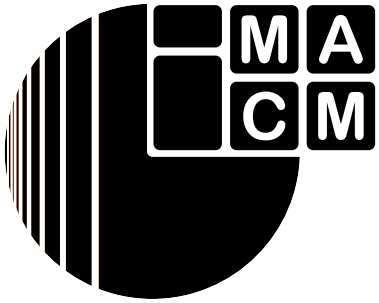


Bergische Universität Wuppertal

Fachbereich Mathematik und Naturwissenschaften

Institute of Mathematical Modelling, Analysis and Computational Mathematics
(IMACM)

Preprint BUW-IMACM 18/03
This version: July 2019

Long Teng

A Review of Tree-based Approaches to solve Forward-Backward Stochastic Differentl Equations

April, 2018

http://www.math.uni-wuppertal.de

A Review of Tree-based Approaches to solve Forward-Backward Stochastic Differential Equations

Long Teng

Lehrstuhl für Angewandte Mathematik und Numerische Analysis,
Fakultät für Mathematik und Naturwissenschaften,
Bergische Universität Wuppertal, Gaußstr. 20, 42119 Wuppertal, Germany,
teng@math.uni-wuppertal.de

Abstract

In this work, we study solving (decoupled) forward-backward stochastic differential equations (FBSDEs) numerically using the regression trees. Based on the general theta-discretization for the time-integrands, we show how to efficiently use regression tree-based methods to solve the resulting conditional expectations. Several numerical experiments including high-dimensional problems are provided to demonstrate the accuracy and performance of the tree-based approach. For the applicability of FBSDEs in financial problems, we apply our tree-based approach to the Heston stochastic volatility model, the high-dimensional pricing problems of a Rainbow option and an European financial derivative with different interest rates for borrowing and lending.

Keywords *forward-backward stochastic differential equations (FBSDEs), high-dimensional problem, regression tree*

1 Introduction

It is well-known that many problems (e.g., pricing, hedging) in the field of financial mathematics can be represented in terms of FBSDEs, which makes problems easier to solve but exhibits usually no analytical solution, see e.g., [19]. However, compared to the forward stochastic differential equations (SDEs), it is more challenged to efficiently find an accurate numerical solution of the FBSDEs. In this work, we show how to solve FBSDEs using the regression tree-based methods.

The general form of (decoupled) FBSDEs reads

$$\begin{cases} dX_t = a(t, X_t)\,dt + b(t, X_t)\,dW_t, \quad X_0 = x_0, \\ -dY_t = f(t, X_t, Y_t, Z_t)\,dt - Z_t\,dW_t, \\ Y_T = \xi = g(X_T), \end{cases} \tag{1}$$

where $X_t, a \in \mathbb{R}^n$, b is a $n \times d$ matrix, $f(t, X_t, Y_t, Z_t) : [0, T] \times \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{m\times d} \to \mathbb{R}^m$ is the driver function and ξ is the square-integrable terminal condition. We see that the

terminal condition Y_T depends on the final value of a forward SDE. For $a = 0$ and $b = 1$, namely $X_t = W_t$, one obtains a backward stochastic differential equation (BSDE) of the form

$$\begin{cases} -dY_t = f(t, Y_t, Z_t)\, dt - Z_t\, dW_t, \\ Y_T = \xi = g(W_T), \end{cases} \tag{2}$$

where $Y_t \in \mathbb{R}^m$, $W_t = (W_t^1, \cdots, W_t^d)^T$ is a d-dimensional Brownian motion and $f : [0, T] \times \mathbb{R}^m \times \mathbb{R}^{m\times d} \to \mathbb{R}^m$.

The existence and uniqueness of solutions of such equations under the Lipschitz conditions on $f, a(t, X_t), b(t, X_t)$ and g are proved by Pardoux and Peng [29, 30]. Since then, many works try to relax this condition, e.g., the uniqueness of solution is extended under more general assumptions for f in [22] but only in the one dimensional case. The solution of a (F)BSDE is a pair of adapted processes (Y, Z), the role of Z, namely $Z_t\, dW_t$ is to render the process Y be adapted. Moreover, in the application, the process Z can possess some useful information. For example, in option pricing problems, the process Z represents the hedging portfolio while the process Y corresponds to the option price.

In recent years, many numerical methods have been proposed for coupled and decoupled (F)BSDEs. For the numerical algorithms with (least-squares) Monte-Carlo approaches we refer to [3, 7, 16, 21, 37], the multilevel Monte Carlo method based on Picard approximation for high-dimensional nonlinear BSDEs can be found in [13]. Some numerical methods for BSDEs applying binomial tree are investigated in [23]. There exists connection between BSDEs and PDEs, see [20, 31], some numerical schemes with the aid of this connection can be found e.g., in [11, 24, 28]. For the deep-learning-based numerical method we refer to [12]. The approach based on the Fourier method for BSDEs is developed in [32]. See also [10] for the numerical schemes using cubature methods and [34] for the tree-based approach. And many others e.g., [1, 4, 14, 15, 25, 26, 36, 35, 42, 38].

In this paper, we show how to efficiently use regression tree-based approaches to find accurate approximations of (F)BSDEs (1) and (2). We apply the general theta-discretization method for the time-integrands and approximate the resulting conditional expectations using the regression tree-based approach. The schemes with different theta values are analyzed for the tree-based approach. Several numerical experiments of different types including high-dimensional problems and applications in pricing financial derivatives are performed to demonstrate our findings. We show numerical examples of 100-dimensional FBSDE to check the performance and applicability of our tree-based approach for a high-dimensional problem.

In the next section, we start with notation and definitions and discuss in Section 3 the discretization of time-integrands using the theta-method, and derive the reference equations according to the tree-based method. Section 4 is devoted to how to use the regression tree-based approaches to approximate the conditional expectations. In Section 5, several numerical experiments on different types of (F)BSDEs including financial applications are provided to show the accuracy and applicability for high-dimensional problems. Finally, Section 6 concludes this work.

2 Preliminaries

Throughout the paper, we assume that $(\Omega, \mathcal{F}, P; \{\mathcal{F}_t\}_{0\le t\le T})$ is a complete, filtered probability space. In this space, a standard d-dimensional Brownian motion W_t with a finite terminal time T is defined, which generates the filtration $\{\mathcal{F}_t\}_{0\le t\le T}$, i.e., $\mathcal{F}_t = \sigma\{X_s, 0 \le s \le t\}$ for FBSDEs or $\mathcal{F}_t = \sigma\{W_s, 0 \le s \le t\}$ for BSDEs. And the usual hypotheses should be satisfied. We denote the set of all $\mathcal{F}_t$-adapted and square integrable processes in $\mathbb{R}^d$ with $L^2 = L^2(0, T; \mathbb{R}^d)$. A pair of process (Y_t, Z_t) is the solution of the (F)BSDEs (1) or (2) if it is $\mathcal{F}_t$-adapted and square integrable and satisfies (1) or (2) as

$$Y_t = \xi + \int_t^T f(s, (X_s), Y_s, Z_s)\, ds - \int_t^T Z_s\, dW_s, \quad t \in [0, T], \tag{3}$$

where $f(t, (X_s), Y_s, Z_s) : [0, T]\ (\times\mathbb{R}^n) \times \mathbb{R}^m \times \mathbb{R}^{m\times d} \to \mathbb{R}^m$ is $\mathcal{F}_t$ adapted, $\xi = g(X_T) : \mathbb{R}^n \to \mathbb{R}^m$ or $\xi = g(W_T) : \mathbb{R}^d \to \mathbb{R}^m$. As mentioned above, these solutions exist uniquely under Lipschitz conditions.

Suppose that the terminal value Y_T is of the form $g(X_T^{t,x})$, where $X_T^{t,x}$ denotes the solution of dX_t in (1) starting from x at time t. Then the solution $(Y_t^{t,x}, Z_t^{t,x})$ of FBSDEs (1) can be represented [20, 26, 30, 31] as

$$Y_t^{t,x} = u(t, x), \quad Z_t^{t,x} = (\nabla u(t, x))b(t, x) \quad \forall t \in [0, T), \tag{4}$$

which is solution of the semi-linear parabolic PDE of the form

$$\frac{\partial u}{\partial t} + \sum_i^n a_i \partial_i u + \frac{1}{2}\sum_{i,j}^n (bb^T)_{i,j}\partial_{i,j}^2 u + f(t, x, u, (\nabla u)b) = 0 \tag{5}$$

with the terminal condition $u(T, x) = g(x)$. Clearly, the corresponding PDE to the BSDEs (2) with $\xi = g(W_T) : \Omega \times \mathbb{R}^d \to \mathbb{R}^m$ reads

$$\begin{cases} \frac{\partial u}{\partial t} + \frac{1}{2}\sum_i^d \partial_{i,i}^2 u + f(t, u, (\nabla u)b) = 0, \\ u(T, x) = g(x). \end{cases} \tag{6}$$

In turn, suppose (Y, Z) is the solution of (F)BSDEs, $u(t, x) = Y_t^{t,x}$ is a viscosity solution to the PDEs. As mentioned above, BSDE is a special case of FBSDE with $a = 0$ and $b = 1$. Thus, we introduce the numerical schemes concerning FBSDEs in the sequel.

3 Discretization of the FBSDE using theta-method

For simplicity, we discuss the discretization with one-dimensional processes, namely $m = n = d = 1$. And the extension to higher dimensions is possible and straightforward. We introduce the time partition for the time interval $[0, T]$

$$\Delta_t = \{t_i | t_i \in [0, T], i = 0, 1, \cdots, N_T, t_i < t_{i+1}, t_0 = 0, t_{N_T} = T\}. \tag{7}$$

Let $\Delta t_i = t_{i+1} - t_i$ be the time step, and denote the maximum time step with Δt. For the FBSDEs, one needs to additionally discretize the forward SDE in (1)

$$X_t = x_0 + \int_0^t a(s, X_s)\, ds + \int_0^t b(s, X_s)\, dW_s. \tag{8}$$

Suppose that the forward SDE (8) can be already discretized by a process $X_{t_i}^{\Delta_t}$ such that

$$E\left[\max_{t_i} \left| X_{t_i} - X_{t_i}^{\Delta_t} \right|^2\right] = \mathcal{O}(\Delta_t) \tag{9}$$

which means strong mean square convergence of order $1/2$. In the case of that X_t follows a known distribution (e.g., geometric Brownian motion), one can obtain good samples on Δ_t using the known distribution, otherwise the Euler scheme can be employed.

Then one needs to discretize the backward process (3), namely

$$Y_t = \xi + \int_t^T f(s, \mathbb{X}_s)\, ds - \int_t^T Z_s\, dW_s, \quad t \in [0, T), \tag{10}$$

where $\xi = g(X_T), \mathbb{X}_s = (X_s, Y_s, Z_s)$. Let (Y_t, Z_t) be the adapted solution of (10), we thus have

$$Y_i = Y_{i+1} + \int_{t_i}^{t_{i+1}} f(s, \mathbb{X}_s)\, ds - \int_{t_i}^{t_{i+1}} Z_s\, dW_s, \tag{11}$$

where Y_{t_i} is denoted by Y_i for simple notation. To obtain adaptability of the solution (Y_t, Z_t), we could use conditional expectations $E_i[\cdot](= E[\cdot|\mathcal{F}_{t_i}])$. We consider firstly to find the reference equation for Z. By multiplying both sides of the equation (11) by $\Delta W_{i+1} := W_{t_{i+1}} - W_{t_i}$ and taking the conditional expectations $E_i[\cdot]$ on the both sides of the derived equation we obtain

$$-E_i[Y_{i+1}\Delta W_{i+1}] = \int_{t_i}^{t_{i+1}} E_i[f(s, \mathbb{X}_s)\Delta W_s]\, ds - \int_{t_i}^{t_{i+1}} E_i[Z_s]\, ds, \tag{12}$$

where the Itô isometry and Fubini's theorem are used and $\Delta W_s := W_s - W_{t_i}$. Obviously, with respect to the filtration $\mathcal{F}_{t_i}$, the integrands on the right-hand side of (12) is deterministic of time s. Thus, applying the theta-method gives

$$\begin{aligned} -E_i[Y_{i+1}\Delta W_{i+1}] &= \Delta t_i(1-\theta_1)E_i[f(t_{i+1}, \mathbb{X}_{i+1})\Delta W_{i+1}] - \Delta t_i\theta_2 Z_i \\ &\quad - \Delta t_i(1-\theta_2)E_i[Z_{i+1}] + R_\theta^{Z_i}, \\ &\approx \Delta t_i(1-\theta_1)E_i[f(t_{i+1}, \mathbb{X}_{i+1})\Delta W_{i+1}] - \Delta t_i\theta_2 Z_i - \Delta t_i(1-\theta_2)E_i[Z_{i+1}], \end{aligned} \tag{13}$$

where $\theta_1 \in [0,1], \theta_2 \in [0,1)$ and $R_\theta^{Z_i}$ is the discretization error of the integrals in (12). Therefore, the equation (13) lead to a time discrete approximation Z^{Δ_t} for Z

$$\begin{aligned} Z_i^{\Delta_t} &= \frac{\theta_2^{-1}}{\Delta t_i} E_i[Y_{i+1}^{\Delta_t}\Delta W_{i+1}] + \theta_2^{-1}(1-\theta_1)E_i[f(t_{i+1}, \mathbb{X}_{i+1}^{\Delta_t})\Delta W_{i+1}] \\ &\quad - \theta_2^{-1}(1-\theta_2)E_i[Z_{i+1}^{\Delta_t}]. \end{aligned} \tag{14}$$

We start now finding the reference equation for Y. We could take the conditional expectations $E_i[\cdot]$ on the both sides of (10) to obtain

$$Y_i = E_i[Y_{i+1}] + \int_{t_i}^{t_{i+1}} E_i[f(s, \mathbb{X}_s)]\, ds. \tag{15}$$

Again, the integrand on the right-hand side of (15) is deterministic of time s with respect to the filtration $\mathcal{F}_{t_i}$. We use again the theta-method and obtain

$$\begin{aligned} Y_i &= E_i[Y_{i+1}] + \Delta t_i \theta_3 f(t_i, \mathbb{X}_i) + \Delta t_i (1-\theta_3) E_i[f(t_{i+1}, \mathbb{X}_{i+1})] + R_\theta^{Y_i}, \quad \theta_3 \in [0,1] \\ &\approx E_i[Y_{i+1}] + \Delta t_i \theta_3 f(t_i, \mathbb{X}_i) + \Delta t_i (1-\theta_3) E_i[f(t_{i+1}, \mathbb{X}_{i+1})], \end{aligned} \tag{16}$$

where $R_\theta^{Y_i}$ is the discretization error of the integral in (10). Due to $\mathbb{X}_i^{\Delta_t} = (X_i^{\Delta_t}, Y_i^{\Delta_t}, Z_i^{\Delta_t})$, obviously, we have obtained an implicit scheme which can be directly solved by using iterative methods, e.g., Newton's method or Picard scheme.

