Machine Learning for Pricing American Options in High-Dimensional Markovian and non-Markovian models

Ludovic Goudenège *†

Andrea Molent‡

Antonino Zanette§

Premia 22

Abstract

In this paper we propose two efficient techniques which allow one to compute the price of American basket options. In particular, we consider a basket of assets that follow a multi-dimensional Black-Scholes dynamics. The proposed techniques, called GPR Tree (GRP-Tree) and GPR Exact Integration (GPR-EI), are both based on Machine Learning, exploited together with binomial trees or with a closed formula for integration. Moreover, these two methods solve the backward dynamic programming problem considering a Bermudan approximation of the American option. On the exercise dates, the value of the option is first computed as the maximum between the exercise value and the continuation value and then approximated by means of Gaussian Process Regression. The two methods mainly differ in the approach used to compute the continuation value: a single step of binomial tree or integration according to the probability density of the process. Numerical results show that these two methods are accurate and reliable in handling American options on very large baskets of assets. Moreover we also consider the rough Bergomi model, which provides stochastic volatility with memory. Despite this model is only bidimensional, the whole history of the process impacts on the price, and handling all this information is not obvious at all. To this aim, we present how to adapt the GPR-Tree and GPR-EI methods and we focus on pricing American options in this non-Markovian framework.

Keywords: Machine Learning, American Options, Multi-dimensional Black-Scholes Model, Rough Bergomi Model, Binomial Tree Method, Exact Integration.

*This work was supported by a public grant as part of the Investissement d'avenir project, reference ANR-11-LABX-0056-LMH, LabEx LMH.

†Féderation de Mathématiques de CentraleSupélec - CNRS FR3487, France - ludovic.goudenege@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

1 Introduction

Pricing American options is clearly a crucial question of finance but also a challenging one since computing the optimal exercise strategy is not an evident task. This issue is even more exacting when the underling of the option is a multi-dimensional process, such as a baskets of d assets, since in this case the direct application of standard numerical schemes, such as finite difference or tree methods, is not possible because of the exponential growth of the calculation time and the required working memory.

Common approaches in this field can be divided in four groups: techniques which rely on recombinant trees to discretize the underlyings (see [4], [10] and [23]), techniques which employ regression on a truncated basis of L^2 in order to compute the conditional expectations (see [27] and [32]), techniques which exploit Malliavin calculus to obtain representation formulas for the conditional expectation (see [1], [3], [9], and [26]) and techniques which make use of duality-based approaches for Bermudan option pricing (see [20], [25] and [31]).

Recently, Machine Learning algorithms (Rasmussen and Williams [33]) and Deep Learning techniques (Nielsen [29]) have found great application in this sector of option pricing.

Neural networks are used by Kohler et al. [24] to price American options based on several underlyings. Deep Learning techniques are nowadays widely used in solving large differential equations, which is intimately related to option pricing. In particular, Han et al. [19] introduce a Deep Learning-based approach that can handle general high-dimensional parabolic PDEs. E et al. [13] propose an algorithm for solving parabolic partial differential equations and backward stochastic differential equations in high dimension. Beck et al. [7] introduce a method for solving high-dimensional fully nonlinear second-order PDEs. As far as American options in high dimension are concerned, Becker et al. [8] develop a Deep Learning method for optimal stopping problems which directly learns the optimal stopping rule from Monte Carlo samples.

Also Machine Learning techniques have made their contribution. For example, Dixon and Crépey present a multi-Gaussian process regression for estimating portfolio risk, and in particular the associated CVA. De Spiegeleer et al. [12] propose to apply Gaussian Process Regression (GPR) to predict the price of the derivatives from a training set made of observed prices for particular combinations of model parameters. Ludkovski [28] proposes to use GPR meta-models for fitting the continuation values of Bermudan options. Similarly, Goudenège et al. [17] propose the GPR-MC, which is a backward induction algorithm that employs Monte Carlo simulations and GPR to compute the price of American options in very high dimension (up to 100). In the insurance context, Gan [15] studies the pricing of a large portfolio of Variable Annuities in the Black-Scholes model by using clustering and GPR. Moreover, Gan and Lin [16] propose a novel approach that combines clustering technique and GPR to efficiently evaluate policies considering nested simulations.

In this paper we present two numerical techniques which upgrade the GPR-MC approach by replacing the Monte Carlo based computation of the continuation value respectively with a tree step and with an exact integration step. In particular, the algorithms we propose proceed backward over time and compute the price function only on a set of predetermined points. At each time step, a binomial tree step or a closed formula for integration are used together with 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 and the algorithms proceed backward. For the sake of simplicity, we name these new approaches Gaussian Process Regression - Tree (GPR-Tree) and Gaussian Process Regression - Exact Integration (GPR-EI). We observe that the use of the GPR method to extrapolate the option value is particularly efficient in terms of computing time with respect to other techniques such as Neural Networks, especially because a small dataset is considered here. Moreover, Le Gratiet et Garnier [18] developed recent convergence results about GPR, extending the outcomes of Rasmussen and Williams [33], and founding the convergence rate when different

kernels are employed.

In comparison to the Monte Carlo based methods introduced in [28] and [17], the techniques proposed here show some significant differences. Specifically, as far as the GPR-Tree method is concerned, using a tree step in place of a Monte Carlo simulation to estimate the continuation value can be very efficient when small baskets are considered since in this case the number of employed samples is very small. In reverse, such a number, which grows exponentially with the size d of the basket, becomes huge when d is large, which makes the algorithm unsuitable for very large baskets. On the contrary, the number of Monte Carlo simulations can be managed with much more freedom in any dimension. As far as the GPR-EI method is concerned, the use of an exact integration formula speeds up the computation of the continuation value because it avoids the time demanding repeated evaluations of the GPR surface at next time step. In reverse, the GPR models employed within the GPR-EI method cannot be trained by using as predictors the underlyings values directly, as done in GPR-MC or GPR-Tree, but a log type transformation must be applied before. Numerical results show that because of this, more input points are required to obtain the same accuracy, but this is not bothersome at all since the method is still faster than the others, especially in when large baskets are considered.

In order to demonstrate the wide applicability of the GPR methods, we also consider the rough Bergomi model, which is a non-Markovian model with stochastic volatility. Such a model, introduced by Bayer et al. [5] stood out for explaining implied volatility smiles and other phenomena in the pricing of European options. The non-Markovian property of the model makes it difficult to implement a methodologically correct approach to address the valuation of American options. The literature in this framework is really poor. Horvat et al. [22] propose an approach based on Donsker's approximation for fractional Brownian motion and on a tree with exponential complexity. More recently, Bayer et al. [6] introduce a method based on Monte Carlo simulation and exercise Rate Optimization.

Numerical results show that both the GPR-Tree and the GPR-EI methods are accurate and reliable in the multi-dimensional Black-Scholes model. Moreover the computational times with respect to the GPR-MC method are improved. The GPR-Tree and the GPR-EI methods prove its accuracy also when applied to the rough Bergomi model.

The reminder of the paper is organized as follows. Section 2 presents American options in the multi-dimensional Black-Scholes model. Section 3 introduces the GPR method. Section 4 and Section 5 introduce the GPR-Tree and the GPR-EI methods for the multi-dimensional Black-Scholes model respectively. Section 6 presents the American options in the rough Bergomi model. Section 7 and Section 8 introduce the GPR-Tree and the GPR-EI methods for the rough Bergomi model. Section 9 reports some numerical results. Section 10 draws some 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 before maturity. Let $\mathbf{S} = (\mathbf{S}_t)_{t\in[0,T]}$ denote the d-dimensional underlying process, which is supposed to randomly evolve according to the multi-dimensional Black-Scholes model: under the risk neutral probability, such a model is given by the following equation

$$dS_t^i = r\, S_t^i\, dt + \sigma_i\, S_t^i\, dW_t^i, \qquad i = 1, \dots, d, \tag{2.1}$$

with $\mathbf{S}_0 = \left(s_0^1, \dots, s_0^d\right)^\top \in \mathbb{R}_+^d$ the spot price, r the (constant) interest rate, $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_d)^\top$ the vector of 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 at maturity T. Thus, the price at time t of an American option having maturity T and payoff function $\Psi : \mathbb{R}_+^d \to \mathbb{R}$ is then

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

where $\mathcal{T}_{t,T}$ stands for the set of all the stopping times taking values on $[t,T]$ and $\mathbb{E}_{t,\mathbf{x}}[\cdot]$ represents the expectation given all the information at time t and in particular 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 $i = 1, \ldots, d$ we can write

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

where $\mathbf{B}$ is a d-dimensional uncorrelated 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

$$\Gamma = \begin{pmatrix} 1 & \rho_{12} & \ldots & \rho_{1d} \\ \rho_{21} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \rho_{d1} & \ldots & \ldots & 1 \end{pmatrix}. \tag{2.4}$$

3 Gaussian Process Regression

We briefly recall the main features of Gaussian Process Regression. The interested reader can find more details in the seminal book of Rasmussen and Williams [33]. Gaussian Process Regression (GPR), also known as Kriging, is a class of non-parametric kernel-based probabilistic models which represents the input data as the random observations of a Gaussian stochastic process in order to perform predictions. One of the main advantages of this technique is that it allows one to exploit a complex dataset in high dimension. Moreover, data do not need to be sampled according to a regular grid.

Let $X = \{\mathbf{x}_p,\ p = 1, \ldots, P\} \subset \mathbb{R}^d$ be the set of predictors and $Y = \{y_p,\ p = 1, \ldots, P\} \subset \mathbb{R}$ be the set of scalar outputs. These observations are known and they are modeled as the realization of the sum of a Gaussian process $\mathcal{G}$ and a Gaussian noise source ε. Specifically, $y_p = f_p + \varepsilon_p$ where $\{f_p = \mathcal{G}(\mathbf{x}_p), p = 1, \ldots, P\}$ is a Gaussian process and $\{\varepsilon_p, p = 1, \ldots, P\}$ are i.i.d. random variables such that $\varepsilon_p \sim \mathcal{N}(0, \sigma_P^2)$. Then, the distribution of $\mathbf{y} = (y_1 \ldots y_P)^\top$ is assumed to be given by

$$\mathbf{y} \sim \mathcal{N}\left(\mathbf{0}, K(X,X) + \sigma_P^2 I_P\right), \tag{3.1}$$

with I_P the $P \times P$ identity matrix and K a $P \times P$ matrix determined by $K(X,X)_{p,q} = k(\mathbf{x}_p, \mathbf{x}_q)$, for $p, q \in \{1, \ldots, P\}$. As customary, we set the mean of $\mathbf{y}$ to zero but other options are possible, such as a linear function of the predictors. In this paper, we consider two types of kernel function. The first one is the Squared Exponential (SE) kernel which is given by

$$k_{SE}(\mathbf{a}, \mathbf{b}) = \sigma_f^2 \exp\left(-\sum_{i=1}^d \frac{(a_i - b_i)^2}{2\sigma_l^2} \right),\ \mathbf{a}, \mathbf{b} \in \mathbb{R}^d, \tag{3.2}$$

where σ_f^2 is the signal variance and σ_l is the length-scale parameter. The second kernel function is the Automatic Relevance Determination Squared Exponential (ARD SE) kernel, which is given by

$$k_{ASE}(\mathbf{a}, \mathbf{b}) = \sigma_f^2 \exp\left(-\sum_{i=1}^d \frac{(a_i - b_i)^2}{2\sigma_i^2} \right),\ \mathbf{a}, \mathbf{b} \in \mathbb{R}^d, \tag{3.3}$$

where σ_f^2 is the signal variance and σ_i is the length-scale along the i direction. This second kernel is more effective when the predictors have different relevance in the output, and some authors have used it for removing irrelevant input. Furthermore, we stress out that if $d = 1$, then k_{ASE} and k_{SE} are exactly the same kernel.

Now, in addition, let us consider a test set $\tilde{X}$ of M points $\{\tilde{\mathbf{x}}_m,\ m = 1, \ldots, M\}$. The realizations $\tilde{f}_m = \mathcal{G}(\tilde{\mathbf{x}}_m) + \varepsilon_m$ are not known but we want to predict them by considering the observed realizations of $\mathcal{G}$ in $\mathcal{D}$. The prediction of $\tilde{\mathbf{f}}$ is obtained by its conditional mean value $\mathbb{E}\left[\tilde{\mathbf{f}}|\tilde{X}, \mathbf{y}, X\right]$:

$$\mathbb{E}\left[\tilde{\mathbf{f}}|\tilde{X}, \mathbf{y}, X\right] = K\left(\tilde{X}, X\right) A, \tag{3.4}$$

with $K\left(\tilde{X}, X\right)$ a $M \times P$ matrix given by $K\left(\tilde{X}, X\right)_{m,p} = k\left(\tilde{\mathbf{x}}_m, \mathbf{x}_p\right)$ for $m \in \{1, \ldots, M\}$ and $p \in \{1, \ldots, P\}$, while $A = \left[K(X, X) + \sigma_P^2 I_P\right]^{-1} \mathbf{y}$ is a P-dimensional vector. The parameters σ_f^2, σ_l^2 of the SE kernel (or σ_f^2, $\sigma_1^2, \ldots \sigma_d^2$ of the ARD SE kernel) and σ_P^2 of the noise source are termed hyperparameters and they are estimated by log-likelihood maximization. In this regard, we employ a method which uses a trust-region algorithm with a dense, symmetric rank-1-based, quasi-Newton approximation to the Hessian.

Furthermore, if one considers the set $\tilde{X} = \{\tilde{\mathbf{x}}\}$ as a singleton, then Equation (3.4) allows one to define the GPR prediction function by

$$f^{GPR}(\tilde{\mathbf{x}}) = K(\{\tilde{\mathbf{x}}\}, X) A, \tag{3.5}$$

with $K(\{\tilde{\mathbf{x}}\}, X)$ a P-dimensional vector. Moreover, such a function can be written as a linear combination of P kernel functions as follows

$$f^{GPR}(\tilde{\mathbf{x}}) = \sum_{p=1}^{P} k(\tilde{\mathbf{x}}, \mathbf{x}_p)\, \omega_p, \tag{3.6}$$

where the weights are given by $\omega_p = A_p$, for $p = 1, \ldots, P$.

The development of the GPR method consists in two steps: training and evaluation (also called testing). The former consists in estimating the hyperparameters and computing A while the latter can be performed only after the training and it consists in obtaining the predictions via (3.4).

4 The GPR-Tree method in the multi-dimensional Black-Scholes model

The GPR-Tree method is similar to the GPR-MC method but the diffusion of the underlyings is performed through a step of a binomial tree. In particular, the algorithm proceeds backward over time, approximating the price of the American option with the price of a Bermudan option on the same basket. At each time step, the price function is evaluated only on a set of predetermined points, through a binomial tree step together with ouGPR to approximate the continuation value. Finally, the optionality is exploited by computing the option value as the maximum between the continuation value and the exercise value.

