Weierstraß-Institut

für Angewandte Analysis und Stochastik

Leibniz-Institut im Forschungsverbund Berlin e. V.

Preprint ISSN 2198-5855

Dynamic programming for optimal stopping via pseudo-regression

Christian Bayer, Martin Redmann, John G. M. Schoenmakers

submitted: August 16, 2018

Weierstrass Institute
Mohrenstr. 39
10117 Berlin
E-Mail: christian.bayer@wias-berlin.de
martin.redmann@wias-berlin.de
john.schoenmakers@wias-berlin.de

No. 2532
Berlin 2018

2010 *Mathematics Subject Classification.* 60G40, 65C05, 62J05.

Key words and phrases. American options, optimal stopping, linear regression.

Support from the DFG through the research unit FOR2402 is gratefully acknowledged.

Edited by
Weierstraß-Institut für Angewandte Analysis und Stochastik (WIAS)
Leibniz-Institut im Forschungsverbund Berlin e. V.
Mohrenstraße 39
10117 Berlin
Germany

Fax: +49 30 20372-303
E-Mail: preprint@wias-berlin.de
World Wide Web: http://www.wias-berlin.de/

Dynamic programming for optimal stopping via pseudo-regression

Christian Bayer, Martin Redmann, John G. M. Schoenmakers

Abstract

We introduce new variants of classical regression-based algorithms for optimal stopping problems based on computation of regression coefficients by Monte Carlo approximation of the corresponding L^2 inner products instead of the least-squares error functional. Coupled with new proposals for simulation of the underlying samples, we call the approach "pseudo regression". We show that the approach leads to asymptotically smaller errors, as well as less computational cost. The analysis is justified by numerical examples.

1 Introduction

Stochastic optimal stopping problems (in discrete time) play an important role in the theoretical as well as in the numerical literature on stochastic optimal control, since they are both generally considered difficult to solve and have many practical applications, in particular in energy and finance (where American or Bermudan options can naturally be understood as stochastic optimal stopping problems).

Many numerical methods have been suggested, ranging from PDE techniques (based on the Hamilton-Jacobi-Bellman equation of the associated continuous-time problem), to Monte Carlo (simulation) based approaches involving regression techniques, policy iteration, duality, and more. For an overview, see for instance [Gla04], [BS18b].

In this paper, we consider stochastic approaches based on the Bellman equation. A key ingredient of the classical algorithms such as the ones proposed by Longstaff and Schwartz [LS01] or Tsitsiklis and Van Roy [TVR01] is (global) regression, used to compute a conditional expectation of, say, $u(z) \coloneqq \mathbb{E}[Y|Z=z]$ for some random variables Y and Z. Given basis functions $\psi_1, \dots, \psi_K$, one thus looks for the best approximation of the unknown function u in the linear $\operatorname{span}\{\psi_1, \dots, \psi_K\}$ with respect to the distribution of Z denoted by μ, i.e., we would ideally like to solve the minimization problem

$$\beta^* \coloneqq \operatorname*{arg\,min}_{\beta \in \mathbb{R}^K} \mathbb{E}\left[\left|Y - \sum_{k=1}^K \beta_k \psi_k(Z)\right|^2\right],$$

in order to find an approximation $u(\cdot) \approx \sum_{k=1}^K \beta_k^* \psi_k(\cdot) =: u^K(\cdot)$. Classically, the above minimization problem is directly translated into the corresponding least-squares problem based on Monte Carlo approximation of the expectation, i.e., for i.i.d. samples (Y^i, Z^i), $i = 1, \dots, M$, one solves

$$\widehat{\beta} \coloneqq \operatorname*{arg\,min}_{\beta \in \mathbb{R}^K} \sum_{i=1}^M \left|Y^i - \sum_{k=1}^K \beta_k \psi_k(Z^i)\right|^2. \tag{1}$$

While well-understood by now, it is worth-while to recall that the analysis of the convergence of $\widehat{\beta}$ as $M \to \infty$ is not trivial due to the reliance on random matrix theory, see, for instance, [GKKW02].

DOI 10.20347/WIAS.PREPRINT.2532 Berlin 2018

Instead of approximating the minimization problem by Monte Carlo simulation it is also possible to directly approximate the solution β^*. Indeed, note that u^K is, of course, the linear projection of u to $\operatorname{span}\{\psi_1, \dots, \psi_K\}$ in the $L^2(\mu)$-sense. Hence, assuming for ease of notation that the basis functions $\psi_1, \dots, \psi_K$ are orthonormal w.r.t. μ—the general case requires multiplication with the Gram matrix formed by $\langle \psi_k \, , \psi_l \rangle_{L^2(\mu)}$—we have

$$\beta_k^* = \langle u \, , \psi_k \rangle_{L^2(\mu)} = \mathbb{E}\left[\mathbb{E}[Y|Z]\psi_k(Z)\right] = \mathbb{E}[Y\psi_k(Z)].$$

This formula, however, can be immediately approximated by Monte Carlo simulation giving

$$\overline{\beta}_k := \frac{1}{M}\sum_{i=1}^{M} Y^i \psi_k(Z^i), \quad k = 1, \dots, K. \tag{2}$$

From a technical point of view, convergence analysis of $\overline{\beta}$ is relatively straightforward and leads to squared error terms of the order $\frac{K}{M}$ (see Theorem 4.1). On the other hand, the squared error due to the solution $\widehat{\beta}$ of the least squares problem is of order $\frac{(1+\ln M)K}{M}$ (see Theorem 4.2). At the same time, computing $\overline{\beta}$ is also cheaper compared to computing $\widehat{\beta}$, as we avoid computing a system of linear equations (see the discussions in Section 5). However, as we see in the later Section 3.1, computation of $\overline{\beta}$ does rely on knowledge of the Gram matrix associated to the basis functions and the measure μ.

Another important detail of regression based algorithms, especially as consecutive regression steps are required, is the choice of random variables (Y, Z). Clearly, the result of the regression procedure (just as the conditional expectation) only depends on the conditional distribution of Y given Z, but not on the distribution of Z itself, which gives us considerable freedom.

In the context of Bermudan options, let X_j denote the underlying process at time j (see Section 2 for more details) and let v_j denote the option value at time j. Then, in the simplest case of dynamical programming, we need to evaluate conditional expectations $\mathbb{E}[v_j(X_j)|X_{j-1} = z]$. Hence, a very natural implementation of the regression procedure above will be based on M samples $(X_0^i, \dots, X_{\mathcal{J}}^i)$ of the whole trajectory until the expiry time $\mathcal{J}$ of the option, iteratively using slices $Y^i \equiv v_j(X_j^i)$ and $X^i \equiv X_{j-1}^i$ in the above notation, for $j = 1, \dots, \mathcal{J}$. Hence, the distribution μ of X will depend on j.

An alternative approach, especially advantageous when X_j is a homogeneous Markov process, i.e., when the conditional distribution of X_j given $X_{j-1} = z$ does not depend on j, is to fix an (carefully chosen) probability measure μ for all j. Now sample r.v.s U^i from μ and X^i from the conditional distribution of X_j given $X_{j-1} = U^i$. Hence, we obtain

$$\mathbb{E}[v_j(X)|U = z] = \mathbb{E}[v_j(X_j)|X_{j-1} = z],$$

and we can use the same batch of samples for each of the consecutive regression steps for $j = 1, \dots, \mathcal{J}$, considerably reducing the computational time of the algorithm. As an added benefit, we are now free to choose the probability measure μ. This allows us to specifically choose both μ and the basis function $\psi_1, \dots, \psi_K$ such that the basis functions are already orthogonal w.r.t. μ, implying a trivial Gram matrix.

In what follows, we call the combination of using a fixed set of sampled trajectories $X_0^i, \dots, X_{\mathcal{J}}^i$ with the least-squares estimator (1) *standard regression*, and we call a combination of samples (U^i, X^i) based on a arbitrarily chosen measure μ together with the L^2-projection estimator (2) *pseudo regression*. We argue that pseudo regression has both theoretical and numerical advantages compared with standard regression for many Bermudan option problems. Indeed,

- the convergence rates for the number of samples $M \to \infty$ are better due to the missing $\ln(M)$-term (see Theorems 4.1 and 4.2);

- the asymptotic number of floating point operations necessary is smaller (see Section 5);
- numerical examples indicate lower computational costs for fixed error tolerance, in line with the theory, see Section 6.

Outline

In Section 2 we recapitulate some theory of optimal stopping in discrete time and recall the (classical) Tsitsiklis–van Roy and Longstaff–Schwartz algorithms. In Section 3 we describe in detail the one-step regression procedures involved for both *standard* and *pseudo regression* and provide a convergence proof for the latter case. In Section 4 we give a proof of convergence for the full Longstaff–Schwartz algorithm based on pseudo regression. We discuss the computational cost for the different variants of the algorithms in Section 5 and give numerical examples in Section 6. We conclude with a summary and an outline of future research in Section 7.

2 Recap of optimal stopping in discrete time

2.1 Theory of optimal stopping in discrete time

Let us recall some facts about the optimal stopping problem in discrete time. Suppose $(Z_j : j = 0, 1, \dots, \mathcal{J})$ is a nonnegative adapted stochastic process in discrete time on a filtered probability space $(\Omega, \mathcal{F}_j, 0 \le j \le \mathcal{J}, P)$, which satisfies

$$\sum_{j=1}^{\mathcal{J}} \mathbb{E}\left[Z_j\right] < \infty.$$

In the context of a (discrete time) American or Bermudan option Z may be regarded as a (discounted) cash-flow process that may be exercised once by the option holder. More specifically, one may think of P as a pricing measure corresponding to some numéraire $\mathcal{N}$ (with $\mathcal{N}_0 = 1$ for simplicity), and $Z = R/\mathcal{N}$, where $(R_j : j = 0, 1, \dots, \mathcal{J})$ is a real (not discounted) cash-flow process. Then, from general no arbitrage principles it is well known that a fair price of the American option is given by

$$Y_0 := \sup_{\tau \in \mathcal{S}_0} \mathbb{E}\left[Z_\tau\right], \tag{3}$$

where $\mathcal{S}_0$ denotes the set of $\mathcal{F}$-stopping times taking values in $\{0, \dots, \mathcal{J}\}$. The *Snell envelope* of Z is defined as

$$Y_j := \operatorname{ess\,sup}_{\tau \in \mathcal{S}_j} \mathbb{E}_{\mathcal{F}_j}\left[Z_\tau\right], \quad j = 0, ..., \mathcal{J}, \tag{4}$$

where $\mathcal{S}_j$ denotes the set of $\mathcal{F}$-stopping times taking values in $\{j, \dots, \mathcal{J}\}$. We recall the following classical facts (e.g. see [Nev75]):

1 The Snell envelope Y of Z is the smallest super-martingale that dominates Z. It can be constructed recursively by the *Backward Dynamic Program principle* or *Bellman principle*:

$$\begin{aligned} Y_{\mathcal{J}} &= Z_{\mathcal{J}} \\ Y_j &= \max\left(Z_j, \mathbb{E}_{\mathcal{F}_j}\left[Y_{j+1}\right]\right), \quad 0 \le j < \mathcal{J}. \end{aligned} \tag{5}$$

2 An optimal stopping time for (3) is given by

$$\tau^* = \min\{j : \ 0 \leq j \leq \mathcal{J}, \ Z_j \geq \mathbb{E}_{\mathcal{F}_j}[Y_{j+1}]]\}$$

with $Y_{\mathcal{J}+1} := 0$. That is,

$$Y_0 = \sup_{\tau \in \mathcal{S}_0} \mathbb{E}[Z_\tau] = \mathbb{E}[Z_{\tau^*}].$$

Thus, in principle, one may arrive at the solution to (3) by carrying out (5) backwardly from $j = \mathcal{J}$ down to $j = 0$. However, straightforwardly, this leads to a high degree nested expression of conditional expectations that is virtually impossible to evaluate in practice.

