American options in the multi-dimensional Black-Scholes model

An American option with maturity T is a derivative instrument whose holder can exercise the intrinsic optionality at any moment, from inception up to maturity. Let $\mathbf{S} = (\mathbf{S}_t)_{t \in [0, T]}$ denote the d -dimensional underlying process. Such a stochastic process is assumed to randomly evolve according to the multidimensional Black-Scholes model: under the risk neutral measure, such a model is given by the following equation

$$dS_t^i = (r - \eta_i) S_t^i dt + \sigma_i S_t^i dW_t^i, \quad i = 1, \dots, d, \quad (2.1)$$

with $\mathbf{S}_0 = (s_{0,1}, \dots, s_{0,d})^\top \in \mathbb{R}_+^d$ the spot price, r the (constant) spot interest rate, $\boldsymbol{\eta} = (\eta_1, \dots, \eta_d)^\top$ the vector of (constant) dividend rates, $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_d)^\top$ the vector of (constant) volatilities, \mathbf{W} a d -dimensional correlated Brownian motion and ρ_{ij} the instantaneous correlation coefficient between W_t^i and W_t^j . Moreover, let $\Psi(\mathbf{S}_T)$ denote the cash-flow associated with the option. Thus, the price at time t of an American option having maturity T and payoff function $\Psi : \mathbb{R}_+^d \rightarrow \mathbb{R}$ is then

$$v^{AM}(t, \mathbf{x}) = \sup_{\tau \in \mathcal{T}_{t,T}} \mathbb{E}_{t,\mathbf{x}} \left[e^{-r(\tau-t)} \Psi(\mathbf{S}_\tau) \right], \quad (2.2)$$

where $\mathcal{T}_{t,T}$ stands for the set of all the stopping times taking values on $[t, T]$ and $\mathbb{E}_{t,\mathbf{x}}[\cdot]$ is the expectation given all the information at time t and assuming $\mathbf{S}_t = \mathbf{x}$.

For simulation purposes, the d -dimensional Black-Scholes model can be written alternatively using the Cholesky decomposition. Specifically, for $i \in \{1, \dots, d\}$ we can write

$$dS_t^i = S_t^i ((r - \eta_i) dt + \sigma_i \Sigma_i d\mathbf{B}_t), \quad (2.3)$$

where \mathbf{B} is a d -dimensional Brownian motion and Σ_i is the i -th row of the matrix Σ defined as a square root of the correlation matrix Γ , given by

$$\Gamma = \begin{pmatrix} 1 & \rho_{12} & \dots & \rho_{1d} \\ \rho_{21} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \rho_{d1} & \dots & \dots & 1 \end{pmatrix}. \quad (2.4)$$

3 Machine Learning for American options in the multi-dimensional Black-Scholes model

3.1 Gaussian Process Regression

In this Section, we present a brief review of Gaussian Process Regression and for a comprehensive treatment we refer to Rasmussen and Williams [?].

Gaussian Process Regression (GPR), also known as Kriging (see Matheron [?], Journel and Huijbregts [?]), is a class of non-parametric kernel-based probabilistic models which represents the input data as the random observations of a Gaussian stochastic process. The most important advantage of this approach in relation to other parametric regression techniques is that it is possible to effectively exploit a complex dataset which may consist of points sampled randomly in a multidimensional space.

In general, a Gaussian process \mathcal{G} is a collection of random variables defined on a common probability space (Ω, \mathcal{F}, P) , any finite number of which have consistent joint Gaussian distributions. We are interested in Gaussian processes for which the random variables in \mathcal{G} are indexed by a point $\mathbf{x} \in \mathbb{R}^d$, $d \in \mathbb{N}$. Therefore, for all $\mathbf{x} \in \mathbb{R}^d$, $\mathcal{G}(\mathbf{x}) : \Omega \rightarrow \mathbb{R}$ is a Gaussian random variable and if $X = \{\mathbf{x}_p, p = 1, \dots, P\} \subset \mathbb{R}^d$ then $(\mathcal{G}(\mathbf{x}_1), \dots, \mathcal{G}(\mathbf{x}_P))^T$ is a random Gaussian vector. Moreover, a Gaussian process is fully specified by its mean function $\mu(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{R}$ (which is usually assumed to be zero) and by its covariance function $k(\mathbf{x}, \mathbf{x}') : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$.

Now, let us consider a training set \mathcal{D} of P observations (the input data), $\mathcal{D} = \{(\mathbf{x}_p, y_p), p = 1, \dots, P\}$ where $X = \{\mathbf{x}_p, p = 1, \dots, P\} \subset \mathbb{R}^d$ denotes the set of input vectors and $Y = \{y_p, p = 1, \dots, P\} \subset \mathbb{R}$ denotes the set of scalar outputs. These observations are modeled as the realization of the sum of a Gaussian process and a noise source. Specifically,

$$y_p = f_p + \varepsilon_p, \quad (3.1)$$

where $\{f_p = \mathcal{G}(\mathbf{x}_p), p = 1, \dots, P\}$ is a Gaussian process and $\{\varepsilon_p, p = 1, \dots, P\}$ are i.i.d. random variables such that $\varepsilon_p \sim \mathcal{N}(0, \sigma_p^2)$. Moreover, the distribution of $\mathbf{f} = (f_1 \dots f_P)^T$ is assumed to be given by

$$\mathbf{f} \sim \mathcal{N}(\mathbf{0}, K(X, X)), \quad (3.2)$$

where $K(X, X)$ is a $P \times P$ matrix with $K(X, X)_{p_1, p_2} = k(\mathbf{x}_{p_1}, \mathbf{x}_{p_2})$ for $p_1, p_2 = 1, \dots, P$ with $k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ the so called kernel function. Thus

$$\mathbf{y} \sim \mathcal{N}(\mathbf{0}, K(X, X) + \sigma_P^2 I_P), \quad (3.3)$$

where I_P is the $P \times P$ identity matrix.

Now, in addition, let us consider a test set \tilde{X} of M points $\{\tilde{\mathbf{x}}_m, m = 1, \dots, M\}$. The realizations $\tilde{f}_m = \mathcal{G}(\tilde{\mathbf{x}}_m)$ are not known but rather we want to estimate them by exploiting the observed realizations of \mathcal{G} in \mathcal{D} . The *a priori* joint distribution of \mathbf{y} and $\tilde{\mathbf{f}} = (\tilde{f}_1, \dots, \tilde{f}_M)^T$ is given by

$$\begin{bmatrix} \mathbf{y} \\ \tilde{\mathbf{f}} \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} \mathbf{0}_P \\ \mathbf{0}_M \end{bmatrix}, \begin{bmatrix} K(X, X) + \sigma_P^2 I_P & K(X, \tilde{X}) \\ K(\tilde{X}, X) & K(\tilde{X}, \tilde{X}) \end{bmatrix} \right) \quad (3.4)$$

where $K(\tilde{X}, \tilde{X})$ is a $M \times M$ matrix given by $K(\tilde{X}, \tilde{X})_{m_1, m_2} = k(\tilde{\mathbf{x}}_{m_1}, \tilde{\mathbf{x}}_{m_2})$ for $m_1, m_2 = 1, \dots, M$, $K(X, \tilde{X})$ is a $P \times M$ matrix given by $K(X, \tilde{X})_{p, m} = k(\mathbf{x}_p, \tilde{\mathbf{x}}_m)$ for $p = 1, \dots, P$, $m = 1, \dots, M$ and $K(\tilde{X}, X)$ is a $M \times P$ matrix given by $K(\tilde{X}, X)_{m, p} = k(\tilde{\mathbf{x}}_m, \mathbf{x}_p)$ for $m = 1, \dots, M$, $p = 1, \dots, P$.