Let N denote the number of time steps, $\Delta t = T/N$ be the time increment and $t_n = n\,\Delta t$ represent the discrete exercise dates for $n = 0, 1, \ldots, N$. At any exercise date t_n, the value of the option is determined by the vector of the underlying prices $\mathbf{S}_{t_n}$ as follows:

$$v(t_n, \mathbf{S}_{t_n}) = \max\left(\Psi(\mathbf{S}_{t_n}), C(t_n, \mathbf{S}_{t_n})\right), \tag{4.1}$$

where C denotes the continuation value of the option and it is given by the following relation:

$$C(t_n, \mathbf{S}_{t_n}) = \mathbb{E}_{t_n, \mathbf{S}_{t_n}}\left[e^{-r\Delta t} v\left(t_{n+1}, \mathbf{S}_{t_{n+1}}\right)\right]. \tag{4.2}$$

We observe that, if the function $v(t_{n+1},\cdot)$ is known, then it is possible to compute $v(t_n,\cdot)$ by approximating the expectation in (4.2). In order to obtain such an approximation, we consider a set X of P points whose elements represent certain possible values for the underlyings $\mathbf{S}$:

$$X = \left\{ \mathbf{x}^p = (x_1^p, \ldots, x_d^p)^\top, p = 1, \ldots, P \right\} \subset \mathbb{R}_+^d, \tag{4.3}$$

where $\mathbb{R}_+^d =]0, +\infty[^d$. Such a set is determined as done by Goudenège et al. [17], that is the elements of X are obtained through a quasi-random simulation of $\mathbf{S}_T$ based on the Halton sequence, which allows one to cover the region of interest, avoiding to leave some uncovered areas and to create useless clusters of points (other low discrepancy sequences may be considered, such as Sobol's one). In particular, let H^p be the p-th point of the Halton quasi-random sequence in $\mathbb{R}^d$ and Φ^{-1} be the inverse cumulative distribution of a standard normal distribution. Then,

$$\mathbf{x}^1 = \mathbf{S}_0, \tag{4.4}$$

and for $p \in \{2, \ldots, P\}$ the components of $\mathbf{x}^p$ are given by

$$x_i^p = S_0^i e^{\left(r - \frac{1}{2}\sigma_i^2\right)T + \sigma_i\sqrt{T}\sum_{j=1}^d \Sigma_{d,j}\Phi^{-1}\left(H_j^p\right)}, \tag{4.5}$$

for $p = 1, \ldots P$ and $i = 1, \ldots, d$. The GPR-Tree method assesses the continuation value through one step of the binomial tree proposed by Ekval [14]. In this case, the set $\tilde{X}^p$ consists of $M = 2^d$ possible values for $\mathbf{S}_{t_{n+1}}$ which are computed as follows:

$$\tilde{x}_i^{p,m} = x_i^p \exp\left(\left(r - \frac{\sigma_i^2}{2}\right)\Delta t + \sigma_i\sqrt{\Delta t}\Sigma_i \mathbf{G}_k\right), \tag{4.6}$$

being $\mathbf{G}_m$ the m-th point of the space $\{-1, +1\}^d$. In particular, if $\mathbf{Y}_m \in \{0, 1\}^d$ is the vector whose components are the digits of the binary representation of $2^d - 1$, then $\mathbf{G}_m = 2\mathbf{Y}_m - 1$. It is worth noticing that, as pointed out in [14], the elements of $\tilde{X}^p$ are equally likely and this simplifies the evaluation of the expected value to the computation of the arithmetic mean of the future values.

Generally speaking, let $\tilde{v}_{n+1}(\cdot)$ be a suitable approximation of $v(t_{n+1},\cdot)$. So, by using the tree step, the price function may be approximated by

$$v_n^{Tree}(\mathbf{x}^p) = \max\left(\Psi(\mathbf{x}^p), \frac{e^{-r\Delta t}}{2^d}\sum_{m=1}^{2^d}\tilde{v}_{n+1}(\tilde{\mathbf{x}}^{p,m})\right). \tag{4.7}$$

If we proceed backward, the function $v(t,\cdot)$ is known at maturity since it is given by the payoff function $\Psi(\cdot)$ and so $v_{N-1}^{Tree}(\cdot)$ can be computed at all the points of X through equation (4.7) by replacing $\tilde{v}_{n+1}(\cdot)$ with $\Psi(\cdot)$. In order to compute $v_{N-2}^{Tree}(\cdot)$ at X, and thus going on up to $t = 0$, we have to compute a suitable approximation $\tilde{v}_{N-1}(\cdot)$ of $v(t_{N-1},\cdot)$ and then evaluate it at the points of the set $\tilde{X} = \bigcup_{p=1}^P \tilde{X}^p$. Since we know $v_{N-1}^{Tree}(\cdot)$ at X, we propose to employ the GPR method to obtain the required approximation. Specifically, let $v_{N-1}^{GPR}(\cdot)$ denote the GPR prediction of $v_{N-1}^{Tree}(\cdot)$, obtained by considering the predictor set X and the response $\mathbf{y} \in \mathbb{R}^P$ given by

$$y^p = v_{N-1}^{Tree}(\mathbf{x}^p), \ p \in \{1, \ldots, P\}, \tag{4.8}$$

and by employing the Squared Exponential kernel. Then, the GPR-Tree approximation $v_{N-2}^{GPR-Tree}(\cdot)$ of the value function $v(t_{N-2},\cdot)$ at time t_{N-2} can be computed as follows:

$$v_{N-2}^{GPR-Tree}\left(\mathbf{x}^p\right) = \max\left(\Psi\left(\mathbf{x}^p\right), \frac{e^{-r\Delta t}}{2^d}\sum_{m=1}^{2^d} v_{N-1}^{GPR}\left(\tilde{\mathbf{x}}^{p,m}\right)\right),\ p \in \{1,\dots,P\}. \tag{4.9}$$

Following the same steps, the dynamic programming problem can be solved by proceeding backward. Specifically, let $n \in \{1,\dots,N-3\}$ and let $v_{n+1}^{GPR}(\cdot)$ denote the GPR prediction of $v_{n+1}^{GPR-Tree}(\cdot)$ obtained from predictor set X and the response $\mathbf{y} \in \mathbb{R}^P$ given by

$$y^p = v_{n+1}^{GPR-Tree}\left(\mathbf{x}^p\right). \tag{4.10}$$

Then, the function $v_n^{GPR-Tree}$ can be obtained as

$$v_n^{GPR-Tree}\left(\mathbf{x}^p\right) = \max\left(\Psi\left(\mathbf{x}^p\right), \frac{e^{-r\Delta t}}{2^d}\sum_{m=1}^{2^d} v_{n+1}^{GPR}\left(\tilde{\mathbf{x}}^{p,m}\right)\right). \tag{4.11}$$

Finally, once the GPR approximation $v_1^{GPR}(\cdot)$ of $v_1^{GPR-Tree}(\cdot)$ is available, the GPR-Tree estimate of the initial option price is given by

$$\max\left(\Psi\left(\mathbf{S}_0\right), \frac{e^{-r\Delta t}}{2^d}\sum_{m=1}^{2^d} v_1^{GPR}\left(\tilde{\mathbf{x}}^{1,m}\right)\right). \tag{4.12}$$

We stress out that the GPR-MC method in [17] is similar to the GPR-Tree method here proposed. The main difference consists in the use of random simulations in place of a tree step to compute the continuation value. In particular, for each time step t_n and for each point $\mathbf{x}^p$, M random simulations of $\mathbf{S}_{t_{n+1}}|\mathbf{S}_{t_n} = \mathbf{x}^p$ are generated and they are employed to estimate the continuation value by its discounted empirical mean. We stress out that, as far as the GPR-MC is considered, the number M of future values employed in the estimation of the continuation value can be chosen as a sufficiently large value, while in the case of GPR-Tree it is determined by $M = 2^d$.

5 The GPR-EI method in the multi-dimensional Black-Scholes model

The GPR-EI method differs from both the GPR-MC and GPR-Tree methods for two reasons. First of all, the predictors employed in the GPR step are related to the logarithms of the predictors used in the GPR-Tree method. Secondly, the continuation value at these points is computed through a closed formula which comes from an exact integration.

Let $X = \{\mathbf{x}^p, p = 1,\dots,P\} \subset \mathbb{R}_+^d$ be the same set as in (4.3) and define $\log\left(\mathbf{x}^p\right)$ as the vector obtained by applying the natural logarithm to all the components of $\mathbf{x}^p$, that is $\log\left(\mathbf{x}^p\right) = \left(\log\left(x_1^p\right),\dots,\log\left(x_d^p\right)\right)^\top$. Moreover, let us define the set

$$Z = \left\{\mathbf{z}^p = \left(z_1^p,\dots,z_d^p\right)^\top, p = 1,\dots,P\right\}. \tag{5.1}$$

We set $\mathbf{z}^1 = \log\left(\mathbf{S_0}\right)$ and for $p = 2,\dots,P$ and for $i = 1\dots,d$ we set

$$z_i^p = \log\left(S_0^i\right) + \sigma_i\sqrt{T}\sum_{j=1}^{d}\Sigma_{d,j}\Phi^{-1}\left(H_j^p\right) \tag{5.2}$$

or equivalently, with reference to (4.3), $\mathbf{z}^p = \log(\mathbf{x}^p) - \left(r - \frac{1}{2}\boldsymbol{\sigma}^2\right)T$ (please, observe that adding $\log(\mathbf{S}_0)$ in the z_i^p definition simply shifts the predictor set, with no effects in GPR training, but it simplifies the notation in the proofs). The elements of Z are determined in order to represent some possible values of the process $\mathbf{Z}_t = \log(\mathbf{S}_0) + \left(\sigma_1 W_t^1, \ldots, \sigma_d W_t^d\right)^\top$. This is done to profit from the Gaussian properties of the process $\mathbf{Z}_t$ and to be able to calculate the expected value by using a closed formula as described below. In this case, we do not work directly with the function v, but we rather consider the function $u : [0,T] \times Z \to \mathbb{R}$ defined as

$$u(t, \mathbf{z}) := v\left(t, \exp\left(\mathbf{z} + \left(r - \frac{1}{2}\boldsymbol{\sigma}^2\right)t\right)\right). \tag{5.3}$$

In a nutshell, the main idea is to approximate the function u at $t_N, t_{N-1}, \ldots, t_1$ by using the GPR method on the fixed grid Z and by using the Squared Exponential kernel. In accordance with equation (3.6), the GPR approach allows one to approximate the function $u(t_n, \cdot)$ at time t_n by

$$u_n^{GPR}(\mathbf{z}) = \sum_{q=1}^{P} k_{SE}(\mathbf{z}^q, \mathbf{z})\,\omega_q, \tag{5.4}$$

where $\omega_1, \ldots, \omega_P$ are appropriate real weights. The continuation value can be computed by integrating the function u^{GPR} against a d-dimensional probability density. This calculation can be done easily by means of a closed formula.

Specifically, the GPR-EI method relies on the following Proposition.

Proposition 1. *Let $u_N^{GPR-EI} = \Psi$ and consider the following sequence of functions defined on Z recursively as follows: for each $n \in \{0, \ldots, N-1\}$ we define*

$$u_n^{GPR-EI}(\mathbf{z}^p) = \max\left(\Psi\left(\exp\left(\mathbf{z}^p + \left(r - \frac{1}{2}\boldsymbol{\sigma}^2\right)t_n\right)\right), e^{-r\Delta t}\sum_{q=1}^{P}\omega_q\sigma_f^2\sigma_l^d\frac{e^{-\frac{1}{2}(\mathbf{z}^q-\mathbf{z}^p)^\top\left(\Pi+\sigma_l^2 I_d\right)^{-1}(\mathbf{z}^q-\mathbf{z}^p)}}{\sqrt{\det\left(\Pi+\sigma_l^2 I_d\right)}}\right) \tag{5.5}$$

σ_f, σ_l, and $\omega_1, \ldots, \omega_P$ are certain constants determined by the GPR approximation of the function u_{n+1}^{GPR-EI} for $k = 1, \ldots, P$, considering Z as the predictor set, and $\Pi = (\Pi_{i,j})$ is the $d \times d$ covariance matrix of the log-increments defined by $\Pi_{i,j} = \rho_{i,j}\sigma_i\sigma_j\Delta t$. Then, for all $n \in \{0, \ldots, N-1\}$, the function $u_n^{GPR-EI}(\cdot)$, approximates the function $u(t_n, \cdot)$ in Z.

The proof of Proposition 1 is reported in the Appendix A. Equation (5.5) allows one to compute the option price at time $t = 0$ by proceeding backward. In fact, the function $u(t_N, \cdot)$ is known at time $t_N = T$ throught (5.3) since the price function $v(t_N, \cdot)$ is equal to the payoff function $\Psi(\cdot)$. Moreover, if an approximation of $u(t_{n+1}, \cdot)$ is available, then one can approximate $u(t_n, \cdot)$ at Z by means of relation (5.5). Finally, the option price at time $t = 0$ is approximated by $u_0^{GPR-EI}(\mathbf{z}^1)$.

6 American options in the rough Bergomi model

The rough Bergomi model, introduced by Bayer et al. [5], shapes the underlying process S_t and its volatility V_t through the following relations:

$$dS_t = rS_t dt + \sqrt{V_t}S_t dW_t^1 \tag{6.1}$$

$$V_t = \xi_0(t)\exp\left(\eta\widetilde{W}_t^H - \frac{1}{2}\eta^2 t^{2H}\right), \tag{6.2}$$

with r the (constant) interest rate, η a positive parameter and $H \in (0,1)$ the Hurst parameter. The deterministic function $\xi_0(t)$ represents the forward variance curve and following Bayer et al. [5] we consider it as constant. The process W_t^1 is a Brownian motion, whereas $\widetilde{W}_t^H$ is a Riemann-Liouville fractional Brownian motion that can be expressed as a stochastic integral:

$$\widetilde{W}_t^H = \sqrt{2H} \int_0^t (t-s)^{H-\frac{1}{2}} \, dW_t^2, \tag{6.3}$$

with W_t^2 a Brownian motion and ρ the instantaneous correlation coefficient between W_t^1 and W_t^2.

The rough Bergomi model stood out for its ability to explain implied volatility and other phenomena related to European options. Moreover, it is particularly interesting from a computational point of view as it is a non-Markovian model and therefore it is not possible to apply standard techniques for American options.

In this framework, the price at time t of an American option having maturity T and payoff function $\Psi : \mathbb{R}_+ \to \mathbb{R}$ is then

$$v(t, \mathcal{F}_t) = \sup_{\tau \in \mathcal{T}_{t,T}} \mathbb{E}\left[e^{-r(\tau - t)} \Psi(S_\tau) | \mathcal{F}_t \right], \tag{6.4}$$

where $\mathcal{F}_t$ is the natural filtration generated by the couple $\left(W_s^1, \widetilde{W}_s^H\right)$ for $s \in [0,t]$. We point out that, as opposed to the multi-dimensional Brownian motion, in this case, the stopping time τ does not only depend from the actual values of S and V but, since these are non-Markovian processes, it depends on the whole filtration, that is from the whole observed history of the processes.