By choosing the different values for θ_1 and θ_2, one can obtain different schemes. For example, one receives the Crank-Nicolson scheme by setting $\theta_1 = \theta_2 = \theta_3 = 1/2$, which is second-order accurate. When $\theta_1 = \theta_2 = \theta_3 = 1$, the scheme is first-order accurate, see [37, 41, 39]. In our experiments we find that the numerical second-order convergence rate can only be achieved when the number of samples is sufficiently large. The convenience rate of the tree-based method is one divided by the square root of sample size, to receive the accuracy $(\Delta t)^2 = (\frac{T}{N_T})^2$, the number of samples should be around $(\frac{N_T}{T})^4$. For example, when $T = 0.5$ and $N_T = 32$, one needs 64^4 samples to obtain that accuracy, that is a quite large integer. Therefore, to evaluate the performance of the tree-based methods with smaller sample size, in this work we will consider the first-order accurate scheme for solving the FBSDEs by choosing $\theta_1 = 1/2, \theta_2 = 1, \theta_3 = 1/2$:

$$Y_{N_T}^{\Delta_t} = g(X_{N_T}^{\Delta_t}),\ Z_{N_T}^{\Delta_t} = g_x(X_{N_T}^{\Delta_t}), \tag{17}$$

$$\textbf{For } i = N_T - 1, \cdots, 0:$$

$$Z_i^{\Delta_t} = \frac{1}{\Delta t_i} E_i[Y_{i+1}^{\Delta_t} \Delta W_{i+1}] + \frac{1}{2} E_i[f(t_{i+1}, \mathbb{X}_{i+1}^{\Delta_t}) \Delta W_{i+1}], \tag{18}$$

$$Y_i^{\Delta_t} = E_i[Y_{i+1}^{\Delta_t}] + \frac{\Delta t_i}{2} f(t_i, \mathbb{X}_i^{\Delta_t}) + \frac{\Delta t_i}{2} E_i[f(t_{i+1}, \mathbb{X}_{i+1}^{\Delta_t})]. \tag{19}$$

The error estimates for the scheme above is given in Section 4.3.

4 Computation of conditional expectations with the tree-based approach

In this section we introduce how to use the tree-based approach to compute the conditional expectations included in the schemes introduced above, which actually are all of the form $E[Y|X]$ for square integrable random variables X and Y. Therefore, we present the regression approach based on the form $E[Y|X]$ throughout this section.

4.1 Non-parametric regression

We assume that the model in non-parametric regression reads

$$Y = \eta(X) + \epsilon, \tag{20}$$

where ϵ has a zero expectation and a constant variance. Obviously, it can be thus implied that

$$E[Y|X = x] = \eta(x). \tag{21}$$

To approximate the conditional expectations, our goal in regression is to find the estimator of this function, $\hat{\eta}(x)$. By non-parametric regression, we are not assuming a particular form for η. Instead of, $\hat{\eta}$ is represented in a regression tree. Suppose we have a set of samples, $(\hat{x}_{\mathcal{M}}, \hat{y}_{\mathcal{M}})$, $\mathcal{M} = 1, \cdots, M$, for (X, Y), where X denotes a predictor variable and Y presents the corresponding response variable. With such samples we construct a regression tree, which can then be used to determine $E[Y|X = x]$ for an arbitrary x, whose value is not necessarily equal to one of samples $\hat{x}_{\mathcal{M}}$.

As an example, we specify the procedure for (18) in case of FBSDEs, namely where $\mathbb{X}_{i+1}^{\Delta_t} = (X_{i+1}^{\Delta_t}, Y_{i+1}^{\Delta_t}, Z_{i+1}^{\Delta_t})$. We assume that $(X_i^{\Delta_t}, \mathcal{F}_{t_i})_{t_i \in \Delta_t}$ is Markovian. Hence, (18) can be rewritten as

$$Z_i^{\Delta_t} = E\left[\frac{1}{\Delta t_i} Y_{i+1}^{\Delta_t} \Delta W_{i+1} + \frac{1}{2} f(t_{i+1}, \mathbb{X}_{i+1}^{\Delta_t}) \Delta W_{i+1} | X_i^{\Delta_t}\right], \quad i = N_T - 1, \cdots, 0. \tag{22}$$

And there exist deterministic functions $z_i^{\Delta_t}(x)$ such that

$$Z_i^{\Delta_t} = z_i^{\Delta_t}(X_i^{\Delta_t}). \tag{23}$$

Starting from the time T, we construct the regression tree $\hat{T}_z$ for the conditional expectation in (22) using samples $(\hat{x}_{N_T-1,\mathcal{M}}, \frac{1}{\Delta t_{N_T-1}} \hat{y}_{N_T,\mathcal{M}} \Delta \hat{w}_{N_T,\mathcal{M}} + \frac{1}{2} \hat{f}_{N_T,\mathcal{M}} \Delta \hat{w}_{N_T,\mathcal{M}})$. Thereby, the function

$$z_{N_T-1}^{\Delta_t}(x) = E\left[\frac{1}{\Delta t_{N_T-1}} Y_{N_T}^{\Delta_t} \Delta W_{N_T} + f(t_{N_T}, \mathbb{X}_{N_T}^{\Delta_t}) \Delta W_{N_T} | X_{N_T-1}^{\Delta_t} = x\right], \tag{24}$$

is estimated and presented by a regression tree. Based on the constructed tree, by applying (24) to the samples $\hat{x}_{N_T-1,\mathcal{M}}$ one can directly obtain the samples $\hat{z}_{N_T-1,\mathcal{M}}$ of the random variable $Z_{N_T-1}^{\Delta_t}$, for $\mathcal{M} = 1, \cdots, M$. Recursively, backward in time, these samples $\hat{z}_{N_T-1,\mathcal{M}}$ will be used to generate samples $\hat{z}_{N_T-2,\mathcal{M}}$ of the random variables $Z_{N_T-2}^{\Delta_t}$ at the time t_{N_T-2}. At the initial time $t = 0$, we have a fix initial value x_0 for dX_t, no samples are needed. Using the regression trees constructed at time t_1 we obtain the solution $Z_0^{\Delta_t} = z_0^{\Delta_t}(x_0)$. For the BSDEs, X_t is just the Brownian motion W_t, which has the zero initial value. Following the same procedure to the conditional expectations in (19), one obtains implicitly $Y_0^{\Delta_t}$.

4.2 Binary regression tree

As mentioned above, regression tree is used to estimate relationship between the predictor variable X and the response variable Y, namely to find the estimator $\hat{\eta}$ of η in (21) and then to predict given future samples of X. In this section, we review the procedure in [8, 27] for constructing a best regression tree based on the given samples. Basically, we need to grow, prune and finally select the tree. We firstly give the notation:

- $(\hat{x}_{\mathcal{M}}, \hat{y}_{\mathcal{M}})$ denote samples, namely observed data.
- $\hat{t}$ is a node in the tree $\hat{T}$, $\hat{t}_L$ and $\hat{t}_R$ are the left and right child nodes.
- $\mathcal{T}$ is the set of terminal nodes in the the tree $\hat{T}$ with the number $|\mathcal{T}|$
- $n(\hat{t})$ represents the number of samples in node $\hat{t}$.
- $\bar{y}(\hat{t})$ is the average of samples falling into node $\hat{t}$, namely predicted response

Growing a Tree We define predicted response as the average value of the samples which are contained in a node $\hat{t}$, namely

$$\bar{y}(\hat{t}) = \frac{1}{n(\hat{t})} \sum_{\hat{x}_{\mathcal{M}} \in \hat{t}} \hat{y}_{\mathcal{M}}. \tag{25}$$

Obviously, the squared error in the node $\hat{t}$ reads

$$R(\hat{t}) = \frac{1}{n(\hat{t})} \sum_{\hat{x}_{\mathcal{M}} \in \hat{t}} (\hat{y}_{\mathcal{M}} - \bar{y}(\hat{t}))^2. \tag{26}$$

The mean squared error for the tree $\hat{T}$ is defined as the sum of the squared errors in all the terminal nodes and given by

$$R(\hat{T}) = \sum_{\hat{t} \in \mathcal{T}} R(\hat{t}) = \frac{1}{n(\hat{t})} \sum_{\hat{t} \in \mathcal{T}} \sum_{\hat{x}_{\mathcal{M}} \in \hat{t}} (\hat{y}_{\mathcal{M}} - \bar{y}(\hat{t}))^2. \tag{27}$$

Basically, the tree is constructed by partitioning the space for the samples $\hat{x}$ using a sequence of binary splits. For a split s and node $\hat{t}$, the change in the mean squared error can be thus calculated as

$$\Delta R(s, \hat{t}) = R(\hat{t}) - R(\hat{t}_L) - R(\hat{t}_R). \tag{28}$$

The regression tree is thus obtained by iteratively splitting nodes with s, which yields the largest $\Delta R(s, \hat{t})$. Thereby, decrease in $R(\hat{T})$ is maximized. Obviously, the optimal stopping criterion is that all responses in a terminal node are the same, but that is not really realistic. There are some other criteria are available, e.g., growing the tree until number of samples in a terminal node is five, which is suggested in [8].

Pruning a tree When using the optimal stopping criterion, all responses in a terminal node are same, i.e., each terminal node contains only one response, then the error $R(\hat{t})$, therewith $R(\hat{T})$, will be zero. However, first of all, this is unrealistic as already mentioned. Secondly, the samples is thereby over fitted and the regression tree will thus not generalize well to new observed samples. Breiman et al. [8] suggested growing an overly large regression tree $\hat{T}_{\max}$ and then to find nested sequence of sub-trees by successively pruning branches of the tree. This procedure is called pruning a tree. We define an error-complexity measure as

$$R_\alpha(\hat{T}) = R(\hat{T}) + \alpha|\mathcal{T}|, \quad \alpha \geq 0, \tag{29}$$

where α represents the complexity cost per terminal node. The error-complexity should be minimized by looking for trees. Let $\hat{T}_{\max}$ be the overly large tree, in which each terminal node contains only one response. Thus, we have $R_\alpha(\hat{T}_{\max}) = \alpha|\mathcal{T}|$ which indicates a high cost of complexity, while the error is small. To minimize the cost we delete the branches with the weakest link $\hat{t}_k^*$ in tree $\hat{T}_k$, which is defined as

$$g_k(\hat{t}_k^*) = \min_{\hat{t}}\{g_k(\hat{t})\}, \quad g_k(\hat{t}) = \frac{R(\hat{t}) - R(\hat{T}_{k\hat{t}})}{|\mathcal{T}_{k\hat{t}}| - 1}, \tag{30}$$

where $\hat{T}_{k\hat{t}}$ is the branch $\hat{T}_{\hat{t}}$ corresponding to the internal node $\hat{t}$ of sub-tree $\hat{T}_k$. Then, we prune the branch defined by the node $\hat{t}_k^*$

$$\hat{T}_{k+1} = \hat{T}_k - \hat{T}_{\hat{t}_k^*}, \tag{31}$$

and thus obtain a finite sequence of sub-trees with fewer terminal nodes and decreasing complexity until the root node as

$$\hat{T}_{\max} > \hat{T}_1 > \hat{T}_2 > \cdots > \hat{T}_K = \text{root}. \tag{32}$$

On the other hand, we set

$$\alpha_{k+1} = g_k(\hat{t}_k^*) \tag{33}$$

and thus obtain an increasing sequence of values for the complexity parameter α, namely

$$0 = \alpha_1 < \cdots < \alpha_k < \alpha_{k+1} < \cdots \alpha_K. \tag{34}$$

By observing the both sequences (32) and (34), it is not difficult to find: for $k \geq 1$, the tree $\hat{T}_k$ is the one which has the minimal cost complexity for $\alpha_k \leq \alpha < \alpha_{k+1}$.