Since we know the values for the training set, we can consider the conditional distribution of $\tilde{\mathbf{f}}$ given \mathbf{y} . It is possible to prove that $\tilde{\mathbf{f}}|\tilde{X}, \mathbf{y}, X$ follows a Gaussian distribution given by

$$\tilde{\mathbf{f}}|\tilde{X}, \mathbf{y}, X \sim \mathcal{N}(\mathbb{E}[\tilde{\mathbf{f}}|\tilde{X}, \mathbf{y}, X], \text{Cov}[\tilde{\mathbf{f}}|\tilde{X}, \mathbf{y}, X]), \quad (3.5)$$

where

$$\mathbb{E}[\tilde{\mathbf{f}}|\tilde{X}, \mathbf{y}, X] = K(\tilde{X}, X) [K(X, X) + \sigma_P^2 I_P]^{-1} \mathbf{y} \quad (3.6)$$

and

$$\text{Cov}[\tilde{\mathbf{f}}|\tilde{X}, \mathbf{y}, X] = K(\tilde{X}, \tilde{X}) - K(\tilde{X}, X) [K(X, X) + \sigma_P^2 I_P]^{-1} K(X, \tilde{X}). \quad (3.7)$$

Therefore, a natural choice consists in predicting the values $\tilde{\mathbf{f}}$ through $\mathbb{E}[\tilde{\mathbf{f}}|\tilde{X}, \mathbf{y}, X]$. Moreover, by using equation (3.6), one can define a function $f^{GPR} : \mathbb{R}^d \rightarrow \mathbb{R}$ that approximates the function $\mathbf{x}_p \mapsto y_p$ by setting

$$f^{GPR}(\tilde{\mathbf{x}}) = \mathbb{E}[\tilde{\mathbf{f}}|\{\tilde{\mathbf{x}}\}, \mathbf{y}, X] \quad (3.8)$$

$$= \sum_{p=1}^P k(\tilde{\mathbf{x}}, \mathbf{x}^p) \omega_p, \quad (3.9)$$

where $\boldsymbol{\omega} = (\omega_1, \dots, \omega_P)^\top$ is a vector of weights determined by

$$\boldsymbol{\omega} = [K(X, X) + \sigma_P^2 I_P]^{-1} \mathbf{y}. \quad (3.10)$$

The computation in (3.6) requires the knowledge of the covariance function K and of the noise variance σ_P^2 . A commonly used covariance function is the Matern 3/2 kernel $k_{Ma} : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$, which is given by

$$k_{Ma}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \left(1 + \frac{\sqrt{3} \|\mathbf{x} - \mathbf{x}'\|_2}{\sigma_l} \right) \exp \left(-\frac{\sqrt{3} \|\mathbf{x} - \mathbf{x}'\|_2}{\sigma_l} \right) \quad \text{for } \mathbf{x}, \mathbf{x}' \in \mathbb{R}^d, \quad (3.11)$$

where σ_f^2 is called the signal variance and σ_l is called the length-scale. Another possible choice is the Squared Exponential kernel $k_{SE} : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$, which is given by

$$k_{SE}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp \left(-\frac{\|\mathbf{x} - \mathbf{x}'\|_2^2}{2\sigma_l^2} \right) \quad \text{for } \mathbf{x}, \mathbf{x}' \in \mathbb{R}^d. \quad (3.12)$$

In general, the choice of kernel function is performed by using a log-likelihood criterion. The parameters σ_f^2 , σ_l of the kernel function and σ_P^2 of the noise are called hyperparameters and need to be estimated. A common approach is to consider the maximum likelihood estimates which can be obtained by maximizing the log-likelihood function of the training data, that is by maximizing the following function:

$$-\frac{1}{2} \log(\det(K(X, X) + \sigma_P^2 I_P)) - \frac{1}{2} \mathbf{y}^\top [K(X, X) + \sigma_P^2 I_P]^{-1} \mathbf{y}. \quad (3.13)$$

The development of the GPR model can be divided in the training step and the evaluation step (also called testing step). The training step only requires the knowledge of the training set \mathcal{D} and it consists in estimating the hyperparameters and computing the vector of weights $\boldsymbol{\omega}$. The evaluation step can be computed only after the training step has been accomplished and it consists in obtaining the predictions via the computation of $K(\tilde{X}, X) \boldsymbol{\omega}$. We stress out that the training step is independent of the test set \tilde{X} . Thus one can store the values computed during the training step and perform the evaluation step many times with a small computational cost, which is $\mathcal{O}(P \cdot M)$.

Remark 1. We observe that the computation time depends only marginally on the size d of the space where the points lie, as the value of d only impacts in the time taken to calculate distances between the points which appears in the covariance matrix K , that is $\|\mathbf{x} - \mathbf{x}'\|_2$.

3.2 Machine Learning Exact Integration for European options

In order to improve the GPR-MC approach, we employ the European option price as a control variate. Here, we propose to compute such a price by means of the semi-analytical formula introduced by Goudenège et al. [14], that we term GPR-EI formula. This computation is based on two steps. First of all, the payoff function is approximated by means of GPR. Then, the European price is computed as the discounted expected value of the final cash flow, that is a multidimensional integral of the payoff function with respect to the log-underlying process density. Such an integral can be computed by means of a closed formula when replacing the true payoff function with its GPR approximation.

Let us consider a set $Z = \{\mathbf{z}^q, q = 1, \dots, Q\}$ consisting of Q points in \mathbb{R}^d quasi-randomly distributed according to the law of the vector $(\sigma_1 W_T^1, \dots, \sigma_d W_T^d)^\top$. In particular, we define

$$\mathbf{z}_i^q = \sqrt{T} \sigma_i \Sigma_i \mathbf{h}^q, \quad (3.14)$$

where Σ_i is i -th row of the matrix Σ and \mathbf{h}^q is the q -th point of the Halton sequence in \mathbb{R}^d . The Halton sequence allows one to generate points that cover the space more evenly than a random sequence (other low-discrepancy sequence can be considered, such as Solob's or Faure's ones). Let $u : Z \rightarrow \mathbb{R}$ be the function defined by

$$u(\mathbf{z}) := \Psi \left(\mathbf{S}_0 \exp \left(\left(r - \boldsymbol{\eta} - \frac{1}{2} \boldsymbol{\sigma}^2 \right) T + \mathbf{z} \right) \right). \quad (3.15)$$

In a nutshell, the main idea is to approximate the function u by training the GPR method on the set Z . In particular, we employ the Squared Exponential kernel defined in (3.12). Equation (3.9) allows one to approximate the function $u(\cdot)$ by

$$u^{GPR}(\mathbf{z}) = \sum_{q=1}^Q k_{SE}(\mathbf{z}^q, \mathbf{z}) \omega_q, \quad (3.16)$$

where $\omega_1, \dots, \omega_P$ are weights. The continuation value can be computed by integrating the function u^{GPR} against a d -dimensional probability density. The use of the Squared Exponential kernel allows one to easily perform such a calculation by means of a closed formula. Specifically, the GPR-EI method relies on the following Proposition.

Proposition 1. *Let us consider an European option with payoff function Ψ , inception $t = 0$, maturity T , and multidimensional underlying following the dynamics in (2.1) with spot price \mathbf{S}_0 . The price of such an option at $t = 0$ can be approximated by*

$$v^{EU} = e^{-rT} E_{0, \mathbf{S}_0} [\Psi(\mathbf{S}_T)] \approx e^{-rT} \sum_{q=1}^Q \omega_q \sigma_f^2 \sigma_l^d \frac{e^{-\frac{1}{2}(\mathbf{z}^q)^\top (T \cdot \Pi + \sigma_l^2 I_d)^{-1} (\mathbf{z}^q)}}{\sqrt{\det(T \cdot \Pi + \sigma_l^2 I_d)}} \quad (3.17)$$

where σ_f , σ_l , and $\omega_1, \dots, \omega_Q$ are certain constants determined by the GPR approximation of the function $\mathbf{z} \mapsto u(\mathbf{z})$ considering Z as the predictor set, and $\Pi = (\Pi_{i,j})$ is the $d \times d$ covariance matrix of the vector $(\sigma_1 W_T^1, \dots, \sigma_d W_T^d)^\top$, that is $\Pi_{i,j} = \rho_{i,j} \sigma_i \sigma_j$.

The proof of this Proposition is very similar to the one reported in [14].

Despite the GPR-EI formula (3.17) is adapted to compute the option price supposing the spot price to be \mathbf{S}_0 and the time to maturity to be T , it works quite well also for spots close to \mathbf{S}_0 and time to maturity smaller than T . The following Proposition states how to do that.

Proposition 2. *Let us consider and European option with payoff function Ψ , inception $0 < \tilde{t} < T$, maturity T , and multidimensional underlying following the dynamics in (2.1). Let $\tilde{\mathbf{S}}$ be the vector of the spot prices at time \tilde{t} and define $\tilde{\mathbf{z}} \in \mathbb{R}^d$ such that*

$$\tilde{\mathbf{S}} = \mathbf{S}_0 \exp \left(\left(r - \boldsymbol{\eta} - \frac{1}{2} \boldsymbol{\sigma}^2 \right) \tilde{t} + \tilde{\mathbf{z}} \right). \quad (3.18)$$

The price of such an option at \tilde{t} can be approximated by

$$v^{EU} = e^{-r(T-\tilde{t})} E_{\tilde{t}, \tilde{\mathbf{S}}} [\Psi(\mathbf{S}_T)] \approx e^{-r(T-\tilde{t})} \sum_{q=1}^Q \omega_q \sigma_f^2 \sigma_l^d \frac{e^{-\frac{1}{2}(\mathbf{z}^q - \tilde{\mathbf{z}})^\top ((T-\tilde{t}) \cdot \Pi + \sigma_l^2 I_d)^{-1} (\mathbf{z}^q - \tilde{\mathbf{z}})}}{\sqrt{\det((T-\tilde{t}) \cdot \Pi + \sigma_l^2 I_d)}} \quad (3.19)$$

where σ_f , σ_l , and $\omega_1, \dots, \omega_Q$ and $\Pi = (\Pi_{i,j})$ are defined according to Proposition 1.