7 The GPR-Tree method in the rough Bergomi model

The GPR-Tree method can be adapted to price American options in the rough Bergomi model. Despite the dimension of the model is only two, it is a non-Markovian model which obliges one to take into account the past history when evaluating the price of an option. So, the price of an option at a certain moment depends on all the filtration at that moment. Clearly, evaluating an option by considering the whole history of the process (a continuous process) is not possible. To overcome such an issue, we simulate the process on a finite number of dates and we consider the sub-filtration induced by these observations. First of all, we consider a finite number N of time steps that determines the time increment $\Delta t = \frac{T}{N}$, and we employ the scheme presented in Bayer et. al [5] to generate a set of P simulations of the couple (S_t, V_t) at $t_n = n\,\Delta t$ for $n = 1, \ldots, N$. In particular, if we set $\Delta W_n^1 = W_{t_n}^1 - W_{t_{n-1}}^1$, then the $2N$-dimensional random vector $\mathbf{R}$, given by

$$\mathbf{R} = \left(\Delta W_1^1, \widetilde{W}_{t_1}^H, \ldots, \Delta W_N^1, \widetilde{W}_{t_N}^H \right)^\top, \tag{7.1}$$

follows a zero-mean Gaussian distribution. Moreover, using the relations stated in Appendix B, one can calculate the covariance matrix Υ of $\mathbf{R}$ and its lower triangular square root Λ by using the Cholesky factorization. The vector $\mathbf{R}$ can be simulated by computing $\Lambda \mathbf{G}$, where $\mathbf{G} = (G_1, \ldots, G_{2N})^\top$ is a vector of independent standard Gaussian random variables. Finally, a simulation for $(S_{t_n}, V_{t_n})_{n=0,\ldots,N}$ can be obtained from $\mathbf{R}$ by considering the initial values

$$\hat{S}_{t_0} = S_0, \; V_{t_0} = \xi_0, \tag{7.2}$$

and the Euler–Maruyama scheme given by

$$S_{t_{n+1}} = S_{t_n} \exp\left(\left(r - \frac{1}{2}V_{t_n}\right)\Delta t + \sqrt{V_{t_n}}\Delta W^1_{n+1}\right), \tag{7.3}$$

$$V_{t_{n+1}} = \xi_0 \exp\left(-\frac{1}{2}\eta^2 \left(t_{n+1}\right)^{2H} + \eta \widetilde{W}^H_{t_{n+1}}\right). \tag{7.4}$$

We stress out that the scheme proposed in equations (7.3) and (7.4) is exact for V and approximated for S. In order to avoid overloading the notation, we do not distinguish between the real process S at time t_n and its Euler-Marayuama approximation, also relying on the strong and weak convergence properties of such a scheme (see Pavliotis [30]).

First of all, the GPR-Tree method simulates P different samples for the vector $\mathbf{G}$, namely $\mathbf{G}^p$ for $p = 1, \ldots, P$, and it computes the corresponding paths $\left(S^p_{t_1}, V^p_{t_1}, \ldots, S^p_{t_N}, V^p_{t_N}\right)$ according to (7.2), (7.3) and (7.4). To summarize the values assumed by S and V, let us define the vector

$$\mathbf{SV}^p_{i:j} = \left(S^p_{t_i}, V^p_{t_i}, S^p_{t_{i+1}}, V^p_{t_{i+1}}, \ldots, S^p_{t_j}, V^p_{t_j}\right)^\top \tag{7.5}$$

for $i, j \in \{0, \ldots, N\}$ and $i < j$. Moreover, we also define

$$\log\left(\mathbf{SV}^p_{i:j}\right) = \left(\log\left(S^p_{t_i}\right), \log\left(V^p_{t_i}\right), \log\left(S^p_{t_{i+1}}\right), \log\left(V^p_{t_{i+1}}\right), \ldots, \log\left(S^p_{t_j}\right), \log\left(V^p_{t_j}\right)\right)^\top, \tag{7.6}$$

where log stands for the natural logarithm.

Then, the GPR-Tree method computes the option value for each of these P trajectories, proceeding backward in time and considering the past history coded into the filtration. Since we consider only a finite number of steps, we approximate the filtration $\mathcal{F}_{t_n}$ with the natural filtration $\hat{\mathcal{F}}_{t_n}$ generated by the $2n$ variables $W^1_{t_1}, \widetilde{W}^H_{t_1}, \ldots, W^1_{t_n}, \widetilde{W}^H_{t_n}$. Moreover, $\hat{\mathcal{F}}_{t_n}$ is equal to the filtration generated by $S_{t_1}, V_{t_1}, \ldots, S_{t_n}, V_{t_n}$ because there exists a deterministic bijective function that allows one to obtain $W^1_{t_1}, \widetilde{W}^H_{t_1}, \ldots, W^1_{t_n}, \widetilde{W}^H_{t_n}$ from $S_{t_1}, V_{t_1}, \ldots, S_{t_n}, V_{t_n}$ and vice versa. Therefore, when we calculate the option value conditioned by filtration $\hat{\mathcal{F}}_{t_n}$, it is enough to conditioning with respect to the knowledge of the variables $S_{t_1}, V_{t_1}, \ldots, S_{t_n}, V_{t_n}$.

The GPR-Tree method proceeds backward in time, using a tree method and the GPR to calculate the option price with respect to the initially simulated trajectories. As opposed to the multi-dimensional Black-Scholes model, here we perform more than one single tree step, so as to reduce the number of GPR regressions and thus increasing the computational efficiency. In particular, we consider $N = N^{Tree} \cdot m$ with N^{Tree} and m natural numbers that represent how many times the tree method is used and the number of time steps employed, respectively.

After simulating the P random paths $\{\mathbf{SV}^p_{1:N},\ p = 1, \ldots, P\}$, we compute the tree approximation of the option value $v\left(t_{N-m}, \mathbf{SV}^p_{1:(N-m)}\right)$ at time t_{N-m} for each path as follows:

$$v^{Tree}_{N-m}\left(\mathbf{SV}^p_{1:(N-m)}\right) = \max\left(\Psi\left(S^p_{t_{N-m}}\right), C^{Tree}_{N-m}\left(\mathbf{SV}^p_{1:(N-m)}\right)\right), \tag{7.7}$$

with C^{Tree}_{N-m} stands for the the approximation of the continuation value function at time t_{N-m} obtained by means of a tree approach, which discretizes each component of the Gaussian vector $\mathbf{G}_{[2(N-m)+1]:2N}$ that generates the process. As opposed to the multi-dimensional Black-Scholes model, the approximation of the independent Gaussian components of $\mathbf{G}$ through the equiprobable couple $\{-1, +1\}$ is not suitable since the convergence to the right price is too slow. So, we propose to use the same discrete approximation employed by Alfonsi in [2], which is stated in the following Lemma.

Lemma 2. *The discrete variable A defined by* $\mathbb{P}\left(A=\sqrt{3+\sqrt{6}}\right)=\mathbb{P}\left(A=-\sqrt{3+\sqrt{6}}\right)=\frac{\sqrt{6}-2}{4\sqrt{6}}$ *and* $\mathbb{P}\left(A=\sqrt{3-\sqrt{6}}\right)=\mathbb{P}\left(A=-\sqrt{3-\sqrt{6}}\right)=\frac{1}{2}-\frac{\sqrt{6}-2}{4\sqrt{6}}$ *fits the first seven moments of a standard Gaussian random variable.*

So, for each path p, we consider a quadrinomial tree with m time steps, and we use it to compute the continuation value. In particular, we consider the discrete time process $\left(\hat{S}_k^p,\hat{V}_k^p\right)_{k\in\{N-m,\dots,N\}}$ defined through

$$\hat{S}_{N-m}^p=S_{t_{N-m}}^p,\hat{V}_{N-m}^p=V_{t_{N-m}}^p \tag{7.8}$$

$$\hat{S}_{k+1}^p=\hat{S}_k^p\exp\left(\left(r-\frac{1}{2}\hat{V}_k^p\right)\Delta t+\sqrt{\hat{V}_k^p}\Lambda_{2k+1}\hat{\mathbf{G}}^p\right), \tag{7.9}$$

$$\hat{V}_{k+1}^p=\xi_0\exp\left(-\frac{1}{2}\eta^2\left(t_{k+1}\right)^{2H}+\eta\Lambda_{2k+2}\hat{\mathbf{G}}^p\right), \tag{7.10}$$

where Λ_{2k+1} is the $2k+1$-th rows of the matrix Λ and Λ_{2k+2} the $2k+2$-th row. Moreover, $\hat{G}_j^p=G_j^p$ for $j=1,\dots,2\left(N-m\right)$ and the other components, that is $\hat{G}_j^p$ for $j=2\left(N-m\right)+1,\dots,2N$, are sampled by using the random variable A of Lemma 2.

An option value is assigned to each node of the tree: at maturity, that is for $k=N$, it is equal to the payoff $\Psi\left(\hat{S}_N^p\right)$, and for $k=N-m,\dots,N-1$ it can be obtained as the maximum between the exercise value and the discounted mean value at the future nodes, weighted according to the transition probabilities determined by the probability distribution of A.

This approach allows us to compute the function $v_{N-m}^{GPR-Tree}\left(\mathbf{SV}_{1:(N-m)}^p\right)$ for $p=1,\dots,P$. We point out that, since the quadrinomial tree is not recombinant, the number of nodes grows exponentially with the number of time steps m. Therefore, m must be small. A similar problem arises with the tree approach proposed by Horvat et al. [22]. In order to overcome such an issue, we apply the GPR method to approximate the function $u_{N-m}^{GPR-Tree}\left(\log\left(\mathbf{SV}_{1:(N-m)}^p\right)\right)=v_{N-m}^{GPR-Tree}\left(\mathbf{SV}_{1:(N-m)}^p\right)$. Specifically, we consider a natural number J and we train the GPR method considering the predictor set given by

$$X=\left\{\mathbf{x}^p=\log\left(\mathbf{SV}_{\max\{1,N-m-J\}:N-m}^p\right),p=1,\dots,P\right\}\subset\mathbb{R}^{d_{N-m}} \tag{7.11}$$

where d_{N-m} is the number of predictors, that is the dimension of the regression, and the response $\mathbf{y}\in\mathbb{R}^P$ given by

$$y^p=v_{N-m}^{Tree}\left(\mathbf{SV}_{1:(N-m)}^p\right). \tag{7.12}$$

In particular, the employed predictors for such a regression, that is the components of $\mathbf{x}^p$, include the logarithms of the actual values $\left(\log\left(S_{t_{N-m}}^p\right),\log\left(V_{t_{N-m}}^p\right)\right)$ and at most J observed past values. Specifically, the number of past values included in the predictors at time t_{N-m}, that is the elements of X in (7.11), is equal to $2\left[\left(N-m\right)-\max\left(1,N-m-J\right)\right]$: the factor 2 is there because we are working with the couple (S,V) and the max operator is there because we do not consider as predictors the values $\left(\log\left(S_{t_k}^p\right),\log\left(V_{t_k}^p\right)\right)$ for $k<1$. Thus, the total number of predictors at time t_{N-m} is given by

$$d_{N-m}=2+2\left[\left(N-m\right)-\max\left(1,N-m-J\right)\right]=2\min\left\{N-m,J+1\right\}. \tag{7.13}$$

As opposed to the multi-dimensional Black-Scholes model, in the rough Bergomi case the use of the Squared Exponential kernel is not suitable because it is a isotropic kernel and the predictors employed have different nature (prices and volatilities at different times) and thus changes in each predictor impact

differently on the price. So, we employ the ARD Squared Exponential kernel that has separate length scale for each predictor.

We term u^{GPR}_{N-m} the function obtained by the aforementioned regression, which depends on $\log\left(\mathbf{SV}^p_{\max\{1,N-m-J\}:N-m}\right)$. We stress out that if we consider $J = N - m - 1$ (or greater), then the function u^{GPR}_{N-m} would consider all the observed values of S and V as predictors. Anyway, numerical tests show that it is enough to consider smaller values of J, which reduces the dimension d_{N-m} of the regression and thus improves the numerical efficiency. A similar approach is taken by Bayer et al. [6].

Once we have obtained u^{GPR}_{N-m}, we can approximate the option value $v\left(t_{N-m}, \mathbf{SV}^p_{1:(N-2m)}\right)$ at time t_{N-m} by means of the tree approach again. The only difference in this case is that the value attributed to the terminal nodes is not determined by the payoff function, but through the function u^{GPR}_{N-m}. We term $v^{GPR-Tree}_{N-m}$ the function obtained after this backward tree step. If we train the GPR method considering the predictor set given by

$$Z = \left\{\mathbf{z}^p = \log\left(\mathbf{SV}^p_{\max\{1,N-2m-J\}:N-2m}\right), p = 1, \dots, P\right\} \subset \mathbb{R}^{d_{N-2m}} \tag{7.14}$$

and the response $\mathbf{y} \in \mathbb{R}^P$ given by

$$y^p = v^{Tree}_{N-m}\left(\mathbf{SV}^p_{1:(N-2m)}\right), \tag{7.15}$$

then we obtain the function u^{GPR}_{N-m}, which can be employed to repeat the tree step and the GPR step, proceeding backward up to obtaining the initial option price by backward induction. Finally, we observe that, the number of predictors employed at time t_n for the GPR regression is equal to $d_n = 2\min(n, J+1)$.

8 The GPR-EI method in the rough Bergomi model

The GPR-EI method can be adapted to price American options in the rough Bergomi model. Just like the GPR-Tree approach, the GPR-EI method starts by simulating P different paths $\left(S^p_{t_1}, V^p_{t_1}, \dots, S^p_{t_N}, V^p_{t_N}\right)$ for the processes S and V, and it goes on by solving a backward induction problem, through the use of the GPR method and a closed formula for integration.

Just like the GPR-Tree method in the rough Bergomi model, the GPR-EI method for the rough Bergomi model employs the ARD Squared Exponential kernel defined in (3.3). However, despite the option value at maturity is known (it is equal to the payoff function), in order to compute the continuation value at time t_{N-1}, we have to approximate the payoff function by means of ARD Squared Exponential kernel. Such a function depends on the stock price S_T only and therefore the number of predictors d_N employed for such a task is equal to 1. So, in this case, the ARD Squared Exponential kernel is simply the Squared Exponential kernel. Conversely, at any time-step $t_n < T$ we use the same predictors $d_n = 2\min(n, J+1)$ as in the GPR-Tree method (see Section 7).

The GPR-EI method relies on the following Propositions.

Proposition 3. *The GPR-EI approximation of the option value at time t_{N-1} at $\mathbf{SV}^p_{\max\{1,N-1-J\}:(N-1)}$ is given by:*

$$v^{GPR-EI}_{N-1}\left(\mathbf{SV}^p_{\max\{1,N-1-J\}:(N-1)}\right) = \max\left(\Psi\left(S^p_{t_{N-1}}\right), \sum_{q=1}^{P} \frac{\omega_q e^{-r\Delta t}\sigma_f^2\sigma_l}{\sqrt{\sigma^2_{N,p} + \sigma^2_l}} \exp\left(-\frac{\left(\log\left(S^q_{t_N}\right) - \mu_{N,p}\right)^2}{2\sigma^2_{N,p} + 2\sigma^2_l}\right)\right) \tag{8.1}$$

where σ_f, σ_l, *and* $\omega_1, \ldots, \omega_P$ *are certain constants determined by the GPR approximation of the function* $\log(S_T) \mapsto \Psi(S_T)$. *Moreover,*

$$\mu_{N,p} = \log\left(S^p_{t_{N-1}}\right) + \left(r - \frac{1}{2}\sqrt{V^p_{t_{N-1}}}\right)\Delta t \tag{8.2}$$

and

$$\sigma^2_{N,p} = V^p_{t_{N-1}}\Delta t. \tag{8.3}$$

The proof of Proposition 3 is reported in the Appendix C. Therefore, we can compute the value of the option at time t_{N-1} for each simulated path by using (8.1).