Selecting a Tree We have to make a trade-off between the both criteria of error and complexity, namely we need to choose the best tree from the sequence of pruned sub-trees such that the complexity of tree and squared error are both minimized. To do this, there are two possible ways introduced in [8, 27], namely independent test samples and cross-validation. As an example, we illustrate the independent test sample method, for cross-validation we refer to [8, 27]. Clearly, we need honest estimates of the true

error $R^*(\hat{T})$ to select the right size of the tree. To obtain that estimates, we should use samples that were not used to construct the tree to estimate the error. Suppose we have a set of samples $L = (\hat{x}_{\mathcal{M}}, \hat{y}_{\mathcal{M}})$, which should be randomly divided into two subsets L_1 and L_2. We use the set L_1 to grow a large tree and to obtain the sequence of pruned sub-trees. Thus, the samples in L_2 is used to evaluate the performance of each sub-tree by calculating the error between real response and predicated response. We denote the predicated response using samples $\hat{x}$ to the tree $\hat{T}_k$ with $\bar{y}_k(\hat{x})$, then the estimated error is

$$\hat{R}(\hat{T}_k) = \frac{1}{n_2} \sum_{(\hat{x}_i, \hat{y}_i) \in L_2} (\hat{y}_i - \bar{y}_k(\hat{x}_i))^2, \tag{35}$$

where n_2 is the number of samples in L_2. This estimated error will be calculated for all sub-trees. As mentioned above, if one directly select the tree with the smallest error, then the cost of complexity will be higher. Instead of, we can pick a sub-tree that has the fewest number of nodes, but still keeps the accuracy of the tree with the smallest error, say $\hat{T}_0$ with the error $\hat{R}_{\min}(\hat{T}_0)$. To do this, we define the standard error for this estimate as [8]

$$SE(\hat{R}_{\min}(\hat{T}_0)) := \frac{1}{\sqrt{n_2}} \sqrt{\frac{1}{n_2} \sum_{i=1}^{n_2} (\hat{y}_i - \bar{y}(\hat{x_i}))^4 - (\hat{R}_{\min}(\hat{T}_0))^2}, \tag{36}$$

and then choose the smallest tree $\hat{T}_k^*$ such that

$$\hat{R}(\hat{T}_k^*) \leq \hat{R}_{\min}(\hat{T}_0) + SE(\hat{R}_{\min}(\hat{T}_0)). \tag{37}$$

$\hat{T}_k^*$ is the tree with minimal complexity cost but has equivalent accuracy as the tree with minimum error.

4.3 Practical Applications

Note that we do not need to construct the individual tree for each conditional expectation in the schemes. Due to the linearity of conditional expectation, we construct the trees for all possible combinations of the conditional expectations. We denote the tree's regression error with R_{tr}, the error of used iterative method with R_{impl} and reformulate the scheme (17)-(19) by combining conditional expectations and including all errors as

$$\begin{aligned}
&\hat{y}_{N_T,\mathcal{M}} = g(\hat{x}_{N_T,\mathcal{M}}),\ \hat{z}_{N_T,\mathcal{M}} = g_x(\hat{x}_{N_T,\mathcal{M}}),\\
&\textbf{For } i = N_T - 1, \cdots, 0\ ,\ \mathcal{M} = 1, \cdots, M:\\
&\hat{z}_{i,\mathcal{M}} = E_i^{\hat{x}_{i,\mathcal{M}}} \left[\frac{1}{\Delta t_i} Y_{i+1} \Delta W_{i+1} + \frac{1}{2} f(t_{i+1}, \mathbb{X}_{i+1}) \Delta W_{i+1} \right] + \frac{R_\theta^{Z_i}}{\Delta t_i} + R_{\text{tr}}^{Z_i},\\
&\hat{y}_{i,\mathcal{M}} = E_i^{\hat{x}_{i,\mathcal{M}}} \left[Y_{i+1} + \frac{\Delta t_i}{2} f(t_{i+1}, \mathbb{X}_{i+1}) \right] + \frac{\Delta t_i}{2} \hat{f}_{i,\mathcal{M}} + R_\theta^{Y_i} + R_{\text{impl}}^{Y_i} + R_{\text{tr}}^{Y_i},
\end{aligned}$$

where $E_i^{\hat{x}_{i,\mathcal{M}}}[\mathcal{Y}]$ denotes calculated conditional expectation $E[\mathcal{Y}|X = \hat{x}_{i,\mathcal{M}}]$ using the constructed regression tree with the samples of $\mathcal{Y}$. For example, using samples of the predictor variable X_i (which are $\hat{x}_{i,\mathcal{M}}$) and samples of the response variable $\frac{1}{\Delta t_i}Y_{i+1}\Delta W_{i+1} + \frac{1}{2}f(t_{i+1}, \mathbb{X}_{i+1})\Delta W_{i+1}$ (which are $\frac{1}{\Delta t_i}\hat{y}_{i+1,\mathcal{M}}\Delta\hat{w}_{i+1,\mathcal{M}} + \frac{1}{2}\hat{f}_{i+1,\mathcal{M}}\Delta\hat{w}_{i+1,\mathcal{M}}$) we construct a regression tree. Then, $E_i^{\hat{x}_{i,\mathcal{M}}}[\frac{1}{\Delta t_i}Y_{i+1}\Delta W_{i+1} + \frac{1}{2}f(t_{i+1}, \mathbb{X}_{i+1})\Delta W_{i+1}]$ means the determined value of $E[\frac{1}{\Delta t_i}Y_{i+1}\Delta W_{i+1} + \frac{1}{2}f(t_{i+1}, \mathbb{X}_{i+1})\Delta W_{i+1}|X = \hat{x}_{i,\mathcal{M}}]$ using the constructed tree. Note that, at the initial time $t = 0$, we have $\hat{x}_{0,\mathcal{M}} = x_0$ for $\mathcal{M} = 1, \cdots, M$.

From the errors (27) and (36) we can assume that the approximation error of the tree-based approach is approximately $1/\sqrt{n_2}$ for a large number $n_2 = \frac{M}{2}$, which is the number of samples in L_2 as introduced above. Theoretically, the regression error can be neglected by choosing sufficiently high n_2, namely M. However, the tree-based approach is computationally not that efficient for a quite high value M. For this our idea is to split a quite large set of samples into several small sets of samples, e.g., we can split a set of 20000 samples into 10 sets of 2000 samples. The major reason is that the many times tree-based computations for a small sample number are still more efficient than one computation for a large sample number. We observe, from $t_{N_T} \to t_1$ in the proposed scheme, the samples of $Y_1^{\Delta_t}$ and $Z_1^{\Delta_t}$ are generated backward iteratively starting from the samples of $Y_{N_T}^{\Delta_t}$ and $Z_{N_T}^{\Delta_t}$. When splitting the samples, this procedure can be seen as the projection of samples from $t_{N_T} \to t_1$ but in different groups. Moreover, for the step $t_1 \to t_0$, one has a constant as the predictor variable, namely $W_0 = 0$ for the BSDE or $X_0 = x_0$ for the FBSDE. In fact, in the case of constant predictor, the computation can be done rapidly. We know that the quality of approximations for $Y_0^{\Delta_t}$ and $Z_0^{\Delta_t}$ relies directly on the samples of $Y_1^{\Delta_t}$ and $Z_1^{\Delta_t}$. Our numerical results show that the splitting error of samples projection from $t_{N_T} \to t_1$ could be neglected.

Consequently, we propose to split a large sample size into a few groups of small-size samples at t_{N_T}, for each group we generate backward iteratively the samples for $Y_1^{\Delta_t}$ and $Z_1^{\Delta_t}$[1]. Then, at t_1 we combine the samples of $Y_1^{\Delta_t}$ and $Z_1^{\Delta_t}$ from all groups, which are used as the samples of response variables for the last step $t_1 \to t_0$, whereas the predictor variable is a constant as mentioned already. Note that in the analysis above we have considered a linear regression model, i.e., the proposed scheme is designed to the linear (F)BSDEs.

We summarize our algorithm to solve the FBSDEs as follows.

- Generate M samples and split them into M_g different groups, the sample number in each group is $G = M/M_g$.

[1] Theoretically, the projection of samples in the different groups can be done parallelly. However, the parallelization is not considered in this work.

- For each group, namely $\mathcal{M} = 1, \cdots, M_g$, compute

$$\begin{aligned}
&\hat{y}_{N_T,\mathcal{M}} = g(\hat{x}_{N_T,\mathcal{M}}),\ \hat{z}_{N_T,\mathcal{M}} = g_x(\hat{x}_{N_T,\mathcal{M}}),\\
&\textbf{For } i = N_T - 1, \cdots, 1\ ,\ \mathcal{M} = 1, \cdots, M_g:\\
&\hat{z}_{i,\mathcal{M}} = E_i^{\hat{x}_{i,\mathcal{M}}}\left[\frac{1}{\Delta t_i} Y_{i+1}^{\Delta_t} \Delta W_{i+1} + \frac{1}{2} f(t_{i+1}, \mathbb{X}_{i+1}^{\Delta_t}) \Delta W_{i+1}\right],\\
&\hat{y}_{i,\mathcal{M}} = E_i^{\hat{x}_{i,\mathcal{M}}}\left[Y_{i+1}^{\Delta_t} + \frac{\Delta t_i}{2} f(t_{i+1}, \mathbb{X}_{i+1}^{\Delta_t})\right] + \frac{\Delta t_i}{2} \hat{f}_{i,\mathcal{M}}.
\end{aligned}$$

- Collect all the samples of $(\hat{z}_{1,\mathcal{M}}, \hat{y}_{1,\mathcal{M}})$ for $\mathcal{M} = 1, \cdots, M$ and use all these samples to compute

$$\begin{aligned}
Z_0^{\Delta_t} &= E_0^{x_0}\left[\frac{1}{\Delta t_0} Y_1^{\Delta_t} \Delta W_1 + \frac{1}{2} f(t_1, \mathbb{X}_1^{\Delta_t}) \Delta W_1\right],\\
Y_0^{\Delta_t} &= E_0^{x_0}\left[Y_1^{\Delta_t} + \frac{\Delta t_0}{2} f(t_1, \mathbb{X}_1^{\Delta_t})\right] + \frac{\Delta t_0}{2} \hat{f}_{0,\mathcal{M}}.
\end{aligned}$$

4.4 Error estimates

Suppose that R_{tr} and R_{impl} can be neglected by choosing M and Picard iterations sufficiently high, we consider the discretization errors in the first place. We denote the global errors by

$$\epsilon^{Y_i}(X_i^{\Delta_t}) := Y_i(X_i^{\Delta_t}) - Y_i^{\Delta_t}(X_i^{\Delta_t}), \tag{38}$$

$$\epsilon^{Z_i}(X_i^{\Delta_t}) := Z_i(X_i^{\Delta_t}) - Z_i^{\Delta_t}(X_i^{\Delta_t}), \tag{39}$$

$$\epsilon^{f_i}(X_i^{\Delta_t}) := f(t_i, \mathbb{X}_i) - f(t_i, \mathbb{X}_i^{\Delta_t}). \tag{40}$$

Firstly, we give some remarks concerning related results on the one-step scheme:

- The absolute values of the local errors $R_\theta^{Y_i}$ and $R_\theta^{Z_i}$ in (13) and (16) can be bounded by $C(\Delta t_i)^3$ when $\theta_i = 1/2$, $i = 1, 2, 3$ and by $C(\Delta t_i)^2$ when $\theta_1 = 1/2, \theta_2 = 1, \theta_3 = 1/2$, where C is a constant which can depend on T, a, b and functions f, g in (1), see e.g., [41, 40, 39].

- For notation convenience we might omit the dependency of local and global errors on state of the FBSDEs and the discretization errors of dX_t, namely we assume that $X_i = X_i^{\Delta_t}$.

- For the implicit schemes we will apply Picard iterations which converges for any initial guess when Δt_i is small enough. In the following analysis, we consider the equidistant time discretization Δt.

We start to perform the error analysis for the scheme with $\theta_1 = 1/2, \theta_2 = 1, \theta_3 = 1/2$. The error analysis for other choices of θ_i can be done analogously. For the Z-component $(0 \le i \le N_T - 1)$ we have

$$\epsilon^{Z_i} = E_i^{x_i}[\frac{1}{\Delta t}\epsilon^{Y_{i+1}}\Delta W_{i+1} + \frac{1}{2}\epsilon^{f_{i+1}}\Delta W_{i+1}] + \frac{R_\theta^{Z_i}}{\Delta t}, \tag{41}$$

where the $\epsilon^{f_{i+1}}$ can be bounded using Lipschitz continuity of f by

$$E_i^{x_i}[|\epsilon^{f_{i+1}}|^2] \le E_i^{x_i}[|L(|\epsilon^{Y_{i+1}}| + |\epsilon^{Z_{i+1}}|)|^2] \le 2L^2 E_i^{x_i}[|\epsilon^{Y_{i+1}}|^2 + |\epsilon^{Z_{i+1}}|^2] \tag{42}$$

with Lipschitz constant L. And it holds that

$$|E_i^{x_i}[\epsilon^{Y_{i+1}}\Delta W_{i+1}]|^2 = |E_i^{x_i}[(\epsilon^{Y_{i+1}} - E_i^{x_i}[\epsilon^{Y_{i+1}}])\Delta W_{i+1}]|^2 \le \Delta t(E_i^{x_i}[|\epsilon^{Y_{i+1}}|^2] - |E_i^{x_i}[\epsilon^{Z_{i+1}}]|^2). \tag{43}$$

Consequently, we calculate

$$(\Delta t)^2|\epsilon^{Z_i}|^2 \le 6\Delta t(E_i^{x_i}[|\epsilon^{Y_{i+1}}|^2] - |E_i^{x_i}[\epsilon^{Y_{i+1}}]|^2) + 3L(\Delta t)^3 E_i^{x_i}[|\epsilon^{Y_{i+1}}|^2 + |\epsilon^{Z_{i+1}}|^2] + 6|R_\theta^{Z_i}|^2, \tag{44}$$

where Hölder's inequality is used.

