# Numerical methods for backward stochastic differential equations: A survey[*]

## Jared Chessari

*Appian Corporation, Australia*
*e-mail:* jared.chessari@appian.com

## Reiichiro Kawai

*Graduate School of Arts and Sciences / Mathematics and Informatics Center,*
*The University of Tokyo, Japan*
*e-mail:* raykawai@g.ecc.u-tokyo.ac.jp

## Yuji Shinozaki

*Bank of Japan, Japan*

*Tokyo Institute of Technology, Japan*
*e-mail:* sinozaki.yuji@gmail.com

## Toshihiro Yamada

*Graduate School of Economics, Hitotsubashi University, Japan*

*Japan Science and Technology Agency (JST), Japan*
*e-mail:* toshihiro.yamada@r.hit-u.ac.jp

**Abstract:** Backward Stochastic Differential Equations (BSDEs) have been widely employed in various areas of social and natural sciences, such as the pricing and hedging of financial derivatives, stochastic optimal control problems, optimal stopping problems and gene expression. Most BSDEs cannot be solved analytically and thus numerical methods must be applied to approximate their solutions. There have been a variety of numerical methods proposed over the past few decades as well as many more currently being developed. For the most part, they exist in a complex and scattered manner with each requiring a variety of assumptions and conditions. The aim of the present work is thus to systematically survey various numerical methods for BSDEs, and in particular, compare and categorize them, for further developments and improvements. To achieve this goal, we focus primarily on the core features of each method based on an extensive collection of 333 references: the main assumptions, the numerical algorithm itself, key convergence properties and advantages and disadvantages, to provide an up-to-date coverage of numerical methods for BSDEs, with insightful summaries of each and a useful comparison and categorization.

# Contents

## 1. Introduction

A Backward Stochastic Differential Equation (BSDE) was first introduced by Jean-Michel Bismut in 1973 [34]. The paper used a linear BSDE as an equation for the adjoint process in the stochastic version of the Pontryagin maximum principle. BSDEs were generalized by Pardoux and Peng in 1990 [255] to a general non-linear BSDE of the following form:

$$Y_t = \xi_T + \int_t^T f(s, Y_s, Z_s)\, ds - \int_t^T Z_s\, dW_s, \quad t \in [0, T],$$

where questions regarding the existence and uniqueness of the solution process $(Y_t, Z_t)_{t \in [0,T]}$ were addressed. We also mention here the closely related Forward BSDEs (FBSDEs) given as follows:

$$\begin{cases} Y_t = \Phi(X_T) + \int_t^T f(s, X_s, Y_s, Z_s)ds - \int_t^T Z_s dW_s, \\ X_t = \mathbf{x} + \int_0^t \mu(s, X_s)ds + \int_0^t \sigma(s, X_s)dW_s, \end{cases}$$

for $t \in [0, T]$. They are essentially a BSDE coupled with a (forward) Stochastic Differential Equation (SDE), in the sense that the driver can depend on the solution to the SDE, and the terminal condition of the BSDE is a function of the terminal value of the SDE. We also look in detail at FBSDEs, as these types of BSDEs appear widely in literature (especially in regards to numerical methods). The solution of an FBSDE can be formulated in terms of the solution of a semilinear parabolic PDE, and this equivalence is exploited in many numerical methods. For the remainder of the introduction when we mention BSDEs, this can refer to BSDEs or FBSDEs. In following sections, however, we will be strict on which type of BSDE we are referring to.

BSDEs have been used widely in a number of areas of social and natural sciences. The problem of pricing and hedging a European option can be formulated in terms of a BSDE. In fact, any pricing problem using a replication argument can be written in terms of a BSDE [105, 171]. A BSDE can also take into account portfolio constraints in pricing problems [43, 53, 83, 85]. Thus, BSDEs have often been employed for the valuation of many financial derivatives in both complete and incomplete or constrained markets, including European and American options [103, 104]. BSDEs can provide necessary and sufficient conditions for optimality [260, 261] and enhance implementability [150, 166] in optimal control problems such as utility maximization control problems with constraints and risk-sensitive control problems [101, 171, 172, 176, 273, 307].