Proposition 4. *Let* $n \in \{0, \ldots, N-2\}$ *and define*

$$\boldsymbol{\mu}_{n+1,p} = \left(\log\left(S^p_{t_n}\right) + \left(r - \frac{1}{2}V^p_{t_n}\right)\Delta t, \log(\xi_0) + \eta\Lambda_{2n+2}\underline{\mathbf{G}}^p - \frac{1}{2}\eta^2 t^{2H}_{n+1}\right)^\top, \tag{8.4}$$

where Λ_{2n+2} *is the* $2n+2$*-th row of the matrix* Λ *and* $\underline{\mathbf{G}}^p = (G^p_1, \ldots, G^p_{2n}, 0 \ldots, 0)^\top$, *and*

$$\Sigma_{n+1,p} = \begin{pmatrix} \Delta t V^p_{t_n} & \eta\sqrt{\Delta t V^p_{t_n}}\Lambda_{2n+2,2n+1} \\ \eta\sqrt{\Delta t V^p_{t_n}}\Lambda_{2n,2n+1} & \eta^2\left(\Lambda^2_{2n+2,2n+2} + \Lambda^2_{2n+2,2n+1}\right) \end{pmatrix}, \tag{8.5}$$

where $\Lambda_{i,j}$ *stands for the element of* Λ *in position* i, j. *Moreover, consider a natural number* $J \in \mathbb{N}$ *and set* $d_{n+1} = 2\min\{n+1, J+1\}$. *Moreover, consider the sequence of functions defined recursively as follows:* v^{GPR-EI}_{N-1} *is the same function as in Proposition (3) and, for* $n \in \{0, \ldots, N-2\}$, v^{GPR-EI}_n *is defined as follows:*

$$v^{GPR-EI}_n\left(\mathbf{SV}^p_{\max\{1,n-J\}:n}\right) = \max\left(\Psi\left(S^p_{t_n}\right), e^{-r\Delta t}\sigma^2_f\sigma_{d_{n+1}-1}\sigma_{d_{n+1}}\sum_{q=1}^{P}\omega_q h^p_q f^p_q\right), \tag{8.6}$$

where $\sigma_{d_{n+1}-1}$, $\sigma_{d_{n+1}}$, σ_f *and* $\omega_1, \ldots, \omega_P$ *are certain constants determined by the GPR approximation of the function* $\log(\mathbf{SV}_{1:n+1}) \mapsto v^{GPR-EI}_{n+1}(\mathbf{SV}_{1:n+1})$ *considering* $\left\{\mathbf{SV}^p_{\max\{1,n+1-J\}:n+1}, p = 1, \ldots, P\right\}$ *as the predictor set. Moreover,* h^p_q *and* f^p_q *are two factors given by*

$$h^p_q = \begin{cases} \exp\left(-\sum_{i=1}^{d_{n+1}-2}\frac{\left(z^p_i - z^q_i\right)^2}{2\sigma^2_i}\right) & \text{if } n > 0 \\ 1 & \text{if } n = 0 \end{cases} \tag{8.7}$$

and

$$f^p_q = \frac{\exp\left(-\frac{1}{2}\left(\begin{pmatrix} z^q_{d_{n+1}-1} \\ z^q_{d_{n+1}} \end{pmatrix} - \boldsymbol{\mu}_{n+1,p}\right)^\top\left(\Sigma_{n+1,p} + \begin{pmatrix} \sigma^2_{d_{n+1}-1} & 0 \\ 0 & \sigma^2_{d_{n+1}} \end{pmatrix}\right)^{-1}\left(\begin{pmatrix} z^q_{d_{n+1}-1} \\ z^q_{d_{n+1}} \end{pmatrix} - \boldsymbol{\mu}_{n+1,p}\right)\right)}{\sqrt{\det\left(\Sigma_{n+1,p} + \begin{pmatrix} \sigma^2_{d_{n+1}-1} & 0 \\ 0 & \sigma^2_{d_{n+1}} \end{pmatrix}\right)}}, \tag{8.8}$$

where $z^p_i = \log\left(S^p_{n+1-(i-1)/2}\right)$ *if* i *is even and* $z^p_i = \log\left(V^p_{n+1-i/2}\right)$ *if* i *is odd, for* $i = 1, \ldots, d_{n+1}$. *Then, for all* $n \in \{0, \ldots, N-2\}$, *the function* $v^{GPR-EI}_n(\cdot)$ *approximates the function* $v(t_n, \cdot)$ *in the set* $\left\{\mathbf{SV}^p_{\max\{1,n-J\}:n}, p = 1, \ldots, P\right\}$.

The proof of Proposition 4 is reported in the Appendix D. Relations (8.1) and (8.6) can be used to compute the option price at time $t = 0$ by backward induction.

9 Numerical results

In this Section we present some numerical results about the effectiveness of the proposed algorithms. The first section is devoted to the numerical tests about the multi-dimensional Black-Scholes model, while the second is devoted to the rough Bergomi model. The algorithms have been implemented in MATLAB and computations have been preformed on a server which employs a 2.40 GHz Intel® Xenon® processor (Gold 6148, Skylake) and 20 GB of RAM.

9.1 Multi-dimensional Black-Scholes model

Following Goudenège et al. [17], we consider an Arithmetic basket Put, a Geometric basket Put and a Call on the Maximum of d-assets.

In particular, we use the following parameters $T = 1$, $S_0^i = 100$, $K = 100$, $r = 0.05$, constant volatilities $\sigma_i = 0.2$, constant correlations $\rho_{ij} = 0.2$ and $N = 10$ exercise dates. Moreover, we consider $P = 250,\ 500$ or 1000 points. As opposed to the other input parameters, we vary the dimension d, considering $d =$ 2, 5, 10, 20, 40 and 100.

We present now the numerical results obtained with the GPR-Tree and the GPR-EI methods for the three payoff examples.

9.1.1 Geometric basket Put option

Geometric basket Put is a particularly interesting option since it is possible to reduce the problem of pricing it in the d-dimensional model to a one dimensional American Put option in the Black-Scholes model which can be priced straightforwardly, for example using the CRR algorithm with 1000 steps (see Cox et al. [11]). Therefore, in this case, we have a reliable benchmark to test the proposed methods. Moreover, when d is smaller than 10 we can also compute the price by means of a multi-dimensional binomial tree (see Ekvall [14]). In particular, the number of steps employed for the multi-dimensional 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. Furthermore, we also report the prices obtained with the GPR-MC method, employing $P = 1000$ points and $M = 10^5$ Monte Carlo simulations, for comparison purposes. As far as the GPR-Tree is concerned, we compute the prices only for the values of d smaller than 40 since for higher values of d the tree step becomes over time demanding. In fact, the computation of the continuation value with the tree step grows exponentially with the dimension d and for $d = 40$ it requires the evaluation of the GPR approximation at $2^{40} \approx 10^{12}$ points for every times step and for every point of X.

Results are reported in Table 1. We observe that the two proposed methods provide accurate and stable results and the computational time is generally very small, except for the GPR-Tree method at $d = 20$. Moreover, the computer processing time of the GRP-EI method increases little with the size of the problem and this makes the method particularly effective when the dimension of the problem is high. This is because the computation of the expected value and the training of the GPR model are minimally affected by the dimension of the problem.

The GPR-Tree method seems to be the more accurate for $d \leq 10$. When $d = 20$, GPR-Tree is still the most accurate but the computational time required is extremely greater than that required by GPR-EI with the same number P of points considered. Furthermore, when $d > 20$, GPR-Tree still works, but the computational time becomes excessive.

		GPR-Tree			GPR-EI			GPR-MC	Ekvall	Benchmark
d	P	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)	4.57	4.62	4.62
5		3.44 (9)	3.43 (15)	3.44 (23)	3.40 (4)	3.43 (14)	3.41 (27)	3.41	3.44	3.45
10		3.00 (10)	2.96 (33)	2.93 (60)	2.85 (4)	2.88 (9)	2.93 (30)	2.90		2.97
20		2.80 (4220)	2.72 (14304)	2.72 (49609)	2.63 (4)	2.73 (9)	2.63 (29)	2.70		2.70
40					2.45 (4)	2.52 (10)	2.53 (38)	2.57		2.56
100					2.27 (5)	2.32 (15)	2.39 (45)	2.40		2.47

Table 1: Results for a Geometric basket Put option using the GPR-Tree method and the GPR-EI method. In the last columns, the prices obtained by using the GPR-MC method, the Ekvall multi-dimensional tree and the exact benchmark (d is the dimension and P is the number of points). Values in brackets are the computational times (in seconds).

Figure 9.1 summarizes the behavior of the GPR methods while changing the dimension d. As we can see, all the considered methods produce relatively accurate estimates that converge at the exact price (within the limits of the approximation of an American option with a Bermudan). Furthermore, the relative error is small with all the considered methods, but the computational time required by the GPR-Tree method and the GPR-EI method is generally smaller with respect to the GPR-MC method.

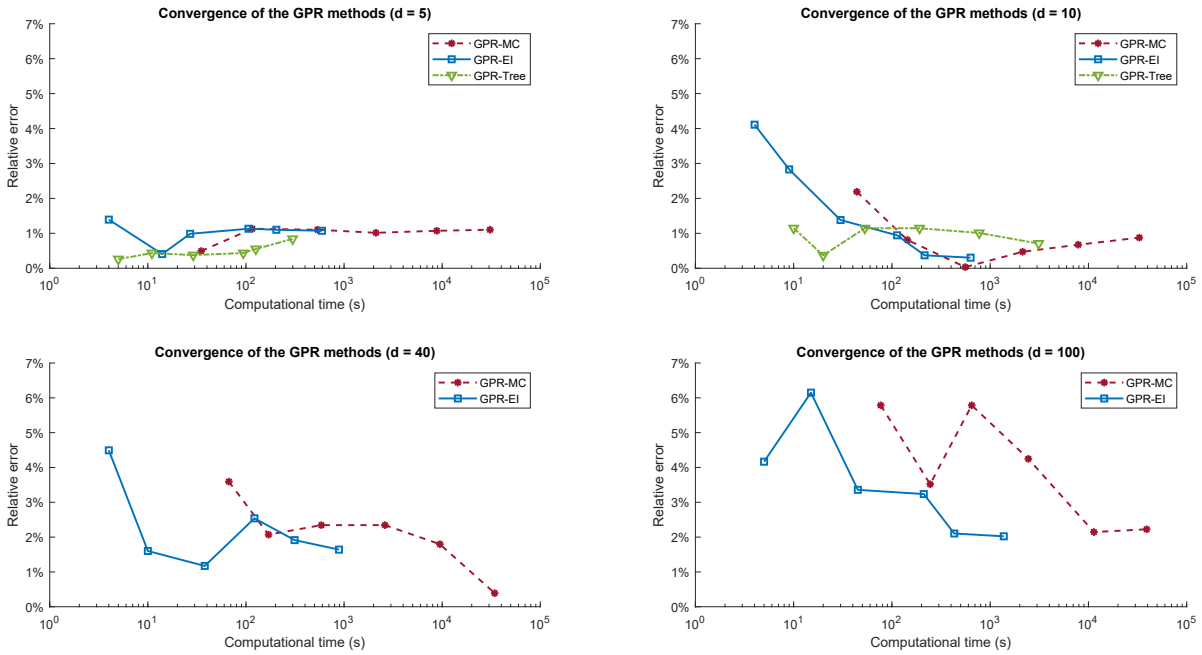

Figure 9.1: Comparison among the GPR methods changing the dimension d and doubling the number of points from $P = 250$ to $P = 8000$. As far as the GPR-MC method is concerned $M = 10^4$ Monte Carlo simulations are employed.

9.1.2 Arithmetic basket Put option

As opposed to the Geometric basket Put option, in this case we have no method to obtain a fully reliable benchmark. Therefore we only consider the prices obtained by means of the GPR-MC method, employed with $P = 1000$ points and $M = 10^5$ Monte Carlo simulations. Moreover, for small values of d, a benchmark can be obtained by means of a multi-dimensional tree method (see Ekvall [14]), just as shown for the Geometric case. Results are reported in Table 2. Similarly to the Geometric basket Put, the prices obtained are reliable and they do not change much with respect to the number P of points. As opposed to the GPR-Tree method, which can not be applied for high values of d, the GPR-EI method requires a small computational time for all the values concerned of d.

9.1.3 Call on the Maximum

As for the Arithmetic basket Put, in this case we have no numerical methods to obtain a fully reliable benchmark. However, for small values of d, we can approximate the price obtained by means of a multi-dimensional tree method. Moreover, we also consider the price obtained with the GPR-MC method. Results, which are shown in Table 3, have an accuracy comparable to the one obtained for the Arithmetic basket Put option.

		GPR-Tree			GPR-EI			GPR-MC	Ekvall
d	P	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)	4.37	4.42
5		3.15 (5)	3.12 (9)	3.13 (24)	3.09 (6)	3.12 (9)	3.10 (44)	3.09	3.15
10		2.71 (10)	2.64 (21)	2.62 (70)	2.49 (5)	2.56 (9)	2.60 (38)	2.58	
20		2.37 (4259)	2.35 (16343)	2.40 (57399)	2.26 (6)	2.31 (14)	2.28 (42)	2.38	
40					2.18 (4)	2.18 (10)	2.16 (31)	2.17	
100					2.35 (7)	2.01 (13)	2.06 (42)	1.92	

Table 2: Results for an Arithmetic basket Put option using the GPR-Tree method and the GPR-EI method. In the last columns, the prices obtained by using the GPR-MC method and the Ekvall multi-dimensional tree (d is the dimension and P is the number of points). Values in brackets are the computational times (in seconds).

		GPR-Tree			GPR-EI			GPR-MC	Ekvall
d	P	250	500	1000	250	500	1000		
2		16.94 (5)	16.94 (8)	16.93 (20)	16.75 (4)	16.81 (10)	16.82 (28)	16.86	16.86
5		27.14 (5)	27.17 (10)	27.19 (26)	26.92 (4)	27.15 (9)	26.95 (27)	27.20	27.20
10		35.27 (11)	34.97 (21)	35.08 (106)	35.66 (4)	34.98 (10)	34.84 (29)	35.17	
20		43.26 (4126)	43.21 (15025)	43.00 (51090)	45.05 (4)	42.74 (11)	42.62 (35)	42.76	
40					51.79 (5)	50.36 (10)	49.53 (41)	50.70	
100					59.03 (5)	60.72 (13)	60.96 (42)	59.69	

Table 3: Results for a Call on Maximum Put option using the GPR-Tree method and the GPR-EI method. In the last columns, the prices obtained by using the GPR-MC method and the Ekvall multi-dimensional tree (d is the dimension and P is the number of points). Values in brackets are the computational times (in seconds).

9.2 Rough Bergomi model

Following Bayer et al. [6], we consider an American Put option and we use the same parameters: $T = 1$, $H = 0.07$, $\rho = -0.90$, $\xi_0 = 0.09$, $\eta = 1.9$, $S_0 = 100$, $r = 0.05$ and strike $K = 70, 80, \ldots, 120, 130$ or 140. As far as the GPR-Tree is concerned, we employ $N = 50$ or $N = 100$ time steps with $m = 1$ or $m = 2$, $P = 500, 1000, 2000$ or 4000 random paths, and $J = 0, 1, 3, 7$ or 15 past values. As far as the GPR-EI is concerned, we employ $N = 50$ or $N = 100$ time steps, $P = 1000, 2000, 4000$ or 8000 random paths, and $J = 0, 1, 3, 7$ or 15 past values. Similar to what observed by Bayer et al. [6], the difference changing the value of J does not impact significantly on the price, which indicates that considering the non-Markovian nature of the processes in the formulation of the exercise strategies is not particularly relevant. Conversely, using a large number of predictors significantly increases computational time. Numerical results are reported in Tables 4, 5 and 6, together with the results reported by Bayer et al. in [6]. Prices are very close to the benchmark, except for the case $K = 120$: in this case with both the two GPR methods we obtain a price which is close to 20.20 while Bayer et al. obtain 20.00. Anyway, it is worth noticing that the relative gap

between these two results is less than 1% .

The values of the H parameter usually estimated from market data are close to 0. Anyway, we also investigate how our methods behave when the value of the Hurst exponent H is above $\frac{1}{2}$. In particular, we consider $H = 0.80$. Moreover, we also consider $P = 8000$ and $P = 16000$ points for the GPR-Tree and GPR-EI methods respectively. Results for American options are reported in Table 7 and 8. Also in this case, adding only a few J previous values is enough, in particular when the strike K is low.