For the Y-component in the implicit scheme we have

$$\epsilon^{Y_i} = E_i^{x_i}[\epsilon^{Y_{i+1}} + \frac{\Delta t}{2}\epsilon^{f_{i+1}}] + \frac{\Delta t}{2}\epsilon^{f_i} + R_\theta^{Y_i}, \tag{45}$$

Again using Lipschitz continuity, this error can be bounded by

$$|\epsilon^{Y_i}| \le |E_i^{x_i}[\epsilon^{Y_{i+1}}]| + \frac{\Delta t L}{2}(|\epsilon^{Y_i}| + |\epsilon^{Z_i}|) + \frac{\Delta t L}{2}E_i^{x_i}[|\epsilon^{Y_{i+1}}| + |\epsilon^{Z_{i+1}}|] + R_\theta^{Y_i}. \tag{46}$$

By the inequality $(a+b)^2 \le a^2 + b^2 + \gamma\Delta t a^2 + \frac{1}{\gamma\Delta t}b^2$ we calculate

$$\begin{aligned}|\epsilon^{Y_i}|^2 \le{}& (1+\gamma\Delta t)|E_i^{x_i}[\epsilon^{Y_{i+1}}]|^2 + \frac{3(\Delta t L)^2}{2}(|\epsilon^{Y_i}|^2 + |\epsilon^{Z_i}|^2) + \frac{3(\Delta t L)^2}{2}(|\epsilon^{Y_{i+1}}|^2 + |\epsilon^{Z_{i+1}}|^2) \\ &+ 3|R_\theta^{Y_i}|^2 + \frac{1}{\gamma}\left(\frac{3\Delta t L^2}{2}(|\epsilon^{Y_i}|^2 + |\epsilon^{Z_i}|^2) + \frac{3\Delta t L^2}{2}(|\epsilon^{Y_{i+1}}|^2 + |\epsilon^{Z_{i+1}}|^2) + \frac{3|R_\theta^{Y_i}|^2}{\Delta t}\right).\end{aligned} \tag{47}$$

Theorem 4.1. *Given*

$$E_{N_T-1}^{x_{N_T-1}}[|\epsilon^{Z_{N_T}}|^2] \sim \mathcal{O}((\Delta t)^2), \quad E_{N_T-1}^{x_{N_T-1}}[|\epsilon^{Y_{N_T}}|^2] \sim \mathcal{O}((\Delta t)^2),$$

It holds then

$$E_0^{x_0}\left[|\epsilon^{Y_i}|^2 + \frac{\Delta t}{6}|\epsilon^{Z_i}|^2\right] \le Q(\Delta t)^2, \quad 0 \le i \le N_T - 1, \tag{48}$$

where Q is a constant which only depend on T, f, g and a, b in (1).

Proof. By combining both (44) and (47) we straightforwardly obtain

$$
\begin{aligned}
E_i^{x_i}[|\epsilon^{Y_i}|^2] + \frac{\Delta t}{6}E_i^{x_i}[|\epsilon^{Z_i}|^2] \leq (1+\gamma\Delta t)|E_i^{x_i}[\epsilon^{Y_{i+1}}]|^2 + \frac{3(\Delta t L)^2}{2}(E_i^{x_i}|\epsilon^{Y_i}|^2 + E_i^{x_i}|\epsilon^{Z_i}|^2) \\
+ \frac{3(\Delta t L)^2}{2}(E_i^{x_i}|\epsilon^{Y_{i+1}}|^2 + E_i^{x_i}|\epsilon^{Z_{i+1}}|^2) + 3E_i^{x_i}[|R_\theta^{Y_i}|^2] \\
+ \frac{1}{\gamma}\left(\frac{3\Delta t L^2}{2}(E_i^{x_i}|\epsilon^{Y_i}|^2 + E_i^{x_i}|\epsilon^{Z_i}|^2) + \frac{3\Delta t L^2}{2}(E_i^{x_i}|\epsilon^{Y_{i+1}}|^2 + E_i^{x_i}|\epsilon^{Z_{i+1}}|^2) + \frac{3E_i^{x_i}[|R_\theta^{Y_i}|^2]}{\Delta t}\right) \\
+ (E_i^{x_i}[|\epsilon^{Y_{i+1}}|^2] - |E_i^{x_i}[\epsilon^{Y_{i+1}}]|^2) + \frac{L}{2}(\Delta t)^2 E_i^{x_i}[|\epsilon^{Y_{i+1}}|^2 + |\epsilon^{Z_{i+1}}|^2] + \frac{E_i^{x_i}[|R_\theta^{Z_i}|^2]}{\Delta t}
\end{aligned}
$$

which implies

$$
\begin{aligned}
&\left(1 - \frac{3(\Delta t L)^2}{2} - \frac{3\Delta t L^2}{2\gamma}\right)E_i^{x_i}[|\epsilon^{Y_i}|^2] + \left(\frac{\Delta t}{6} - \frac{3(\Delta t L)^2}{2} - \frac{3\Delta t L^2}{2\gamma}\right)E_i^{x_i}[|\epsilon^{Z_i}|^2] \\
&\leq \left(1 + \gamma\Delta t + 2(\Delta t L)^2 + \frac{3\Delta t L^2}{2\gamma}\right)E_i^{x_i}[|\epsilon^{Y_{i+1}}|^2] + \left(2(\Delta t L)^2 + \frac{3\Delta t L^2}{2\gamma}\right)E_i^{x_i}[|\epsilon^{Z_{i+1}}|^2] \\
&+ 3E_i^{x_i}[|R_\theta^{Y_i}|^2] + \frac{3E_i^{x_i}[|R_\theta^{Y_i}|^2]}{\gamma\Delta t} + \frac{E_i^{x_i}[|R_\theta^{Z_i}|^2]}{\Delta t}.
\end{aligned}
$$

We choose γ such that $\frac{\Delta t}{6} - \frac{3\Delta t L^2}{2\gamma} \geq \frac{3\Delta t L^2}{2\gamma}(i.e. \gamma \geq 18L^2)$, by which the latter inequality can be rewritten as

$$
\begin{aligned}
&\left(1 - \frac{3(\Delta t L)^2}{2} - \frac{3\Delta t L^2}{2\gamma}\right)E_i^{x_i}[|\epsilon^{Y_i}|^2] + \left(\frac{\Delta t}{6} - \frac{3(\Delta t L)^2}{2} - \frac{3\Delta t L^2}{2\gamma}\right)E_i^{x_i}[|\epsilon^{Z_i}|^2] \\
&\leq \left(1 + \gamma\Delta t + 2(\Delta t L)^2 + \frac{3\Delta t L^2}{2\gamma}\right)E_i^{x_i}[|\epsilon^{Y_{i+1}}|^2] + \left(2(\Delta t L)^2 + \frac{\Delta t}{6} - \frac{3\Delta t L^2}{2\gamma}\right)E_i^{x_i}[|\epsilon^{Z_{i+1}}|^2] \\
&+ 3E_i^{x_i}[|R_\theta^{Y_i}|^2] + \frac{3E_i^{x_i}[|R_\theta^{Y_i}|^2]}{\gamma\Delta t} + \frac{E_i^{x_i}[|R_\theta^{Z_i}|^2]}{\Delta t},
\end{aligned}
$$

which implies

$$
\begin{aligned}
E_i^{x_i}[|\epsilon^{Y_i}|^2] + \frac{\Delta t}{6}E_i^{x_i}[|\epsilon^{Z_i}|^2] \leq \frac{1+C\Delta t}{1-C\Delta t}\left(E_i^{x_i}[|\epsilon^{Y_{i+1}}|^2] + \frac{\Delta t}{6}E_i^{x_i}[|\epsilon^{Z_{i+1}}|^2]\right) \\
+ 3E_i^{x_i}[|R_\theta^{Y_i}|^2] + \frac{E_i^{x_i}[|R_\theta^{Y_i}|^2]}{6L^2\Delta t} + \frac{E_i^{x_i}[|R_\theta^{Z_i}|^2]}{\Delta t}.
\end{aligned}
$$

By induction, we obtain then

$$\begin{aligned}
E_i^{x_i}[|\epsilon^{Y_i}|^2] + \frac{\Delta t}{6} E_i^{x_i}[|\epsilon^{Z_i}|^2] \leq & \left(\frac{1+C\Delta t}{1-C\Delta t}\right)^{N_T-i} \left(E_{N_T-1}^{x_{N_T-1}}[|\epsilon^{Y_{N_T}}|^2] + \frac{\Delta t}{6} E_{N_T-1}^{x_{N_T-1}}[|\epsilon^{Z_{N_T}}|^2]\right) \\
& + \sum_{j=i}^{N_T-1} \left(\frac{1+C\Delta t}{1-C\Delta t}\right)^{j-i} \left(3E_i^{x_i}[|R_\theta^{Y_j}|^2] + \frac{E_i^{x_i}[|R_\theta^{Y_j}|^2]}{6L^2\Delta t} + \frac{E_i^{x_i}[|R_\theta^{Z_j}|^2]}{\Delta t}\right) \\
& \leq \exp(2CT) \left(E_{N_T-1}^{x_{N_T-1}}[|\epsilon^{Y_{N_T}}|^2] + \frac{\Delta t}{6} E_{N_T-1}^{x_{N_T-1}}[|\epsilon^{Z_{N_T}}|^2]\right) \\
& + \exp(2CT) \sum_{j=i}^{N_T-1} \left(3E_i^{x_i}[|R_\theta^{Y_j}|^2] + \frac{E_i^{x_i}[|R_\theta^{Y_j}|^2]}{6L^2\Delta t} + \frac{E_i^{x_i}[|R_\theta^{Z_j}|^2]}{\Delta t}\right).
\end{aligned}$$

With the known conditions and bounds of the local errors we complete the proof. □

5 Numerical experiments

In this section we use some numerical examples to show the accuracy of our methods for solving the (F)BSDEs. As already introduced above, N_T and M are the total discrete time steps and sampling number, respectively. For all the examples, we consider an equidistant time and perform 20 Picard iterations. We ran the algorithms 10 times independently and take average value of absolute error, whereas the two different seeds are used for every five simulations. Numerical experiments were performed with an Intel(R) Core(TM) i5-8500 CPU @ 3.00GHz and 15 GB RAM.

5.1 Example of BSDE

The first BSDE we consider is

$$\begin{cases} -dY_t = (\frac{Y_t}{2} - \frac{Z_t}{2})\, dt - Z_t\, dW_t, \\ Y_T = \sin(W_T + \frac{T}{2}), \end{cases} \tag{49}$$

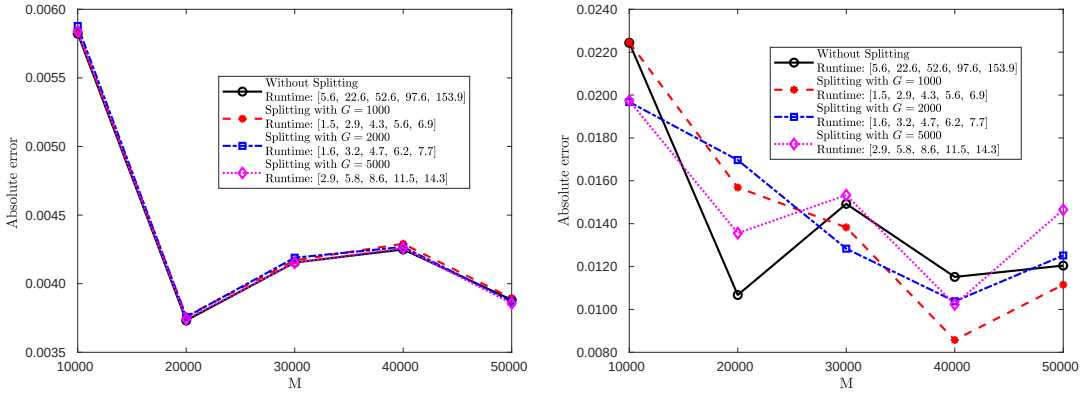
with the analytical solution

$$\begin{cases} Y_t = \sin(W_t + \frac{t}{2}), \\ Z_t = \cos(W_t + \frac{t}{2}). \end{cases} \tag{50}$$

The generator f is highly oscillatory function and contains the component Z_t. For this example we set $T = \frac{1}{2}$, the analytical solution of (Y_0, Z_0) is $(0, 1)$.