Let us now assume the presence of an underlying Markovian process $X := X^{0,x} := (X_j^{0,x} : j = 0, 1, \ldots, \mathcal{J})$, adapted to $(\mathcal{F}_j)$, living in $\mathbb{R}^d$, and starting at $X_0^{0,x} = x$ a.s. More generally, $(X_r^{j,z} : r = j, ..., \mathcal{J})$ denotes a random trajectory with $X_j^{j,z} = z$ a.s. Let us further assume that the cash-flow has the form

$$Z_j(\omega) = f_j(X_j(\omega)), \ \ 0 \leq j \leq \mathcal{J}$$

for some functions $f_j(\cdot) : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$. Then, due to Markovianity, there exist functions $v_j(\cdot) : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$, such that we may similarly write

$$Y_j(\omega) = v_j(X_j(\omega)), \ \ 0 \leq j \leq \mathcal{J}.$$

The Bellman principle now simply says that

$$v_j(X_j) = \max\left(f_j(X_j), \mathbb{E}[v_{j+1}(X_{j+1})|X_j]\right), \qquad j < \mathcal{J}.$$

Henceforth

$$c_j(x) \coloneqq \mathbb{E}[v_{j+1}(X_{j+1})|X_j = x]$$

is called the *continuation value function*. The numerically challenging task is, of course, the computation of the c_j for $0 \leq j < \mathcal{J}$.

2.2 Standard regression algorithms

For clarity, let us describe the classical Tsitsiklis–van Roy algorithm in full detail. Let $(X_0^{(m)}, \ldots, X_{\mathcal{J}}^{(m)})$, $m = 1, \ldots, M$, denote M independent trajectories from the Markov process X. Initialize $\widehat{v}_{\mathcal{J}} \coloneqq f_{\mathcal{J}}$, $\widehat{c}_{\mathcal{J}} \coloneqq 0$. If $\widehat{v}_j$ and $\widehat{c}_j$ are already constructed, iteratively construct (backward in time)

$$\widehat{\beta}^{(j-1)} \coloneqq \operatorname*{arg\,min}_{\beta \in \mathbb{R}^K} \sum_{m=1}^{M} \left(\widehat{v}_j(X_j^{(m)}) - \sum_{k=1}^{K} \beta_k \psi_k(X_{j-1}^{(m)}) \right)^2, \tag{6}$$

$$\widehat{c}_{j-1}(\cdot) \coloneqq \sum_{k=1}^{K} \widehat{\beta}_k \psi_k(\cdot), \quad \widehat{v}_{j-1}(\cdot) \coloneqq \max(f_{j-1}(\cdot), \widehat{c}_{j-1}(\cdot)). \tag{7}$$

After this construction, we can either simply return the approximate value $\widehat{v}_0(X_0)$, or refine the estimate by simulating the expected pay-off due to the nearly optimal stopping time,

$$\widehat{\tau} = \min\left\{j : 0 \leq j \leq \mathcal{J}, \quad f_j(X_j) > \widehat{c}_j(X_j)\right\},$$

using newly generated independent samples from the process X.

The Longstaff–Schwartz algorithm is defined similarly, except that the regression step (6) does not use the previously constructed value function $\widehat{v}_j$, but rather the nearly optimal stopping time induced by $\widehat{c}_j, \dots, \widehat{c}_{\mathcal{J}}$. More precisely, the Longstaff–Schwartz algorithm goes as follows: Initialize for $m = 1, ..., M$, $\tau_{\mathcal{J}}^{(m)} := \mathcal{J}$, $\widehat{c}_{\mathcal{J}} := 0$. If the $\tau_j^{(m)}$ and $\widehat{c}_j$ are already constructed, iteratively construct (backward in time)

$$\widehat{\beta}^{(j-1)} := \operatorname*{arg\,min}_{\beta \in \mathbb{R}^K} \sum_{m=1}^{M} \left(f_{\tau_j^{(m)}}(X^{(m)}_{\tau_j^{(m)}}) - \sum_{k=1}^{K} \beta_k \psi_k(X^{(m)}_{j-1}) \right)^2, \tag{8}$$

$$\widehat{c}_{j-1}(\cdot) := \sum_{k=1}^{K} \widehat{\beta}_k \psi_k(\cdot), \tag{9}$$

$$\text{If } f_{j-1}(X^{(m)}_{j-1}) > \widehat{c}_{j-1}(X^{(m)}_{j-1}) \quad \text{then } \tau^{(m)}_{j-1} = j-1 \text{ else } \tau^{(m)}_{j-1} = \tau^{(m)}_j. \tag{10}$$

In both algorithms, the regression step itself only relies on two random variables, which we might as well denote by $(X, Y) \in \mathbb{R}^d \times \mathbb{R}$, living on some probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Consider the problem of estimating the function $u : \mathbb{R}^d \to \mathbb{R}$, satisfying

$$u(X) = \mathbb{E}\left[Y|X\right]. \tag{11}$$

As indicated, we solve the least squares minimization problem

$$\widehat{\beta} := \operatorname*{arg\,inf}_{\beta \in \mathbb{R}^K} \sum_{m=1}^{M} \left(Y^{(m)} - \sum_{k=1}^{K} \beta_k \psi_k\left(X^{(m)}\right) \right)^2, \tag{12}$$

and consider the estimation

$$\widehat{u}\left(x\right) = \sum_{k=1}^{K} \widehat{\beta}_k \psi_k\left(x\right). \tag{13}$$

It is well-known that by defining the design matrix $\mathcal{N} \in \mathbb{R}^{M \times K}$ by

$$\mathcal{N}_{mk} := \psi_k\left(X^{(m)}\right), \quad m = 1, ..., M, \ k = 1, ..., K,$$

and the vector $\mathcal{Y} \in \mathbb{R}^M$ by

$$\mathcal{Y}_m = Y^{(m)}, \quad m = 1, ..., M,$$

that the solution to (12) may be written as

$$\widehat{\beta} = \frac{1}{M} \left(\frac{1}{M} \mathcal{N}^\top \mathcal{N} \right)^{-1} \mathcal{N}^\top \mathcal{Y}, \tag{14}$$

provided that $\mathcal{N}$ has full rank K. The latter is typically almost surely the case when $K \leq M$. Note that,

$$\begin{aligned} \frac{1}{M} \left[\mathcal{N}^\top \mathcal{N}\right]_{k,l=1,...,K} &= \frac{1}{M} \sum_{m=1}^{M} \psi_k\left(X^{(m)}\right) \psi_l\left(X^{(m)}\right) \\ &\approx \mathbb{E}\left[\psi_k\left(X\right) \psi_l\left(X\right)\right]. \end{aligned} \tag{15}$$

In general, the marginal distribution of X is not explicitly known and the inversion of the matrix $\frac{1}{M}\mathcal{N}^\top \mathcal{N}$ in (14) is a main delicate issue since it has random nonnegative eigenvalues that can be arbitrary close to zero by chance. Furthermore, the computation of $\mathcal{N}$ requires about KM function calls and the computation of (14) requires about $K^2 M$ elementary operations.

3 The pseudo regression approach

As an alternative to the well-known methods in [LS01] and [TVR01] we now propose a backward algorithm for approximating the continuation functions c_j (respectively v_j) by functions $\overline{c}_j$, (respectively $\overline{v}_j$,) $j = \mathcal{J}, ..., 0$, in the present setup that is based on *pseudo regression*. Let us assume that we have chosen a set of basis functions $\psi_k : \mathbb{R}^d \to \mathbb{R}$, $k = 1, ..., K$, and a measure μ concentrated on $\mathcal{D} \subset \mathbb{R}^d$, such that the Gram matrix $\mathcal{G}$ defined by

$$\mathcal{G}_{kl} := \langle \psi_k, \psi_l \rangle := \int \psi_k(z)\psi_l(z)\mu(dz) \tag{16}$$

together with its inverse $\mathcal{G}^{-1}$ is explicitly known, or can be efficiently computed. In the algorithm spelled out below we construct a set of approximative continuation functions $\overline{c}_j$, $j = \mathcal{J}, ..., 0$, which satisfy

$$\overline{c}_j(z) \approx c_j(z) := \mathbb{E}\left[v_{j+1}(X_{j+1}^{j,z})\right] = \mathbb{E}\left[v_{j+1}(X_{j+1}^{0,\cdot}) \,|\, X_j^{0,\cdot} = z\right].$$

Moreover, it is assumed (for simplicity) that we are able to sample trajectories $X_\cdot^{0,x}$ exactly.

The probability measure μ is used to measure the regression error, cf. (2), i.e., we try to minimize the difference between c_j and $\overline{c}_j$ in the sense of the $L^2(\mu)$-norm. From that perspective, a natural choice of μ as induced by the problem at hand would be the distribution of X_{J+1}, but that choice runs afoul of the requirement that the Gram matrix is known explicitly. We shall see in Section 6 that the problem is not very sensitive to the choice of μ, such that we can often even choose a uniform (and simple) reference measure μ for all j without significant sacrifice in overall accuracy.