			GPR-Tree								Bayer et al.
		N	50				100				
K	J	P	500	1000	2000	4000	500	1000	2000	4000	
70	0		1.85 (40)	1.88 (137)	1.88 (380)	1.87 (574)	1.84 (89)	1.84 (308)	1.86 (714)	1.86 (1201)	
	1		1.85 (52)	1.88 (153)	1.88 (554)	1.87 (947)	1.84 (119)	1.86 (354)	1.86 (1611)	1.86 (3396)	1.88
	3		1.85 (79)	1.87 (266)	1.88 (1501)	1.87 (1581)	1.84 (181)	1.85 (688)	1.86 (1734)	1.86 (4742)	
	7		1.88 (127)	1.89 (736)	1.88 (3248)	1.87 (3630)	1.83 (748)	1.84 (1350)	1.86 (4660)	1.87 (6837)	
	15		1.88 (222)	1.94 (1167)	1.89 (9302)	1.87 (5783)	1.81 (703)	1.84 (2249)	1.85 (12823)	1.81 (11248)	
80	0		3.16 (44)	3.19 (112)	3.20 (349)	3.20 (547)	3.14 (82)	3.16 (266)	3.18 (1112)	3.19 (1037)	
	1		3.17 (59)	3.19 (162)	3.20 (644)	3.20 (880)	3.15 (97)	3.18 (551)	3.18 (1076)	3.20 (1955)	3.22
	3		3.16 (68)	3.19 (326)	3.20 (1154)	3.20 (1537)	3.15 (165)	3.16 (613)	3.19 (2111)	3.22 (3603)	
	7		3.18 (169)	3.18 (600)	3.21 (2812)	3.21 (4339)	3.14 (228)	3.18 (1636)	3.19 (4792)	3.23 (9257)	
	15		3.18 (229)	3.19 (975)	3.17 (7951)	3.20 (7140)	3.15 (640)	3.17 (4083)	3.14 (11502)	3.19 (11487)	
90	0		5.23 (43)	5.24 (121)	5.25 (350)	5.26 (563)	5.21 (103)	5.25 (254)	5.25 (756)	5.28 (976)	
	1		5.23 (56)	5.24 (154)	5.27 (667)	5.26 (971)	5.21 (93)	5.27 (355)	5.28 (1349)	5.30 (2360)	5.31
	3		5.26 (97)	5.28 (353)	5.27 (1499)	5.29 (3597)	5.23 (153)	5.31 (970)	5.30 (2691)	5.33 (3751)	
	7		5.25 (194)	5.24 (617)	5.25 (1991)	5.30 (4015)	5.25 (442)	5.26 (1142)	5.32 (10235)	5.33 (7039)	
	15		5.23 (238)	5.25 (1231)	5.26 (7572)	5.28 (6781)	5.25 (696)	5.28 (3180)	5.34 (15784)	5.31 (13639)	
100	0		8.35 (43)	8.36 (108)	8.38 (338)	8.39 (513)	8.36 (95)	8.42 (270)	8.40 (686)	8.45 (972)	
	1		8.34 (48)	8.36 (147)	8.38 (520)	8.39 (838)	8.38 (95)	8.44 (536)	8.43 (1219)	8.45 (2123)	8.50
	3		8.35 (84)	8.40 (304)	8.43 (1409)	8.39 (1277)	8.40 (164)	8.46 (697)	8.40 (2043)	8.44 (3227)	
	7		8.36 (165)	8.37 (582)	8.42 (2741)	8.47 (4000)	8.37 (275)	8.42 (1070)	8.47 (5772)	8.52 (7643)	
	15		8.39 (290)	8.39 (1127)	8.42 (5372)	8.41 (7860)	8.36 (540)	8.42 (5360)	8.50 (13825)	8.52 (15144)	
110	0		13.02 (45)	13.06 (124)	13.07 (390)	13.11 (483)	13.10 (96)	13.16 (317)	13.15 (679)	13.19 (1067)	
	1		13.02 (48)	13.06 (154)	13.09 (555)	13.13 (819)	13.10 (95)	13.15 (317)	13.15 (1194)	13.19 (1686)	13.23
	3		13.09 (102)	13.06 (274)	13.09 (1081)	13.11 (1389)	13.10 (166)	13.19 (657)	13.18 (2777)	13.22 (3101)	
	7		13.16 (193)	13.11 (629)	13.12 (2848)	13.13 (3556)	13.10 (241)	13.15 (1068)	13.20 (5674)	13.27 (11147)	
	15		13.04 (305)	13.06 (1004)	13.12 (5481)	13.19 (7704)	13.11 (562)	13.22 (3291)	13.25 (13552)	13.22 (12832)	
120	0		20.18 (45)	20.18 (124)	20.19 (348)	20.21 (545)	20.19 (90)	20.23 (285)	20.21 (771)	20.22 (1151)	
	1		20.18 (44)	20.18 (154)	20.19 (502)	20.21 (803)	20.21 (99)	20.24 (373)	20.21 (1094)	20.22 (1764)	20.00
	3		20.19 (93)	20.18 (268)	20.19 (972)	20.22 (1478)	20.20 (162)	20.23 (886)	20.21 (2444)	20.22 (3045)	
	7		20.18 (169)	20.18 (495)	20.18 (2521)	20.24 (3756)	20.20 (395)	20.23 (1048)	20.19 (4764)	20.22 (5881)	
	15		20.18 (250)	20.18 (1082)	20.16 (13516)	20.15 (7354)	20.19 (658)	20.23 (2569)	20.17 (14221)	20.25 (17789)	
130			Always 30.00				Always 30.00				30.00
140			Always 40.00				Always 40.00				40.00

Table 4: Results for an American Put option in the rough Bergomi model using the GPR-Tree method with a single tree step ($m = 1$). N represents the number of time steps, P the number of the simulated paths and J the number of past values employed in the regression. Values in brackets are the computational times (in seconds).

			GPR-Tree								Bayer et al.
		N	50				100				
K	J	P	500	1000	2000	4000	500	1000	2000	4000	
70	0		1.87 (28)	1.88 (97)	1.88 (391)	1.86 (876)	1.87 (71)	1.86 (236)	1.86 (646)	1.86 (1337)	
	1		1.86 (44)	1.87 (183)	1.88 (607)	1.87 (1672)	1.86 (95)	1.87 (310)	1.87 (1222)	1.87 (2265)	1.88
	3		1.87 (71)	1.86 (296)	1.87 (1084)	1.87 (3742)	1.86 (163)	1.87 (594)	1.88 (1962)	1.88 (4222)	
	7		1.91 (168)	1.87 (563)	1.86 (1930)	1.87 (3501)	1.85 (275)	1.87 (1141)	1.88 (4579)	1.88 (7997)	
	15		1.94 (248)	1.88 (986)	1.87 (4841)	1.87 (7806)	1.87 (541)	1.88 (2171)	1.87 (10169)	1.86 (16466)	
80	0		3.18 (31)	3.19 (117)	3.20 (376)	3.20 (823)	3.17 (85)	3.18 (216)	3.19 (603)	3.19 (1368)	
	1		3.19 (47)	3.19 (152)	3.20 (569)	3.20 (1166)	3.18 (119)	3.20 (322)	3.20 (1107)	3.20 (2396)	3.22
	3		3.19 (93)	3.19 (287)	3.20 (1070)	3.20 (2095)	3.17 (167)	3.21 (617)	3.21 (1966)	3.22 (4043)	
	7		3.21 (136)	3.20 (624)	3.20 (2653)	3.21 (3721)	3.19 (301)	3.21 (1134)	3.21 (4179)	3.23 (8031)	
	15		3.24 (322)	3.20 (1186)	3.20 (7011)	3.21 (7392)	3.18 (633)	3.20 (1940)	3.20 (10317)	3.23 (20584)	
90	0		5.24 (28)	5.24 (102)	5.25 (359)	5.26 (702)	5.25 (82)	5.28 (223)	5.26 (707)	5.28 (1504)	
	1		5.25 (44)	5.25 (163)	5.26 (512)	5.27 (1185)	5.27 (109)	5.29 (283)	5.28 (1144)	5.30 (3954)	5.31
	3		5.27 (94)	5.26 (330)	5.28 (1058)	5.28 (1756)	5.30 (177)	5.31 (555)	5.31 (1833)	5.32 (4226)	
	7		5.28 (150)	5.30 (561)	5.29 (2253)	5.29 (3595)	5.30 (315)	5.33 (1089)	5.33 (4319)	5.33 (7073)	
	15		5.28 (269)	5.27 (1000)	5.28 (4348)	5.29 (7411)	5.28 (533)	5.33 (2127)	5.34 (17098)	5.34 (16804)	
100	0		8.36 (29)	8.37 (103)	8.37 (329)	8.39 (748)	8.42 (70)	8.45 (190)	8.42 (584)	8.46 (1313)	
	1		8.39 (47)	8.40 (177)	8.39 (510)	8.42 (1145)	8.43 (89)	8.46 (325)	8.44 (969)	8.48 (2058)	8.50
	3		8.42 (89)	8.42 (302)	8.43 (986)	8.45 (1844)	8.47 (167)	8.50 (551)	8.49 (2322)	8.51 (4439)	
	7		8.43 (173)	8.43 (552)	8.44 (2083)	8.45 (3926)	8.47 (322)	8.51 (1117)	8.49 (4120)	8.53 (8324)	
	15		8.44 (340)	8.44 (1134)	8.45 (4637)	8.46 (7013)	8.51 (684)	8.53 (2229)	8.48 (8403)	8.53 (14183)	
110	0		13.04 (32)	13.06 (90)	13.08 (334)	13.12 (695)	13.15 (77)	13.18 (237)	13.16 (572)	13.20 (1364)	
	1		13.09 (67)	13.09 (180)	13.12 (544)	13.15 (1135)	13.17 (95)	13.20 (296)	13.19 (1192)	13.22 (2207)	13.23
	3		13.11 (78)	13.14 (282)	13.17 (1119)	13.18 (1896)	13.18 (158)	13.23 (575)	13.23 (1917)	13.26 (4028)	
	7		13.11 (157)	13.14 (520)	13.19 (2083)	13.19 (3659)	13.20 (318)	13.22 (1058)	13.24 (4508)	13.29 (7440)	
	15		13.09 (254)	13.15 (1007)	13.17 (4449)	13.20 (7668)	13.22 (625)	13.26 (2582)	13.27 (9055)	13.24 (13191)	
120	0		20.19 (37)	20.19 (121)	20.20 (304)	20.22 (692)	20.21 (79)	20.24 (206)	20.22 (662)	20.23 (1484)	
	1		20.20 (49)	20.21 (180)	20.21 (494)	20.25 (1047)	20.21 (95)	20.24 (283)	20.24 (959)	20.26 (2029)	20.00
	3		20.19 (98)	20.19 (268)	20.25 (1120)	20.26 (2077)	20.23 (156)	20.26 (588)	20.26 (2075)	20.24 (3705)	
	7		20.20 (152)	20.18 (511)	20.17 (1935)	20.26 (3592)	20.25 (363)	20.25 (1139)	20.23 (4395)	20.28 (6293)	
	15		20.19 (278)	20.17 (1036)	20.22 (4161)	20.24 (7844)	20.18 (624)	20.22 (1951)	20.19 (8057)	20.28 (15643)	
130			Always 30.00				Always 30.00				30.00
140			Always 40.00				Always 40.00				40.00

Table 5: Results for an American Put option in the rough Bergomi model using the GPR-Tree method with a double tree step ($m = 2$). N represents the number of time steps, P the number of the simulated paths and J the number of past values employed in the regression. Values in brackets are the computational times (in seconds).

			GPR-EI								Bayer et al.
		N	50				100				
K	J	P	1000	2000	4000	8000	1000	2000	4000	8000	
70	0		1.82 (101)	1.84 (253)	1.85 (351)	1.85 (533)	1.86 (162)	1.88 (579)	1.87 (689)	1.88 (1011)	
	1		1.82 (96)	1.85 (525)	1.85 (636)	1.85 (884)	1.86 (184)	1.88 (816)	1.87 (913)	1.88 (1551)	1.88
	3		1.83 (263)	1.85 (1305)	1.83 (1118)	1.84 (1630)	1.86 (369)	1.88 (2389)	1.88 (2831)	1.89 (2994)	
	7		1.81 (497)	1.85 (2706)	1.85 (3014)	1.85 (3447)	1.80 (657)	1.87 (4848)	1.88 (5576)	1.86 (4132)	
	15		1.78 (820)	1.84 (4939)	1.83 (5802)	1.83 (6006)	1.79 (1932)	1.83 (11876)	1.85 (14703)	1.88 (5870)	
80	0		3.14 (86)	3.16 (271)	3.18 (348)	3.17 (558)	3.22 (162)	3.24 (549)	3.21 (602)	3.22 (1065)	
	1		3.14 (127)	3.16 (409)	3.19 (601)	3.18 (865)	3.23 (212)	3.24 (984)	3.21 (847)	3.22 (1285)	3.22
	3		3.14 (160)	3.18 (1334)	3.19 (1190)	3.19 (1476)	3.22 (357)	3.24 (1411)	3.23 (2739)	3.21 (2387)	
	7		3.15 (453)	3.18 (3263)	3.19 (3197)	3.18 (3252)	3.22 (631)	3.24 (5813)	3.23 (5327)	3.25 (5035)	
	15		3.12 (947)	3.16 (7107)	3.19 (5650)	3.16 (7575)	3.17 (2103)	3.12 (17466)	3.23 (15258)	3.22 (5974)	
90	0		5.19 (77)	5.22 (271)	5.24 (353)	5.24 (517)	5.29 (166)	5.30 (470)	5.28 (608)	5.29 (993)	
	1		5.19 (89)	5.22 (416)	5.24 (455)	5.25 (748)	5.31 (223)	5.32 (887)	5.30 (1146)	5.29 (1266)	5.31
	3		5.22 (239)	5.26 (1036)	5.27 (1259)	5.24 (1230)	5.33 (493)	5.34 (2624)	5.28 (1427)	5.33 (2387)	
	7		5.19 (307)	5.23 (2490)	5.26 (2348)	5.25 (2534)	5.32 (1584)	5.30 (3909)	5.30 (4560)	5.34 (5803)	
	15		5.23 (1189)	5.25 (5729)	5.26 (6236)	5.27 (6503)	5.28 (2120)	5.28 (9220)	5.28 (9943)	5.29 (6216)	
100	0		8.30 (81)	8.33 (260)	8.36 (472)	8.38 (566)	8.44 (189)	8.46 (466)	8.45 (625)	8.45 (1099)	
	1		8.30 (93)	8.33 (402)	8.36 (413)	8.38 (732)	8.44 (191)	8.46 (742)	8.48 (1189)	8.46 (1362)	8.50
	3		8.37 (250)	8.35 (851)	8.43 (1412)	8.38 (1028)	8.44 (362)	8.46 (1256)	8.51 (2344)	8.45 (1886)	
	7		8.39 (476)	8.39 (2957)	8.44 (3366)	8.42 (3556)	8.44 (670)	8.47 (3808)	8.53 (4867)	8.52 (5165)	
	15		8.30 (573)	8.34 (3808)	8.42 (6466)	8.45 (10222)	8.44 (1361)	8.46 (9213)	8.49 (12488)	8.51 (11531)	
110	0		13.05 (84)	13.07 (229)	13.10 (325)	13.10 (519)	13.20 (216)	13.18 (486)	13.17 (646)	13.17 (1048)	
	1		13.08 (190)	13.09 (444)	13.12 (476)	13.14 (796)	13.20 (182)	13.18 (737)	13.17 (857)	13.17 (1770)	13.23
	3		13.06 (180)	13.08 (728)	13.17 (1162)	13.10 (1111)	13.24 (454)	13.20 (1635)	13.21 (2035)	13.27 (2751)	
	7		13.05 (360)	13.16 (4208)	13.13 (2252)	13.13 (2111)	13.20 (772)	13.25 (4496)	13.24 (4532)	13.21 (4336)	
	15		13.05 (812)	13.09 (5221)	13.19 (6290)	13.12 (5257)	13.27 (2118)	13.21 (9895)	13.28 (13941)	13.22 (6948)	
120	0		20.19 (86)	20.20 (281)	20.21 (307)	20.21 (704)	20.24 (174)	20.21 (535)	20.21 (620)	20.21 (1087)	
	1		20.19 (93)	20.20 (372)	20.21 (454)	20.21 (736)	20.24 (200)	20.21 (776)	20.22 (1025)	20.21 (1311)	20.00
	3		20.19 (180)	20.20 (675)	20.20 (1002)	20.19 (1286)	20.25 (468)	20.22 (1825)	20.21 (1418)	20.21 (1971)	
	7		20.19 (323)	20.22 (2411)	20.21 (2307)	20.22 (3043)	20.24 (696)	20.20 (5008)	20.21 (2715)	20.19 (4496)	
	15		20.16 (1227)	20.20 (5759)	20.19 (3662)	20.21 (7173)	20.24 (1300)	20.22 (9185)	20.20 (5834)	20.21 (7580)	
130			Always 30.00				Always 30.00				30.00
140			Always 40.00				Always 40.00				40.00

Table 6: Results for an American Put option in the rough Bergomi model using the GPR-EI method. N represents the number of time steps, P the number of the simulated paths and J the number of past values employed in the regression. Values in brackets are the computational times (in seconds).