Firstly, in order to see the computational acceleration by using the samples-splitting introduced above, we compare the scheme between using and not using the samples-splitting in Figure 1. Since the algorithm without splitting are slow, we thus compare them up to the sample size 50000, whereas N_T is fixed to 10. Let $Y_{0,k}^{\Delta_t}$ and $Z_{0,k}^{\Delta_t}$ denote the result on the k-th run of the algorithm, $k = 1, \cdots, 10$, the approximations read

as $Y_0^{\Delta_t} = \frac{1}{10}\sum_{k=1}^{10} Y_{0,k}^{\Delta_t}$ and $Z_0^{\Delta_t} = \frac{1}{10}\sum_{k=1}^{10} Z_{0,k}^{\Delta_t}$. In our tests we consider average of the absolute errors, i.e., $\frac{1}{10}\sum_{k=1}^{10} |Y_{0,k}^{\Delta_t} - Y_0|$ and $\frac{1}{10}\sum_{k=1}^{10} |Z_{0,k}^{\Delta_t} - Z_0|$. We see that there



(a) Absolute error: $\frac{1}{10}\sum_{k=1}^{10} |Y_{0,k}^{\Delta_t} - Y_0|$ (b) Absoulte error: $\frac{1}{10}\sum_{k=1}^{10} |Z_{0,k}^{\Delta_t} - Z_0|$

Figure 1: Comparison of absolute errors among schemes not using and using sample-splitting ($\theta_1 = \frac{1}{2}, \theta_2 = 1, \theta_3 = \frac{1}{2}$) with different sample sizes of group (G), the average runtimes are given in seconds.

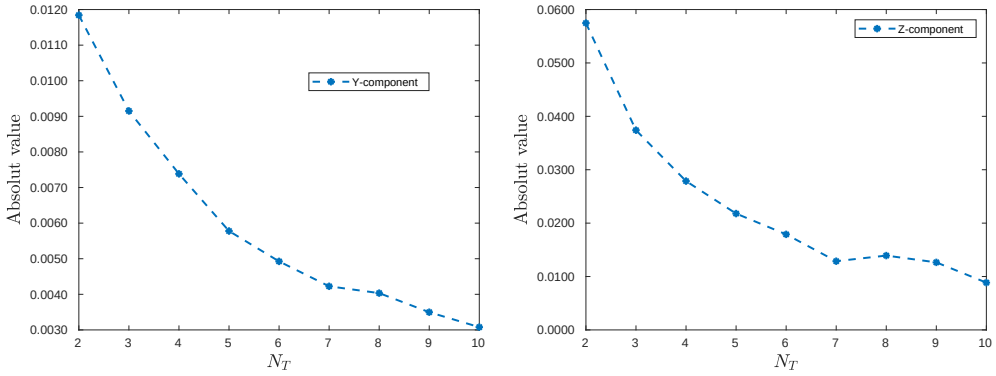
are no considerable differences between using and not using the sample-splitting for approximating Y_0. And the approximations of Z_0 with the sample-splitting against M converge in a very stable fashion. Furthermore, the application of sample-splitting allows a much efficient computation, e.g., when $M = 50000$, the scheme without splitting used 153.9 seconds while it used only 6.9 seconds by using the splitting with $G = 1000$. In the remaining of this paper we perform all the schemes always using the splitting with $G = 1000$, unless otherwise specified.

Next we study the influence of M on the error. This is a good example to test performances of the tree-based approach based on different schemes by choosing θ_i's values, since the generator f is linear and the exact solutions of (Y_T, Z_T) are known. For this we fix the number of steps to 2 and test all possible values of θ_i. We find that the explicit schemes for $\theta_3 = 0, \theta_2 = 1, \theta_1 = 1/2, 1$ and the implicit schemes for $\theta_3 = 1/2, 1, \theta_2 = 1, \theta_1 = 1/2, 1$ can converge for a small M, all others need a very large number M. As an example we report the absolute errors $\frac{1}{10}\sum_{k=1}^{10} |Y_{0,k}^{\Delta_t} - Y_0|$, $\frac{1}{10}\sum_{k=1}^{10} |Z_{0,k}^{\Delta_t} - Z_0|$ and the empirical standard deviations $\sqrt{\frac{1}{9}\sum_{k=1}^{10} |Y_{0,k}^{\Delta_t} - Y_0^{\Delta_t}|^2}$, $\sqrt{\frac{1}{9}\sum_{k=1}^{10} |Z_{0,k}^{\Delta_t} - Z_0^{\Delta_t}|^2}$ for some chosen schemes in Table 1. We observe, even for $N_T = 2$, the second-order scheme ($\theta_1 = \frac{1}{2}, \theta_2 = \frac{1}{2}, \theta_3 = \frac{1}{2}$) converges only for a quite large M. In particular, the error $|Z_0 - Z_0^{\Delta_t}|$ approaches the convergence value first from $M = 200000$. Since error for the scheme ($\theta_1 = \frac{1}{2}, \theta_2 = 1, \theta_3 = \frac{1}{2}$) is smallest of all the schemes, which converge for a small value of M. This is the reason why we will consider the scheme for ($\theta_1 = \frac{1}{2}, \theta_2 = 1, \theta_3 = \frac{1}{2}$) for the following analysis and almost all the examples. To take this a step further, we fix now $M = 200000$ and plot the absolute error against the

N_T	2						
M	2000	5000	10000	50000	100000	200000	300000
				$\frac{1}{10}\sum_{k=1}^{10}\lvert Y_{0,k}^{\Delta_t} - Y_0\rvert$			
$(\theta_1 = 1, \theta_2 = 1, \theta_3 = \frac{1}{2})$	0.0254	0.0220	0.0251	0.0248	0.0247	0.0244	0.0246
standard deviation	0.0193	0.0147	0.0099	0.0023	0.0021	0.0016	8.3067e-04
$(\theta_1 = \frac{1}{2}, \theta_2 = 1, \theta_3 = \frac{1}{2})$	0.0177	0.0128	0.0125	0.0123	0.0121	0.0118	0.0120
standard deviation	0.0196	0.0151	0.0102	0.0023	0.0021	0.0017	9.1699e-04
$(\theta_1 = \frac{1}{2}, \theta_2 = \frac{1}{2}, \theta_3 = \frac{1}{2})$	0.0169	0.0129	0.0073	0.0020	0.0019	0.0017	7.2826e-04
standard deviation	0.0197	0.0162	0.0113	0.0025	0.0022	0.0019	0.0011
$(\theta_1 = 1, \theta_2 = 1, \theta_3 = 1)$	0.0171	0.0124	0.0070	0.0022	0.0020	0.0019	0.0017
standard deviation	0.0197	0.0159	0.0110	0.0024	0.0021	0.0019	0.0010
				$\frac{1}{10}\sum_{k=1}^{10}\lvert Z_{0,k}^{\Delta_t} - Z_0\rvert$.			
$(\theta_1 = 1, \theta_2 = 1, \theta_3 = \frac{1}{2})$	0.1221	0.1210	0.1190	0.1166	0.1187	0.1197	0.1201
standard deviation	0.0303	0.0199	0.0147	0.0079	0.0037	0.0032	0.0025
$(\theta_1 = \frac{1}{2}, \theta_2 = 1, \theta_3 = \frac{1}{2})$	0.0579	0.0578	0.0562	0.0537	0.0561	0.0575	0.0578
standard deviation	0.0319	0.0235	0.0165	0.0081	0.0042	0.0036	0.0027
$(\theta_1 = \frac{1}{2}, \theta_2 = \frac{1}{2}, \theta_3 = \frac{1}{2})$	0.0991	0.0453	0.0295	0.0158	0.0144	0.0074	0.0058
standard deviation	0.1111	0.0550	0.0312	0.0171	0.0173	0.0077	0.0060
$(\theta_1 = 1, \theta_2 = 1, \theta_3 = 1)$	0.1114	0.1111	0.1095	0.1072	0.1095	0.1107	0.1112
standard deviation	0.0300	0.0230	0.0151	0.0079	0.0042	0.0035	0.0028

Table 1: Comparison of absolute errors for $N_T = 2$ against the sample size M.

number of steps in Figure 2 when using $(\theta_1 = \frac{1}{2}, \theta_2 = 1, \theta_3 = \frac{1}{2})$. We see that the scheme

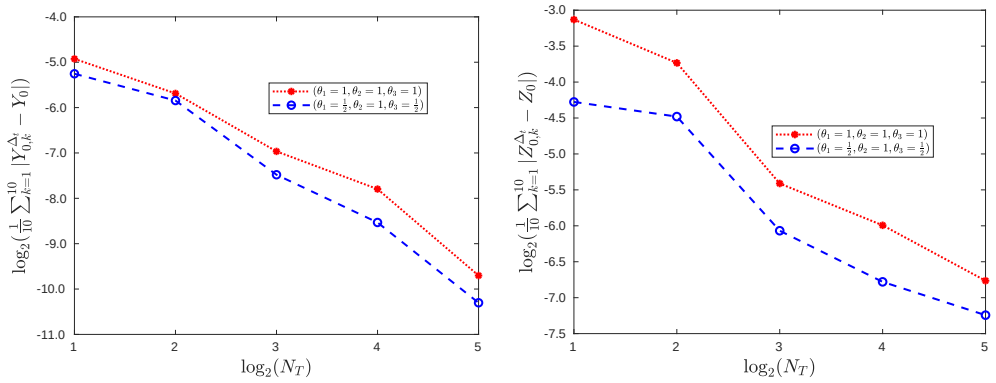


(a) Absolute error: $\frac{1}{10}\sum_{k=1}^{10}|Y_{0,k}^{\Delta_t} - Y_0|$ (b) Absoulte error: $\frac{1}{10}\sum_{k=1}^{10}|Z_{0,k}^{\Delta_t} - Z_0|$

Figure 2: Comparison of absolute errors against the number of steps N_T for $\theta_1 = \frac{1}{2}, \theta_2 = 1, \theta_3 = \frac{1}{2}$, and $M = 200000$.

converges meaningfully.

For the convergence with respect to the time step we refer to Figure 3, where we plot $\log_2\left(\frac{1}{10}\sum_{k=1}^{10}|Y_{0,k}^{\Delta_t} - Y_0|\right)$ and $\log_2\left(\frac{1}{10}\sum_{k=1}^{10}|Z_{0,k}^{\Delta_t} - Z_0|\right)$ with respect to $\log_2(N_T)$. To estimate the convergence rate with respect to the time step sizes we adjust roughly sample sizes M according to the time partitions, i.e., larger M for smaller dt, the used sample sizes M are listed in Table 2. The results shown in Table 2 and Figure 1 are



(a) Y-component (b) Z-component

Figure 3: The plots of average of the absolute values with respect to $\log_2(N_T)$.

consistent with the conclusions in Theorem 4.1. Actually, when we use the absolute value $|Y_0 - Y_0^{\Delta_t}|$ and $|Z_0 - Z_0^{\Delta_t}|$, where $Y_0^{\Delta_t} = \frac{1}{10}\sum_{k=1}^{10} Y_{0,k}^{\Delta_t}$ and $Z_0^{\Delta_t} = \frac{1}{10}\sum_{k=1}^{10} Z_{0,k}^{\Delta_t}$. the obtained numerical convergence rates in Table 2 are higher.

N_T	2	4	8	16	32	
M	1000	2000	20000	100000	300000	
			$\frac{1}{10}\sum_{k=1}^{10}\lvert Y_{0,k}^{\Delta_t} - Y_0\rvert$			CR
$(\theta_1 = 1, \theta_2 = 1, \theta_3 = \frac{1}{2})$	0.0329	0.0194	0.0080	0.0045	0.0012	1.17
standard deviation	0.0276	0.0224	0.0051	0.0020	9.8927e-04	
$(\theta_1 = \frac{1}{2}, \theta_2 = 1, \theta_3 = \frac{1}{2})$	0.0262	0.0174	0.0056	0.0027	7.9174e-04	1.28
standard deviation	0.0279	0.0226	0.0052	0.0020	9.9436e-04	
			$\frac{1}{10}\sum_{k=1}^{10}\lvert Z_{0,k}^{\Delta_t} - Z_0\rvert$			CR
$(\theta_1 = 1, \theta_2 = 1, \theta_3 = \frac{1}{2})$	0.1142	0.0752	0.0235	0.0157	0.0092	0.95
standard deviation	0.0273	0.0271	0.0193	0.0078	0.0055	
$(\theta_1 = \frac{1}{2}, \theta_2 = 1, \theta_3 = \frac{1}{2})$	0.0516	0.0448	0.0149	0.0091	0.0066	0.82
standard deviation	0.0285	0.0265	0.0180	0.0086	0.0056	
			average runtime in seconds			
$(\theta_1 = 1, \theta_2 = 1, \theta_3 = \frac{1}{2})$	0.1	0.5	2.3	23.9	147.0	
$(\theta_1 = \frac{1}{2}, \theta_2 = 1, \theta_3 = \frac{1}{2})$	0.1	0.2	2.3	25.5	144.4	

Table 2: Absolute errors, standard deviations, average runtimes in seconds and convergence rates (CR) for the Example of BSDE (49).