BSDEs have also been tied to nonlinear expectations and thus have been used to construct the risk measures [262]. The more complex reflected BSDEs [103] and doubly reflected BSDEs [84, 214] have been developed in connection with zero-sum Dynkin games, optimal stopping problems, recallable options, mixed differential games, and mixed stochastic control [157, 159]. In addition, BSDEs have found applications in filtering [14, 15, 16] and physics [93, 278]. More examples of applications in these areas can be found in [79, 102, 105] and image processing [36].

Due to the complex nature of BSDEs, it is rarely possible to find an analytical solution. Thus, one must often resort to numerical methods in order to solve these equations. There have been a variety of numerical methods proposed over the past few decades and also many more currently in progress. For the most part they exist in a complex and scattered manner, with each requiring a variety of assumptions and conditions. The ultimate goal of this survey is to further facilitate the research activity on numerical methods for BSDEs by providing an up-to-date overview of the different types in an organized structure. Before providing the systematic survey of numerical methods, we will first review some basic theory regarding BSDEs (Section 2) in order to better prepare for the goal. Broadly speaking, we will then look at the three main classes of numerical methods, namely backward (Section 3), forward (Section 5), and deep learning based methods (Section 6). By a backward method, we mean a numerical method which works backwards in time and requires the computation of conditional expectations (Section 4), whereas for a forward method, we refer to a method that does not inherently work backward in time so as to (originally, at least) avoid the computation of conditional expectations. In sharp contrast to backward and forward methods, the method of deep learning (Section 6) is quite distinctive in its rather mixed structure and effectiveness in high-dimensional problems.

We then summarize in Section 7 the major components of each category to discuss the power and limitations of various classes of numerical methods in collective comparison. We next provide in Section 8 a further survey on numerical methods for BSDEs with a variety of nonstandard features, such as coupled FBSDEs (Section 8.1), reflected BSDEs (Section 8.2), BSDEs with jumps (Section 8.3), non-Lipschtiz BSDEs (Section 8.4), quadratic BSDEs (Section 8.5), second-order BSDEs (Section 8.6), McKean-Vlasov BSDEs (Section 8.7) and backward stochastic partial differential equations (Section 8.8), each of which has attracted increasing interest for addressing various emerging problems in social and natural sciences and nonlinear partial differential equations. Finally, in Section 9, we summarize our discourse and highlight some future directions.

In order to achieve our main goal of describing and discussing the main idea of each category, we will only focus on one or two relevant representative methods in each subsection, followed by an overview of other methods. To avoid overloading the paper with non-essential technical details, we omit the rather lengthy technical intricacies of the methods in most instances. In particular, the algorithms we provide in each section are described without going into detail and are presented for illustrative purposes only.

## 2. Preliminaries

The aim of this section is to review the basic theory of BSDEs (Section 2.1) and FBSDEs (Section 2.2) in brief. In particular, we give the standard sufficient conditions that ensure the existence and uniqueness of their solutions. Let $(\Omega, \mathcal{F}, \mathbb{F}, \mathbb{P})$ be a given filtered probability space which satisfies the usual conditions of completeness and right continuity. On this space, we define the $d$-dimensional standard Brownian motion $(W_t)_{t \in [0,T]}$, whose natural filtration, augmented by the class of $\mathbb{P}$-null sets of $\mathcal{F}_T$, is $\mathbb{F} = (\mathcal{F}_t)_{t \in [0,T]}$. We reserve $T$ for a strictly positive constant which indicates a fixed terminal time of the interval $[0, T]$, and denote by $\pi_n := \{0 =: t_0 < \cdots < t_k < t_{k+1} < \cdots < t_n := T\}$ an arbitrary yet fixed partition of the interval $[0, T]$ for all $n \in \mathbb{N}$, with $|\pi_n| := \max_{k \in \{0, \cdots, n-1\}} (t_{k+1} - t_k)$. For the sake of brevity, we often suppress the subscript $n$ from the notation $\pi_n$. We define the following probabilistic spaces and sets:

- $\mathcal{P}$: the $\sigma$-field of predictable sets in $\Omega \times [0, T]$.
- $\mathcal{L}_m^2(\mathcal{F}_t)$: the set of all $\mathbb{R}^m$-valued random vectors $X$ that are $\mathcal{F}_t$-measurable and square-integrable, that is, $\mathbb{E}[\|X\|^2] < \infty$.
- $\mathcal{S}_m^2(0, T)$: the space of all continuous predictable processes $X : \Omega \times [0, T] \to \mathbb{R}^m$, satisfying $\mathbb{E}[\sup_{t \in [0,T]} \|X_t\|^2] < \infty$.
- $\mathcal{H}_m^2(0, T)$: the space of all predictable processes $X : \Omega \times [0, T] \to \mathbb{R}^m$, which satisfy $\|X\|_0^2 := \mathbb{E}[\int_0^T \|X_t\|^2 dt] < \infty$.

We reserve $C$ for a positive constant, whose value changes from line to line, depending on the context at hand. Let $|\cdot|$ and $\|\cdot\|$ denote the magnitude and Euclidean norm respectively, where the latter is understood to be a suitable matrix norm in the context of matrices. We denote $D^{\otimes 2} := DD^\top$ and by $\mathcal{B}(D)$ the Borel $\sigma$-field of a set $D$. We define the following function spaces which appear throughout:

- $\mathcal{C}_b^{k,l}$: set of continuously differentiable and bounded functions which are $k$ times continuously differentiable in their first coordinate and $l$ times in their second, with bounded partial derivatives up to order $l$.
- $\mathcal{C}_p^k$: set of $\mathcal{C}^{k-1}$ functions with piecewise continuous $k$-th derivative.
- $\mathcal{C}^{k+\alpha}$: set of $\mathcal{C}^k$ functions whose $k$-th derivative is Hölder continuous of order $\alpha \in [0, 1]$.

### 2.1. Backward stochastic differential equations

We aim here to review the basics of the following BSDE:

$$\begin{cases} -dY_t = f(t, Y_t, Z_t)dt - Z_t dW_t, & t \in [0, T], \\ Y_T = \xi, \end{cases} \tag{2.1}$$

where $(Y, Z)_{t \in [0,T]}$ takes values in $\mathbb{R}^m \times \mathbb{R}^{m \times d}$. The function $f : \Omega \times [0, T] \times \mathbb{R}^m \times \mathbb{R}^{m \times d} \to \mathbb{R}^m$ is called the generator or driver. The random variable $\xi = Y_T$ is the

terminal condition for the process $Y$ and the pair $(f, \xi)$ are called the parameters of the BSDE. A pair $(Y, Z)$ of stochastic processes is a (not necessarily unique) solution to the BSDE (2.1) if

- $(Y_t)_{t \in [0,T]}$ is an $\mathbb{F}$-adapted continuous process in $\mathbb{R}^m$.
- $(Z_t)_{t \in [0,T]}$ is an $\mathbb{F}$-predictable process in $\mathbb{R}^{m \times d}$ satisfying $\mathbb{P}(\int_0^T \|Z_t\|^2 dt < \infty) = 1$.
- $Y_t = \xi + \int_t^T f(s, Y_s, Z_s)\, ds - \int_t^T Z_s\, dW_s$ for all $t \in [0, T]$, $a.s.$

We will only consider solutions $(Y, Z)$ in the space $\mathcal{S}_m^2(0, T) \times \mathcal{H}_{m \times d}^2(0, T)$. The following theorem [255, 318] identifies the standard sufficient conditions imposed to ensure the existence and uniqueness of a solution to the BSDE (2.1) in $\mathcal{S}_m^2(0, T) \times \mathcal{H}_{m \times d}^2(0, T)$.