			GPR-Tree									
		N	50					100				
K	J	P	500	1000	2000	4000	8000	500	1000	2000	4000	8000
70	0		1.91 (30)	1.87 (148)	1.75 (548)	1.81 (794)	1.82 (3181)	1.85 (59)	1.84 (192)	1.71 (797)	1.84 (1495)	1.81 (4938)
	1		1.85 (44)	1.87 (140)	1.84 (579)	1.83 (1169)	1.83 (2816)	1.85 (77)	1.86 (322)	1.84 (1112)	1.84 (2022)	1.81 (7273)
	3		1.85 (119)	1.86 (265)	1.84 (934)	1.84 (1824)	1.84 (6558)	1.88 (134)	1.86 (549)	1.75 (1912)	1.84 (3560)	1.84 (10084)
	7		1.86 (155)	1.85 (804)	1.84 (3623)	1.84 (4119)	1.84 (11826)	1.86 (236)	1.87 (1164)	1.81 (4730)	1.84 (8633)	1.84 (17666)
	15		1.85 (258)	1.85 (985)	1.84 (4400)	1.83 (12173)	1.84 (14662)	1.87 (407)	1.83 (1845)	1.79 (9945)	1.80 (20841)	1.85 (33776)
80	0		3.13 (29)	3.14 (98)	3.10 (362)	3.10 (752)	3.09 (2296)	3.13 (59)	3.10 (219)	3.06 (660)	3.03 (1523)	3.10 (4540)
	1		3.11 (41)	3.13 (144)	3.11 (490)	3.10 (1087)	3.10 (3294)	3.15 (82)	3.11 (278)	3.11 (988)	3.10 (2243)	3.10 (5547)
	3		3.13 (101)	3.12 (256)	3.11 (844)	3.11 (1725)	3.10 (5564)	3.16 (174)	3.13 (504)	3.10 (2301)	3.10 (3396)	3.10 (12266)
	7		3.12 (147)	3.12 (512)	3.11 (2246)	3.11 (4735)	3.10 (10252)	3.15 (266)	3.13 (1051)	3.11 (4292)	3.10 (7185)	3.10 (21590)
	15		3.12 (237)	3.11 (950)	3.11 (5034)	3.10 (7476)	3.10 (15470)	3.15 (537)	3.13 (2272)	3.12 (8707)	3.10 (14591)	3.09 (43726)
90	0		5.13 (42)	5.13 (137)	5.10 (351)	5.07 (686)	5.08 (1822)	5.13 (69)	5.09 (193)	5.08 (717)	5.08 (1542)	5.07 (4604)
	1		5.13 (61)	5.12 (142)	5.10 (536)	5.10 (1119)	5.09 (2774)	5.16 (91)	5.10 (277)	5.08 (996)	5.08 (2132)	5.06 (7586)
	3		5.11 (74)	5.13 (402)	5.10 (949)	5.10 (1735)	5.10 (10086)	5.15 (135)	5.12 (503)	5.11 (1790)	5.09 (3774)	5.06 (11529)
	7		5.13 (135)	5.12 (493)	5.11 (2217)	5.10 (3852)	5.10 (10736)	5.16 (361)	5.13 (1070)	5.11 (3709)	5.09 (6594)	5.10 (18562)
	15		5.12 (423)	5.13 (789)	5.11 (5634)	5.10 (7094)	5.10 (14285)	5.13 (545)	5.13 (1893)	5.11 (8520)	5.10 (12872)	5.10 (35265)
100	0		8.21 (31)	8.21 (146)	8.18 (485)	8.18 (703)	8.18 (2286)	8.19 (64)	8.17 (213)	8.16 (654)	8.16 (1477)	8.15 (4895)
	1		8.23 (42)	8.23 (131)	8.22 (701)	8.21 (1130)	8.21 (5189)	8.23 (88)	8.21 (402)	8.17 (844)	8.18 (2254)	8.18 (6878)
	3		8.23 (78)	8.24 (252)	8.19 (787)	8.21 (1781)	8.21 (4406)	8.15 (140)	8.22 (471)	8.19 (1405)	8.20 (3953)	8.19 (9540)
	7		8.23 (121)	8.24 (700)	8.23 (2604)	8.22 (3207)	8.22 (10692)	8.22 (237)	8.20 (846)	8.24 (3405)	8.21 (8021)	8.20 (19457)
	15		8.25 (263)	8.24 (855)	8.23 (4350)	8.23 (7599)	8.23 (14081)	8.21 (545)	8.23 (1941)	8.22 (7384)	8.21 (14074)	8.21 (35054)
110	0		12.96 (33)	12.97 (96)	12.94 (338)	12.94 (774)	12.94 (2236)	12.95 (64)	12.93 (210)	12.90 (619)	12.89 (1482)	12.89 (4553)
	1		13.00 (47)	13.04 (149)	12.99 (497)	12.99 (1079)	12.99 (3206)	13.00 (89)	13.01 (265)	12.98 (1001)	12.95 (1888)	12.90 (6762)
	3		12.87 (60)	13.05 (250)	13.04 (853)	13.03 (1918)	13.03 (6225)	13.02 (150)	13.01 (506)	12.98 (2244)	12.98 (3481)	12.90 (22390)
	7		13.04 (140)	13.04 (482)	13.05 (1933)	13.03 (4024)	13.03 (6225)	13.04 (241)	13.02 (1274)	13.02 (3438)	12.98 (6442)	12.98 (16022)
	15		13.03 (244)	13.07 (1007)	13.05 (4595)	13.04 (6551)	13.06 (17346)	13.06 (560)	13.05 (1873)	13.04 (8116)	13.00 (12929)	13.01 (38808)
120	0		20.17 (32)	20.17 (144)	20.17 (278)	20.17 (604)	20.16 (2255)	20.17 (65)	20.16 (223)	20.14 (535)	20.13 (1453)	20.13 (3882)
	1		20.23 (46)	20.24 (134)	20.17 (466)	20.22 (1052)	20.23 (3847)	20.17 (131)	20.16 (297)	20.21 (938)	20.19 (2203)	20.21 (7013)
	3		20.17 (70)	20.24 (382)	20.25 (1525)	20.23 (1562)	20.25 (6177)	20.17 (158)	20.24 (451)	20.24 (1653)	20.19 (3776)	20.20 (12783)
	7		20.22 (204)	20.26 (480)	20.24 (2486)	20.26 (3076)	20.27 (8032)	20.17 (269)	20.24 (855)	20.24 (3019)	20.24 (7113)	20.23 (16677)
	15		20.19 (255)	20.26 (843)	20.27 (2753)	20.27 (5460)	20.26 (13966)	20.17 (452)	20.25 (1892)	20.23 (5783)	20.16 (12247)	20.16 (34612)
130			Always 30.00					Always 30.00				
140			Always 40.00					Always 40.00				

Table 7: Results for an American Put option in the rough Bergomi model (case $H = 0.80$) using the GPR-Tree method and with a double tree step ($m = 2$). N represents the number of time steps, P the number of the simulated paths and J the number of past values employed in the regression. Values in brackets are the computational times (in seconds).

			GPR-EI									
		N	50					100				
K	J	P	1000	2000	4000	8000	16000	1000	2000	4000	8000	16000
70	0		1.84 (83)	1.82 (336)	1.84 (572)	1.81 (600)	1.83 (2141)	1.77 (145)	1.81 (534)	1.83 (708)	1.84 (1052)	1.83 (4792)
	1		1.85 (112)	1.84 (487)	1.84 (533)	1.84 (791)	1.83 (2746)	1.82 (183)	1.83 (900)	1.84 (1007)	1.84 (1454)	1.84 (5773)
	3		1.85 (188)	1.80 (1086)	1.84 (978)	1.83 (1589)	1.83 (4887)	1.82 (433)	1.84 (2086)	1.84 (2166)	1.84 (2265)	1.83 (5816)
	7		1.80 (473)	1.84 (2748)	1.82 (2496)	1.84 (3250)	1.83 (4887)	1.81 (1082)	1.85 (4885)	1.85 (3492)	1.84 (5284)	1.82 (13684)
	15		1.81 (777)	1.83 (5850)	1.82 (5317)	1.84 (4918)	1.84 (7696)	1.75 (2446)	1.82 (11467)	1.84 (11605)	1.84 (12387)	1.84 (15105)
80	0		3.11 (86)	3.10 (354)	3.06 (369)	3.10 (614)	3.09 (2518)	3.06 (177)	3.08 (484)	3.10 (817)	3.10 (1047)	3.09 (4268)
	1		3.12 (119)	3.11 (480)	3.11 (548)	3.10 (724)	3.10 (2220)	3.09 (208)	3.10 (1077)	3.11 (901)	3.10 (1585)	3.10 (4604)
	3		3.11 (216)	3.11 (924)	3.11 (1135)	3.10 (1712)	3.10 (2976)	3.10 (443)	3.11 (2384)	3.11 (2078)	3.10 (2405)	3.10 (6348)
	7		3.13 (471)	3.11 (3716)	3.05 (4801)	3.10 (2295)	3.10 (4523)	3.06 (1086)	3.09 (5049)	3.11 (5091)	3.11 (5435)	3.10 (9036)
	15		3.11 (1019)	3.10 (4239)	3.11 (5049)	3.10 (5466)	3.09 (7634)	3.10 (2630)	3.12 (10232)	3.11 (15950)	3.09 (10985)	3.11 (18322)
90	0		5.11 (108)	5.10 (298)	5.09 (363)	5.08 (536)	5.07 (1952)	5.08 (146)	5.09 (392)	5.09 (712)	5.08 (933)	5.06 (4095)
	1		5.11 (123)	5.12 (515)	5.10 (506)	5.09 (692)	5.09 (3204)	5.08 (201)	5.11 (813)	5.10 (911)	5.09 (1371)	5.08 (5265)
	3		5.12 (177)	5.11 (982)	5.09 (1398)	5.10 (1182)	5.09 (2993)	5.12 (517)	5.11 (2442)	5.10 (2050)	5.09 (2159)	5.08 (5608)
	7		5.10 (294)	5.12 (2403)	5.11 (2724)	5.10 (2209)	5.10 (5164)	5.11 (779)	5.11 (4771)	5.03 (7348)	5.10 (5465)	5.09 (12761)
	15		5.12 (799)	5.11 (5672)	5.10 (7810)	5.09 (4394)	5.09 (7597)	5.12 (2907)	5.05 (11564)	5.10 (11152)	5.10 (11409)	5.08 (25286)
100	0		8.19 (69)	8.18 (278)	8.17 (307)	8.17 (570)	8.15 (1939)	8.18 (179)	8.18 (448)	8.17 (484)	8.15 (947)	8.14 (4916)
	1		8.22 (94)	8.21 (454)	8.18 (424)	8.20 (682)	8.20 (2555)	8.21 (206)	8.20 (701)	8.19 (1064)	8.17 (1358)	8.18 (4697)
	3		8.19 (166)	8.18 (918)	8.22 (1143)	8.22 (1327)	8.21 (2843)	8.21 (442)	8.20 (1625)	8.20 (1951)	8.18 (2345)	8.18 (6309)
	7		8.23 (416)	8.22 (2306)	8.24 (2956)	8.22 (2762)	8.22 (5105)	8.22 (840)	8.22 (4472)	8.22 (4074)	8.18 (4724)	8.19 (10050)
	15		8.20 (831)	8.23 (3783)	8.22 (5024)	8.21 (7017)	8.20 (9061)	8.22 (2251)	8.21 (9295)	8.18 (10871)	8.20 (12163)	8.18 (15536)
110	0		12.95 (74)	12.92 (231)	12.92 (309)	12.92 (538)	12.91 (1847)	12.91 (197)	12.90 (489)	12.90 (560)	12.89 (1280)	12.88 (3985)
	1		12.99 (104)	13.00 (377)	13.01 (543)	13.00 (778)	12.91 (2166)	12.98 (184)	12.95 (796)	12.93 (909)	12.96 (1353)	12.88 (5367)
	3		13.00 (196)	12.98 (683)	13.03 (914)	13.00 (1611)	13.00 (2937)	12.98 (369)	13.00 (1908)	13.00 (1951)	12.95 (3190)	12.94 (6431)
	7		13.02 (369)	13.02 (2049)	13.04 (3056)	13.02 (2103)	13.03 (4324)	12.99 (1108)	13.02 (3690)	13.01 (3738)	13.01 (4953)	13.00 (8514)
	15		13.00 (1141)	13.02 (4043)	13.08 (4610)	13.04 (5089)	13.02 (6459)	13.02 (2243)	12.90 (6836)	13.02 (7513)	13.01 (8240)	13.00 (15697)
120	0		20.17 (101)	20.15 (255)	20.14 (277)	20.14 (474)	20.14 (2264)	20.14 (161)	20.13 (386)	20.12 (573)	20.12 (1001)	20.12 (4868)
	1		20.21 (126)	20.21 (415)	20.17 (403)	20.22 (737)	20.21 (2279)	20.19 (190)	20.19 (723)	20.19 (881)	20.19 (1233)	20.19 (4393)
	3		20.19 (236)	20.25 (929)	20.22 (847)	20.21 (1289)	20.22 (2650)	20.18 (387)	20.15 (1626)	20.21 (1897)	20.21 (2062)	20.19 (5487)
	7		20.27 (317)	20.24 (1569)	20.24 (2500)	20.22 (3148)	20.23 (3789)	20.20 (852)	20.15 (2243)	20.22 (3352)	20.23 (3991)	20.19 (7113)
	15		20.21 (630)	20.25 (3051)	20.26 (3230)	20.24 (4027)	20.20 (5392)	20.24 (1538)	20.19 (4785)	20.26 (6696)	20.20 (7249)	20.28 (12195)
130			Always 30.00					Always 30.00				
140			Always 40.00					Always 40.00				

Table 8: Results for an American Put option in the rough Bergomi model (case $H = 0.80$) using the GPR-EI method. N represents the number of time steps, P the number of the simulated paths and J the number of past values employed in the regression. Values in brackets are the computational times (in seconds).