3.1 Pseudo regression variant of Tsitsiklis–van Roy

We start with $\overline{v}_\mathcal{J} = v_\mathcal{J} = f_\mathcal{J}$ and $\overline{c}_\mathcal{J} = c_\mathcal{J} = 0$. The backward iteration step $j \to j-1$ works as follows: First generate M i.i.d. copies $\mathcal{U}^{(m)}$, $m = 1, \ldots, M$ with $\mathcal{U}^{(1)} \sim \mu$. Simulate for $m = 1, \ldots, M$, the r.v. $X_j^{j-1,\mathcal{U}^{(m)}}$, and consider the $M \times K$ matrix $\mathcal{M}^{(j)}$ defined by

$$\mathcal{M}_{mk}^{(j)} := \psi_k\left(\mathcal{U}^{(m)}\right). \tag{17}$$

Define the vector $\mathcal{Y}^{(j)} \in \mathbb{R}^M$ by

$$\mathcal{Y}_m^{(j)} := \overline{v}_j\left(X_j^{j-1,\mathcal{U}^{(m)}}\right). \tag{18}$$

Following (2), the coefficients of the basis functions are given by

$$\overline{\beta}^{(j)} := \frac{1}{M}\mathcal{G}^{-1}\left(\mathcal{M}^{(j)}\right)^\top \mathcal{Y}^{(j)} \tag{19}$$

and then we obtain the approximate continuation value and solution, respectively, by

$$\overline{c}_{j-1}(z) := \sum_{k=1}^{K} \overline{\beta}_k^{(j)} \psi_k(z) \quad \text{and} \quad \overline{v}_{j-1}(z) := \max\left(f_{j-1}(z), \overline{c}_{j-1}(z)\right). \tag{20}$$

A pseudo-code representation of the algorithm is given in Algorithm 1.

The pseudo regression algorithm for Bermudan options is related to the well-known Tsitsiklis–van Roy algorithm (see [TVR01]), but differs essentially because of the pseudo regression step (19). In contrast, Tsitsiklis–van Roy compute the coefficients (19) by using standard global regression. Another striking difference is that in Algorithm 1 the basis functions ψ_k have to be evaluated much less times, since only one sample of M drawings from the distribution μ serves for all exercise dates. The merits of standard versus pseudo regression in a general setting are explained and discussed in detail in Section 4 below.

Data: $\mu, M, \psi_1, \ldots, \psi_K, \mathcal{G}, f_0, \ldots, f_{\mathcal{J}}$.
Result: Value function $\overline{v}_j$ and continuation value $\overline{c}_j$, $j = 0, \ldots, \mathcal{J}$.
begin
  $\overline{v}_{\mathcal{J}} \longleftarrow v_{\mathcal{J}} = f_{\mathcal{J}}$
  $\overline{c}_{\mathcal{J}} \longleftarrow c_{\mathcal{J}} = 0$
  for $m \longleftarrow 1$ **to** M **do**
    Generate $\mathcal{U}^{(m)} \sim \mu$
  end
  $\mathcal{M} \longleftarrow \left(\psi_k(\mathcal{U}^{(m)})\right)_{\substack{m=1,\ldots,M \\ k=1,\ldots,K}} \in \mathbb{R}^{M \times K}$
  for $j \longleftarrow \mathcal{J}$ **to** 1 **do**
    for $m \longleftarrow 1$ **to** M **do**
      Generate $X_j^{j-1,\mathcal{U}^{(m)}}$
      `// These r.v. are understood to be independent conditional` $\mathcal{U}^{(m)}$
    end
    $\mathcal{Y}^{(j)} \longleftarrow \left(\overline{v}_j\left(X_j^{j-1,\mathcal{U}^{(m)}}\right)\right)_{m=1,\ldots,M} \in \mathbb{R}^M$
    $\overline{\beta}^{(j)} \longleftarrow \frac{1}{M}\mathcal{G}^{-1}\mathcal{M}^\top \mathcal{Y}^{(j)}$
    $\overline{c}_{j-1}(\cdot) \longleftarrow \sum_{k=1}^{K} \overline{\beta}_k^{(j)} \psi_k(\cdot)$
    $\overline{v}_{j-1}(\cdot) \longleftarrow \max\left(f_{j-1}(\cdot),\, \overline{c}_{j-1}(\cdot)\right)$
  end
end

Algorithm 1: Pseudo regression variant of TV for Bermudan options

3.2 Pseudo regression variant of Longstaff–Schwartz

In order to obtain a pseudo regression variant of the Longstaff–Schwartz algorithm we modify the backward construction of the approximative continuation functions $\overline{c}_j$, $j = \mathcal{J}, ..., 0$ (initialized with $\overline{c}_{\mathcal{J}} = 0$ again) in the following way. Let us assume that $\overline{c}_j, ..., \overline{c}_{\mathcal{J}}$ are constructed. Simulate for $m = 1, ..., M$ at time $j-1$ the trajectory

$$X_r^{j-1,\mathcal{U}^{(m)}}, \quad r = j, ..., \mathcal{J}, \tag{21}$$

and modify (18) to

$$\begin{aligned}\mathcal{Y}_m^{(j)} :=& f_\tau(X_\tau^{j-1,\mathcal{U}^{(m)}}), \quad \text{where} \\ & \tau \equiv \min\left\{r : r \geq j,\ f_r(X_r^{j-1,\mathcal{U}^{(m)}}) \geq \overline{c}_r\left(X_r^{j-1,\mathcal{U}^{(m)}}\right)\right\}.\end{aligned}$$

Then compute (19) and set $\overline{c}_{j-1}(z)$ according to (20). The corresponding modification of Algorithm 1 is obvious.

At the first glance this procedure is significantly more costly. However, if the chain X is autonomous, which we may assume w.l.o.g. in fact, we simulate first

$$X_r^{0,\mathcal{U}^{(m)}}, \quad r = 0, ..., \mathcal{J},$$

and then take in (21)

$$X_r^{j-1,\mathcal{U}^{(m)}} = X_{r-j+1}^{0,\mathcal{U}^{(m)}}, \quad r = j, ..., \mathcal{J}. \tag{22}$$

So, for the autonomous case, one set of full trajectories, just as in the standard LS algorithm, is sufficient for this algorithm as well.

4 Accuracy analysis of pseudo regression

In the next section we analyze an alternative and potentially more efficient pseudo regression procedure for computing $\mathbb{E}[Y|X]$, i.e. (11), given that we may sample Y from its conditional distribution given X (although we generally do not know $\mathbb{E}[Y|X]$ explicitly of course).

4.1 A general framework

Suppose that in (11) it is possible to sample Y from its conditional distribution given X, say $\nu(dy|X)$. A canonical example is the setup in Section 3 where

$$X = X_j^{0,x} \quad \text{and} \quad Y = g\left(X_{j+1}^{0,x}\right) = g\left(X_{j+1}^{j,X_j^{0,x}}\right),$$

for some arbitrary x. Let us consider a random variable $\mathcal{U}$ with values in some domain $\mathcal{D} \subset \mathbb{R}^d$, distributed according to some probability measure $\mu(dz)$ concentrated on $\mathcal{D}$. We then generate i.i.d. copies $\mathcal{U}^{(m)}$, $m = 1, ..., M$ of $\mathcal{U}$, and sample for each $m = 1, ..., M$, independently $Y^{(m)}$ from $\nu\left(dy|\mathcal{U}^{(m)}\right)$. Then define the vector $\mathcal{Y} \in \mathbb{R}^M$ as

$$\mathcal{Y} := \left[Y^{(1)}, ..., Y^{(M)}\right]^\top.$$

Now for a linearly independent system $(\psi_k : k = 1, 2, ...)$, with

$$\int \psi_k^2(z)\mu(dz) < \infty,$$

consider the $M \times K$ matrix

$$\mathcal{M}_{mk} := \psi_k\left(\mathcal{U}^{(m)}\right).$$

Assuming that we know explicitly the matrix $\mathcal{G}$ defined by the scalar products $\mathcal{G}_{kl} := \langle \psi_k, \psi_l \rangle$ (cf. (16)), we now compute the pseudo regression coefficients

$$\overline{\beta} = \frac{1}{M}\mathcal{G}^{-1}\mathcal{M}^\top \mathcal{Y}, \tag{23}$$

and consider the pseudo regression approximation

$$\overline{u}(z) = \sum_{k=1}^{K} \overline{\beta}_k \psi_k(z) \approx \mathbb{E}\left[Y \,|\, \mathcal{U} = z\right], \quad z \in \mathcal{D}. \tag{24}$$

Clearly, the difference with standard regression is that the random matrix $\frac{1}{M}\mathcal{N}^\top\mathcal{N}$ in (14) is replaced by $\mathcal{G}$ in view of (15). In general $\mathcal{G}^{-1}$ can be pre-computed outside the Monte Carlo simulation with arbitrary accuracy or is explicitly known due to a suitable choice of the system $(\psi_k : k = 1, 2, ...)$ and the measure μ. So the computation of (23) only involves KM elementary operations and no random matrix inversion is required. Moreover, naturally, we may assume w.l.o.g. that the system $(\psi_k : k = 1, 2, ...)$ is an orthonormal system with respect to $L_2(\mathcal{D}, \mu)$ and then (23) simplifies to

$$\overline{\beta} = \frac{1}{M}\mathcal{N}^\top \mathcal{Y}.$$

4.2 Accuracy analysis of the regression

For the convergence properties of the pseudo-regression method we could basically refer to [ABE+17, BBR+18], where pseudo regression is applied in the context of global solutions for random PDEs. For the convenience of the reader, however, let us here recap the analysis in condensed form, consistent with the present terminology and a somewhat less involved setup.

Theorem 4.1. *(Accuracy pseudo regression) Suppose that in (11)*

$$\begin{gathered} |u(z)| \le D \quad \textit{and} \quad \mathrm{Var}\left[Y \,|\, X = z\right] < \sigma^2, \quad \textit{for all } z \in \mathcal{D}, \\ 0 < \underline{\lambda_{\min}} \le \lambda_{\min}\left(\mathcal{G}^K\right) \le \lambda_{\max}\left(\mathcal{G}^K\right) \le \overline{\lambda_{\max}}, \quad \textit{for all } K = 1, 2, ..., \end{gathered}$$

where $\lambda_{\min}\left(\mathcal{G}^K\right)$, and $\lambda_{\max}\left(\mathcal{G}^K\right)$, denote the smallest, respectively largest, eigenvalue of the positive symmetric matrix $\mathcal{G}$. Then it holds,

$$\begin{aligned} &\mathbb{E}\int_{\mathcal{D}} |\overline{u}(z) - u(z)|^2 \mu(dz) \\ &\le \frac{\overline{\lambda_{\max}}}{\underline{\lambda_{\min}}}\left(\sigma^2 + D^2\right)\frac{K}{M} + \inf_{w \in \mathrm{span}\{\psi_1,...,\psi_K\}} \int_{\mathcal{D}} |w(z) - u(z)|^2 \mu(dz). \end{aligned} \tag{25}$$

The proof of Theorem 4.1 is provided in Appendix A.1.