**Theorem 2.1** (**Existence and uniqueness of the solution to a BSDE**). Consider the BSDE given by (2.1) with parameters $(f, \xi)$, satisfying:

- $f : \Omega \times [0, T] \times \mathbb{R}^m \times \mathbb{R}^{m \times d} \to \mathbb{R}^m$ is $\mathcal{P} \otimes \mathcal{B}(\mathbb{R}^m) \otimes \mathcal{B}(\mathbb{R}^{m \times d})$-measurable.
- The process $(f(t, 0, 0))_{t \in [0,T]}$ belongs to $\mathcal{H}_m^2(0, T)$.
- The driver is uniformly Lipschitz continuous with respect to $(\mathbf{y}, \mathbf{z})$, that is, there exists a constant $C > 0$ such that for all $\mathbf{y}_1, \mathbf{y}_2 \in \mathbb{R}^m$ and $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{R}^{m \times d}$, $\|f(\omega, t, \mathbf{y}_1, \mathbf{z}_1) - f(\omega, t, \mathbf{y}_2, \mathbf{z}_2)\| \le C(\|\mathbf{y}_1 - \mathbf{y}_2\| + \|\mathbf{z}_1 - \mathbf{z}_2\|)$, $dt \otimes d\mathbb{P}$-$a.e.$
- $\xi \in \mathcal{L}_m^2(\mathcal{F}_T)$.

Then, there exists a unique solution $(Y, Z) \in \mathcal{S}_m^2(0, T) \times \mathcal{H}_{m \times d}^2(0, T)$ which solves the BSDE (2.1).

Before moving on, we briefly present some intuition regarding the solution $(Y, Z)$ of the BSDE (2.1). Consider the simple case when $f \equiv 0$ in (2.1) giving us the following BSDE:

$$\begin{cases} dY_t = Z_t dW_t, & t \in [0, T], \\ Y_T = \xi. \end{cases} \tag{2.2}$$

Clearly, the degenerate driver $f \equiv 0$ satisfies the sufficient assumptions for uniqueness and existence. Hence, under the assumption that $\xi \in \mathcal{L}_m^2(\mathcal{F}_T)$, it holds by Theorem 2.1 that the BSDE with null generator (2.2) admits a unique solution $(Y, Z) \in \mathcal{S}_m^2(0, T) \times \mathcal{H}_{m \times d}^2(0, T)$. If $(Y, Z)$ is the unique solution in $\mathcal{S}_m^2(0, T) \times \mathcal{H}_{m \times d}^2(0, T)$ of the BSDE with null driver (2.2), then $Y$ satisfies

$$Y_t = \mathbb{E}\left[Y_T | \mathcal{F}_t\right] = \mathbb{E}\left[\xi | \mathcal{F}_t\right],$$

and thus $Y$ is an $(\mathcal{F}_t)_{t \in [0,T]}$-martingale. Moreover, based on the formulation (2.2), the martingale $Y$ can be written as follows:

$$Y_t = \mathbb{E}\left[\xi|\mathcal{F}_t\right] = Y_0 + \int_0^t Z_s dW_s = \mathbb{E}\left[\xi|\mathcal{F}_0\right] + \int_0^t Z_s dW_s = \mathbb{E}\left[\xi\right] + \int_0^t Z_s dW_s.$$