5.2 Example of FBSDE

In the remaining examples we always use the scheme for $(\theta_1 = \frac{1}{2}, \theta_2 = 1, \theta_3 = \frac{1}{2})$, unless otherwise specified. For the example of FBSDE we compute the price of a European call option $V(t, S_t)$ via a FBSDE where the underlying asset as the forward process, which follows a geometric Brownian motion given by

$$dS_t = \mu S_t\, dt + \sigma S_t dW_t. \tag{51}$$

It is well-known that the exact solution is analytically given in [6], namely Black-Scholes price. We assume that the asset pays dividends with the rate d. As introduced in [20], the corresponding FBSDE for the price of option can be derived by setting up a self-financing portfolio Y_t, which consists of π_t assets and $Y_t - \pi_t$ bonds with risk-free return rate r, which reads

$$\begin{cases} \quad dS_t = \mu S_t\, dt + \sigma S_t\, dW_t, \\ -dY_t = \left(-rY_t - \frac{\mu - r + d}{\sigma} Z_t\right) dt - Z_t\, dW_t, \\ \quad Y_T = \xi = \max(S_T - K, 0). \end{cases} \tag{52}$$

Y_t corresponds to the option value $V(t, S_t)$, Z_t is related to the hedging strategy, $Z_t = \sigma S_t \pi_t = \sigma S_t \frac{\partial V}{\partial S}$.

For S^{Δ_t}, we simulate the forward process dS_t by using Euler-Method, although its analytical solution is available. Note that, although the function $g(x) = \max(x, 0)$ is not differentiable in this example, we still use it to generate samples for (Y_T, Z_T) in our

N_T	2	4	8	12	16	20	
M	2000	10000	30000	60000	100000	250000	
			$\frac{1}{10}\sum_{k=1}^{10}\frac{\lvert Y_{0,k}^{\Delta_t}-Y_0\rvert}{\lvert Y_0\rvert}$				CR
$(\theta_1=\frac{1}{2},\theta_2=1,\theta_3=\frac{1}{2})$	0.0230	0.0091	0.0052	0.0038	0.0027	0.0011	1.15
standard deviation	0.1311	0.0501	0.0279	0.0255	0.0143	0.0055	
			$\frac{1}{10}\sum_{k=1}^{10}\frac{\lvert Z_{0,k}^{\Delta_t}-Z_0\rvert}{\lvert Z_0\rvert}$				CR
$(\theta_1=\frac{1}{2},\theta_2=1,\theta_3=\frac{1}{2})$	0.0492	0.0246	0.0151	0.0113	0.0091	0.0056	0.86
standard deviation	0.6187	0.3161	0.1950	0.1307	0.1218	0.0708	
				average runtime in seconds			
	0.1	0.5	3.1	9.5	21.3	66.7	

Table 3: Relative errors, standard deviations, average runtimes in seconds and convergence rates for the Black-Scholes model.

tree-based approaches:

$$\begin{cases} Y_{N_T,\mathcal{M}}^{\Delta_t} = \max(S_{N_T,\mathcal{M}}^{\Delta_t} - K, 0), \\ Z_{N_T,\mathcal{M}}^{\Delta_t} = \begin{cases} \sigma S_{N_T,\mathcal{M}}^{\Delta_t} & \text{when } S_{N_T,\mathcal{M}}^{\Delta_t} > K \\ 0, \quad \text{otherwise} \end{cases} \end{cases} \tag{53}$$

where $\mathcal{M} = 1, \cdots, M$. For the comparison purpose, we take the parameter values, which are used in [37]

$$K = S_0 = 100,\ r = 0.03,\ \mu = 0.05,\ d = 0.04,\ \sigma = 0.2,\ T = 0.33 \tag{54}$$

with the exact solution $(Y_0, Z_0) = (4.3671, 10.0950)$. For the following financial applications we consider the relative error $\frac{1}{10}\sum_{k=1}^{10}\frac{\lvert Y_{0,k}^{\Delta_t}-Y_0\rvert}{\lvert Y_0\rvert}$, see Table 3. Note that we have in this example the simulation error by using the Euler-Method for the forward process dS_t. Furthermore, terminal condition for Y_T is not differentiable at the point $S_T = K$, which leads to a jump for the component Z_T at that points. Although that, without any smoothing techniques we still obtain the satisfactory results using the tree-based approach.

5.3 Example of two-dimensional FBSDE

For the two-dimensional FBSDE we consider the Heston stochastic volatility model [17] which reads

$$\begin{cases} dS_t = \mu S_t\, dt + \sqrt{\nu_t} S_t\, dW_t^S, \\ d\nu_t = \kappa_\nu(\mu_\nu - \nu_t)\, dt + \sigma_\nu \sqrt{\nu_t}\, dW_t^\nu, \\ dW_t^S dW_t^\nu = \rho\, dt, \end{cases} \tag{55}$$

where S_t is the spot price of the underlying asset, ν_t is the volatility. It is well-known that the Heston model (55) can be reformulated as

$$d\mathbf{X}_t = \begin{pmatrix} d\nu_t \\ dS_t \end{pmatrix} = \begin{pmatrix} \kappa_\nu(\mu_\nu - \nu_t) \\ \mu S_t \end{pmatrix} dt + \begin{pmatrix} \sigma_\nu\sqrt{\nu_t} & 0 \\ S_t\rho\sqrt{\nu_t} & S_t\sqrt{1-\rho^2}\sqrt{\nu_t} \end{pmatrix} \begin{pmatrix} d\tilde{W}_t^\nu \\ d\tilde{W}_t^S \end{pmatrix}, \quad (56)$$

where $\tilde{W}_t^\nu$ and $\tilde{W}_t^S$ are independent Brownian motions. To find the FBSDE form for the Heston Model we consider the following self-financing strategy

$$dY_t = a_t\, dU(t,\nu_t,S_t) + b_t dS_t + c_t dP_t, \quad (57)$$

$$= a_t\, dU(t,\nu_t,S_t) + b_t dS_t + \frac{(Y_t - a_t U(t,\nu_t,S_t) - b_t S_t)}{P_t} dP_t, \quad (58)$$

where $U(t,\nu_t,S_t)$ is the value of another option for hedging volatility, $dP_t = rP_t\, dt$ is used for the risk-free asset, a_t, b_t and c_t are numbers of the option, underlying asset and risk-free asset, respectively. We assume that

$$dU(t,\nu_t,S_t) = \eta(t,\nu_t,S_t)dt, \quad (59)$$

which can be substituted into (58) to obtain

$$-dY_t = \left(a_t r U(t,\nu_t,S_t) - a_t\eta(t,\nu_t,S_t) - \frac{(\mu - r)}{\sqrt{1-\rho^2}\sqrt{\nu_t}} Z_{t,2} - rY_t \right) dt - \mathbf{Z}_t \begin{pmatrix} d\tilde{W}_t^\nu \\ d\tilde{W}_t^S \end{pmatrix} \quad (60)$$

with

$$\mathbf{Z}_t = (Z_t^1, Z_t^2) = \left(a_t\sigma_\nu\sqrt{\nu_t} + b_t S_t\rho\sqrt{\nu_t}, b_t S_t\sqrt{1-\rho^2}\sqrt{\nu_t} \right). \quad (61)$$

In the Heston model [17], the market price of the volatility risk is assumed to $\lambda\nu_t$. With the notations used in (56), the Heston pricing PDE including λ reads

$$\frac{\partial V}{\partial t} + rS\frac{\partial V}{\partial S} + (\kappa_\nu(\mu_\nu - \nu) - \lambda\nu)\frac{\partial V}{\partial \nu} + \frac{1}{2}\nu S^2\frac{\partial^2 V}{\partial S^2} + \rho\sigma_\nu\nu S\frac{\partial^2 V}{\partial S\partial\nu} + \frac{1}{2}\sigma_\nu^2\nu\frac{\partial^2 V}{\partial\nu^2} - rV = 0. \quad (62)$$

The solution of the FBSDE (60) is exactly the solution of the Heston PDE (62) by choosing $rU(t,\nu_t,S_t) - \eta(t,\nu_t,S_t) \equiv -\lambda\nu_t$. The equations (60) and (61) can thus be reformulated as

$$-dY_t = \left(-a_t\lambda\nu_t - \frac{(\mu - r)}{\sqrt{1-\rho^2}\sqrt{\nu_t}} Z_t^2 - rY_t \right) dt - \mathbf{Z}_t \begin{pmatrix} d\tilde{W}_t^\nu \\ d\tilde{W}_t^S \end{pmatrix} \quad (63)$$

$$= \left(-\frac{\lambda\sqrt{\nu_t}}{\sigma_\nu} Z_t^1 + \left(\frac{\rho\lambda\sqrt{\nu_t}}{\sqrt{1-\rho^2}\sigma_\nu} - \frac{(\mu - r)}{\sqrt{1-\rho^2}\sqrt{\nu_t}} \right) Z_t^2 - rY_t \right) dt - \mathbf{Z}_t \begin{pmatrix} d\tilde{W}_t^\nu \\ d\tilde{W}_t^S \end{pmatrix} \quad (64)$$

with $\mathbf{Z}_t$ defined in (61). Note that the generator in this example can be not Lipschitz continuous. The European-style option can be replicated by hedging this portfolio. We

consider e.g., a call option whose value at time t is same to the portfolio value Y_t, and $Y_T = \xi = \max(S_T - K, 0)$. Hence, Y_t is the Heston option value $V(t, \nu_t, S_t)$, $\mathbf{Z}_t$ presents the hedging strategies, where $Z_t^1 = \frac{\partial V}{\partial \nu}\sigma_\nu\sqrt{\nu_t} + \frac{\partial V}{\partial S}S_t\rho\sqrt{\nu_t}$ and $Z_t^2 = \frac{\partial V}{\partial S}S_t\sqrt{1-\rho^2}\sqrt{\nu_t}$. The semi-analytical solution of the Heston model is available, the corresponding Delta hedging $\frac{\partial V}{\partial S}$ can thus be obtained also in a closed form. However, the Vega hedging against volatility risk is defined as the derivative of option value with respect to the volatility ν_t, which is driven by the Cox-Ingersoll-Ross process in the Heston model and thus not analytically available. For this reason we can only consider the approximation of Y-component, namely the option price in the Heston model. The parameter values used for this numerical test are

$$\begin{aligned} &K = S_0 = 50,\ r = 0.03,\ \mu = 0.05,\ \lambda = 0,\ T = 0.5,\\ &\nu_0 = \mu_\nu = 0.04,\ \kappa_\nu = 1.9,\ \sigma_\nu = 0.1,\ \rho = -0.7, \end{aligned}$$

which give the exact solution $Y_0 = 3.1825$. The forward processes dS_t and $d\nu_t$ are simulated using the Euler-method, for the final values at the maturity T we take

$$\begin{cases} Y_{N_T,\mathcal{M}}^{\Delta_t} = \max(S_{N_T,\mathcal{M}}^{\Delta_t} - K, 0), \\ Z_{N_T,\mathcal{M}}^{1,\Delta_t} = \begin{cases} S_{N_T,\mathcal{M}}^{\Delta_t}\rho\sqrt{\nu_{N_T,\mathcal{M}}^{\Delta_t}} & \text{when } S_{N_T,\mathcal{M}}^{\Delta_t} > K \\ 0, \quad \text{otherwise} \end{cases} \\ Z_{N_T,\mathcal{M}}^{2,\Delta_t} = \begin{cases} S_{N_T,\mathcal{M}}^{\Delta_t}\sqrt{1-\rho^2}\sqrt{\nu_{N_T,\mathcal{M}}^{\Delta_t}} & \text{when } S_{N_T,\mathcal{M}}^{\Delta_t} > K \\ 0, \quad \text{otherwise} \end{cases} \end{cases} \tag{65}$$

where $\mathcal{M} = 1, \cdots, M$. The corresponding relative errors are reported in Table 4. We obtain quite accurate approximation for the Heston option price by solving the two-dimensional FBSDE, although the generator is not Lipschitz continuous. It is well-known that a splitting scheme of the Alternating Direction Implicit (ADI) type has been widely analyzed and applied to efficiently find the numerical solution of a two-dimensional parabolic partial differential equation (PDE). We thus compare our tree-based approach to the Craig-Sneyd (CS) Crank-Nicolson ADI finite difference scheme [9] for solving the Heston model in Table 4. We denote N_S and N_ν as number of points for the stock price and the volatility grid, respectively. The ADI scheme is performed in domain $[0, 2K]$ for S and $[0, 0.5]$ for ν with a uniform grid $N_S = N_\nu = 40$, the time steps N_T are given in Table 4. One can observe that the tree-based approach gives a better at least compatible result.