The proof of Proposition 2 derives directly from Proposition 1 by considering $\tilde{\mathbf{S}}$ in place of \mathbf{S}_0 . The hyperparameters σ_f , σ_l , and the weights $\omega_1, \dots, \omega_Q$ need to be computed only once and then we can use formulas (3.17) and (3.19) to compute the European prices. The resolution of the linear systems within the exponential factors and the computation of the matrix determinant in (3.17) and (3.19) can be done quite fast by computing the Cholesky decomposition of the matrices $(T - \tilde{t}) \cdot \Pi + \sigma_l^2 I_d$ for each of the few possible values of t , that is $t = 0, t_1, \dots, t_{N-1}$. For this reason, turns out to be faster than repeated Monte Carlo simulations to compute the many European prices to be used as control variate.

3.3 Machine Learning Control Variate algorithm for American options

3.3.1 The GPR Monte Carlo Method

Let us introduce the GPR Monte Carlo approach. We approximate the price of an American option with the price of a Bermudan option on the same basket. Specifically, let N be the number of time steps and $\Delta t = T/N$ the time increment. The discrete exercise dates are $t_n = n \Delta t$, as $n = 1, \dots, N$. If \mathbf{x} represents the vector of the underlying prices at the exercise date t_n , then the price of the Bermudan option v^{BM} is given by

$$v^{BM}(t_n, \mathbf{x}) = \max \left(\Psi(\mathbf{x}), E_{t_n, \mathbf{x}} \left[e^{-r \Delta t} v^{BM}(t_{n+1}, \mathbf{S}_{t_{n+1}}) \right] \right). \quad (3.20)$$

First of all, by knowing the function $v^{BM}(t_{n+1}, \cdot)$, one can compute $v^{BM}(t_n, \cdot)$ by approximation of the expectation in (3.20). In order to do that, we consider a set X^n of P points whose coordinates represent certain possible values for the underlyings at time t_n :

$$X^n = \{\mathbf{x}^{n,p} = (x_1^{n,p}, \dots, x_d^{n,p}), p = 1, \dots, P\} \subset \mathbb{R}^d. \quad (3.21)$$

Suppose now we want to compute $v^{BM}(t_n, \cdot)$ but only for $\mathbf{x}^{n,p} \in X^n$. This goal can be achieved by means of a one step Monte Carlo simulation. In particular, for each $\mathbf{x}^{n,p} \in X^n$, we simulate a set of points \tilde{X}_p^n

$$\tilde{X}_p^n = \{\tilde{\mathbf{x}}^{n,p,m} = (\tilde{x}_1^{n,p,m}, \dots, \tilde{x}_d^{n,p,m}), m = 1, \dots, M\} \subset \mathbb{R}^d \quad (3.22)$$

of M possible values for $\mathbf{S}_{t_{n+1}}$ according to the law of $\mathbf{S}_{t_{n+1}} | \mathbf{S}_{t_n} = \mathbf{x}$. In particular, for $i = 1, \dots, d$, $n = 1, \dots, N$, $p = 1, \dots, P$, $m = 1, \dots, M$, we define

$$\tilde{x}_i^{n,p,m} = x_i^{n,p} e^{(r - \eta_i - \frac{1}{2} \sigma_i^2) \Delta t + \sqrt{\Delta t} \sigma_i \Sigma_i \mathbf{G}^{n,p,m}}, \quad (3.23)$$

where $\mathbf{G}^{n,p,m} \sim \mathcal{N}(0, I_d)$ is a standard Gaussian random vector and Σ_i is the i -th row of the matrix Σ , just as in (2.3). Then, the option price can be approximated for each $\mathbf{x}^{n,p} \in X^n$ by

$$\hat{v}^{BM}(t_n, \mathbf{x}^{n,p}) = \max \left(\Psi(\mathbf{x}^{n,p}), \frac{e^{-r\Delta t}}{M} \sum_{m=1}^M v^{BM}(t_{n+1}, \tilde{\mathbf{x}}^{n,p,m}) \right), \quad (3.24)$$

if the quantities $v^{BM}(t_{n+1}, \tilde{\mathbf{x}}^{n,p,m})$ are known for all of these simulated points $\tilde{\mathbf{x}}^{n,p,m}$. If we proceed backward, the function $v^{BM}(t, \cdot)$ is known for $t = T$ since it is equal to the payoff function $\Psi(\cdot)$ and thanks to (3.24) it is known, through an approximation, also for $t = t_{N-1}$ and $\mathbf{x}^{N-1,p} \in X^{N-1}$. In order to assess $v^{BM}(t_{N-2}, \mathbf{x}^{N-2,p})$ for all $\mathbf{x}^{N-2,p} \in X^{N-2}$, and thus going on up to $t = 0$, it is necessary to evaluate the function $v^{BM}(t_{N-2}, \cdot)$ for all the points in $\tilde{X}^{N-2} = \bigcup_{p=1}^P \tilde{X}^{N-2,p}$. This cannot be done directly since we know $\hat{v}^{BM}(t_{N-1}, \cdot)$ only for the points in X^{N-1} and not for all those in \tilde{X}^{N-2} . To overcome this issue, we compute the approximation of the function $\hat{v}^{BM}(t_{N-1}, \cdot)$ by means of the GPR technique. In particular the set X^{N-1} serves as the predictor set and $\{\hat{v}^{BM}(t_{N-1}, \mathbf{x}^{N-1,p}), p = 1, \dots, P\}$ as the response set.

More generally, let $v_n^{BM,GPR}(\cdot)$ be the GPR approximation of $\hat{v}^{BM}(t_n, \cdot)$ trained by considering X^n as the predictor set and $\{\hat{v}^{BM}(t_n, \mathbf{x}^{n,p}), p = 1, \dots, P\}$ as the response set, where \hat{v}^{BM} is defined as in (3.24). Then, we can proceed backward by computing

$$\hat{v}^{BM}(t_{n-1}, \mathbf{x}^{n-1,p}) = \max \left(\Psi(\mathbf{x}^{n-1,p}), \frac{e^{-r\Delta t}}{M} \sum_{m=1}^M v_n^{BM,GPR}(\tilde{\mathbf{x}}^{n-1,p,m}) \right). \quad (3.25)$$

and by computing $v_{n-1}^{BM,GPR}$, that is the GPR approximation of $\hat{v}^{BM}(t_{n-1}, \cdot)$. Finally, the option price at time $t = 0$ is computed through

$$\hat{v}^{BM}(0, \mathbf{S}_0) = \max \left(\Psi(\mathbf{S}_0), \frac{e^{-r\Delta t}}{M} \sum_{m=1}^M \tilde{v}_1^{BM,GPR}(\tilde{\mathbf{x}}^{0,m}) \right) \quad (3.26)$$

where the points $\tilde{\mathbf{x}}^{0,1}, \dots, \tilde{\mathbf{x}}^{0,M}$ are random simulations of \mathbf{S}_{t_1} given by

$$\tilde{x}_i^{0,m} = S_0^i e^{(r - \eta_i - \frac{1}{2}\sigma_i^2)\Delta t + \sqrt{\Delta t}\sigma_i\Sigma_i\mathbf{G}^{0,m}}, \quad (3.27)$$

where $\mathbf{G}^{0,m} \sim \mathcal{N}(0, I_d)$ is a standard Gaussian random vector for any $m \in \{1, \dots, M\}$.

The choice of the sets $X^n, n = 1, \dots, N-1$ is a sensitive question. Similarly to what proposed by Ludkovski [?], here we use a deterministic space-filling sequence based on the Halton sequence. Specifically, let \mathbf{h}^p be the p -th point of the Halton quasi-random sequence in \mathbb{R}^d and Φ^{-1} the inverse cumulative distribution of a standard normal distribution. We define the points $\mathbf{x}^{n,p}$ as follows:

$$\mathbf{x}_i^{n,p} = S_0^i e^{(r - \eta_i - \frac{1}{2}\sigma_i^2)t_n + \sqrt{t_n}\sigma_i\Sigma\Phi^{-1}(H^p)}, \quad (3.28)$$

for $i = 1, \dots, d, n = 1, \dots, N-1$, and $p = 1, \dots, P$. This choice for the sets X^n proves to be the most effective, since the points used to train the GPR algorithm at time t_n are sampled according to the density function of the process \mathbf{S}_{t_n} .