It is interesting to compare Theorem 4.1 with a corresponding theorem that holds for the standard regression estimate (13):

Theorem 4.2. *(Accuracy standard regression) Suppose that,*

$$|u(x)| \leq D \quad \textit{and} \quad \operatorname{Var}\left[Y \,|\, X = x\right] < \sigma^2, \quad \textit{for all } x \in \mathbb{R}^d,$$

then for

$$\widetilde{u}_D(x) = \begin{cases} \widetilde{u}(x) & \textit{if} \quad |\widetilde{u}(x)| \leq D \\ D & \textit{if} \quad \widetilde{u}(x) > D \\ -D & \textit{if} \quad \widetilde{u}(x) < -D \end{cases}$$

and some universal constant $c > 0$, it holds that

$$\begin{aligned} &\mathbb{E} \int |\widetilde{u}_D(x) - u(x)|^2 \, \mu_X(dx) \\ &\leq c \max\left(\sigma^2, D^2\right) \frac{(1 + \ln M)\, K}{M} + 8 \inf_{w \,\in\, \operatorname{span}\{\psi_1, \ldots, \psi_K\}} \int_{\mathcal{D}} |w(x) - u(x)|^2 \, \mu_X(dx), \end{aligned} \tag{26}$$

where μ_X denotes the distribution of X in (11).

The proof of Theorem 4.2 is much more complicated than the proof of Theorem 4.1 and relies heavily on uniform laws of large numbers from the theory of empirical processes. For details see [GKKW02].

4.3 Convergence of the pseudo Longstaff–Schwartz algorithm

In this section we investigate the convergence properties of the pseudo Longstaff–Schwartz algorithm in the spirit of [BS18a] and [Zan13]. More specifically, we adapt the proof in the paper [BS18a] on optimal stopping in the context of interacting particle systems. The proof for the Tsitsiklis–van Roy algorithm is very similar and, hence, omitted.

Let us consider the algorithm based on (21), that is, for every exercise date the sample (21) is simulated independently, and consider the information set

$$\mathcal{G}_j := \sigma\left\{\mathbf{X}^{j;M}, \ldots, \mathbf{X}^{\mathcal{J}-1;M}\right\} \text{ with } \mathbf{X}^{j;M} := \left(X_r^{j,\mathcal{U}^{(m)},m}, \quad r = j, ..., \mathcal{J},\ m = 1, ..., M\right).$$

Let us define for a generic dummy trajectory $(X_l)_{l=0,\ldots,\mathcal{J}}$ corresponding to the (exact) solution independent of $\mathcal{G}_j$,

$$\widetilde{c}_j(x) := \mathbb{E}_{\mathcal{G}_{j+1}}\left[f_{\overline{\tau}_{j+1}}\left(X_{\overline{\tau}_{j+1}}\right) \middle|\, X_j = x\right], \tag{27}$$

where $\overline{\tau}_{\mathcal{J}} = \mathcal{J}$, and

$$\overline{\tau}_j := j\, 1_{\left\{f_j(X_j) \geq \overline{c}_j(X_j)\right\}} + \overline{\tau}_{j+1} 1_{\left\{f_j(X_j) < \overline{c}_j(X_j)\right\}}.$$

It is important to note that, in (27), $\widetilde{c}_j(\cdot)$ is a $\mathcal{G}_{j+1}$-measurable random function while the estimation $\overline{c}_j(\cdot)$ is a $\mathcal{G}_j$-measurable one as the construction of $\overline{c}_j$ also depends on $\mathbf{X}^{j;M}$, see (20). After proceeding backwardly from $j = \mathcal{J}$ down to $j = 1$, we thus have a sequence of approximative continuation functions $\overline{c}_j(\cdot)$, and a sequence of corresponding conditional expectations $\widetilde{c}_j(\cdot)$. The convergence analysis is based on the following lemma (cf. Lemma 5 in [BS18a]).

Lemma 4.3. *For the conditional expectations (27) we have that,*

$$\left\|\widetilde{c}_j - c_j\right\|_{L_p(\mu)} \leq \sum_{l=j+1}^{\mathcal{J}-1} \left\|\overline{c}_l - c_l\right\|_{L_p(\mu_{j,l})} \tag{28}$$

with $p \geq 1$, $\mu_{j,l}$ being the distribution of $X_l^{j,\mathcal{U}}$, $1 \leq j \leq l \leq \mathcal{J}, \mathcal{U} \sim \mu$, and c_j, being the true continuation functions.

The proof is almost identical with the proof of the similar Lemma 5 in [BS18a]. For the convenience of the reader, it is given in Appendix A.2.

Remark 4.4. *Note that the inequality (28) involves $\mathcal{G}_{j+1}$-measurable objects. It is also interesting to compare (28) with similar (though different) inequalities in [Zan13].*

We now state our convergence theorem connected with the pseudo Longstaff–Schwartz algorithm (cf. Theorem 7 in [BS18a]).

Theorem 4.5. *Assume that the conditions of Theorem 4.1 are fulfilled. In particular assume that the cash-flows f_j are uniformly bounded in $j = 1, \ldots, \mathcal{J}$. Let us define, for a generic measure ν, the norm*

$$\|\cdot\|^2_{L_2(\nu\otimes\mathbb{P})} := \mathbb{E}\left[\|\cdot\|^2_{L_2(\nu)}\right],$$

due to the unconditional expectation with respect to the "all in" probability measure $\mathbb{P}$. One then has for $j = 1, \ldots, \mathcal{J}-1$ and $X_l^{j,\mathcal{U}} \sim \mu_{j,l}$, $j \leq l \leq \mathcal{J}$,

$$\left\|\overline{c}_j - c_j\right\|_{L_2(\mu\otimes\mathbb{P})} \leq \varepsilon_{j,M,K}\eta(2+\eta)^{\mathcal{J}-j-1}, \quad \textit{with} \tag{29}$$

$$\varepsilon_{j,M,K} = \sqrt{\frac{K}{M}} + \max_{j\leq l<\mathcal{J}} \inf_{w \,\in\, \textit{span}\{\psi_1,\ldots,\psi_K\}} \left\|c_l - w\right\|_{L_2(\mu_{j,l})},$$

for some constant $\eta > 0$ not depending on K and M.

The proof follows similar lines as the one in [BS18a] (although there are some differences due to the different setup) and is given in the present context and terminology in Appendix A.3.

5 Computational cost

We will now discuss the advantages and disadvantages of the two different approaches for various use-cases, both in the context of the Tsitsiklis–van Roy algorithm and the Longstaff–Schwartz algorithm. The main issue is, of course, the relation between computational work and accuracy. Comparing Theorems 4.1 and 4.2, we see that the error as function of the number of basis functions K, the choice of basis functions $\psi_1, \ldots, \psi_K$ and the number of samples M is roughly equivalent for both methods.

Remark 5.1. *We ignore the different constants as well as the additional $\ln M$ term in Theorem 4.2. In practice, different constants may, of course, have drastic effects ion run-time, which is why the numerical experiments presented in Section 6 are crucial. A more subtle difference is related to the choice of the measure with respect to which the error is calculated. Also we note that we only focus on the cost of computing the functions c_j, respectively v_j, as the other aspects of the computation have negligible cost, independent of the chosen regression method.*

Let us recall our general setting: we are given a cash-flow process $Z_j = f_j(X_j)$, $j = 0, \dots, \mathcal{J}$, which is based on an $\mathbb{R}^d$-valued Markov process X_j, $j = 0, \dots, \mathcal{J}$, and we would like to compute the corresponding Bermudan option price. In the following, we need to make certain assumptions on the simulation.

Assumption 5.2. *We can exactly simulate the Markov process X. More precisely, given a sample of X_j, we can simulate a sample of X_{j+1}, $j = 0, \dots, \mathcal{J} - 1$, exactly at cost normalized to one.*

Assumption 5.2 seems to restrict us to simple models such as Black-Scholes or Bachelier, for which exact simulation is easily possible, but note that any discretization error would be expected to effect both regression algorithms in the same way, both with respect to accuracy and with respect to computational cost. Therefore, we think that Assumption 5.2 is justified.

Remark 5.3 (Cost model)**.** *In the discussions of computational cost, all estimates shall be understood as counting the number of* function evaluations*. More precisely, each of the following operations incurs one unit cost:*

- *Generating one sample of X_{j+1} conditional on X_j;*
- *Evaluating a basis function ψ_k at one point x;*
- *An elementary floating point calculation such as a product between two floating point numbers.*

Of course, these operations incur very different computational costs in practice. However, note that it is very difficult to realistically bound true computational times any way. These may heavily depend on hardware features (e.g., cache misses), and, in particular, on the implementation details.

5.1 Tsitsiklis–van Roy algorithm

With Assumption 5.2, we can already describe the computational work of the standard regression algorithm.

Proposition 5.4 (Computational cost of standard regression)**.** *The computational cost of the standard regression satisfies*

$$\mathcal{C}_{reg} = \mathcal{O}\left(\mathcal{J}(MK^2 + K^3)\right).$$

Proof. This result is, of course, very well known. The dominating terms for the computational cost the computation of the random matrix $\mathcal{N}^\top \mathcal{N}$ and the computation of the coefficient $\widetilde{\beta}$ by, e.g., LU or Cholesky decomposition. Both operations have to be recomputed for each exercise time $j = 0, \dots, \mathcal{J} - 1$. □

For the pseudo regression approach we will operate under

Assumption 5.5. *The basis functions $\psi_1, \dots, \psi_K$ are chosen such that the matrix $\mathcal{G}$ is given explicitly.*

The assumption is most easily satisfied by choosing the basis function to be orthonormal polynomials w.r.t. μ. Then we obtain

Proposition 5.6 (Computational cost of pseudo regression)**.** *The computational cost of pseudo regression under Assumptions 5.2 and 5.5 satisfies*

$$\mathcal{C}_{\textit{pseudo}} = \mathcal{O}\left(\mathcal{J}MK + \mathcal{J}K^2 + K^3\right).$$

If, in addition, the basis functions are orthonormal w.r.t. μ, then the cost instead satisfies

$$\mathcal{C}_{\textit{pseudo}} = \mathcal{O}\left(\mathcal{J}MK\right).$$

Proof. First we need to compute the LU decomposition of the matrix $\mathcal{G}$, at cost proportional to K^3—independent of $\mathcal{J}$. We also need to simulate the random variables $\mathcal{U}$ and set up the matrix $\mathcal{M}$ at cost $\mathcal{O}(MK)$. In each iteration of the algorithm, we then need to simulate the vector $\mathcal{Y}$ and multiply $\mathcal{M}^\top\mathcal{Y}$ at cost proportional to MK. Finally, assembling the solution of the linear system $\mathcal{G}\overline{\beta} = \frac{1}{M}\mathcal{M}^\top\mathcal{Y}$ incurs costs proportional to K^2.