5.4 Example of high-dimensional FBSDE

It is interesting for us to test performance of tree-based approach in solving high-dimensional FBSDE. For this we consider the pricing problem of Rainbow option [33, 18]. We suppose that D stocks, which are for simplicity assumed to be independent and identically distributed, and driven by

$$dS_{t,d} = \mu S_{t,d}\, dt + \sigma S_{t,d} dW_{t,d}, \quad d = 1, \cdots D, \tag{66}$$

Exact price: 3.1825						
I	The tree-based approach ($\theta_1 = \frac{1}{2}, \theta_2 = 1, \theta_3 = \frac{1}{2}$)					
N_T	2	4	8	16	32	
M	5000	10000	40000	100000	300000	
$\frac{1}{10}\sum_{k=1}^{10}\frac{\lvert Y_{0,k}^{\Delta_t}-Y_0\rvert}{\lvert Y_0\rvert}$	0.0207	0.0115	0.0043	0.0028	0.0015	CR $\approx$0.96
standard deviation	0.0840	0.0307	0.0173	0.0109	0.0051	
average runtime	0.2	0.9	7.4	38.8	241.3	
II	The CS Crank-Nicolson ADI scheme					
N_T	2	4	8	16	32	
$\frac{\lvert Y^{\Delta_t}-Y_0\rvert}{\lvert Y_0\rvert}$	0.0900	0.0103	0.0068	0.0062	**0.0061**	
runtime	0.2	0.7	1.2	2.7	**6.2**	
III	The tree-based approach ($\theta_1 = \frac{1}{2}, \theta_2 = 1, \theta_3 = \frac{1}{2}$)					
N_T	8	8	8	8	8	
M	100	500	1000	5000	10000	
$\frac{1}{10}\sum_{k=1}^{10}\frac{\lvert Y_{0,k}^{\Delta_t}-Y_0\rvert}{\lvert Y_0\rvert}$	0.1181	0.0612	0.0278	0.0162	**0.0051**	
standard deviation	0.4637	0.2137	0.1171	0.0561	**0.0183**	
average runtime	0.1	0.2	0.2	1.0	**1.9**	

Table 4: Relative errors, standard deviations, average runtimes in seconds and convergence rates for the Heston model. Part I: the tree-based approach is used for different values of N_T and M; Part II: the CS Crank-Nicoln ADI finite different scheme is used; Part III: the tree-based approach is used for $N_T = 8$ and different values of M.

where $\sigma > 0$ and $\mu \in \mathbb{R}$. For the terminal condition we take that of a Call on max

$$Y_T = \xi = \max\left(\max_{d=1,\cdots,D}(S_{T,d}) - K, 0\right). \tag{67}$$

The driver f is then defined by

$$f(t,x,y,z) = -ry - \frac{\mu - r}{\sigma}\sum_{d=1}^{D} z_d. \tag{68}$$

In this linear example we take

$$K = S_0 = 100,\ r = 0.04,\ \mu = 0.06,\ T = 0.1.$$

To the best of our knowledge, there is no method available for pricng the high-dimensional Rainbow option, which could allow for a less computational time than direct Monte-Carlo simulation. However, our aim is to show performance of the tree-based approach for pricing a high-dimensional Rainbow option based on the BSDE. Therefore, we compare our approach to the multilevel Monte Carlo method based on Picard approximation proposed in [13]. The reference prices are computed with 7 Picard iterations.

We consider the 10-dimensional pricing problem, i.e., $D = 10$. Firstly, in Table 5 (Part I), we adjust roughly sample sizes M to approximate the convergence rate with respect to the time step sizes. All the relative errors, empirical standard deviation and convergence rate are reported there. The reference price $Y_0 = 10.4689$ is computed by means of the multilevel-Picard approximation method in [13] with 7 Picard iterations, whereas the average runtime are 2249.6 seconds. It is not difficult to see that our results are quite promising, and show that the 10-dimensional problem can be highly effective and accurate approximated using the tree-based approach. The obtained convergence rate of the proposed scheme is 1.9. For a comparison purpose, using the same reference price we report the errors, standard deviations and average rumtimes for the Picard iteration number $\{1, \cdots, 6\}$ using the method in [13] in Table 5 (Part III). To compare the result for the Picard iteration number equals 6 (bold and underlined), in Table 5 (Part II) we show our results for $N_T = 12$ by varying different sample sizes. From our result for $N_T = 12$ and $M = 2000$ (bold and underlined) we see that for this 10-dimensional pricing problem, our scheme is more than 10 times faster than the approximation method in [13]. Note that, in order to see performance of our approach for the problem in which the forward SDE does not exhibit an analytical solution, we simply use the Euler method for dS.

Finally, we test our scheme for the 100-dimensional pricing problem. Note that due to the limitation of memory, we only set $M = 300000$ for $N_T = 20$ in the 100-dimensional case. In Table 6, the average runtime of using the multilevel-Picard method for 100-dimension (2613.9) is not much longer than that (2249.6) in Table 5 for 10-dimension. Especially, by comparing the average runtime in Table 5 in Section 4.3 in [13] for 1-dimensional to that in Table 6 in the same section in [13], it seems that the

Reference price	$Y_0 = 10.4689$ (average runtime 2249.6 seconds)						
I	The tree-based approach ($\theta_1 = \frac{1}{2}, \theta_2 = 1, \theta_3 = \frac{1}{2}$)						
N_T	2	4	8	12	16	20	
M	5000	10000	80000	100000	200000	400000	
$\frac{1}{10}\sum_{k=1}^{10} \frac{\lvert Y_{0,k}^{\Delta_t} - Y_0 \rvert}{\lvert Y_0 \rvert}$	0.0390	0.0195	0.0078	0.0038	0.0013	3.4737e-04	CR ≈1.9
standard deviation	0.0429	0.0356	0.0109	0.0080	0.0045	0.0025	
average runtime in seconds	0.9	4.3	75.7	146.6	602.9	999.2	
II	The tree-based approach ($\theta_1 = \frac{1}{2}, \theta_2 = 1, \theta_3 = \frac{1}{2}$)						
N_T	12	12	12	12	12	12	
M	100	500	1000	2000	5000	10000	
$\frac{1}{10}\sum_{k=1}^{10} \frac{\lvert Y_{0,k}^{\Delta_t} - Y_0 \rvert}{\lvert Y_0 \rvert}$	0.0309	0.0154	0.0100	**0.0061**	0.0059	0.0056	
standard deviation	0.4048	0.1721	0.1326	**0.0762**	0.0552	0.0303	
average runtime in seconds	0.5	0.9	1.6	**3.1**	7.6	15.0	
III	The multilevel Monte Carlo method [13]						
Number of the Picard iteration	1	2	3	4	5	6	
$\frac{1}{10}\sum_{k=1}^{10} \frac{\lvert Y_{0,k}^{\Delta_t} - Y_0 \rvert}{\lvert Y_0 \rvert}$	0.1920	0.2312	0.0759	0.02290	0.0120	**0.0058**	
standard deviation	1.7304	2.8257	0.9796	0.2691	0.1695	**0.0825**	
average runtime in seconds	0.0	0.0	0.0	0.3	3.4	**43.9**	

Table 5: Relative errors, standard deviations, average runtimes in seconds and convergence rate for the max option in the case $D = 10$. Part I: the tree-based approach is used for different values of N_T and M; Part II: the tree-based approach is used for $N_T = 12$ and different values of M; Part III: the multilevel Monte Carlo method is used for different iteration numbers.

multilevel-Picard method in [13] is not sufficiently efficient for a lower dimensional problem. In contrast, in the previous numerical experiments (10-dimensional problem) we have seen that our proposed approach is much more efficient. Although the computational expense in our proposed approach increases for the increasing dimensionality, for this 100-dimensional pricing problem our approach is still two time faster than the method proposed in [13] for the same or better error level, see both the results which are bold and underlined in Table 6. The proposed scheme converges with the rate of 1.09 for the 100-dimensional pricing problem.

5.5 Example of nonlinear FBSDE

In this section we test our scheme for nonlinear high-dimensional problems. We find that nonlinear training data may lead to overfitting when directly using the above introduced procedure. Therefore, to avoid the overfitting for the nonlinear problems, we propose to control the error already while growing a tree. For this, we estimate the cross validation

reference price	$Y_0 = 17.4267$ (average runtime 2613.9 seconds)						
I	The tree-based approach ($\theta_1 = \frac{1}{2}, \theta_2 = 1, \theta_3 = \frac{1}{2}$)						
N_T	2	4	8	12	16	20	
M	5000	10000	80000	100000	200000	300000	
$\frac{1}{10}\sum_{k=1}^{10}\frac{\lvert Y_{0,k}^{\Delta_t} - Y_0\rvert}{\lvert Y_0\rvert}$	0.1920	0.0943	0.0466	0.0297	0.0206	0.0152	CR≈1.09
standard deviation	0.0771	0.0353	0.0180	0.0111	0.0104	0.0082	
average runtime in seconds	16.2	90.1	1621	3162	8529	16180	
II	The tree-based approach ($\theta_1 = \frac{1}{2}, \theta_2 = 1, \theta_3 = \frac{1}{2}$)						
N_T	20	20	20	20	20	20	
M	100	500	1000	2000	5000	10000	
$\frac{1}{10}\sum_{k=1}^{10}\frac{\lvert Y_{0,k}^{\Delta_t} - Y_0\rvert}{\lvert Y_0\rvert}$	0.0240	**0.0140**	0.0159	0.0110	0.0143	0.0137	
standard deviation	0.4819	**0.2275**	0.1858	0.0962	0.0761	0.0704	
average runtime in seconds	8.6	**27.6**	56.1	111.3	270.2	541.9	
III	The multilevel Monte Carlo method [13]						
Number of the Picard iteration	1	2	3	4	5	6	
$\frac{1}{10}\sum_{k=1}^{10}\frac{\lvert Y_{0,k}^{\Delta_t} - Y_0\rvert}{\lvert Y_0\rvert}$	0.1970	0.1368	0.0606	0.0546	0.0249	**0.0165**	
standard deviation	4.2130	3.1551	1.4108	1.2591	0.4458	**0.3539**	
average runtime in seconds	0.0	0.0	0.0	0.4	4.0	**50.1**	

Table 6: Relative errors, standard deviations, average runtimes in seconds and convergence rate for the max option in the case $D = 100$. Part I: the tree-based approach is used for different values of N_T and M; Part II: the tree-based approach is used for $N_T = 20$ and different values of M; Part III: the multilevel Monte Carlo method is used for different iteration numbers.

mean squared errors of the trees, which are constructed with different number of observations in each branch node. Clearly, the best tree, namely the best number of observations in each branch node can be determined by comparing the errors. Theoretically, for the best result, the error control needs to be performed for each time step. However, it will be computationally too expensive. Fortunately, in our test we find the best numbers of observations for each time step are very close to each other. For substantially less computation time, one only needs determine one of them, e.g.,, for the first iteration, and fix it for all other iterations. We note that, the pruning procedure cannot bring considerable improvement when the tree has been grown using the best number of observations, is thus unnecessary in this case.

As an example, we consider a pricing problem of an European option in a financial market with different interest rate for borrowing and lending to hedge the European option. This pricing problem is analyzed in [5] and is used as a standard nonlinear (high-dimensional) example in the many works, see e.g., [12, 13, 16, 2]. Similar but different to (67) and (68), the terminal condition and generator for the option pricing with different interest

rate read as

$$Y_T = \xi = \max\left(\max_{d=1,\cdots,D}(S_{T,d}) - K_1, 0\right) - 2\max\left(\max_{d=1,\cdots,D}(S_{T,d}) - K_2, 0\right) \tag{69}$$

and

$$f(t,x,y,z) = -R^l y - \frac{\mu - R^l}{\sigma}\sum_{d=1}^{D} z_d + (R^b - R^l)\max\left(0, \frac{1}{\sigma}\sum_{d=1}^{D} z_d - y\right), \tag{70}$$

respectively, where R^b, R^l are different interest rates and K_1, K_2 are strikes. Obviously, (69) and (70) are both nonlinear.

We first consider a 1-dimensional case, in which we use $Y_T = \xi = \max(S_T - 100, 0)$ instead of (69) to agree with the setting in [16, 12]. The parameter values are set as: $T = 0.5$, $\mu = 0.06$, $\sigma = 0.02$, $R^l = 0.04$, $R^b = 0.06$. We use $Y_0 = 7.156$ computed using the finite difference method as the reference price. Note that the reference price is confirmed in [16] as well. Firstly, we fix $M = 200000$, $G = 50000$ and plot the relative error against the number of steps in Figure 4. We obtain very good numerical results, and reach an

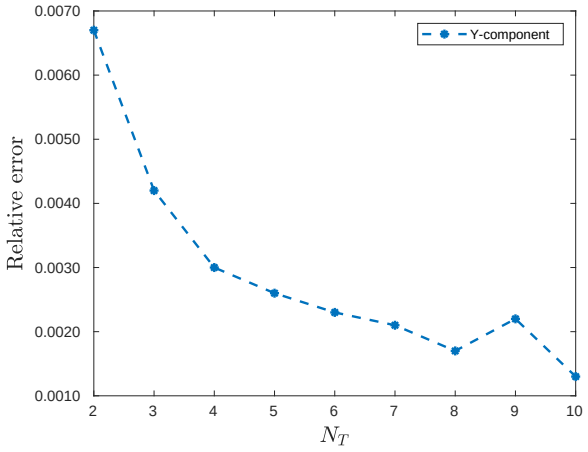


Figure 4: Comparison of relative errors against the number steps N_T and $M = 200000$ for one-dimensional pricing with different interest rate.

error of order 10^{-3}. In Table 7 we compare our results to the results given in Table 5 in [13], and show that the tree-based approach with $N_T = 10$ can reach accuracy level of the multilevel Monte Carlo with 7 Picard iterations for significantly less computational time. Note that the samples-splitting ($G = 50000$) is only used for $M = 100000, 200000$. Finally, we test our scheme for 100-dimensional nonlinear pricing problem. In contrast to the case of 1-dimension, the terminal condition (69) is more challenge to deal with. In our test, (69) can still be used to generate samples of Y_{N_T}. However, for Z_{N_T}, the one-sided derivative of (69), as it in (53) and (65) is not sufficient for the 100-dimensional nonlinear pricing problem. Therefore, for this example we choose the scheme by setting

Reference price	$Y_0 = 7.156$						
I	The tree-based approach ($\theta_1 = \frac{1}{2}, \theta_2 = 1, \theta_3 = \frac{1}{2}$)						
N_T	10	10	10	10	10	10	10
M	2000	4000	10000	20000	50000	100000	200000
$\frac{1}{10}\sum_{k=1}^{10}\frac{\lvert Y_{0,k}^{\Delta_t} - Y_0\rvert}{\lvert Y_0\rvert}$	0.0239	0.0152	0.0107	0.0073	0.0043	0.0035	**0.0013**
standard deviation	0.2089	0.1100	0.0953	0.0686	0.0364	0.0352	**0.0130**
average runtime in seconds	0.1	0.1	0.2	0.3	0.9	1.8	**3.7**
II	The multilevel Monte Carlo method, see Table 5 in [13]						
Number of the Picard iteration	1	2	3	4	5	6	7
$\frac{1}{10}\sum_{k=1}^{10}\frac{\lvert Y_{0,k}^{\Delta_t} - Y_0\rvert}{\lvert Y_0\rvert}$	0.8285	0.4417	0.1777	0.1047	0.0170	0.0086	**0.0019**
standard deviation	7.7805	4.0799	1.6120	0.8106	0.1512	0.0714	**0.0157**
average runtime in seconds	0.0	0.0	0.0	0.3	3.1	38.7	**1915.1**

Table 7: Relative errors, standard deviations, average runtimes in seconds and convergence rate for the 1-dimensional pricing with different interest rates. Part I: the tree-based approach is used for $N_T = 10$ and different values of M; Part II: the multilevel Monte Carlo method is used for different iteration numbers.