3.3.2 The GPR Monte Carlo Control Variate Method

Let us present the GPR Monte Carlo Control Variate method (GPR-MC-CV), that is our proposed algorithm.

The control variate technique is commonly used to reduce the variance of Monte Carlo estimators, but it can also give its contribution in American pricing. Following Bally et al. [2] and Caramellino and Zanette

[?], we employ the European price as a control variate for the American price. Let us consider an American and an European option with the same payoff function Ψ and maturity T , and let v^{AM}, v^{EU} denote their prices respectively. For a fixed time t and underlying stocks \mathbf{x} , we define the American-European price gap as:

$$v(t, \mathbf{x}) = v^{AM}(t, \mathbf{x}) - v^{EU}(t, \mathbf{x}). \quad (3.29)$$

Then

$$v(T, \mathbf{x}) = 0, \quad (3.30)$$

and it is straightforward to see that

$$v(t, \mathbf{x}) = \sup_{\tau \in \mathcal{T}_{t,T}} E_{t,\mathbf{x}} \left[e^{-r(\tau-t)} \hat{\Psi}(\tau, \mathbf{S}_\tau) \right]. \quad (3.31)$$

where $\mathcal{T}_{t,T}$ stands for the set of all stopping times taking values in $[t, T]$ and $\hat{\Psi}$ is defined by

$$\hat{\Psi}(t, \mathbf{x}) = \Psi(\mathbf{x}) - v^{EU}(t, \mathbf{x}). \quad (3.32)$$

We stress out that $\hat{\Psi}(T, \mathbf{x}) = 0$ and the function $\hat{\Psi}$ depends on the time variable also. Therefore, in order to numerically evaluate $v^{AM}(0, \mathbf{S}_0)$, one can arrange a dynamic programming principle, based on Bermudan approximation, actually equal to the one in Section 3.3.1 by replacing Ψ with $\hat{\Psi}$. Once the initial price gap $v(0, \mathbf{S}_0)$ has been calculated, one can retrieve the American price by computing

$$v^{AM}(0, \mathbf{S}_0) = v(0, \mathbf{S}_0) + v^{EU}(0, \mathbf{S}_0). \quad (3.33)$$

The sketch of the GPR-MC-CV algorithm is presented here.

```

Preprocessing: compute  $\mathbf{x}^{n,p}$  and  $\tilde{\mathbf{x}}^{n,p,m}$  by using equations (3.28) and (3.23),
                 $v^{EU}(t_n, \mathbf{x}^{n,p})$  and  $\hat{\Psi}(t_n, \mathbf{x}^{n,p})$  by using (3.17), (3.19) and (3.32)
Step  $N-1$ : shaping of  $v_{N-1}^{GPR}(\cdot)$ :
     $\hookrightarrow$  For  $p = 1, \dots, P$  compute  $\hat{v}(t_{N-1}, \mathbf{x}^{N-1,p}) = \hat{\Psi}(\mathbf{x}^{N-1,p})$ 
     $\hookrightarrow$  Define the training set  $\mathcal{D} = \{(\mathbf{x}^p, \hat{v}(t_{N-1}, \mathbf{x}^{N-1,p}))\}, p = 1, \dots, P\}$ 
     $\hookrightarrow$  Train GPR on  $\mathcal{D}$  to obtain  $v_{N-1}^{GPR}(\cdot)$ 
Step  $N-2$ : shaping of  $v_{N-2}^{GPR}(\cdot)$ :
     $\hookrightarrow$  For  $p = 1, \dots, P$  compute
         $\hat{v}(t_{N-2}, \mathbf{x}^{N-2,p}) = \max \left( \hat{\Psi}(\mathbf{x}^{N-2,p}), \frac{e^{-r\Delta t}}{M} \sum_{m=1}^M v_{N-1}^{GPR}(\tilde{\mathbf{x}}^{N-2,p,m}) \right)$ 
     $\hookrightarrow$  Define the training set  $\mathcal{D} = \{(\mathbf{x}^p, \hat{v}(t_{N-2}, \mathbf{x}^p))\}, p = 1, \dots, P\}$ 
     $\hookrightarrow$  Train GPR on  $\mathcal{D}$  to obtain  $v_{N-2}^{GPR}(\cdot)$ 

 $\vdots \leftarrow$  Steps  $n = N-3, \dots, 1$  [ replace  $N-2$  with  $n$  and  $N-1$  with  $n+1$ ; ]

Step 0: computation of the price:

```

$$\hat{v}(0, \mathbf{S}_0) = \max \left(\Psi(\mathbf{S}_0), \frac{e^{-r\Delta t}}{M} \sum_{m=1}^M v_1^{GPR}(\tilde{\mathbf{x}}^{0,m}) \right)$$

$$v^{BM}(0, \mathbf{S}_0) = \hat{v}(0, \mathbf{S}_0) + v^{EU}(0, \mathbf{S}_0)$$

Remark 2. We observe that the Monte Carlo evaluation of the continuation value can be easily parallelized since the summations in (3.25), are independent of each other and can be calculated separately. Thus, this feature allows one to significantly reduce the computational time.

Remark 3. As observed by Ludkovsi [?], the main computational cost is due to the training of the GPR model, which is proportional to the cube of the observation amount. In our case, this training has to be performed one time to compute the European prices with a cost $\mathcal{O}(Q^3)$ (with Q the number of points employed in European price computation), and $N-2$ times within the algorithm to approximate the American-European gap at a give time, thus $\mathcal{O}(N \cdot P^3)$ (with N the number of time steps and P the number of points used to train the GPR models at each time step). On the other hand, the cost of the Monte Carlo step depends on both the number of evaluations to be performed and to the number of points employed: the cost for such a step is $\mathcal{O}(N \cdot P \cdot M)$ (with M the number of Monte Carlo simulations employed in estimating the continuation gap value). Finally, we observe that if we compute the European prices by using M' Monte Carlo simulations instead of by using the GPR-EI formula, then the cost $\mathcal{O}(Q^3)$ is replaced by $\mathcal{O}(N \cdot P \cdot M')$.

3.3.3 The Control Variate for GPR-Tree and GRP-EI

Although the control variable technique was initially conceived as a variance reduction techniques for Monte Carlo methods, it can also be a valid support in other contexts. We investigate the benefits brought by this technique to the GPR-Tree and GPR-EI techniques introduced by Goudenège et al. [14] for pricing American options in high dimension. In particular, as proposed for the GPR-MC method, we use the European price as a control variate and we employ GPR-Tree (or GPR-EI) to compute the American-European price gap. Let us give a brief introduction of these two numerical approaches. We refer the interested reader to [14] for more details.

The GPR-Tree method is similar to the GPR-MC method here proposed. The main difference consists in the use of a tree step in place of random simulations to compute the continuation value. In particular, for each time step t_n and for each point \mathbf{x}^p , 2^d future values are generated according to the tree method proposed by Ekvall [12], in place of Monte Carlo simulations. Such a method is particularly efficient when the dimension d is low (that is, indicatively, it does not exceed 10).

The GPR-EI method differs from both the GPR-MC and GPR-Tree methods for three reasons. First of all, the predictors employed in the GPR step are related to the logarithms of the underlying value. Then, the continuation value at these points is computed through a closed formula which comes from an exact integration. Finally, the GPR-EI method employs the Squared Exponential kernel, which is given by

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2\sigma_l^2} \|\mathbf{x} - \mathbf{x}'\|_2^2\right), \quad (3.34)$$

for $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$, where d is the dimension of the regression problem.

4 Numerical Results

In this Section we report some numerical results in order to investigate the effectiveness of the proposed Machine Learning algorithm for pricing American options in the multi-dimensional Black-Scholes model.

First of all, we compare the GPR-MC and GPR-MC-CV methods considering Geometric and Arithmetic basket put options and then we focus on a Call on the Maximum option. Moreover, we study the benefits of using the control variable also for GPR-Tree and GPR-EI methods. Finally, we investigate the variance

of the price estimators about the two methods. We stress out that the GPR-Tree method is interesting only for low dimension options: when d exceeds 10, the method still works, but computational times grow exponentially.