In the orthonormal case, we have $\mathcal{G} = \mathcal{G}^{-1} = \mathrm{Id}_K$, and the cost of setting up $\mathcal{M}$ and multiplying $\mathcal{M}^\top\mathcal{Y}$ becomes dominant. □

In practice, even better cost savings are possible under

Assumption 5.7. *The Markov process X is homogeneous in time, i.e., the conditional distribution of X_{j+1} given X_j does not depend on j.*

This condition is very often satisfied in financial models, and it has drastic implications for the pseudo regression algorithm (but not for the standard regression). Indeed, since the conditional distribution does not depend on j, and we always re-sample the starting points (at step j) from the same distribution μ—instead of the distribution of X_j—, we can simply use the *same* samples for setting up $\mathcal{Y}$ for each time-step in (23). Formally, the asymptotic cost does not change compared to Proposition 5.6, but in practice we do observe major effects due to decreasing constants.

Remark 5.8. *It is well-understood in practice that it is generally beneficial to add the payoff function itself to the set of basis functions. This may cause problems for the pseudo regression, as the inner products of the payoff function with the other (typically polynomial) basis functions cannot be expected to be given in closed form, thereby violating Assumption 5.5. However, we can compute those scalar products numerically, by quadrature, quasi Monte Carlo or even standard Monte Carlo, at negligible extra cost, especially in the setting of Assumption 5.7. With some additional work, we can still achieve orthonormality by Gram-Schmidt.*

Let us summarize the findings of this section by looking at the most typical case. Arguably, this is the case when $M \gg K, \mathcal{J}$. We may always choose basis functions to be orthonormal, hence we consider the second case in Proposition 5.6. For standard regression, the computational costs are, hence, asymptotically proportional to $\mathcal{J}MK^2$, whereas the pseudo regression only incurs costs proportional to $\mathcal{J}MK$. This will lead to a computational advantage, especially when K is large.

5.2 Longstaff–Schwartz algorithm

Asymptotically, the Longstaff–Schwartz algorithm based on standard regression usually incurs the same cost as the Tsitsiklis–van Roy algorithm based on standard regression (Proposition 5.4).

Proposition 5.9. *The computational cost of the Longstaff–Schwartz algorithm due to (8)–(10), using standard regression, is*

$$\mathcal{C}_{reg} = \mathcal{O}\left(\mathcal{J}MK^2 + \mathcal{J}K^3\right).$$

Proof. First we simulate all trajectories at cost $\mathcal{O}(\mathcal{J}M)$ and evaluate the basis functions along all simulated values at cost $\mathcal{O}(\mathcal{J}MK)$. For each step j in the backward iteration we need to set up the matrix $\mathcal{N}^\top\mathcal{N}$ at (individual) cost $\mathcal{O}(MK^2)$. Then we need to compute the right hand side $\mathcal{Y}^{(j)}$ at cost $\mathcal{O}(\mathcal{J}M)$, which assumes that the values of the continuation function at times $j+1,\dots,\mathcal{J}$ have been pre-computed in the earlier stages of the backward iteration. Finally, compute the coefficients at cost $\mathcal{O}(MK^2 + K^3)$. □

If we apply the Longstaff–Schwartz algorithm with pseudo-regression we note an important difference compared to Tsitsiklis–van Roy: in the standard case of the algorithm (presented in Section 3.2) we potentially have to evaluate the basis functions $\psi_1,\dots,\psi_K$ for each sample $X_r^{j-1,\mathcal{U}^m}$, $r = j,\dots,\mathcal{J}$. In the worst case, this incurs costs proportional to $\mathcal{J}^2KM$ on top. Hence, we obtain

Proposition 5.10. *The computational cost of the Longstaff–Schwartz algorithm based on pseudo regression is*

$$\mathcal{C}_{pseudo} = \mathcal{O}\left(\mathcal{J}K^2 + K^3 + \mathcal{J}^2MK\right).$$

If the basis functions are orthonormal w.r.t. μ, then the costs reduce to

$$\mathcal{C}_{pseudo} = \mathcal{O}\left(\mathcal{J}^2MK\right).$$

Suppose that we are actually in the setting of Assumption 5.7. Then we may once again duplicate the samples. In this case, we still need to simulate full trajectories starting from the sampled initial points at $j = 0$, but we can then "shift" those samples in time. In this case, we only need to evaluate the basis functions at $X_r^{0,\mathcal{U}^m}$, $r = j,\dots,\mathcal{J}$, which incurs an additional cost $\mathcal{O}(\mathcal{J}MK)$. On the other hand, we will get a cost component $\mathcal{O}(\mathcal{J}^2M)$ simply from assembling $\mathcal{Y}^{(j)}$ for each j. In total, we obtain

Proposition 5.11. *If Assumption 5.7 holds and we duplicate samples, then the computational cost of the Longstaff–Schwartz algorithm based on pseudo regression is*

$$\mathcal{C}_{pseudo} = \mathcal{O}\left(\mathcal{J}K^2 + K^3 + \mathcal{J}MK + \mathcal{J}^2M\right).$$

If the basis functions are orthonormal w.r.t. μ, then the costs reduce to

$$\mathcal{C}_{pseudo} = \mathcal{O}\left(\mathcal{J}MK + \mathcal{J}^2M\right).$$

Let us, once again, summarize the discussion on the computational costs by looking at a typical case. For true Bermudan options, $\mathcal{J}$ is, of course, fixed, while M and K need to be increased in order to improve the accuracy of the estimator. Hence, the typical case for asymptotic considerations is probably $M \gg K \gg \mathcal{J}$. Once again, we may very well assume to have chosen orthonormal basis functions together with Assumption 5.7. Hence, regarding pseudo regression, we are in the second case of Proposition 5.11. Under these conditions, the computational cost of the Longstaff–Schwartz algorithm with standard regression is asymptotically proportional to $\mathcal{J}MK^2$, while the pseudo regression incurs cost asymptotically proportional to $\mathcal{J}MK$. Again, the costs of the standard regression dominate in the long run.

6 Numerical experiments

Unless otherwise stated, the numerical experiments below are run on a laptop computer with an Intel® Core™ i7-6500U processor and 8GB RAM. All algorithms are implemented and executed in GNU Octave version 4.0.3 running on openSUSE Leap 42.3. Moreover, all the codes are single-threaded.

We consider a Bermudan Max-Call option on n assets which, for instance, has already been considered in [AB04]. The assets X^i are identically distributed and yield dividends with rate δ. They are given as the solutions to

$$dX_t^i = (r-\delta)X_t^i dt + \sigma X_t^i dW_t^i, \quad X_0^i = x_0, \quad t \in [0,T], \quad i = 1,\dots,n, \tag{30}$$

where W^i are independent scalar Brownian motions. The interest rate r as well as σ are constant. We assume to have $\mathcal{J}+1$ exercise dates $0 \le t_0 < t_1, \dots < t_{\mathcal{J}} \le T$ in which the option holder may exercise to obtain the payoff

$$h(X_t) = \left(\max(X_t^1,\dots,X_t^n) - \kappa\right)^+, \tag{31}$$

where $\kappa > 0$. Moreover, we introduce the discounted payoff function by $f_t(X_t) = \mathrm{e}^{-rt}\, h(X_t)$.

Throughout the remainder of this section, it is assumed that $T = 3$, $r = 0.05$, $\delta = 0.1$, $\sigma = 0.2$ and $\kappa = 100$. We further choose $t_j = j\frac{T}{\mathcal{J}}$ for $j = 0,\dots,\mathcal{J}$.

6.1 Option pricing using Tsitsiklis–van Roy

We aim to determine an approximation $\bar{v}_0$ of the value Y_0 of the Bermudan Max-Call option above. To do so, we use the algorithm of Tsitsiklis–van Roy [TVR01], where the computation of the continuation functions $\bar{c}_j$ ($j = 0,\dots,\mathcal{J}-1$) is based on the standard regression (SR). On the other hand we use the algorithm explained in Section 3.1 (see also Algorithm 1), in which SR is replaced by a pseudo regression (PR) method.

In this section, we set $\mathcal{J} = 9$. We choose random initial values $\mathcal{U} \sim \mu$ within the PR for each component X^i given by (30). In fact, we set

$$\mathcal{U} = \mathrm{e}^{\mathrm{m}_n + \hat{\sigma}_n Z},$$

where $Z \sim \mathcal{N}(0,1)$, i.e., $\mathcal{U}$ is log-normal distributed. We choose different mean parameter m_n and a variance parameter $\hat{\sigma}_n$ depending on n. This is because we observe that we obtain somewhat better results if these parameter are slightly modified with the number of assets n.

We choose orthonormal polynomials $(\psi_k)_{k=1,\dots,K}$ with respect to μ. To be more precise, we introduce Hermite polynomials on $\mathbb{R}$ of degree i which we denote by H_i. We then define $\psi_1,\dots,\psi_K$ via a suitable ordering of the functions

$$\prod_{j=1}^{n} H_{i_j}\left(\frac{\ln(y_j) - \mathrm{m}_n}{\hat{\sigma}_n}\right)$$

with $i_1 + i_2 + \ldots + i_n \le p$, where $p \in \mathbb{N}$ is the largest polynomial degree and $i_j \in \mathbb{N}$ for all j. Thus, the total number of basis functions is $K = \frac{(p+n)!}{p!\,n!}$. In fact, we just pick all the products of

Hermite polynomials with total degree up to p, and use some bijection to assign them to an index $k = 1, \dots, K$. In the following, the above orthonormal functions are not just used for the PR but also within the SR ansatz.

We now determine $\bar{v}_0$ of the Bermudan option for the initial values $x_0 = 90, 100, 110$. Moreover, we conduct the numerical experiments for $n = 2, 3, 4$ assets and we set $p = 5$. We aim to achieve the same accuracy for both the PR and the SR based algorithm in order to be able to compare both schemes. This can be done by using a different number of samples for PR than for SR. However, it is hard to find these numbers of samples such that both algorithms yield exactly the same output $\bar{v}_0$. Therefore, for both methods, $M = 2\text{e}{+06}$ samples are used to derive the continuation functions, since we observe that in this case the PR and the SR based method lead to approximately the same output. Notice that the algorithm with a slightly lower number for $\bar{v}_0$ can always be improved by using more samples.

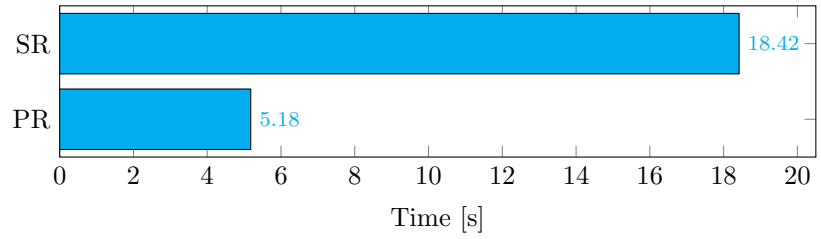
We start with the case $n = 2$. We observe that X_t^i has a different mean and a different variance over time t. Consequently, we choose m_2 and $\hat{\sigma}_2$ such that $\mathcal{U}$ approximately covers the important areas of the range of X_t^i regardless of the particular time $t \in [0, T]$ chosen. We see that both PR and SR perform almost equally well for the case $n = 2$, see first block of Table 1. It seems that PR even yields slightly better results for $\bar{v}_0$. The respective computational times can be found in Figure 1a. It turns out that the PR algorithm is more than three times faster.