As the process $Y$ is a martingale, it is certainly a local martingale. Thus, by the martingale representation theorem, there exists a unique process $\gamma \in \mathcal{H}_{m \times d}^2(0, T)$ satisfying $Y_t = Y_0 + \int_0^t \gamma_s dW_s$ for all $t \in [0, T]$. Hence, we obtain

$$\xi = Y_T = Y_0 + \int_0^T \gamma_s dW_s = Y_0 + \int_0^t \gamma_s dW_s + \int_t^T \gamma_s dW_s = Y_t + \int_t^T \gamma_s dW_s,$$

which shows that $(Y, \gamma)$ solves the BSDE (2.2). By the uniqueness of the solution to the BSDE, we get $Z = \gamma$. For this simple case, we see that an adapted solution to the BSDE can only be given by a pair, that is, the $Z$ component is needed to ensure that the process $Y$ is adapted. In a sense, the $Z$ component "steers" the system and is thus called the control process. One cannot simply revert time as in deterministic ODEs, as the filtration can only go in one direction.

To better illustrate the interpretations of $Y$, $Z$ and $\xi$, we provide a simple example in a tangible problem in finance. Consider a financial market with one risky asset $S$, whose price at time t, $S_t$, follows the following SDE:

$$dS_t = S_t \mu_t dt + S_t \sigma_t dW_t. \tag{2.3}$$

A trader can either invest in the risky asset $S$ or borrow/invest money at an instantaneous risk free interest rate, denoted by $r$. Here, we assume that $\mu$, $\sigma$ and $r$ are bounded and predictable processes. If the amount of money invested in $S$ at time $t$ is $\phi_t$ and the total wealth of the trade is $Y$, then the magnitude $|Y_t - \phi_t|$ represents the amount of money that is borrowed (if $Y_t - \phi_t$ is negative) or invested (if $Y_t - \phi_t$ is positive) at time $t$. The wealth process $Y$ can thus be shown to follow the dynamics:

$$dY_t = \frac{\phi_t}{S_t} dS_t + r_t(Y_t - \phi_t)dt = \left(\phi_t(\mu_t - r_t) + r_t Y_t\right) dt + \phi_t \sigma_t dW_t. \tag{2.4}$$

Consider a European option with payoff at time $T$ given by a random variable $\xi \in \mathcal{L}_m^2(\mathcal{F}_T)$. We note that the payoff for a European option $\xi$ will usually be a function of the asset price $S$ at time $T$, that is $\xi = \Phi(S_T)$. Although this would move us to the context of a FBSDE (Section 2.2), we ignore this fact here for illustrative purposes. A trader who wants to sell this option must identify the minimal initial amount of capital $Y_0$ in such a way that the payoff $\xi = \Phi(S_T)$ can be replicated. If a process $\phi$ is found such that $Y_T = \xi$, then this initial amount is $Y_0$. In other words, we look for a couple $(Y, \phi)$ such that

$$Y_t = \xi - \int_t^T \left(\phi_s(\mu_s - r_s) + r_s Y_s\right) ds - \int_t^T \phi_s \sigma_s dW_s.$$

If there exists a predictable process $\pi$ such that $(\mu - r) = \sigma\pi$, then by setting $Z = \phi\sigma$, the above equation becomes

$$Y_t = \xi - \int_t^T (Z_s\pi_s + r_s Y_s)\, ds - \int_t^T Z_s dW_s.$$

Hence, the problem of pricing and hedging is reduced to finding a solution to the above (linear) BSDE, where $Y$ is the wealth process (which is equivalent to the price of the option), $\xi = \Phi(S_T)$ is the payoff of a European option and $Z$ is proportional to the volatility in the model and the amount invested in the risky asset $S$.