4.1 Geometric and Arithmetic Basket Put Options

In this test we focus on two payoff that depend on the mean of the underlyings. Specifically, we consider the following payoff examples:

- Geometric basket Put

$$\Psi(\mathbf{S}_T) = \left(K - \left(\prod_{i=1}^d S_T^i \right)^{\frac{1}{d}} \right)_+,$$

- Arithmetic basket Put

$$\Psi(\mathbf{S}_T) = \left(K - \frac{1}{d} \sum_{i=1}^d S_T^i \right)_+.$$

We consider both the GPR-MC and the GPR-MC-CV method in order to investigate the benefits induced by the control variate technique. We consider the same parameters as in [14]: $T = 1$, $S_i = 100$, $K = 100$, $r = 0.05$, equal (null) dividend rates $\eta_i = 0.0$, equal volatilities $\sigma_i = 0.2$, equal correlations $\rho_{ij} = 0.2$ and $N = 10$ exercise dates. Moreover, we consider $P = 250, 500$ or 1000 points, $M = 10^3, 10^4$ or 10^5 Monte Carlo simulations and $Q = 10000$ points for the computation of the European prices with the GPR-EI formula. As opposed to the other input parameters, we vary the dimension d , considering $d = 2, 5, 10, 20, 40$ and 100 . The algorithm has been implemented in MATLAB and computations have been performed on a server which employs a 2.40 GHz Intel[®] Xeon[®] processor (Gold 6148, Skylake) and 20 GB of RAM.

We present now the numerical results for the two payoff examples. First of all, let us present the European results, obtained by means of the GPR-EI formula. Table 1 reports the prices, changing the dimension d and the number of employed points Q . Moreover, we also report a Benchmark price computed by Monte Carlo simulation considering 10^6 samples (95% confidence intervals are ± 0.01 for all the benchmark values.). As we can see with only 1000 points we can obtain accurate results in any considered dimension.

Let us now focus on the American results. As far as the Geometric basket Put is considered, it is possible to reduce the problem of pricing in the d -dimensional model to a one dimensional American Put option in the Black-Scholes model with opportune parameters. The price of such a one dimensional American option can be computed in a easy way, for example by using the CRR algorithm with 1000 steps (see Cox et al. [10]). Therefore in this case we have a reliable benchmark to test the algorithm. Moreover, when d is less than 10 we can also compute the price by means of a multi-dimensional binomial tree (see Ekvall [12]). In particular, the number of steps employed for the binomial tree is equal to 200 when $d = 2$ and to 50 when $d = 5$. For values of d larger than 5, prices cannot be approximated via such a tree, because the memory required for the calculations would be too large. Results are reported in Tables 2 and 3. We observe that both the two algorithms are very accurate in low dimension, despite we are approximating an American option with a Bermudan one. When larger baskets are considered, say $d \geq 40$, the prices obtained with the GPR-MC are less accurate and less stable while changing the number of points P and the number of Monte Carlo simulations M . The computer processing time of the GPR-MC-CV method are a little higher than those of the GPR-MC because European prices need to be computed.

d	Q	Geometric Basket Put					Arithmetic Basket Put				
		GPR-EI				Bm	GPR-EI				Bm
		250	500	1000	8000		250	500	1000	8000	
2		4.10 (2)	4.11 (3)	4.13 (15)	4.17 (44)	4.18	3.83 (2)	3.85 (1)	3.86 (12)	3.90 (41)	3.92
5		2.90 (2)	2.98 (1)	3.01 (3)	3.04 (24)	3.06	2.49 (1)	2.57 (1)	2.60 (3)	2.63 (26)	2.64
10		2.48 (1)	2.45 (1)	2.52 (3)	2.59 (26)	2.59	2.01 (1)	2.03 (1)	2.08 (3)	2.13 (25)	2.14
20		2.28 (1)	2.33 (1)	2.26 (4)	2.31 (31)	2.33	1.81 (1)	1.84 (1)	1.80 (4)	1.85 (26)	1.86
40		2.12 (1)	2.18 (1)	2.21 (5)	2.17 (44)	2.20	1.73 (1)	1.74 (1)	1.73 (4)	1.71 (37)	1.72
100		2.03 (1)	2.07 (1)	2.09 (7)	2.08 (43)	2.11	1.93 (1)	1.63 (1)	1.67 (5)	1.62 (35)	1.63

Table 1: European price results for the Geometric and Arithmetic Basket Put option obtained by using the GPR-EI formula. In the last column the prices obtained by using a Monte Carlo simulation. The values in brackets are the computational times (in seconds).

We also stress out that the computer processing time increase little with the size of the problem. This is due to the fact that the dimension affects significantly only the computational time of the Monte Carlo step while the GPR step is only minimally distressed (see Remark 1).

Table 4 and 5 report the results for the GPR-Tree and GPR-EI methods employing or not the control variate technique. By comparing the results of the two Tables, we observe that the option prices for $d \leq 10$ are very similar: in this case control variate technique is not crucial to improve convergence. As opposed to that, GPR-EI benefits sensitively from control variate technique when high values of d are considered.

As opposed to the Geometric basket Put option, we have no method to obtain a fully reliable benchmark when dealing with an Arithmetic basket Put option. However, for small values of d , a reference price can be obtained by means of a multidimensional tree method (see Ekvall [12]), just as shown for the Geometric case. Results are reported in Tables 6 and 7. The conclusions that we can draw in this case are similar to those for the Geometric case: both the two methods are accurate in low dimension, while the control variate method is more effective in high dimension.

Table 8 and 9 report the results for the GPR-Tree and GPR-EI methods employing or not the control variate technique. Just as for the Geometric put option we observe that the option prices for $d \leq 10$ are very similar: in this case control variate technique is not crucial to improve convergence. As opposed to that, control variate technique has an impact on GPR-EI results when high values of d are considered. Anyway, in this case, due to the lack of a benchmark price, it is difficult to draw clear cut conclusions.

4.2 Call on the Maximum option

Let us consider a Call on the Maximum of d -assets American option, whose payoff is given by:

$$\Psi(\mathbf{S}_T) = \left(\max_{i=1 \dots d} S_T^i - K \right)_+.$$

The Call on the Maximum setting is particularly interesting for investigating scalability of our approaches in the dimension d of the problem. As observed by Ludkovski [?], as opposed to basket Put options, the

d	P			250			500			1000			Ekvall	Bm
	10^3	10^4	10^5	10^3	10^4	10^5	10^3	10^4	10^5	10^3	10^4	10^5		
2	4.54 (8)	4.57 (36)	4.58 (347)	4.57 (19)	4.59 (140)	4.57 (1340)	4.52 (72)	4.60 (605)	4.57 (5121)				4.62	4.62
5	3.55 (8)	3.43 (42)	3.44 (498)	3.43 (25)	3.45 (146)	3.44 (1378)	3.45 (56)	3.42 (508)	3.43 (4761)				3.44	3.45
10	2.99 (9)	3.03 (46)	3.02 (431)	2.96 (29)	2.98 (142)	2.95 (1517)	2.98 (55)	2.95 (616)	2.96 (5281)					2.97
20	2.68 (10)	2.68 (85)	2.69 (463)	2.75 (30)	2.70 (210)	2.72 (1441)	2.72 (64)	2.69 (597)	2.70 (5598)					2.70
40	2.71 (14)	2.58 (104)	2.58 (621)	2.60 (24)	2.61 (263)	2.62 (2094)	2.51 (74)	2.55 (655)	2.54 (6373)					2.56
100	2.50 (26)	2.51 (110)	2.50 (1822)	2.48 (42)	2.45 (321)	2.45 (3817)	2.43 (112)	2.45 (892)	2.43 (12410)					2.47

Table 2: American price results for a Geometric basket Put option obtained by using the GPR-MC method. In the last column the exact benchmark. The values in brackets are the computational times (in seconds).

d	P			250			500			1000			Ekvall	Bm
	10^3	10^4	10^5	10^3	10^4	10^5	10^3	10^4	10^5	10^3	10^4	10^5		
2	4.58 (19)	4.57 (45)	4.57 (382)	4.58 (37)	4.57 (133)	4.57 (1445)	4.57 (73)	4.57 (656)	4.57 (6420)				4.62	4.62
5	3.41 (19)	3.41 (43)	3.41 (387)	3.41 (33)	3.40 (129)	3.41 (1445)	3.40 (79)	3.41 (649)	3.41 (5206)				3.44	3.45
10	2.95 (18)	2.95 (44)	2.95 (423)	2.94 (35)	2.95 (157)	2.95 (1680)	2.94 (78)	2.94 (764)	2.95 (5782)					2.97
20	2.68 (22)	2.69 (52)	2.68 (523)	2.69 (34)	2.69 (174)	2.69 (1629)	2.71 (86)	2.71 (687)	2.71 (5950)					2.70
40	2.55 (28)	2.54 (71)	2.54 (752)	2.55 (50)	2.54 (233)	2.55 (2028)	2.55 (121)	2.56 (760)	2.55 (8616)					2.56
100	2.46 (46)	2.46 (117)	2.46 (1398)	2.50 (82)	2.48 (322)	2.48 (4211)	2.47 (175)	2.48 (875)	2.48 (10171)					2.47