It is possible to also take the same mean and variance parameter for $n > 2$. However, enlarging the variance slightly for $n = 3$ leads to results which are a little bit better. The mean parameter remains the same as for $n = 2$. From the second block in Table 1, it can be seen that both algorithms lead to approximately the same value $\bar{v}_0$. The advantage of using PR is the much lower computational time. We know from Figure 1b that we save a factor of more than five compared to SR.

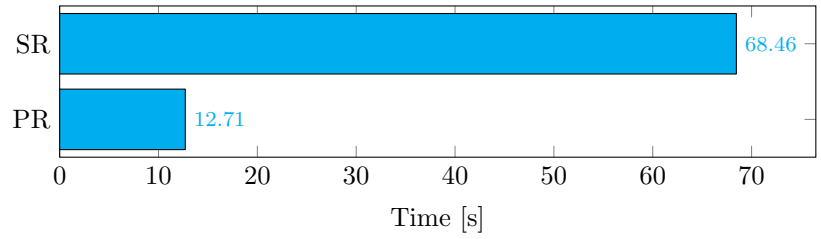
In the case of $n = 4$, the variance parameter is again enlarged compared to the case of having $n = 3$ assets. We modify the mean parameter as well, for details see Table 1. Again, SR and PR yield results of the same quality, see third block of Table 1, whereas the PR based algorithm is extremely fast compared to the SR. Figure 1c shows that a factor of more than nine can be saved.

x_0	$\bar{v}_0(x_0)$ based on PR	$\bar{v}_0(x_0)$ based on SR	Parameter used
$n = 2$			
90	8.046 (0.006)	8.030 (0.006)	$K = 21$, $\hat{\sigma}_n = 0.26$
100	13.884 (0.008)	13.868 (0.008)	and
110	21.322 (0.009)	21.314 (0.009)	$\mathrm{m}_n = \ln(x_0) - 0.105$
$n = 3$			
90	11.238 (0.007)	11.234 (0.007)	$K = 56$, $\hat{\sigma}_n = 0.29$
100	18.640 (0.009)	18.640 (0.009)	and
110	27.533 (0.010)	27.520 (0.010)	$\mathrm{m}_n = \ln(x_0) - 0.105$
$n = 4$			
90	14.045 (0.008)	14.049 (0.008)	$K = 126$, $\hat{\sigma}_n = 0.32$
100	22.638 (0.009)	22.638 (0.010)	and
110	32.527 (0.011)	32.531 (0.011)	$\mathrm{m}_n = \ln(x_0) - 0.179$

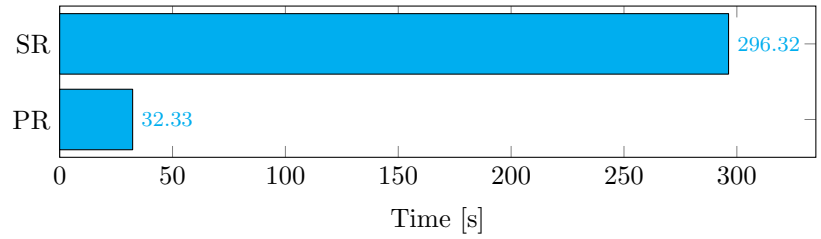
Table 1: Approximative value $\bar{v}_0$ of the Bermudan option based on SR and PR for $\mathcal{J} = 9$ using Tsitsiklis–van Roy. The computation of the continuation functions is based on $M = 2\text{e}{+06}$ samples.



(a) Computational times for $n = 2$.



(b) Computational times for $n = 3$.



(c) Computational times for $n = 4$.

Figure 1: Time to compute continuation functions using Tsitsiklis–van Roy and $M = 2\text{e}{+06}$ samples; standard versus pseudo regression for $n = 2, 3, 4$ and $\mathcal{J} = 9$.

We conclude this section by considering the case $n = 5$ separately. The reason is that in this case 8 GB RAM are not sufficient such that this experiment cannot be run on the laptop computer. Therefore, we use a computer with an HP BL460c Gen8 2xXeon Eight-Core processor with 128GB RAM. Again, the same version of Octave is used.

Notice that the experiments for $n = 2, 3, 4$ have a different relative computational time on this computer when comparing PR and SR. The relative computational gain of using PR is then 15 to 25 percent lower than the factors obtained in Figure 1. This indicates that the computational gain also depends on the architecture of the underlying computer.

Another reason to investigate this case separately is that we use a larger number of samples for the PR based algorithm in order to achieve the same value $\bar{v}_0$ for both methods. In fact, we use $M = 8\text{e}{+06}$ samples for PR and $M = 2\text{e}{+06}$ samples for SR. It can be seen from Table 2 that both algorithms yield the same value for $\bar{v}_0$. Although four times more samples are used for PR, the algorithm is still more than two times faster, see Figure 2. Notice that the absolute computational time on the 128GB RAM computer is much larger than the one on the 8GB RAM laptop computer. This explains the large values in Figure 2.

Remark 6.1. *Comparing the values $\bar{v}_0$ from the PR variant of the Tsitsiklis–van Roy algorithm in Tables 1 and 2 with the lower bounds in [AB04, Table 2], it can be seen that these values are very close to each other. To be more precise, we achieve $99.76\% - 99.94\%$ ($n = 2$), $99.63\% - 100.01\%$ ($n = 3$) and $99.71\% - 99.98\%$ ($n = 5$) of the values in [AB04] depending on the particular initial value. For*

x_0	$\bar{v}_0(x_0)$ based on PR	$\bar{v}_0(x_0)$ based on SR	Parameter used
$n = 5$			
90	16.569 (0.008)	16.570 (0.008)	$K = 252$, $\hat{\sigma}_n = 0.34$
100	26.090 (0.010)	26.088 (0.010)	and
110	36.718 (0.011)	36.715 (0.011)	$\mathrm{m}_n = \ln(x_0) - 0.21$

Table 2: Approximative value $\bar{v}_0$ of the Bermudan option based on SR and PR for $\mathcal{J} = 9$ and $n = 5$ using Tsitsiklis–van Roy. The computation of the continuation functions is based on $M = 2\mathrm{e}{+06}$ samples for SR and on $M = 8\mathrm{e}{+06}$ samples for PR.

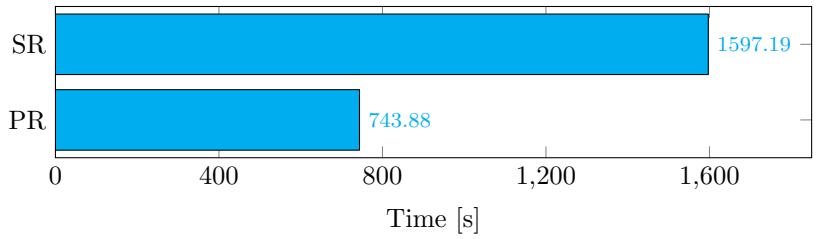


Figure 2: Time to compute continuation functions using Tsitsiklis–van Roy on a computer with 128GB RAM; standard ($M = 2\mathrm{e}{+06}$) versus pseudo ($M = 8\mathrm{e}{+06}$) regression for $n = 5$ and $\mathcal{J} = 9$.

the case $n = 4$ no reference value is known to us.

6.2 Option pricing using Longstaff–Schwartz

In this section, we study the same problem as in Section 6.1. However, we now determine the approximation $\bar{v}_0$ for the value of the Bermudan Max-Call option based on Longstaff–Schwartz. Again, we investigate a PR based version (see Section 3.2) and a SR based method (see [LS01]) for this approach in order to derive the continuation functions $\bar{c}_j$.

We observe from the simulations that PR is clearly faster than SR for Longstaff–Schwartz for $n > 2$ using the same parameter as in Section 6.1. Instead of presenting the same experiments for Longstaff–Schwartz again, we only point out one case in which we see a very large gain in terms of computational time. For that reason, we study the case $n = 4$ only and modify the number of exercise dates to five, i.e., $\mathcal{J} = 4$. All other parameter remain as before. Table 3 shows us that both types of regression provide a similar value for $\bar{v}_0$ but using SR is more than two times more expensive, compare Figure 3.

x_0	$\bar{v}_0(x_0)$ based on PR	$\bar{v}_0(x_0)$ based on SR	Parameter used
$n = 4$			
90	13.719 (0.008)	13.708 (0.008)	$K = 126$, $\hat{\sigma}_n = 0.32$
100	22.170 (0.010)	22.163 (0.010)	and
110	31.914 (0.011)	31.915 (0.011)	$\mathrm{m}_n = \ln(x_0) - 0.179$

Table 3: Approximative value $\bar{v}_0$ of the Bermudan option based on SR and PR for $\mathcal{J} = 4$ using Longstaff–Schwartz.

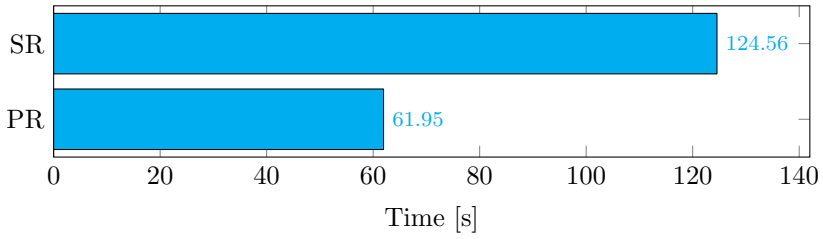


Figure 3: Time to compute continuation functions using Longstaff–Schwartz; standard versus pseudo regression for $n = 4$ and $\mathcal{J} = 4$.

Remark 6.2. *The gain of the pseudo-regression version of the Longstaff–Schwartz algorithm with respect to the standard one is generally smaller than the gain in Section 6.1, in the context of Tsitsiklis–van Roy. The reason is clear: The backward steps (8)–(10) in the standard algorithm cannot be straightforwardly modified to a pseudo-regression based setting. Therefore in order to construct a vector $\mathcal{Y}^{(j)}$ of stopped cash-flows in a pseudo-regression based setup analogue to the standard method, new trajectories are simulated for each exercise date starting from an initial state simulated under μ. Of course, this makes the procedure more costly, however, the generally expensive inversion of a random matrix at each exercise date is avoided this way.*

7 Conclusions

We compare the classical regression method for solving Bermudan options in the form of either the Tsitsiklis–van Roy or the Longstaff–Schwartz algorithm with a new variant based on pseudo regression, i.e., Monte Carlo simulation of L^2 inner products based on samples simulated from an artificial base measure μ (not directly related to the distribution of the underlying stock price at any point in time).