$\theta_1 = \theta_2 = \theta_3 = 1$ such that Z-component will be not directly needed for the iterations. In Figure 5, the results of using $M = 200000$, $G = 50000$ against the number steps N_T are reported. Again, in Table 8 we compare our results to them in Table 6 in [13].

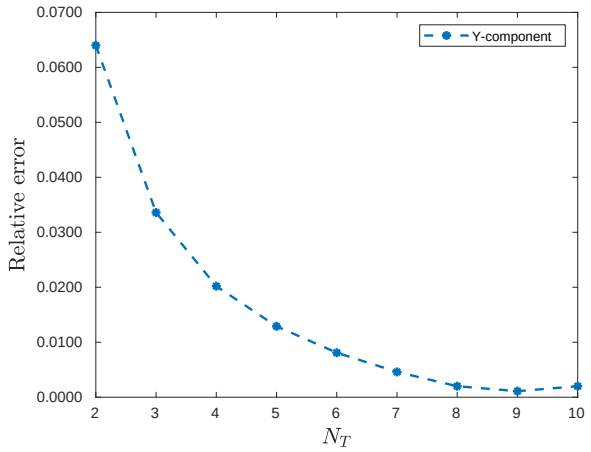


Figure 5: Comparison of relative errors against the number steps N_T and $M = 200000$ for 100-dimensional pricing with different interest rate.

The reference price $Y_0 = 21.2988$ is computed using the multilevel Monte Carlo with 7 Picard iterations, whereas $K_1 = 120, K_2 = 150$, and values of other parameters are the same as those for the 1-dimensional case. We only use the samples-splitting ($G = 50000$) when $M > 50000$. We see that our result with $N_T = 10, M = 2000$ is already better

Reference price	$Y_0 = 21.2988$(average runtime 2725.1 seconds)					
I	The tree-based approach ($\theta_1 = 1, \theta_2 = 1, \theta_3 = 1$)					
N_T	10	10	10	10	10	10
M	10000	50000	100000	200000	300000	400000
$\frac{1}{10}\sum_{k=1}^{10}\frac{\lvert Y_{0,k}^{\Delta_t} - Y_0\rvert}{\lvert Y_0\rvert}$	**0.0212**	0.0022	0.0023	0.0020	0.0017	0.0022
standard deviation	**0.0629**	0.0286	0.0243	0.0199	0.0143	0.0129
average runtime in seconds	**19.6**	115.9	233.3	464.9	703.5	947.7
II	The multilevel Monte Carlo method, see Table 6 in [13]					
Number of the Picard iteration	1	2	3	4	5	6
$\frac{1}{10}\sum_{k=1}^{10}\frac{\lvert Y_{0,k}^{\Delta_t} - Y_0\rvert}{\lvert Y_0\rvert}$	0.4415	0.4573	0.1798	0.1042	0.0509	**0.0474**
standard deviation	8.7977	11.3167	4.4920	2.9533	1.4486	**1.3757**
average runtime in seconds	0.0	0.0	0.0	0.4	4.2	**52.9**

Table 8: Relative errors, standard deviations, average runtimes in seconds and convergence rate for pricing with different interest rates in the case D=100. Part I: the tree-based approach is used for $N_T = 10$ and different values of M; Part II: the multilevel Monte Carlo method is used for different iteration numbers.

than the approximation of multilevel Monte-Carlo with 6 iterations for almost same computational time. Furthermore, a better approximation (smaller standard deviations) can always be achieved with a larger number of M. Note that the same reference price is used to compare the deep learning-based numerical methods for high-dimensional BSDEs in [12] (Table 3),which has achieved a relative error of 0.0039 in a runtime of 566 seconds.

6 Conclusion

In this work, we have studied solving forward-backward stochastic differential equations numerically using the regression tree-based methods. We show how to use the regression tree to approximate the conditional expectations arising by discretizing the time-integrands using the general theta-discretization method. We have performed several numerical experiments for different types of (F)BSDEs including its application to 100-dimensional nonlinear pricing problem. Our numerical results are quite promising and indicate that the tree-based approach is very attractive to solve high-dimensional nonlinear (F)BSDEs.

References

[1] V. Bally. Approximation scheme for solutions of bsde. In N. El Karoui and L. Mazliak, editors, *Backward stochastic differential equations*. Addison Wesley Longman, Harlow, UK, 1997.

[2] C. Bender, N. Schweizer, and J. Zhuo. A primal-dual algorithm for bsdes. *Math. Financ.*, 27(3):866–901, 2017.

[3] C. Bender and J. Steiner. Least-squares monte carlo for backward sdes. *Numer. Methods Finance*, 12:257–289, 2012.

[4] C. Bender and J. Zhang. Time discretization and markovian iteration for coupled fbsdes. *Ann. Appl. Probab.*, 18:143–177, 2008.

[5] Y. Z. Bergman. Option pricing with differential interest rates. *Rev. Financ. Stud.*, 8(2):475–500, 1995.

[6] F. Black and M. Scholes. The pricing of options and corporate liabilities. *J. Political Economy*, 81:637–654, 1973.

[7] B. Bouchard and N. Touzi. Discrete-time approximation and monte-carlo simulation of backward stochastic differential equations. *Stoch. Proc. Appl.*, 111:175–206, 2004.

[8] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and regression trees*. Taylor & Francis, 1984.

[9] I. J. D. Craig and A. D. Sneyd. An alternating-direction implicit scheme for parabolic equations with mixed derivatives. *Comput. Math. Appl.*, 16:341–350, 1988.

[10] D. Crisan and K. Manolarakis. Solving backward stochastic differential equations using the cubature method: Application to nonlinear pricing. *SIAM J. Finan. Math.*, 3(1):534–571, 2010.

[11] J. Douglas, J. Ma, and P. Protter. Numerical methods for forward-backward stochastic differential equations. *Ann. Appl. Probab.*, 6:940–968, 1996.

[12] W. E., J. Han, and A. Jentzen. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Commun. Math. Stat.*, 5(4), 2017.

[13] W. E., M. Hutzenthaler, A. Jentzen, and T. Kruse. On multilevel picard numerical approximations for high-dimensional nonlinear parabolic partial differential equations and high-dimensional nonlinear backward stochastic differential equations. *J. Sci. Comput.*, 2019.

[14] Y. Fu, W. Zhao, and T. Zhou. Efficient spectral sparse grid approximations for solving multi-dimensional forward backward sdes. *Discrete Cont. Dyn-B.*, 22(9):3439–3458, 2017.

[15] E. Gobet and C. Labart. Solving bsde with adaptive control variate. *SIAM J. Numer. Anal.*, 48(1):257–277, 2010.

[16] E. Gobet, J. P. Lemor, and X. Warin. A regression-based monte carlo method to solve backward stochastic differential equations. *Ann. Appl. Probab.*, 15:2172–2202, 2005.

[17] S. L. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Rev. Fin. Stud.*, 6(2):327–343, 1993.

[18] H. Johnson. Options on the maximum or the minimum of several assets. *J. Financ. Quant. Anal.*, 22(3):277–283, 1987.

[19] N. EL Karoui, C. Kapoudjan, E. Pardoux, S. Peng, and M. C. Quenez. Reflected solutions of backward stochastic differential equations and related obstacle problems for pdes. *Ann. Probab.*, 25:702–737, 1997.

[20] N. EL Karoui, S. Peng, and M. C. Quenez. Backward stochastic differential equations in finance. *Math. Finance*, 7(1):1–71, 1997.

[21] J.P. Lemor, E. Gobet, and X. Warin. Rate of convergence of an empirical regression method for solving generalized backward stochastic differential equations. *Bernoulli*, 12:889–916, 2006.

[22] J. P. Lepeltier and J. San Martin. Backward stochastic differential equations with continuous generator. *Statist. Probab. Lett.*, 32(425–430), 1997.

[23] J. Ma, P. Protter, J. San Martín, and S. Torres. Numerical method for backward stochastic differential equations. *Ann. Appl. Probab.*, 12:302–316, 2002.

[24] J. Ma, P. Protter, and J. Yong. Solving forward-backward stochastic differential equations explicity-a four step scheme. *Probab. Theory Related Fields*, 98(3):339–359, 1994.

[25] J. Ma, J. Shen, and Y. Zhao. On numerical approximations of forward-backward stochastic differential equations. *SIAM J. Numer. Anal.*, 46:2636–2661, 2009.

[26] J. Ma and J. Zhang. Representations and regularities for solutions to bsdes with reflections. *Stoch. Proc. Appl.*, 115:539–569, 2005.

[27] W. L. Martinez and A. R. Martinez. *Computational statistics handbook with Matlab.* CRC Press, Taylor & Francis Group, Boca Raton, US, 2007. Second Edition.

[28] G. N. Milsetin and M. V. Tretyakov. Numerical algorithms for forward-backward stochastic differential equations. *SIAM J. Sci. Comput.*, 28:561–582, 2006.

[29] E. Pardoux and S. Peng. Adapted solution of a backward stochastic differential equations. *System and Control Letters*, 14:55–61, 1990.

[30] E. Pardoux and S. Peng. Backward stochastic differential equation and quasilinear parabolic partial differential equations. *Lectures Notes in CSI.*, 176:200–217, 1992.

[31] S. Peng. Probabilistic interpretation for systems of quasilinear parabolic partial differential equations. *Stochastics and Stochastic Reports*, 37(1–2):61–74, 1991.

[32] M. J. Ruijter and C. W. Oosterlee. A fourier cosine method for an efficient computation of solutions to bsdes. *SIAM J. Sci. Comput.*, 37(2):A859–A889, 2015.

[33] R. M. Stulz. Options on the minimum or the maximum of two risky assets. *J. Financial Econ.*, 10(2):161–185, 1982.

[34] L. Teng. A multi-step scheme based on cubic spline for solving backward stochastic differential equations. Preprint 18/13, University of Wuppertal, available on webpage at `https://www.imacm.uni-wuppertal.de/fileadmin/imacm/preprints/2018/imacm_18_13.pdf`, April 2018.

[35] G. Zhang, M. Gunzburger, and W. Zhao. A sparse-grid method for multi-dimensional backward stochastic differential equations. *J. Comput. Math.*, 31(3):221–248, 2013.

[36] J. Zhang. A numerical scheme for bsdes. *Ann. Appl. Probab.*, 14:459–488, 2004.

[37] W. Zhao, L. Chen, and S. Peng. A new kind of accurate numerical method for backward stochastic differential equations. *SIAM J. Sci. Comput.*, 28(4):1563–1581, 2006.

[38] W. Zhao, Y. Fu, and T. Zhou. New kinds of high-order multistep schemes for coupled forward backward stochastic differential equations. *SIAM J. Sci. Comput.*, 36(4):A1731–A1751, 2014.

[39] W. Zhao, Y. Li, and L. Ju. Error estimates of the crank-nicolson scheme for solving backward stochastic differential equations. *Int. J. Numer. Anal. Model.*, 10(4):876–898, 2013.

[40] W. Zhao, Y. Li, and G. Zhang. A generalized θ-scheme for solving backward stochastic differential equations. *Discrete Cont. Dyn-B.*, 17(5):1585–1603, 2012.

[41] W. Zhao, J. Wang, and S. Peng. Error estimates of the θ-scheme for backward stochastic differential equations. *Discrete Contin. Dyn. Syst. Ser. B*, 12:905–924, 2009.

[42] W. Zhao, G. Zhang, and L. Ju. A stable multistep scheme for solving backward stochastic differential equations. *SIAM J. Numer. Anal.*, 48:1369–1394, 2010.