Another typical example in quantitative finance, that is more general than the simple European case and which justifies the use of nonlinear BSDEs, is the so-called the differential rates problem (for instance, [313] among many others). Consider again, a financial market with one risky asset $S$ given by the SDE (2.3), with $\phi_t$ representing the amount of money invested in $S$ at time $t$. We thus have that the dynamics of the total wealth $Y$ are again given by (2.4). Let a lending interest rate $r$ be applied to the lending case $Y_t - \phi_t \geq 0$ and let a borrowing interest rate $R$ such that $R > r$ be applied to the borrowing case $Y_t - \phi_t \leq 0$. In a similar manner to the European case above, if a process $\phi$ is found such that $Y_T = \xi$ with target payoff $\xi \in \mathcal{L}_m^2(\mathcal{F}_T)$ and with $Z = \phi\sigma$, we then get a backward process:

$$Y_t = \xi - \int_t^T \left[ \frac{\mu_s}{\sigma_s} Z_s + r_s \left( Y_s - \frac{Z_s}{\sigma_s} \right)_+ - R_t \left( Y_s - \frac{Z_s}{\sigma_s} \right)_- \right] ds - \int_t^T Z_s dW_s,$$

which lies in the framework of nonlinear BSDEs.

Stochastic control is undoubtedly a major field of application. To briefly describe stochastic control in our notation, consider the stochastic optimization problem $\sup_k J(k)$ in the weak formulation [105]:

$$J(k) := \mathbb{E}_k \left[ \Phi(X_T) + \int_0^T f(s, X_s, k_s) ds \right], \tag{2.5}$$

where $\Phi$ and $f$ take values in $\mathbb{R}$, and $k$ controls the SDE as in

$$X_t = \mathbf{x} + \int_0^t b(s, X_s, k_s) ds + \int_0^t \sigma(s, X_s) dB_s^k, \quad t \in [0, T],$$

and $\mathbb{E}_k$ is the expectation with respect to the probability measure $\mathbb{P}_k$ under which $B_t^k := B_t - \int_0^t \sigma^{-1}(s, X_s) b(s, X_s, k_s) ds$ is a Brownian motion. Then, the following linear BSDE under $\mathbb{P}_k$ has a unique solution $(Y^k, Z^k)$:

$$Y_t^k = \Phi(X_T) + \int_t^T f(s, X_s, k_s) ds - \int_t^T Z_s^k dB_s^k,$$

due to Theorem 2.1 with $\xi = \Phi(X_T)$, provided that all the conditions are met. In view of $J(k) = Y_0^k$, the stochastic control problem (2.5) is well within the framework (2.1).

### *2.2. Forward backward stochastic differential equations*

We next review forward-backward stochastic differential equations (FBSDEs), which have already made a brief implicit appearance in the example on European option pricing (Section 2.1). Consider the following BSDE:

$$\begin{cases} -dY_t = f(t, X_t, Y_t, Z_t)dt - Z_t dW_t, & t \in [0, T], \\ Y_T = \Phi(X_T), \end{cases} \tag{2.6}$$

where $(Y, Z)_{t \in [0,T]}$ takes values in $\mathbb{R}^m \times \mathbb{R}^{m \times d}$, $X$ is the $\mathbb{R}^q$-valued diffusion process solving the standard stochastic differential equation:

$$\begin{cases} dX_t = \mu(t, X_t)dt + \sigma(t, X_t)dW_t, & t \in [0, T], \\ X_0 = \mathbf{x}, \end{cases} \tag{2.7}$$

and $f : [0, T] \times \mathbb{R}^q \times \mathbb{R}^m \times \mathbb{R}^{m \times d} \to \mathbb{R}^m$, $\Phi : \mathbb{R}^q \to \mathbb{R}^m$, $\mu : [0, T] \times \mathbb{R}^q \to \mathbb{R}^q$ and $\sigma : [0, T] \times \mathbb{R}^q \to \mathbb{R}^{q \times d}$ are all given functions, and $\mathbf{x}_0$ is a suitable point in $\mathbb{R}^q$. We note that the dimension of the $Y$ component is set to one ($m = 1$) in many instances, particularly when dealing with deterministic PDEs (Theorems 2.2 and 2.3). The pair of equations (2.6)-(2.7) is known as the Markovian BSDE, or the (uncoupled) FBSDE. We now introduce the "Markovian" counterparts of the assumptions made in Theorem 2.1 which ensure the existence and uniqueness of solutions to the FBSDE (2.6)-(2.7), as well as an equivalence between the solution of the FBSDE to a viscosity solution of the parabolic PDE (2.9). We refer the reader to, for instance, [256, 318] for more detail and proofs.