Table 3: American price results for a Geometric basket Put option obtained by using the GPR-MC-CV method. In the last column the exact benchmark. The values in brackets are the computational times (in seconds).

d	P	GPR-Tree			GPR-EI			Ekvall	Bm
		250	500	1000	250	500	1000		
2		4.61 (4)	4.61 (7)	4.61 (22)	4.58 (4)	4.58 (9)	4.57 (26)	4.62	4.62
5		3.44 (9)	3.43 (15)	3.44 (23)	3.40 (4)	3.43 (14)	3.41 (27)	3.44	3.45
10		3.00 (10)	2.96 (33)	2.93 (60)	2.85 (4)	2.88 (9)	2.93 (30)		2.97
20					2.63 (4)	2.73 (9)	2.63 (29)		2.70
40					2.45 (4)	2.52 (10)	2.53 (38)		2.56
100					2.27 (5)	2.32 (15)	2.39 (45)		2.47

Table 4: American price results for a Geometric basket Put option obtained by using the GPR-Tree and GPR-EI methods (without control variate). In the last column the exact benchmark. The values in brackets are the computational times (in seconds).

d	P	GPR-Tree			GPR-EI			Ekvall	Bm
		250	500	1000	250	500	1000		
2		4.58 (16)	4.58 (33)	4.58 (64)	4.57 (17)	4.57 (18)	4.57 (24)	4.62	4.62
5		3.42 (15)	3.41 (19)	3.41 (35)	3.41 (13)	3.41 (15)	3.40 (24)	3.44	3.45
10		2.94 (18)	2.94 (31)	2.94 (72)	2.91 (12)	2.93 (13)	2.93 (25)		2.97
20					2.65 (13)	2.67 (19)	2.64 (41)		2.70
40					2.54 (19)	2.57 (33)	2.54 (57)		2.56
100					2.47 (20)	2.46 (29)	2.47 (58)		2.47

Table 5: American price results for a Geometric basket Put option obtained by using the GPR-Tree and GPR-EI methods (with control variate technique). In the last column the exact benchmark. The values in brackets are the computational times (in seconds).

d	P			250			500			1000			Ekvall
	10^3	10^4	10^5	10^3	10^4	10^5	10^3	10^4	10^5	10^3	10^4	10^5	
2	4.34 (8)	4.37 (43)	4.38 (365)	4.37 (21)	4.39 (145)	4.37 (1588)	4.32 (62)	4.40 (767)	4.37 (5183)				4.42
5	3.25 (9)	3.12 (40)	3.14 (380)	3.12 (20)	3.14 (149)	3.13 (1531)	3.14 (60)	3.11 (565)	3.11 (5713)				3.15
10	2.64 (11)	2.69 (40)	2.67 (419)	2.65 (18)	2.66 (149)	2.64 (1551)	2.66 (55)	2.62 (641)	2.63 (5149)				
20	2.27 (17)	2.27 (68)	2.28 (631)	2.39 (22)	2.35 (164)	2.36 (1626)	2.39 (70)	2.36 (620)	2.37 (5817)				
40	2.21 (16)	2.11 (94)	2.10 (780)	2.17 (35)	2.19 (226)	2.19 (2165)	2.15 (105)	2.19 (692)	2.18 (8739)				
100	1.94 (20)	1.95 (110)	1.94 (1494)	1.94 (34)	1.93 (306)	1.92 (3452)	1.95 (91)	1.97 (884)	1.95 (9820)				

Table 6: American price results for a Arithmetic basket Put option obtained by using the GPR-MC . In the last column the exact benchmark. The values in brackets are the computational times (in seconds).

d	P			250			500			1000			Ekvall
	10^3	10^4	10^5	10^3	10^4	10^5	10^3	10^4	10^5	10^3	10^4	10^5	
2	4.38 (19)	4.37 (46)	4.37 (366)	4.38 (31)	4.37 (122)	4.37 (1532)	4.38 (71)	4.37 (590)	4.37 (5372)				4.42
5	3.10 (15)	3.11 (40)	3.11 (378)	3.11 (31)	3.10 (130)	3.10 (1514)	3.09 (73)	3.11 (630)	3.11 (5338)				3.15
10	2.62 (18)	2.62 (44)	2.62 (422)	2.60 (33)	2.62 (129)	2.61 (1576)	2.60 (78)	2.61 (668)	2.61 (5785)				
20	2.33 (21)	2.35 (53)	2.34 (534)	2.34 (36)	2.35 (182)	2.35 (1839)	2.37 (86)	2.36 (656)	2.36 (5751)				
40	2.19 (29)	2.19 (71)	2.19 (699)	2.20 (50)	2.18 (255)	2.19 (2083)	2.19 (113)	2.20 (870)	2.19 (7009)				
100	2.09 (46)	2.09 (114)	2.09 (1594)	2.12 (80)	2.09 (293)	2.09 (3452)	2.07 (163)	2.09 (908)	2.09 (11731)				

Table 7: American price results for a Arithmetic basket Put option obtained by using the GPR-MC-CV method. In the last column the exact benchmark. The values in brackets are the computational times (in seconds).

d	P	GPR-Tree			GPR-EI			Ekvall
		250	500	1000	250	500	1000	
2		4.42 (5)	4.42 (9)	4.42 (25)	4.38 (4)	4.38 (9)	4.37 (28)	4.42
5		3.15 (5)	3.12 (9)	3.13 (24)	3.09 (6)	3.12 (9)	3.10 (44)	3.15
10		2.71 (10)	2.64 (21)	2.62 (70)	2.49 (5)	2.56 (9)	2.60 (38)	
20					2.26 (6)	2.31 (14)	2.28 (42)	
40					2.18 (4)	2.18 (10)	2.16 (31)	
100					2.35 (7)	2.01 (13)	2.06 (42)	

Table 8: American price results for a Arithmetic basket Put option obtained by using the GPR-Tree and GPR-EI methods (without control variate). In the last column the exact benchmark. The values in brackets are the computational times (in seconds).

d	P	GPR-Tree			GPR-EI			Ekvall
		250	500	1000	250	500	1000	
2		4.39 (16)	4.39 (20)	4.39 (31)	4.37 (16)	4.37 (18)	4.37 (39)	4.42
5		3.11 (15)	3.11 (23)	3.11 (34)	3.10 (18)	3.10 (24)	3.10 (23)	3.15
10		2.60 (17)	2.61 (31)	2.61 (75)	2.57 (12)	2.59 (16)	2.59 (24)	
20					2.30 (14)	2.33 (19)	2.33 (31)	
40					2.19 (23)	2.22 (28)	2.19 (66)	
100					2.16 (36)	2.13 (34)	2.13 (56)	

Table 9: American price results for a Arithmetic basket Put option obtained by using the GPR-Tree and GPR-EI methods (with control variate technique). In the last column the exact benchmark. The values in brackets are the computational times (in seconds).

d	Q	GPR-EI				Bm
		250	500	1000	8000	
2		10.77 (1)	10.94 (2)	10.99 (12)	11.14 (89)	11.19
5		22.36 (1)	22.68 (2)	22.82 (10)	22.99 (43)	23.04
10		34.37 (1)	34.38 (2)	34.86 (8)	35.49 (43)	35.59
20		48.31 (1)	49.60 (1)	48.57 (7)	49.28 (44)	49.45
30		57.47 (1)	57.46 (1)	57.05 (5)	57.62 (28)	57.68
50		66.65 (1)	67.60 (1)	67.94 (1)	68.13 (76)	68.03
100		80.34 (1)	81.20 (1)	81.45 (5)	82.00 (34)	82.14

Table 10: European price results for a Call on the Maximum option, obtained by using the GPR-EI formula. In the last column the prices obtained by using a Monte Carlo simulation. The values in brackets are the computational times (in seconds).

stopping region of a Call on the Maximum consists of several disconnected pieces and this makes the pricing problem particularly challenging. As done in the previous Section, we consider both the GPR-MC and the GPR-MC-CV method in order to investigate the benefits induced by the use of this technique. We consider the same parameters as those employed by Becker et al. [5]: $T = 3$, $S_i = 100$, $K = 100$, $r = 0.05$, equal dividend rates $\eta_i = 0.1$, equal volatilities $\sigma_i = 0.2$, equal (null) correlations $\rho_{ij} = 0.0$ and $N = 9$ exercise dates. Moreover, we consider $P = 250, 500$ or 1000 points, $M = 10^3, 10^4$ or 10^5 Monte Carlo simulations. As opposed to the other input parameters, we vary the dimension d , considering $d = 2, 5, 10, 20, 30, 50$ and 100 . In this particular case, because of the long maturity and unbounded payoff, the GPR-EI formula is not very accurate when considering high dimension and initial points far from the spot \mathbf{S}_0 , and so we prefer computing the European price by means of Quasi-Monte Carlo simulation with 10^6 random simulations.