As a key issue, setting up and inverting a random matrix at every exercise date, can be avoided in the pseudo-regression approach. Therefore, the computational cost can be considerably lower for a similar level of accuracy, as verified in numerical examples in Section 6. This is also motivated from an asymptotic cost analysis, see Section 5. Furthermore, the pseudo-regression based algorithms are much easier to analyze theoretically, and leads to shorter and clearer convergence proofs (see Section 4). At the same time, convergence rates are slightly improved as the logarithmic error terms seen in classical regression error analysis (due to inversion of a random matrix) can be omitted.

The choice of the probability measure μ turns out to be crucial for the success of the pseudo regression algorithm. On the other hand, the procedure is insensitive enough w.r.t. μ that one single measure can in many cases be chosen for each time step. However, μ has to be chosen in an appropriate way since bad choices can lead to vastly increased errors of the ultimately computed option prices. As a rule, μ should be chosen such that $\mathcal{U} \sim \mu$ covers the important areas of the underlying stochastic process X_j for all j. These areas can usually be estimated roughly from the dynamics of the underlying process.

In some sense, the particular choice of μ can be compared to importance sampling, as it allows to change the distribution of the actual sampled points, without inducing bias. As such we expect that the flexibility in the choice of μ could be advantageous in particular in situations like deep out-of-the-money options, when the payoff is positive only on a rare event. We will study these aspects in more details in future work.

A Proofs

A.1 Proof of Theorem 4.1

Let u^K be the projection of u on to the linear span of $\psi_1, ..., \psi_K$, i.e.,

$$u^K = \underset{w \in \operatorname{span}\{\psi_1,...,\psi_K\}}{\arg\inf} \int_{\mathcal{D}} |w(z) - u(z)|^2 \, \mu(dz). \tag{32}$$

Then, with $\gamma := (\gamma_1, ..., \gamma_K)^\top \in \mathbb{R}^K$ defined by

$$u^K = \sum_{k=1}^{K} \gamma_k \psi_k, \tag{33}$$

and $\alpha \in \mathbb{R}^K$ defined by $\alpha_k := \langle \psi_k \, , u \rangle_{L^2(\mu)}$, it follows straightforwardly by taking scalar products that

$$\gamma = \mathcal{G}^{-1}\alpha. \tag{34}$$

By the rule of Pythagoras it follows that,

$$\begin{aligned} &\mathbb{E} \int_{\mathcal{D}} |\overline{u}(z) - u(z)|^2 \, \mu(dz) = \\ &\mathbb{E} \int_{\mathcal{D}} \left|\overline{u}(z)) - u^K(z)\right|^2 \mu(dz) + \int_{\mathcal{D}} \left|u^K(z) - u(z)\right|^2 \mu(dz). \end{aligned} \tag{35}$$

Hence, the second term in (25) is clear due to (32) and (35). With $\psi(z) := (\psi_1(z), ..., \psi_K(z))^\top$ we obtain for the first term in (35) that

$$\begin{aligned} &\mathbb{E} \int_{\mathcal{D}} \left|\overline{u}(z) - u^K(z)\right|^2 \mu(dz) = \int_{\mathcal{D}} \mathbb{E} \left|\overline{\beta}^\top \psi(z) - \gamma^\top \psi(z)\right|^2 \mu(dz) \\ &= \int_{\mathcal{D}} \mathbb{E} \left| \left(\frac{1}{M} \mathcal{Y}^\top \mathcal{M} - \alpha^\top \right) \mathcal{G}^{-1} \psi(z) \right|^2 \mu(dz) \\ &= \int_{\mathcal{D}} \mathbb{E} \left[\left(\frac{1}{M} \mathcal{Y}^\top \mathcal{M} - \alpha^\top \right) \mathcal{G}^{-1} \psi(z) \psi^\top(z) \mathcal{G}^{-1} \left(\frac{1}{M} \mathcal{M}^\top \mathcal{Y} - \alpha \right) \right] \mu(dz) \\ &= \mathbb{E} \left[\left(\frac{1}{M} \mathcal{Y}^\top \mathcal{M} - \alpha^\top \right) \mathcal{G}^{-1} \left(\frac{1}{M} \mathcal{M}^\top \mathcal{Y} - \alpha \right) \right], \end{aligned}$$

using (23), (24), (33), (34), and

$$\int_{\mathcal{D}} \left[\psi(z)\psi^\top(z)\right]_{kl} \mu(dz) = \langle \psi_k, \psi_l \rangle = \mathcal{G}_{kl}.$$

We thus have

$$\begin{aligned} 0 &\leq \mathbb{E} \int_{\mathcal{D}} \left|\overline{u}(z) - u^K(z)\right|^2 \mu(dz) \\ &\leq \frac{1}{\underline{\lambda}_{\min}} \mathbb{E} \left| \frac{1}{M} \mathcal{N}^\top \mathcal{Y} - \alpha \right|^2 = \frac{1}{\underline{\lambda}_{\min}} \sum_{k=1}^{K} \operatorname{Var} \left[\frac{1}{M} \mathcal{M}^\top \mathcal{Y} \right]_k, \end{aligned}$$

since

$$\begin{aligned}\mathbb{E}\left[\frac{1}{M}\mathcal{M}^\top\mathcal{Y}\right]_k &= \frac{1}{M}\mathbb{E}\sum_{m=1}^{M}\psi_k(\mathcal{U}^{(m)})Y^{(m)} \\ &= \mathbb{E}\left(\psi_k(\mathcal{U}^{(1)})Y^{(1)}\right) = \mathbb{E}\left(\psi_k(\mathcal{U}^{(1)})\mathbb{E}\left[Y^{(1)}\,|\mathcal{U}^{(1)}\right]\right) \\ &= \langle\psi_k, u\rangle = \alpha_k.\end{aligned} \tag{36}$$

Now, by observing that

$$\begin{aligned}\operatorname{Var}\left[\frac{1}{M}\mathcal{M}^\top\mathcal{Y}\right]_k &= \operatorname{Var}\left(\frac{1}{M}\sum_{m=1}^{M}\psi_k(\mathcal{U}^{(m)})Y^{(m)}\right) \\ &= \frac{1}{M}\operatorname{Var}\left(\psi_k(\mathcal{U}^{(1)})Y^{(1)}\right) \\ &= \frac{1}{M}\mathbb{E}\operatorname{Var}\left[\psi_k(\mathcal{U}^{(1)})Y^{(1)}|\mathcal{U}^{(1)}\right] + \frac{1}{M}\operatorname{Var}\mathbb{E}\left[\psi_k(\mathcal{U}^{(1)})Y^{(1)}|\mathcal{U}^{(1)}\right] \\ &= \frac{1}{M}\mathbb{E}\left(\psi_k^2(\mathcal{U}^{(1)})\operatorname{Var}\left[Y^{(1)}|\mathcal{U}^{(1)}\right]\right) + \frac{1}{M}\operatorname{Var}\psi_k(\mathcal{U}^{(1)})u\left(\mathcal{U}^{(1)}\right) \\ &\leq \frac{\sigma^2+D^2}{M}\mathcal{G}^K_{kk},\end{aligned} \tag{37}$$

one has

$$\frac{1}{\underline{\lambda_{\min}}}\sum_{k=1}^{K}\operatorname{Var}\left[\frac{1}{M}\mathcal{M}^\top\mathcal{Y}\right]_k \leq \frac{\sigma^2+D^2}{M\underline{\lambda_{\min}}}\mathsf{tr}\left(\mathcal{G}^K\right) \leq \frac{\sigma^2+D^2}{M\underline{\lambda_{\min}}}K\overline{\lambda_{\max}},$$

and then (25) follows.

Remark A.1. *From (36) we see that we are essentially approximating the inner products $\langle\psi_k\,,u\rangle_{L^2(\mu)}$ by a simple Monte Carlo simulation. At a first glance one may estimate the squared error due to (37) as being proportional to K^2/M (up to the projection error itself). Thus, Theorem 4.1 states that actually this error is proportional to K/M, even when the basis functions are not orthogonal.*

A.2 Proof of Lemma 4.3

Let X be a generic trajectory independent of $\mathcal{G}_{j+1}$, and let us represent a family of optimal stopping times τ_j^*, $j=1,...,\mathcal{J}$, by $\tau_{\mathcal{J}}^*=\mathcal{J}$, and for $j<\mathcal{J}$,

$$\tau_j^* := j\,1_{\left\{f_j(X_j)\geq c_j(X_j)\right\}} + \tau_{j+1}^* 1_{\left\{f_j(X_j)<c_j(X_j)\right\}}.$$

For $j<\mathcal{J}$ we then have,

$$\begin{aligned}f_{\tau_{j+1}^*}(X_{\tau_{j+1}^*}) - f_{\overline{\tau}_{j+1}}(X_{\overline{\tau}_{j+1}}) &= \left(f_{j+1}(X_{j+1}) - f_{\overline{\tau}_{j+1}}(X_{\overline{\tau}_{j+1}})\right)1_{\{\tau_{j+1}^*=j+1,\overline{\tau}_{j+1}>j+1\}} \\ &+ \left(f_{\tau_{j+1}^*}(X_{\tau_{j+1}^*}) - f_j(X_{j+1})\right)1_{\{\tau_{j+1}^*>j+1,\overline{\tau}_{j+1}=j+1\}} \\ &+ \left(f_{\tau_{j+1}^*}(X_{\tau_{j+1}^*}) - f_{\overline{\tau}_{j+1}}(X_{\overline{\tau}_{j+1}})\right)1_{\{\tau_{j+1}^*>j+1,\overline{\tau}_{j+1}>j+1\}}.\end{aligned}$$