10 Conclusions

In this paper we have presented two numerical methods to compute the price of American options on a basket of underlyings following the Black-Scholes dynamics. These two methods are based on the GPR-Monte Carlo method and improve its results in terms of accuracy and computational time. The GPR-Tree method can be applied for dimensions up to $d = 20$ and it proves to be very efficient when $d \leq 10$. The GPR-Exact Integration method proves to be particularly flexible and stands out for the small computational cost which allows one to obtain excellent estimates in a very short time. The two methods also turns out to be an effective tool to address non-Markovian problems such as the pricing of American options in the rough Bergomi model. These two methods are thus a step forward in overcoming 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] Aurélien Alfonsi. High order discretization schemes for the CIR process: application to affine term structure and Heston models. *Mathematics of Computation*, 79(269):209–237, 2010. 10

[3] 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

[4] Vlad Bally, Gilles Pagès, and Jacques Printems. First-order schemes in the numerical quantization method. *Mathematical Finance*, 13(1):1–16, 2003. 2

[5] Christian Bayer, Peter Friz, and Jim Gatheral. Pricing under rough volatility. *Quantitative Finance*, 16(6):887–904, 2016. 3, 8, 9

[6] Christian Bayer, Raúl Tempone, and Sören Wolfers. Pricing american options by exercise rate optimization. *arXiv: 1809.07300*, 2018. 3, 12, 17

[7] Christian Beck, Weinan E, and Arnulf Jentzen. Machine Learning approximation algorithms for high-dimensional fully nonlinear partial differential equations and second-order backward stochastic differential equations. *Journal of Nonlinear Science*, pages 1–57, 2017. 2

[8] Sebastian Becker, Patrick Cheridito, and Arnulf Jentzen. Deep optimal stopping. *Journal of Machine Learning Research*, 20(74):1–25, 2019. 2

[9] 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

[10] Mark Broadie and Paul Glasserman. Pricing American-style securities using simulation. *Journal of Economic Dynamics and Control*, 21(8-9):1323–1352, 1997. 2

[11] John C Cox, Stephen A Ross, and Mark Rubinstein. Option pricing: A simplified approach. *Journal of Financial Economics*, 7(3):229–263, 1979. 14

[12] Jan De Spiegeleer, Dilip B Madan, Sofie Reyners, and Wim Schoutens. Machine Learning for quantitative finance: fast derivative pricing, hedging and fitting. *Quantitative Finance*, 18(10):1635–1643, 2018. 2

[13] Weinan E, Jiequn Han, and Arnulf Jentzen. Deep Learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Communications in Mathematics and Statistics*, 5(4):349–380, 2017. 2

[14] Niklas Ekvall. A lattice approach for pricing of multivariate contingent claims. *European Journal of Operational Research*, 91(2):214–228, 1996. 6, 14, 16

[15] Guojun Gan. Application of Data Clustering and Machine Learning in Variable Annuity valuation. *Insurance: Mathematics and Economics*, 53(3):795–801, 2013. 2

[16] Guojun Gan and X. Sheldon Lin. Valuation of large variable annuity portfolios under nested simulation: A functional data approach. *Insurance: Mathematics and Economics*, 62:138 – 150, 2015. 2

[17] 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. 2, 3, 6, 7, 14

[18] Loic Le Gratiet and Josselin Garnier. Regularity dependence of the rate of convergence of the learning curve for gaussian process regression. *arXiv: 1210.2879*, 2012. 2

[19] Jiequn Han, Arnulf Jentzen, and Weinan E. Solving high-dimensional partial differential equations using Deep Learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, 2018. 2

[20] Martin B Haugh and Leonid Kogan. Pricing American options: a duality approach. *Operations Research*, 52(2):258–270, 2004. 2

[21] Robert V Hogg, Joseph McKean, and Allen T Craig. *Introduction to Mathematical Statistics*. Pearson Education, 2005. 27

[22] Blanka Horvath, Antoine Jacquier, and Aitor Muguruza. Functional central limit theorems for rough volatility. *arXiv: 1711.03078*, 2017. 3, 11

[23] 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

[24] Michael Kohler, Adam Krzyżak, and Nebojsa Todorovic. Pricing of high-dimensional American options by neural networks. *Mathematical Finance*, 20(3):383–410, 2010. 2

[25] Jérôme Lelong. Dual pricing of American options by Wiener chaos expansion. *SIAM Journal on Financial Mathematics*, 9(2):493–519, 2018. 2

[26] 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

[27] 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

[28] Mike Ludkovski. Kriging metamodels and experimental design for Bermudan option pricing. *Journal of Computational Finance*, 22(1), 2018. 2, 3

[29] Michael A Nielsen. *Neural Networks and Deep Learning*, volume 25. Determination press San Francisco, CA, USA, 2015. 2

[30] Grigorios A Pavliotis. *Stochastic processes and applications: diffusion processes, the Fokker-Planck and Langevin equations*, volume 60. Springer, 2014. 10

[31] Leonard CG Rogers. Monte Carlo valuation of American options. *Mathematical Finance*, 12(3):271–286, 2002. 2

[32] 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

[33] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian Processes for Machine Learning*. MIT Press Cambridge, MA, 2006. 2, 4

A Proof of Proposition 1

Let $n \in \{0, \dots, N-1\}$ and suppose the function u_{n+1}^{GPR-EI} to be known at Z and to approximate $u(t_N, \cdot)$. This is true for $n = N-1$ since, by definition, $u_N^{GPR-EI} = \Psi$. Let us define the quantity

$$\hat{\mathbf{x}}^p = \exp\left(\mathbf{z}^p + \left(r - \frac{1}{2}\boldsymbol{\sigma}^2\right) t_n\right) \tag{A.1}$$

for $p = 1, \dots, P$. The function $u(t_n, \cdot)$ at time t_n at $\mathbf{z}^p$ follows

$$u(t_n, \mathbf{z}^p) = v(t_n, \hat{\mathbf{x}}^p). \tag{A.2}$$

$$= \max\left(\Psi(\hat{\mathbf{x}}^p), C(t_n, \hat{\mathbf{x}}^p)\right), \tag{A.3}$$

where

$$C(t_n, \hat{\mathbf{x}}^p) = \mathbb{E}_{t_n, \hat{\mathbf{x}}^p}\left[e^{-r\Delta t} v\left(t_{n+1}, \mathbf{S}_{t_{n+1}}\right)\right] \tag{A.4}$$

We can also write

$$C(t_n, \hat{\mathbf{x}}^p) = \mathbb{E}_{t_n, \hat{\mathbf{x}}^p}\left[e^{-r\Delta t} u\left(t_{n+1}, \log\left(\mathbf{S}_{t_{n+1}}\right) - \left(r - \frac{1}{2}\boldsymbol{\sigma}^2\right) t_{n+1}\right)\right] \tag{A.5}$$

$$= \mathbb{E}_{t_n, \hat{\mathbf{x}}^p}\left[e^{-r\Delta t} u\left(t_{n+1}, \mathbf{Z}_{t_{n+1}}\right)\right] \tag{A.6}$$

where $\mathbf{Z}_{t_{n+1}}$ is the random variable defined as

$$\mathbf{Z}_{t_{n+1}} = \log\left(\mathbf{S}_{t_{n+1}}\right) - \left(r - \frac{1}{2}\boldsymbol{\sigma}^2\right) t_{n+1}. \tag{A.7}$$

Let us define $\Pi = (\Pi_{i,j})$ as the $d \times d$ covariance matrix of the log-increments, that is $\Pi_{i,j} = \rho_{i,j}\sigma_i\sigma_j\Delta t$. Moreover, let Λ be a square root of Π and $\mathbf{G}$ as a vector that follows a standard Gaussian law. Then, we observe that $\mathbf{Z}_{t_{n+1}}$ has the following conditional law

$$\mathbf{Z}_{t_{n+1}} | \mathbf{S}_{t_n} = \hat{\mathbf{x}}^p \sim \mathcal{N}(\mathbf{z}^p, \Pi). \tag{A.8}$$

In fact, simple Algebra leads to

$$\mathbf{Z}_{t_{n+1}} = \mathbf{z}^p + \Lambda \mathbf{G}. \tag{A.9}$$

Moreover, relation (A.8) can also be stated as

$$\mathbf{Z}_{t_{n+1}} \left| \left(\log\left(\mathbf{S}_{t_n}\right) - \left(r - \frac{1}{2}\boldsymbol{\sigma}^2\right) t_n = \mathbf{z}^p \right) \sim \mathcal{N}\left(\mathbf{z}^p, \mathit{\Pi}\right). \right. \tag{A.10}$$

Let $f_{\mathbf{z}^p}(\mathbf{z})$ denote the density function of $\mathbf{Z}_{t_{n+1}}$ given $\log\left(\mathbf{S}_{t_n}\right) - \left(r - \frac{1}{2}\boldsymbol{\sigma}^2\right) t_n = \mathbf{z}^p$. Specifically,

$$f_{\mathbf{z}^p}(\mathbf{z}) = \frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{\det\left(\Pi\right)}} \exp\left(-\frac{1}{2}\left(\mathbf{z}-\mathbf{z}^p\right)^\top \Pi^{-1}\left(\mathbf{z}-\mathbf{z}^p\right)\right). \tag{A.11}$$

Then, according to (A.6),we can write

$$C\left(t_n, \hat{\mathbf{x}}^p\right) = e^{-r\Delta t}\int_{\mathbb{R}^d} f_{\mathbf{z}^p}(\mathbf{z})\, u\left(t_{n+1}, \mathbf{z}\right) d\mathbf{z}. \tag{A.12}$$

$$\approx e^{-r\Delta t}\int_{\mathbb{R}^d} f_{\mathbf{z}^p}(\mathbf{z})\, u_{n+1}^{GPR-EI}(\mathbf{z})\, d\mathbf{z}. \tag{A.13}$$

Now, let us consider GPR approximation of the function u_{n+1}^{GPR-EI}, obtained by assuming Z as the predictor set and by employing the Squared Exponential Kernel $k_{SE} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. According to equation (3.6), let

$$u_{n+1}^{GPR}(\mathbf{z}) = \sum_{q=1}^{P} \omega_q k_{SE}\left(\mathbf{z}, \mathbf{z}^q\right) \tag{A.14}$$

be the GPR approximation of the function $u\left(t_{n+1}, \mathbf{z}\right)$. The GPR-EI approximation C_n^{GPR-EI} of the continuation value is then given by

$$C_n^{GPR-EI}\left(\hat{\mathbf{x}}^p\right) = e^{-r\Delta t}\int_{\mathbb{R}^d} f_{\mathbf{z}^p}(\mathbf{z})\, u_{n+1}^{GPR}(\mathbf{z})\, d\mathbf{z}, \tag{A.15}$$

$$= e^{-r\Delta t}\sum_{q=1}^{P} \omega_q \int_{\mathbb{R}^d} f_{\mathbf{z}^p}(\mathbf{z})\, k_{SE}\left(\mathbf{z}^q, \mathbf{z}\right) d\mathbf{z}. \tag{A.16}$$

To compute each integral in (A.16), we observe that

$$\begin{aligned}
&\int_{\mathbb{R}^d} f_{\mathbf{z}^p}(\mathbf{z})\, k_{SE}\left(\mathbf{z}^q, \mathbf{z}\right) d\mathbf{z} = \\
&\quad = (2\pi)^{\frac{d}{2}}\sigma_f^2\sigma_l^d \int_{\mathbb{R}^d} \frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{\det\left(\Pi\right)}} e^{-\frac{1}{2}\left(\mathbf{z}-\mathbf{z}^p\right)^\top \Pi^{-1}\left(\mathbf{z}-\mathbf{z}^p\right)} \frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{\sigma_l^{2d}}} e^{-\frac{1}{2}\left(\mathbf{z}-\mathbf{z}^q\right)^\top \left(\sigma_l^2 I_d\right)^{-1}\left(\mathbf{z}-\mathbf{z}^q\right)} d\mathbf{z} \\
&= (2\pi)^{\frac{d}{2}}\sigma_f^2\sigma_l^d \int_{\mathbb{R}^d} \frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{\det\left(\Pi\right)}} e^{-\frac{1}{2}\left(\mathbf{z}-\mathbf{z}^p\right)^\top \Pi^{-1}\left(\mathbf{z}-\mathbf{z}^p\right)} \frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{\sigma_l^{2d}}} e^{-\frac{1}{2}\left((\mathbf{0}-z)-\left(-\mathbf{z}^q\right)\right)^\top \left(\sigma_l^2 I_d\right)^{-1}\left((\mathbf{0}-\mathbf{z})-\left(-\mathbf{z}^q\right)\right)} d\mathbf{z} \\
&\quad = (2\pi)^{\frac{d}{2}}\sigma_f^2\sigma_l^d f_{\mathbf{z}^p} * g_{-\mathbf{z}^q}(0)
\end{aligned} \tag{A.17}$$

where $*$ is the convolution product and $g_{-\mathbf{z}^q}$ is the density function of a Gaussian random vector which has law given by $\mathcal{N}\left(-\mathbf{z}^q, \sigma_l^2 I_d\right)$. Moreover, the convolution product of the densities of two independent random variables is equal to the density of their sum (see Hogg et al. [21]) and we can obtain the following relation which allows one to exactly compute the integrals in (A.16):

$$f_{\mathbf{x}^p} * g_{-\mathbf{x}^q}(0) = \frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{\det\left(\Pi + \sigma_l^2 I_d\right)}} e^{-\frac{1}{2}\left(\mathbf{z}^q-\mathbf{z}^p\right)^\top \left(\Pi+\sigma_l^2 I_d\right)^{-1}\left(\mathbf{z}^q-\mathbf{z}^p\right)}. \tag{A.18}$$

Therefore, the GPR-EI approximation C_n^{GPR-EI} at $\hat{\mathbf{x}}^p$ reads

$$C_n^{GPR-EI}\left(\hat{\mathbf{x}}^p\right) = e^{-r\Delta t}\sum_{q=1}^{P}\omega_q\sigma_f^2\sigma_l^d\frac{e^{-\frac{1}{2}\left(\mathbf{z}^q-\mathbf{z}^p\right)^\top\left(\Pi+\sigma_l^2 I_d\right)^{-1}\left(\mathbf{z}^q-\mathbf{z}^p\right)}}{\sqrt{\det\left(\Pi+\sigma_l^2 I_d\right)}}, \tag{A.19}$$

and the GPR-EI approximation u_n^{GPR-EI} of the function $u\left(t_n,\cdot\right)$ at time t_n and at $\mathbf{z}^p$ is given by

$$u_n^{GPR-EI}\left(\mathbf{z}^p\right) = \max\left(\Psi\left(\hat{\mathbf{x}}^p\right), e^{-r\Delta t}\sum_{q=1}^{P}\omega_q\sigma_f^2\sigma_l^d\frac{e^{-\frac{1}{2}\left(\mathbf{z}^q-\mathbf{z}^p\right)^\top\left(\Pi+\sigma_l^2 I_d\right)^{-1}\left(\mathbf{z}^q-\mathbf{z}^p\right)}}{\sqrt{\det\left(\Pi+\sigma_l^2 I_d\right)}}\right). \tag{A.20}$$

B Covariance of the vector R in (7.1)