First of all, let us present the European results, obtained by means of the GPR-EI formula. Table 10 reports the prices, changing the dimension d and the number of employed points Q . Moreover, we also report a Benchmark price computed by Monte Carlo simulation considering 10^6 samples (95% confidence intervals are ± 0.01 for all the benchmark values).

The aforementioned testing set has also been considered by Becker et al. [5] and therefore we report their results as reference prices. Furthermore, for small values of d , we can approximate the price obtained by means of a multidimensional tree method. Results, which are reported in Tables 11 and 12, are quite meaningful. Both the two methods perform fine in low dimension, but when large baskets are considered outcomes are strongly different. As far as this particular dataset is considered, the GPR-MC approach gives several null results and others very high, which means that the GPR regression is not able to extrapolate the price surface correctly. In particular, this happens when $d \geq 50$. Increasing the number P of points fixes things when $d = 50$ and for $P = 1000$ results are likely, although outside the confidence interval proposed by Becker et al. [5]. Anyway, when $d = 100$ we always obtain null value, showing all the limits of the GPR-MC approach. As opposed to the GPR-MC, the GPR-MC-CV method performs very well for all the considered dimensions and almost all the values obtained with $P = 1000$ and $M = 10^5$ are within the confidence intervals proposed by Becker et al. [5].

Finally, Tables 13 and 14 report the results for the GPR-Tree and the GPR-EI method. The benefits of

the control variable are evident while comparing these two Tables. Results in Table 13 have been obtained without using the control variate: these two methods seems to be not very effective for the particular Bermudan option considered here. Results in Table 14 have been obtained by using the control variate technique As far as the GPR-EI method is concerned, variates control technique improves significantly the results and allows one to obtain accurate prices even in very high dimension.

P		250			500			1000			Becker et al.
d		10^3	10^4	10^5	10^3	10^4	10^5	10^3	10^4	10^5	95% c.i.
2		14.04 (10)	13.89 (32)	13.91 (404)	13.87 (17)	13.87 (128)	13.92 (1156)	13.67 (53)	13.89 (679)	13.92 (4479)	[13.88, 13.91]
5		26.98 (8)	26.65 (36)	26.76 (362)	26.19 (19)	26.54 (144)	26.39 (1633)	26.47 (57)	26.40 (566)	26.43 (4837)	[26.14, 26.17]
10		38.84 (9)	38.96 (42)	38.86 (430)	39.09 (18)	39.31 (137)	39.37 (1363)	38.42 (54)	38.62 (503)	38.59 (4304)	[38.30, 38.37]
20		60.16 (10)	59.79 (72)	59.87 (378)	59.61 (26)	59.66 (141)	59.61 (1431)	58.23 (58)	58.20 (543)	58.21 (4835)	[51.55, 51.80]
30		73.97 (15)	73.73 (73)	73.75 (439)	80.00 (23)	79.60 (185)	79.69 (1789)	73.40 (65)	73.15 (550)	73.22 (5192)	[59.48, 59.87]
50		93.27 (15)	93.43 (84)	93.26 (1058)	21.12 (27)	21.31 (234)	21.30 (2178)	113.61 (69)	113.24 (604)	113.18 (6247)	[69.56, 69.95]
100		0.01 (19)	0.01 (85)	0.01 (1318)	88.44 (32)	88.05 (284)	88.14 (3105)	138.42 (88)	138.18 (747)	138.25 (8602)	[83.36, 83.86]

Table 11: American price results for a Call on the Maximum option obtained by using the GPR-MC . In the last column the exact benchmark. The values in brackets are the computational times (in seconds). In the last column the confidence intervals reported in [5].

P		250			500			1000			Becker et al.
d	M	10^3	10^4	10^5	10^3	10^4	10^5	10^3	10^4	10^5	95% CI
2		13.93 (8)	13.89 (57)	13.90 (349)	13.89 (25)	13.88 (152)	13.91 (1449)	13.86 (64)	13.90 (495)	13.90 (4979)	[13.88, 13.91]
5		26.16 (9)	26.13 (58)	26.14 (369)	26.07 (22)	26.12 (151)	26.11 (1437)	26.13 (54)	26.13 (517)	26.12 (5423)	[26.14, 26.17]
10		38.07 (9)	38.10 (43)	38.11 (446)	38.20 (29)	38.21 (154)	38.21 (1395)	38.26 (72)	38.29 (503)	38.29 (5215)	[38.30, 38.37]
20		51.27 (11)	51.31 (77)	51.33 (551)	51.60 (26)	51.64 (167)	51.65 (1671)	51.56 (73)	51.58 (636)	51.60 (8046)	[51.55, 51.80]
30		59.25 (12)	59.25 (90)	59.25 (709)	59.33 (26)	59.32 (205)	59.32 (1970)	59.54 (82)	59.54 (656)	59.54 (6436)	[59.48, 59.87]
50		70.29 (13)	70.25 (104)	70.23 (992)	69.61 (30)	69.57 (286)	69.55 (2511)	69.73 (69)	69.69 (663)	69.67 (7733)	[69.56, 69.95]
100		82.16 (17)	82.18 (122)	82.16 (1621)	83.86 (38)	83.86 (350)	83.84 (4196)	83.33 (117)	83.34 (925)	83.31 (10506)	[83.36, 83.86]

Table 12: American price results for a Call on the Maximum option obtained by using the GPR-MC-CV method. In the last column the exact benchmark. The values in brackets are the computational times (in seconds). In the last column the confidence intervals reported in [5].

d	P	GPR-Tree			GPR-EI			Becker
		250	500	1000	1000	2000	4000	et al.
2		13.83 (4)	13.83 (8)	13.85 (18)	13.50 (19)	13.51 (49)	13.51 (53)	[13.88, 13.91]
5		25.95 (5)	25.82 (9)	25.78 (22)	25.23 (20)	25.33 (70)	25.39 (60)	[26.14, 26.17]
10		37.76 (7)	37.79 (18)	37.64 (47)	35.90 (21)	36.69 (67)	37.09 (75)	[38.30, 38.37]
20					46.67 (23)	49.31 (73)	49.74 (100)	[51.55, 51.80]
30					53.66 (29)	54.00 (94)	59.14 (111)	[59.48, 59.87]
50					62.17 (30)	25.84 (86)	71.86 (131)	[69.56, 69.95]
100					70.36 (32)	74.84 (145)	51.74 (262)	[83.36, 83.86]

Table 13: American price results for a Call on the Maximum option by using the GPR-Tree and GPR-EI methods (without control variate). In the last column the exact benchmark. The values in brackets are the computational times (in seconds).

d	P	GPR-Tree			GPR-EI			Becker
		250	500	1000	1000	2000	4000	et al.
2		13.79 (20)	13.78 (48)	13.79 (77)	13.89 (22)	13.89 (76)	13.90 (181)	[13.88, 13.91]
5		25.93 (17)	25.90 (21)	25.88 (37)	26.04 (35)	26.08 (168)	26.12 (219)	[26.14, 26.17]
10		38.73 (20)	38.73 (34)	38.62 (71)	38.18 (49)	38.19 (239)	38.26 (297)	[38.30, 38.37]
20					52.41 (171)	51.71 (124)	51.69 (209)	[51.55, 51.80]
30					59.12 (61)	59.12 (154)	59.16 (266)	[59.48, 59.87]
50					69.38 (123)	69.36 (225)	69.49 (424)	[69.56, 69.95]
100					82.99 (164)	83.26 (400)	83.35 (733)	[83.36, 83.86]

Table 14: American price results for a Call on the Maximum option by using the GPR-Tree and GPR-EI methods (with control variate technique). In the last column the exact benchmark. The values in brackets are the computational times (in seconds).