By denoting temporarily $\mathsf{E} := \mathbb{E}_{\mathcal{G}_{j+1}}$, and denoting $\mathcal{R}_j := \mathsf{E}\left[f_{\tau^*_{j+1}}(X_{\tau^*_{j+1}}) - f_{\overline{\tau}_{j+1}}(X_{\overline{\tau}_{j+1}}) \middle| X_j \right]$, we have $\mathcal{R}_j \geq 0$ almost surely, and

$$\begin{aligned}
\mathcal{R}_j &= \mathsf{E}\left[\left(f_{j+1}(X_{j+1}) - \mathsf{E}\left[f_{\overline{\tau}_{j+2}}(X_{\overline{\tau}_{j+2}}) \middle| X_{j+1} \right] \right) 1_{\{\tau^*_{j+1}=j+1, \overline{\tau}_{j+1}>j+1\}} \middle| X_j \right] \\
&\quad + \mathsf{E}\left[\left(\mathsf{E}\left[f_{\tau^*_{j+2}}(X_{\tau^*_{j+2}}) \middle| X_{j+1} \right] - f_{j+1}(X_{j+1}) \right) 1_{\{\tau^*_{j+1}>j+1, \overline{\tau}_{j+1}=j+1\}} \middle| X_j \right] \\
&\quad + \mathsf{E}\left[\mathsf{E}\left[f_{\tau^*_{j+2}}(X_{\tau^*_{j+2}}) - f_{\overline{\tau}_{j+2}}(X_{\overline{\tau}_{j+2}}) \middle| X_{j+1} \right] 1_{\{\tau^*_{j+1}>j+1, \overline{\tau}_{j+1}>j+1\}} \middle| X_j \right] \\
&=: T_1 + T_2 + \mathsf{E}\left[\mathcal{R}_{j+1} 1_{\{\tau^*_{j+1}>j+1, \overline{\tau}_{j+1}>j+1\}} \middle| X_j \right]. \qquad (38)
\end{aligned}$$

For T_1 we have

$$\begin{aligned}
T_1 &= \mathsf{E}\left[\left(f_{j+1}(X_{j+1}) - \mathsf{E}\left[f_{\tau^*_{j+2}}(X_{\tau^*_{j+2}}) \middle| X_{j+1} \right] \right) 1_{\{\tau^*_{j+1}=j+1, \overline{\tau}_{j+1}>j+1\}} \middle| X_j \right] \\
&\quad + \mathsf{E}\left[\left(\mathsf{E}\left[f_{\tau^*_{j+2}}(X_{\tau^*_{j+2}}) \middle| X_{j+1} \right] - \mathsf{E}\left[f_{\overline{\tau}_{j+2}}(X_{\overline{\tau}_{j+2}}) \middle| X_{j+1} \right] \right) 1_{\{\tau^*_{j+1}=j+1, \overline{\tau}_{j+1}>j+1\}} \middle| X_j \right],
\end{aligned}$$

and since

$$\begin{aligned}
\widetilde{c}_{j+1}(X_{j+1}) &\geq f_{j+1}(X_{j+1}) \geq \mathsf{E}\left[f_{\tau^*_{j+2}}(X_{\tau^*_{j+2}}) \middle| X_{j+1} \right] \\
&= c_{j+1}(X_{j+1}) \geq \mathsf{E}\left[f_{\overline{\tau}_{j+2}}(X_{\overline{\tau}_{j+2}}) \middle| X_{j+1} \right]
\end{aligned}$$

on $\{\tau^*_{j+1} = j+1, \overline{\tau}_{j+1} > j+1\}$, we get

$$\begin{aligned}
0 \leq T_1 &\leq \mathsf{E}\left[\left(\widetilde{c}_{j+1}(X_{j+1}) - c_{j+1}(X_{j+1}) \right) 1_{\{\tau^*_{j+1}=j+1, \overline{\tau}_{j+1}>j+1\}} \middle| X_j \right] \\
&\quad + \mathsf{E}\left[\mathcal{R}_{j+1} 1_{\{\tau^*_{j+1}=j+1, \overline{\tau}_{j+1}>j+1\}} \middle| X_j \right]. \qquad (39)
\end{aligned}$$

Similarly, for T_2, we find

$$0 \leq T_2 \leq \mathsf{E}\left[\left(c_{j+1}(X_{j+1}) - \widetilde{c}_{j+1}(X_{j+1}) \right) 1_{\{\tau^*_{j+1}>j+1, \overline{\tau}_{j+1}=j+1\}} \middle| X_j \right]. \qquad (40)$$

Combining (38), (39), and (40), yields

$$\mathcal{R}_j \leq \mathsf{E}\left[|\widetilde{c}_{j+1}(X_{j+1}) - c_{j+1}(X_{j+1})| \middle| X_j \right] + \mathsf{E}\left[\mathcal{R}_{j+1} \middle| X_j \right].$$

By straightforward induction, using the tower property and the final condition $\mathcal{R}_{\mathcal{J}-1} = 0$, we then obtain

$$0 \leq c_j(X_j) - \overline{c}_j(X_j) \leq \sum_{l=j+1}^{\mathcal{J}-1} \mathsf{E}\left[|\widetilde{c}_l(X_l) - c_l(X_l)| \middle| X_j \right].$$

By now taking $X_l = X_l^{j,\mathcal{U}}$ independent of $\mathcal{G}_{j+1}$, and then on both sides the L_p-norm due to the distribution of $X_j^{j,\mathcal{U}} \sim \mu$, applying the triangle inequality, and by using that

$$\mathsf{E}\left[\mathsf{E}\left[|\widetilde{c}_l(X_l) - c_l(X_l)| \middle| X_j \right]^p \right] \leq \mathsf{E}\left[|\widetilde{c}_l(X_l) - c_l(X_l)|^p \right],$$

we finally obtain (28).

A.3 Proof of Theorem 4.5

The theorem is proven by induction. Due to Theorem 4.1 we have (noting that $\overline{c}_j$ and $\widetilde{c}_j$ are random functions) that

$$\mathbb{E}_{\mathcal{G}_{j+1}}\left[\|\overline{c}_j - \widetilde{c}_j\|^2_{L_2(\mu)}\right] \leq C_1^2 \frac{K}{M} + C_2^2 \inf_{w \,\in\, \mathsf{span}\{\psi_1,\ldots,\psi_K\}} \|\widetilde{c}_j - w\|^2_{L_2(\mu)}\,, \quad \text{hence}$$

$$\|\overline{c}_j - \widetilde{c}_j\|_{L_2(\mu\otimes\mathbb{P})} \leq C_1\sqrt{\frac{K}{M}} + C_2 \inf_{w \,\in\, \mathsf{span}\{\psi_1,\ldots,\psi_K\}} \|\widetilde{c}_j - w\|_{L_2(\mu\otimes\mathbb{P})} \tag{41}$$

almost surely, for some $C_1, C_2 > 0$, which do not depend on j, K, and N. We now prove the statement (29) for $\eta := \max(C_1, C_2)$. Since $\widetilde{c}_{\mathcal{J}-1} = c_{\mathcal{J}-1}$ for time $\mathcal{J}-1$, (29) is implied by (41) with $j = \mathcal{J}-1$. Suppose the statement is proved for $0 < j+1 \leq \mathcal{J}-1$. Let us write,

$$\inf_{w \,\in\, \mathsf{span}\{\psi_1,\ldots,\psi_K\}} \|\widetilde{c}_j - w\|_{L_2(\mu\otimes\mathbb{P})} \leq \|\widetilde{c}_j - c_j\|_{L_2(\mu\otimes\mathbb{P})} + \inf_{w \,\in\, \mathsf{span}\{\psi_1,\ldots,\psi_K\}} \|c_j - w\|_{L_2(\mu)}\,. \tag{42}$$

By using (41), (42), and the unconditional expectation applied to Lemma 4.3 with $p = 2$ we get

$$\begin{aligned}
\|\overline{c}_j - c_j\|_{L_2(\mu\otimes\mathbb{P})} &\leq \|\overline{c}_j - \widetilde{c}_j\|_{L_2(\mu\otimes\mathbb{P})} + \|\widetilde{c}_j - c_j\|_{L_2(\mu\otimes\mathbb{P})} \\
&\leq C_1\sqrt{\frac{K}{M}} + C_2 \inf_{w \,\in\, \mathsf{span}\{\psi_1,\ldots,\psi_K\}} \|c_j - w\|_{L_2(\mu)} \\
&\quad + (C_2+1)\,\|\widetilde{c}_j - c_j\|_{L_2(\mu\otimes\mathbb{P})} \\
&\leq \eta\varepsilon_{j,M,K} + (\eta+1)\sum_{l=j+1}^{\mathcal{J}-1} \|\overline{c}_l - c_l\|_{L_2(\mu_{j,l}\otimes\mathbb{P})}\,.
\end{aligned} \tag{43}$$

Using the induction hypothesis we have,

$$\begin{aligned}
\sum_{l=j+1}^{\mathcal{J}-1} \|\overline{c}_l - c_l\|_{L_2(\mu_{j,l}\otimes\mathbb{P})} &\leq \sum_{l=j+1}^{\mathcal{J}-1} \eta(\eta+2)^{\mathcal{J}-1-l}\varepsilon_{l,M,K} \\
&\leq \eta\varepsilon_{j,M,K}\frac{(2+\eta)^{\mathcal{J}-1-j}-1}{\eta+1}.
\end{aligned}$$

Combining the latter with (43) yields (29).

References

[AB04] L. Andersen and M. Broadie. Primal-Dual Simulation Algorithm for Pricing Multidimensional American Options. *Manage. Sci.*, 50:1222–1234, 2004.

[ABE+17] F. Anker, C. Bayer, M. Eigel, M. Ladkau, J. Neumann, and J. Schoenmakers. SDE based regression for linear random PDEs. *SIAM J. Sci. Comput.*, 39(3):A1168–A1200, 2017.

[BBR+18] C. Bayer, D. Belomestny, M. Redmann, S. Riedel, and J. Schoenmakers. Solving linear parabolic rough partial differential equations. *ArXiv e-prints*, March 2018.

[BS18a] D. Belomestny and J. Schoenmakers. Optimal stopping of McKean-Vlasov diffusions via regression on particle systems. *arXiv preprint arXiv: 1806.09483*, 2018.

[BS18b] Denis Belomestny and John Schoenmakers. *Advanced simulation-based methods for optimal stopping and control*. Palgrave Macmillan, London, 2018. With applications in finance.

[GKKW02] L. Györfi, M. Kohler, A. Krzyżak, and H. Walk. *A distribution-free theory of nonparametric regression*. Springer Series in Statistics. Springer-Verlag, New York, 2002.

[Gla04] Paul Glasserman. *Monte Carlo methods in financial engineering*, volume 53 of *Applications of Mathematics (New York)*. Springer-Verlag, New York, 2004. Stochastic Modelling and Applied Probability.

[LS01] F.A. Longstaff and E.S. Schwartz. Valuing American options by simulation: a simple least-squares approach. *Review of Financial Studies*, 14(1):113–147, 2001.

[Nev75] J. Neveu. *Discrete-parameter martingales*. North-Holland Publishing Co., Amsterdam-Oxford; American Elsevier Publishing Co., Inc., New York, revised edition, 1975. Translated from the French by T. P. Speed, North-Holland Mathematical Library, Vol. 10.

[TVR01] J. Tsitsiklis and B. Van Roy. Regression methods for pricing complex American style options. *IEEE Trans. Neural. Net.*, 12(14):694–703, 2001.

[Zan13] D.Z. Zanger. Quantitative error estimates for a least-squares Monte Carlo algorithm for American option pricing. *Finance and Stochastics*, 17(3):503–534, 2013.