**Assumption 1.** Assume the following conditions on the FBSDE (2.6)-(2.7):

- $\mu$, $\sigma$, $f$ and $\Phi$ are uniformly Lipschitz continuous in $(\mathbf{x}, \mathbf{y}, \mathbf{z})$.
- $\mu(\cdot, 0)$, $\sigma(\cdot, 0)$, $f(\cdot, 0, 0, 0)$ and $\Phi(0)$ are bounded.
- $\mu$, $\sigma$ and $f$ are uniformly Hölder-$(1/2)$ continuous in $t$. $\qquad \square$

**Theorem 2.2 (Existence, uniqueness and PDE equivalence).** Under Assumption 1, there exists a unique solution $(X, Y, Z)$ to the FBSDE (2.6)-(2.7). Define $v : [0, T] \times \mathbb{R}^q \to \mathbb{R}$ by

$$v(t, \mathbf{x}) := Y_t^{t, \mathbf{x}} = \mathbb{E}\left[\Phi(X_T^{t, \mathbf{x}}) + \int_t^T f(s, X_s^{t, \mathbf{x}}, Y_s^{t, \mathbf{x}}, Z_s^{t, \mathbf{x}})ds\right], \tag{2.8}$$

for $(t, \mathbf{x}) \in [0, T] \times \mathbb{R}^q$, where $(X^{t, \mathbf{x}}, Y^{t, \mathbf{x}}, Z^{t, \mathbf{x}})$ denotes the adapted solution to the FBSDE (2.6)-(2.7), restricted to $[t, T]$ with $X_t^{t, \mathbf{x}} = \mathbf{x}$, a.s. Then, $v$ is a

viscosity solution to the parabolic PDE:

$$
\begin{cases}
((\partial/\partial t) + \mathcal{L}_t)v(t, \mathbf{x}) \\
\quad + f(t, \mathbf{x}, v(t, \mathbf{x}), (\nabla v(t, \mathbf{x}))^\top \sigma(t, \mathbf{x})) = 0, & (t, \mathbf{x}) \in [0, T) \times \mathbb{R}^q, \\
v(T, \mathbf{x}) = \Phi(\mathbf{x}), & \mathbf{x} \in \mathbb{R}^q,
\end{cases}
\quad (2.9)
$$

where $\mathcal{L}$ denotes the second-order differential operator

$$
\mathcal{L}_t v(t, \mathbf{x}) := \langle \mu(t, \mathbf{x}), \nabla v(t, \mathbf{x}) \rangle + \frac{1}{2} \mathrm{tr} \left[ \sigma^{\otimes 2}(t, \mathbf{x}) \mathrm{Hess}(v(t, \mathbf{x})) \right].
$$