Let us report the formulas for the covariance of the components of the vector R in (7.1). For all $n=1,\dots,N$, and $m=1,\dots,n-1$, the following relations hold:

$$Cov\left(\Delta W_n^1, \Delta W_n^1\right) = \Delta t, \tag{B.1}$$

$$Cov\left(\Delta W_n^1, \widetilde{W}_{t_n}^H\right) = \frac{2\rho\sqrt{2H}}{2H+1}\left(\Delta t\right)^{H+\frac{1}{2}}, \tag{B.2}$$

$$Cov\left(\widetilde{W}_{t_n}^H, \widetilde{W}_{t_n}^H\right) = \left(t_n\right)^{2H} \tag{B.3}$$

$$Cov\left(\Delta W_m^1, \Delta W_n^1\right) = 0, \tag{B.4}$$

$$Cov\left(\Delta W_n^1, \widetilde{W}_{t_m}^H\right) = 0, \tag{B.5}$$

$$Cov\left(\Delta W_m^1, \widetilde{W}_{t_n}^H\right) = \frac{2\rho\sqrt{2H}}{2H+1}\left(\left(t_n-t_{m-1}\right)^{H+\frac{1}{2}} - \left(t_n-t_m\right)^{H+\frac{1}{2}}\right), \tag{B.6}$$

$$Cov\left(\widetilde{W}_{t_m}^H, \widetilde{W}_{t_n}^H\right) = 2H\left(t_m\right)^{2H}\cdot\int_0^1\frac{ds}{\left(1-s\right)^{\frac{1}{2}-H}\left(\frac{t_m}{t_n}-s\right)^{\frac{1}{2}-H}}. \tag{B.7}$$

C Proof of Proposition 3

Let us denote the random vector $\left(S_{t_i}, V_{t_i}, S_{t_{i+1}}, V_{t_{i+1}}, \dots, S_{t_j}, V_{t_j}\right)^\top$ for $i,j\in\{0,\dots,N\}$ and $i<j$ with $\mathbf{SV}_{i:j}$. We observe that the option value $v\left(t_N,\cdot\right)$ at time t_N is given by the payoff function Ψ, which only depends by the final value of the underlying. The option value $v\left(t_{N-1},\cdot\right)$ at time t_{N-1} about the p-th path is given by

$$v\left(t_{N-1}, \mathbf{SV}_{1:(N-1)}^p\right) = \max\left(\Psi\left(S_{t_{N-1}}^p\right), e^{-r\Delta t}C\left(t_{N-1}, \mathbf{SV}_{1:(N-1)}^p\right)\right) \tag{C.1}$$

where C stands for the continuation value and it is equal to

$$C\left(t_{N-1}, \mathbf{SV}_{1:(N-1)}^p\right) = E\left[e^{-r\Delta t}\Psi\left(S_{t_N}\right)\middle|\left(\mathbf{SV}_{1:(N-1)} = \mathbf{SV}_{1:(N-1)}^p\right)\right]. \tag{C.2}$$

We approximate the continuation value in (C.2) by means of the GPR approximation of Ψ. In particular, let $\Psi^{GPR}\left(z\right)$ be the approximation of the function $z\mapsto\Psi\left(\exp\left(z\right)\right)$ by using the GPR method employing the

Squared Exponential Kernel and considering the log-underlying values at maturity as predictors. Specifically, the predictor set is

$$Z = \left\{ z^p = \log\left(S^p_{t_N}\right), p = 1, \dots, P \right\} \subset \mathbb{R} \tag{C.3}$$

and the response $\mathbf{y} \in \mathbb{R}^P$ is given by

$$y^p = \Psi\left(S^p_{t_N}\right). \tag{C.4}$$

In particular, we can write

$$\Psi^{GPR}(z) = \sum_{q=1}^{P} k_{SE}\left(\log\left(S^q_{t_N}\right), z\right)\omega_q = \sigma_f^2 \sum_{q=1}^{P} \exp\left(-\frac{\left(\log\left(S^q_{t_N}\right) - z\right)^2}{2\sigma_l^2}\right)\omega_q \tag{C.5}$$

where k_{SE} is the Squared Exponential kernel, σ_l is the characteristic length scale, σ_f is the signal standard deviation and $\omega_1, \dots, \omega_P$ are weights.

So we approximate the continuation value $C\left(t_{N-1}, \mathbf{SV}^p_{1:(N-1)}\right)$ with the expression:

$$E\left[e^{-r\Delta t}\Psi^{GPR}\left(\ln\left(S_{t_N}\right)\right) \middle| \left(\mathbf{SV}_{1:(N-1)} = \mathbf{SV}^p_{1:(N-1)}\right)\right]. \tag{C.6}$$

We observe that the law of $\log\left(S_{t_N}\right)$ given $S^p_{t_1}, V^p_{t_1}, \dots, S^p_{t_{N-1}}, V^p_{t_{N-1}}$ is normal

$$\log\left(S_{t_N}\right) \middle| \left(\mathbf{SV}_{1:(N-1)} = \mathbf{SV}^p_{1:(N-1)}\right) \sim \mathcal{N}\left(\mu_{N,p}, \sigma^2_{N,p}\right), \tag{C.7}$$

where

$$\mu_{N,p} = \log\left(S^p_{t_{N-1}}\right) + \left(r - \frac{1}{2}V^p_{t_{N-1}}\right)\Delta t \tag{C.8}$$

and

$$\sigma^2_{N,p} = V^p_{t_{N-1}}\Delta t. \tag{C.9}$$

We stress out that equation (C.7) holds for the Euler approximation of the asset price process in the rough Bergomi model, not for the true process. Therefore, the GPR-EI approximation for the continuation value at time t_{N-1} is as follows:

$$\begin{aligned} C^{GPR-EI}_{N-1}\left(\mathbf{SV}^p_{1:(N-1)}\right) &= e^{-r\Delta t}\int_{\mathbb{R}} \frac{\exp\left(-\frac{(z-\mu_{N,p})^2}{2\sigma^2_{N,p}}\right)}{\sqrt{2\pi\sigma^2_{N,p}}}\Psi^{GPR}(z)\,dz \\ &= e^{-r\Delta t}\sigma_f^2\sqrt{2\pi\sigma_l^2}\sum_{q=1}^{P}\int_{\mathbb{R}} \frac{\exp\left(-\frac{(z-\mu_{N,p})^2}{2\sigma^2_{N,p}}\right)}{\sqrt{2\pi\sigma^2_{N,p}}}\frac{\exp\left(-\frac{\left(\log\left(S^q_{t_N}\right)-z\right)^2}{2\sigma_l^2}\right)}{\sqrt{2\pi\sigma_l^2}}\omega_q dz. \end{aligned} \tag{C.10}$$

Taking advantage of the properties of the convolution between density functions, we obtain

$$C^{GPR-EI}_{N-1}\left(\mathbf{SV}^p_{1:(N-1)}\right) = e^{-r\Delta t}\sum_{q=1}^{P}\frac{\omega_q\sigma_f^2\sigma_l}{\sqrt{\sigma^2_{N,p}+\sigma_l^2}}\exp\left(-\frac{\left(\log\left(S^q_{t_N}\right)-\mu_{N,p}\right)^2}{2\sigma^2_{N,p}+2\sigma_l^2}\right). \tag{C.11}$$

D Proof of Proposition 4

In order to proceed backward, from t_{N-2} up to t_1 we consider an integer positive value J and train the GPR method considering the last $J+1$ observed values of the couple $\left(\log\left(S^p_{t_n}\right), \log\left(V^p_{t_n}\right)\right)$ as predictors, and the option price as response. Specifically, the predictor set is

$$Z = \left\{ \mathbf{z}^p = \log\left(\mathbf{SV}^p_{\max\{1,N-1-J\}:(N-1)}\right), p = 1, \ldots, P \right\} \subset \mathbb{R}^{d_{N-1}} \tag{D.1}$$

where $d_{N-1} = 2\min\{N-1, J+1\}$ and the response $\mathbf{y} \in \mathbb{R}^P$ is given by

$$y^p = v\left(t_{N-1}, \mathbf{SV}^p_{1:(N-1)}\right). \tag{D.2}$$

We term u^{GPR}_{N-1} the obtained function. In particular, $u^{GPR}_{N-1} : \mathbb{R}^{d_{N-1}} \to \mathbb{R}$ and

$$u^{GPR}_{N-1}\left(\log\left(\mathbf{SV}^p_{\max\{1,N-1-J\}:(N-1)}\right)\right) \tag{D.3}$$

approximates $v\left(t_{N-1}, \mathbf{SV}^p_{1:(N-1)}\right)$.

Since the predictors have different nature (log-prices and log-volatilities at different times), we use the Automatic Relevance Determination (ARD) Squared Exponential Kernel k_{ASE} to perform the GPR regression. In particular, if d is the dimension of the space containing the predictors, it holds

$$k_{ASE}(\mathbf{a}, \mathbf{b}) = \sigma_f^2 \exp\left(-\sum_{i=1}^{d} \frac{(a_i - b_i)^2}{2\sigma_i^2}\right), \ \mathbf{a}, \mathbf{b} \in \mathbb{R}^d, \tag{D.4}$$

As opposed to the Squared Exponential kernel, the ARD Squared Exponential kernel considers a different length scale σ_i for each predictor that allows the regression to better learn the impact of each predictor on the response.

We present now how to perform the backward induction. So, let us consider $n \in \{0, \ldots, N-2\}$ and suppose the GPR approximation $u^{GPR}_{n+1} : \mathbb{R}^{d_{n+1}} \to \mathbb{R}$ to be known. In particular, $d_{n+1} = 2\min\{n+1, J+1\}$ and for each $\mathbf{z} = \left(z_1, \ldots, z_{d_{n+1}}\right) \in \mathbb{R}^{d_{n+1}}$, it holds

$$u^{GPR}_{n+1}(\mathbf{z}) = \sigma_f^2 \sum_{q=1}^{P} \omega_q \exp\left(-\sum_{i=1}^{d_n} \frac{\left(z_i^q - z_i\right)^2}{2\sigma_i^2}\right), \tag{D.5}$$

where $z_i^q = \log\left(S^q_{n+1-(i-1)/2}\right)$ if i is even and $z_i^q = \log\left(V^q_{n+1-i/2}\right)$ if i is odd, for $i = 1, \ldots, d_{n+1}$. This means that z_i^q is the observed log-price at time $t_{n+1-(i-1)/2}$ of the q-th path if i is even, and it is the observed log-volatility at time $t_{n+1-(i-1)/2}$ of the q-th path if i is odd.

We explain now how to compute the GPR approximation $v^{GPR-EI}_n : \mathbb{R}^{d_n} \to \mathbb{R}$ of the price function at time t_n. First of all, we observe that the vector $\left(\log\left(S^p_{t_{n+1}}\right), \log\left(V^p_{t_{n+1}}\right)\right)^\top$ is not $\hat{\mathcal{F}}_{t_n}$-measurable whereas $\log\left(\mathbf{SV}^p_{\max\{1,n+1-J\}:n}\right)$ is $\hat{\mathcal{F}}_{t_n}$-measurable. The law of $\left(\log\left(S_{t_{n+1}}\right), \log\left(V_{t_{n+1}}\right)\right)^\top$ given $S^p_{t_n}, V^p_{t_n}, \ldots, S^p_{t_1}, V^p_{t_1}$ is normal:

$$\left(\log\left(S_{t_{n+1}}\right), \log\left(V_{t_{n+1}}\right)\right)^\top | (\mathbf{SV}_{1:n} = \mathbf{SV}^p_{1:n}) \sim \mathcal{N}(\mu_{n+1,p}, \Sigma_{n+1,p}), \tag{D.6}$$

We stress out that equation (D.6) holds for the Euler approximation of the asset price process in the rough Bergomi model, not for the true process. In particular

$$\mu_{n+1,p} = \left(\log\left(S^p_{t_n}\right) + \left(r - \frac{1}{2}V^p_{t_n}\right)\Delta t, \log(\xi_0) + \eta\Lambda_{2n+2}\underline{\mathbf{G}}^p - \frac{1}{2}\eta^2 t_{n+1}^{2H}\right)^\top, \tag{D.7}$$

where Λ_{2n+2} is the $2n+2$-th row of the matrix Λ and $\underline{\mathbf{G}}^p = (G_1^p, \dots, G_{2n}^p, 0 \dots, 0)^\top$. Moreover, the covariance matrix is given by

$$\Sigma_{n+1,p} = \begin{pmatrix} \Delta t V_{t_n}^p & \eta\sqrt{\Delta t V_{t_n}^p}\Lambda_{2n+2,2n+1} \\ \eta\sqrt{\Delta t V_{t_n}^p}\Lambda_{2n+2,2n+1} & \eta^2\left(\Lambda_{2n+2,2n+2}^2 + \Lambda_{2n+2,2n+1}^2\right) \end{pmatrix}, \tag{D.8}$$

where $\Lambda_{i,j}$ stands for the element of Λ in position i, j. Using a similar reasoning as done for the continuation value at time t_{N-1}, one can obtain the following GPR-EI approximation for the continuation value at time t_{n-1}:

$$C_n^{GPR-EI}\left(\mathbf{SV}^p_{\max\{1,n-J\}:n}\right) = e^{-r\Delta t}\sigma_f^2\sigma_{d_{n+1}-1}\sigma_{d_{n+1}}\sum_{q=1}^{P}\omega_q h_q^p f_q^p, \tag{D.9}$$

where h_q^p and f_q^p are two factors given by

$$h_q^p = \exp\left(-\sum_{i=1}^{d_{n+1}-2}\frac{(z_i^p - z_i^q)^2}{2\sigma_i^2}\right) \tag{D.10}$$

and

$$f_q^p = \frac{\exp\left(-\frac{1}{2}\left(\begin{pmatrix} z^q_{d_{n+1}-1} \\ z^q_{d_{n+1}} \end{pmatrix} - \mu_{n+1,p}\right)^\top \left(\Sigma_{n+1,p} + \begin{pmatrix} \sigma^2_{d_{n+1}-1} & 0 \\ 0 & \sigma^2_{d_{n+1}} \end{pmatrix}\right)^{-1}\left(\begin{pmatrix} z^q_{d_{n+1}-1} \\ z^q_{d_{n+1}} \end{pmatrix} - \mu_{n+1,p}\right)\right)}{\sqrt{\det\left(\Sigma_{n+1,p} + \begin{pmatrix} \sigma^2_{d_{n+1}-1} & 0 \\ 0 & \sigma^2_{d_{n+1}} \end{pmatrix}\right)}}. \tag{D.11}$$

In particular, h_q^p measures the impact of the past observed values on the price, whereas f_q^p integrates the changes due to the diffusion of the underlying and its volatility.

Therefore, we obtain

$$v_n^{GPR-EI}\left(\mathbf{SV}^p_{\max\{1,n-J\}:n}\right) = \max\left(\Psi\left(S^p_{t_n}\right), e^{-r\Delta t}\sigma_f^2\sigma_{d_{n+1}-1}\sigma_{d_{n+1}}\sum_{q=1}^{P}\omega_q h_q^p f_q^p\right).$$

Finally, we observe that, in order to compute u_n^{GPR}, we train the GPR method considering the predictor set given by

$$Z = \left\{\mathbf{z}^p = \log\left(\mathbf{SV}^p_{\max\{1,n-J\}:n}\right), p = 1, \dots, P\right\} \subset \mathbb{R}^{d_n} \tag{D.12}$$

and the response $\mathbf{y} \in \mathbb{R}^P$ is given by

$$y^p = v_n^{GPR-EI}\left(\mathbf{SV}^p_{\max\{1,n-J\}:n}\right). \tag{D.13}$$

By induction we can compute the option price value for $n = N-2, \dots, 0$.

To conclude, we observe that the continuation value at time $t = 0$ can be computed by using (D.9) and considering $h_q^p = 1$ for $q = 1, \dots, P$ since in this case, there are no past values to consider.