4.3 Variance Reduction

We conclude our numerical investigations by showing the effect of introducing a control variate on the variance of the estimated prices. In particular, we consider the same Geometric Put option as in Section 4.1 and we price the same option 100 different times, changing the seed of the Monte Carlo generator. This allows us to estimate the variance of the price estimator and to make comparisons. Results are available in Tables 15 and 16, that report the estimated standard deviations and their 95% confidence intervals, computed according to the method suggested by Sheskin [?]. It is evident that the standard deviation (and thus the variance) of the prices obtained with the GPR-MC-CV method is several times lower than the one computed with the GPR-MC method. This is also confirmed for all the considered combination of P and M , by the Hartley's F_{max} test (see Sheskin [?]) with a 99% confidence level.

d	P	250		500		1000	
	M	10^3	10^4	10^3	10^4	10^3	10^4
2		69.8 [61.3,81.1]	22.3 [19.6,25.9]	65.7 [57.6,76.3]	20.5 [18.0,23.8]	64.2 [56.3,74.5]	20.2 [17.7,23.4]
5		53.6 [47.1,62.3]	17.2 [15.1,20.0]	55.2 [48.5,64.2]	18.5 [16.2,21.4]	47.0 [41.3,54.6]	14.5 [12.7,16.9]
10		53.2 [46.7,61.8]	16.1 [14.1,18.7]	50.1 [44.0,58.3]	15.0 [13.2,17.4]	48.7 [42.8,56.6]	14.9 [13.1,17.3]
20		53.1 [46.6,61.7]	16.8 [14.7,19.5]	49.9 [43.8,58.0]	16.4 [14.4,19.1]	46.4 [40.8,53.9]	15.2 [13.3,17.7]
40		75.4 [66.2,87.6]	24.6 [21.6,28.5]	57.1 [50.1,66.3]	17.2 [15.1,19.9]	50.1 [44.0,58.2]	15.1 [13.3,17.6]
100		76.8 [67.5,89.3]	24.3 [21.4,28.3]	68.3 [60.0,79.4]	20.6 [18.0,23.8]	61.1 [53.6,70.9]	19.0 [16.7,23.2]

Table 15: Standard deviation for the prices of an American Geometric Basket Put option computed by means of the GPR-MC method (100 repetitions). Values between brackets are 95% confidence intervals for the standard deviation. All results must be multiplied by 10^{-3} .

d	P	250		500		1000	
	M	10^3	10^4	10^3	10^4	10^3	10^4
2		9.1 [8.0,10.5]	2.7 [2.4,3.2]	8.8 [7.7,10.2]	2.6 [2.3,3.1]	8.5 [7.4,9.8]	2.5 [2.2,2.9]
5		7.5 [6.6,8.8]	2.7 [2.4,3.1]	8.4 [7.4,9.8]	2.7 [2.3,3.1]	8.7 [7.7,10.1]	2.3 [2.0,2.7]
10		7.7 [6.7,8.9]	2.4 [2.1,2.8]	8.1 [7.1,9.4]	2.6 [2.3,3.1]	7.1 [6.3,8.3]	2.7 [2.4,3.2]
20		8.0 [7.1,9.3]	2.7 [2.4,3.1]	7.1 [6.2,8.2]	3.3 [2.9,3.8]	6.9 [6.1,8.1]	2.3 [2.0,2.7]
40		7.6 [6.7,8.9]	2.3 [2.0,2.7]	8.7 [7.7,10.2]	2.4 [2.1,2.8]	8.4 [7.3,9.7]	3.4 [3.0,3.9]
100		7.4 [6.5,8.6]	3.2 [2.9,3.8]	8.3 [7.3,9.6]	2.7 [2.3,3.1]	10.5 [9.3,12.2]	3.0 [2.7,3.5]

Table 16: Standard deviation for the prices of an American Geometric Basket Put option computed by means of the GPR-MC-CV method (100 repetitions). Values between brackets are 95% confidence intervals for the standard deviation. All results must be multiplied by 10^{-3} .

5 Conclusions

In this paper we have proposed a new approach to price American options on baskets of assets, each of them following a Black-Scholes dynamics. The method employs Machine Learning technique, Monte Carlo method and variance reduction technique that exploits the European option price as a control variate. The European prices are computed by means of a semi-analytical formula or Quasi-Monte Carlo simulations. Numerical results show that the method is reliable and fast for baskets including up to 100 assets. The use of a control variate improves the algorithm accuracy and reduces the variance of the estimated prices. In certain cases, also the GPR-Tree and GRP-EI methods benefit from the use of a control variate. The computation time is small and shortly growing with respect to the dimension of the basket. Moreover, the algorithm is partially parallelizable and therefore the computing time can be significantly reduced. Machine Learning seems to be a very promising tool for American option pricing in high dimension, overcoming the problem of the curse of dimensionality.

References

- [1] LA Abbas-Turki and Bernard Lapeyre. American options by Malliavin calculus and nonparametric variance and bias reduction methods. *SIAM Journal on Financial Mathematics*, 3(1):479–510, 2012. 2
- [2] Vlad Bally, Lucia Caramellino, and Antonino Zanette. Pricing and hedging American options by Monte Carlo methods using a Malliavin calculus approach. *Monte Carlo Methods and Application*, 11(2):97–133, 2005. 2, 8
- [3] Vlad Bally, Gilles Pagès, and Jacques Printems. First-order schemes in the numerical quantization method. *Mathematical Finance*, 13(1):1–16, 2003. 2
- [4] Christian Bayer, Markus Siebenmorgen, and Raul Tempone. Smoothing the payoff for efficient computation of basket option prices. *Quantitative Finance*, 18(3):491–505, 2018. 2
- [5] Sebastian Becker, Patrick Cheridito, and Arnulf Jentzen. Deep optimal stopping. *Journal of Machine Learning Research*, 20(74):1–25, 2019. 2, 17, 19
- [6] Bruno Bouchard and Nizar Touzi. Discrete-time approximation and Monte-Carlo simulation of backward stochastic differential equations. *Stochastic Processes and their applications*, 111(2):175–206, 2004. 2
- [7] Mark Broadie and Paul Glasserman. Pricing American-style securities using simulation. *Journal of Economic Dynamics and Control*, 21(8-9):1323–1352, 1997. 2
- [8] Ruggero Caldana, Gianluca Fusai, Alessandro Gnoatto, and Martino Grasselli. General closed-form basket option pricing bounds. *Quantitative Finance*, 16(4):535–554, 2016. 2
- [9] René Carmona and Valdo Durrleman. Generalizing the Black-Scholes formula to multivariate contingent claims. *Journal of Computational Finance*, 9(2):43, 2005. 2
- [10] John C Cox, Stephen A Ross, and Mark Rubinstein. Option pricing: A simplified approach. *Journal of Financial Economics*, 7(3):229–263, 1979. 11
- [11] G. Deelstra, J. Liinev, and M. Vanmaele. Pricing of arithmetic basket options by conditioning. *Insurance: Mathematics and Economics*, 34(1):55–77, 2004. 2

- [12] Niklas Ekvall. A lattice approach for pricing of multivariate contingent claims. *European Journal of Operational Research*, 91(2):214–228, 1996. [10](#), [11](#), [12](#)
- [13] Kathrin Glau, Daniel Kressner, and Francesco Statti. Low-rank tensor approximation for Chebyshev interpolation in parametric option pricing. *arXiv preprint arXiv:1902.04367*, 2019. [2](#)
- [14] Ludovic Goudenège, Andrea Molent, and Antonino Zanette. Variance reduction applied to machine learning for pricing bermudan/american options in high dimension. *arXiv: 1903.11275*, 2019. [3](#), [6](#), [10](#), [11](#)
- [15] Martin B Haugh and Leonid Kogan. Pricing American options: a duality approach. *Operations Research*, 52(2):258–270, 2004. [2](#)
- [16] Shashi Jain and Cornelis W Oosterlee. Pricing high-dimensional Bermudan options using the stochastic grid method. *International Journal of Computer Mathematics*, 89(9):1186–1211, 2012. [2](#)
- [17] Benjamin Jourdain and Jérôme Lelong. Robust adaptive importance sampling for normal random vectors. *The Annals of Applied Probability*, 19(5):1687–1718, 2009. [2](#)
- [18] Michael Kohler, Adam Krzyżak, and Nebojsa Todorovic. Pricing of high-dimensional American options by neural networks. *Mathematical Finance*, 20(3):383–410, 2010. [2](#)
- [19] Ralf Korn and Serkan Zeytun. Efficient basket Monte Carlo option pricing via a simple analytical approximation. *Journal of Computational and Applied Mathematics*, 243:48–59, 2013. [2](#)
- [20] Jérôme Lelong. Dual pricing of American options by Wiener chaos expansion. *SIAM Journal on Financial Mathematics*, 9(2):493–519, 2018. [2](#)
- [21] PL Lions and H Regnier. Calcul du prix et des sensibilités d’une option américaine par une méthode de Monte Carlo. *preprint*, 2, 2001. [2](#)
- [22] Francis A Longstaff and Eduardo S Schwartz. Valuing American options by simulation: a simple least-squares approach. *The Review of Financial Studies*, 14(1):113–147, 2001. [2](#)
- [23] Leonard CG Rogers. Monte Carlo valuation of American options. *Mathematical Finance*, 12(3):271–286, 2002. [2](#)
- [24] John N Tsitsiklis and Benjamin Van Roy. Optimal stopping of Markov processes: Hilbert space theory, approximation algorithms, and an application to pricing high-dimensional financial derivatives. *IEEE Transactions on Automatic Control*, 44(10):1840–1851, 1999. [2](#)