Analytic solutions are rarely available for FBSDEs, especially those of practical relevance. However, for illustrative purposes, we present a few examples of the FBSDE in the form (2.6)-(2.7), for which unique solutions are available in closed form, to indicate how simple the FBSDE must be for an analytic solution to be available. First, set $\mu(t, x) = \theta(a - x)$ and $\sigma(t, x) = b$. With a suitable initial state $X_0 = x_0 \in \mathbb{R}$, the forward component (2.7) is then a diffusion process in the form of $X_t = e^{-\theta t} x_0 + a(1 - e^{-\theta t}) + b \int_0^t e^{-\theta(t-s)} dW_s$, that is, an Ornstein-Uhlenbeck process if $\theta > 0$. With the quadratic terminal $\Phi(x) = x^2$ fixed, if the driver is given by $f(t, x, y, z) = ry + e^{-r(T-t)} b^2 + \theta(a - x)z/b$, for some $r, a, b \in \mathbb{R}$, then the FBSDE (2.6) admits a unique solution given by $(Y_t, Z_t) = (e^{-r(T-t)} X_t^2, 2be^{-r(T-t)} X_t)$ for $t \in [0, T]$. Or, if the driver is set, instead, to $f(t, x, y, z) = 2\theta(a - x)x + b^2$, for some $a, b, \theta \in \mathbb{R}$, then the unique solution to the FBSDE (2.6) is available in closed form as $(Y_t, Z_t) = (X_t^2, 2bX_t)$ for $t \in [0, T]$. Next, if the terminal, the driver and the forward component are given by $\Phi(x) = \ln(x)$, $f(t, x, y, z) = a - b^2/2$, $\mu(t, x) = ax$, $\sigma(t, x) = bx$, and $X_0 = x_0 > 0$ for some $a, b \in \mathbb{R}$, then the FBSDE (2.6)-(2.7) admits the unique solution $(Y_t, Z_t) = (\ln(X_t), b)$ for $t \in [0, T]$ and the geometric Brownian motion $X_t = x_0 \exp[(a - b^2/2)t + bW_t]$ for $t \in [0, T]$. We once again stress that these simple examples are only presented for demonstration purposes. FBSDEs of practical interest are generally not explicitly solvable and thus do require numerical treatment.

We next state the nonlinear Feynman-Kac formula with the representation of the derivative of the PDE solution. We need the following additional assumption to give the representation under the Lipschitz continuous condition of $f$ and $\Phi$. For details (including the definition of the solution $\nabla X$ of a variational equation) and applications we refer the reader to, for instance, [228, 316].

**Assumption 2.** Assume the following conditions:

- $\sigma$ is uniformly elliptic on $[0, T] \times \mathbb{R}^q$, that is, there exists $C > 1$ such that $C^{-1} \|\xi\|^2 \leq \langle \xi, \sigma^{\otimes 2}(t, \mathbf{x})\xi \rangle \leq C \|\xi\|^2$ for all $t \in [0, T]$ and $\mathbf{x}, \xi \in \mathbb{R}^q$.
- $\mu$ and $\sigma$ are in $\mathcal{C}_b^1$ in $\mathbf{x}$. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\square$

**Theorem 2.3 (Nonlinear Feynman-Kac formula with the representation of the derivative of the PDE solution).** Under Assumptions 1 and 2,

the viscosity solution $v$ to the parabolic PDE (2.9) is in $\mathcal{C}_b^{0,1}([0,T] \times \mathbb{R}^q; \mathbb{R})$ and it holds that

$$(Y_t, Z_t) = (v(t, X_t), (\nabla v(t, X_t))^\top \sigma(t, X_t)), \quad t \in [0, T), \tag{2.10}$$

and

$$\nabla v(t, \mathbf{x}) = \mathbb{E}\left[\Phi(X_T^{t,\mathbf{x}})N_T^{t,\mathbf{x}} + \int_t^T f(s, X_s^{t,\mathbf{x}}, Y_s^{t,\mathbf{x}}, Z_s^{t,\mathbf{x}})N_s^{t,\mathbf{x}}ds\right], \tag{2.11}$$

for $(t, \mathbf{x}) \in [0, T) \times \mathbb{R}^q$, where $N_s^{t,\mathbf{x}} := (s-t)^{-1}(\int_t^s (\sigma^{-1}(r, X_r^{t,\mathbf{x}})\nabla X_r^{t,\mathbf{x}})^\top dW_r)^\top$ for $s \in (t, T]$.

From here on out, with some exceptions (including Sections 5.2 and 8.4), all the conditions of Theorems 2.1 and 2.2 are imposed as standing assumptions. Further assumptions (such as Assumption 2) may be imposed in different forms for different numerical methods depending on the BSDEs and FBSDEs of interest.

From a practical perspective, FBSDEs have long been actively studied in the pricing and hedging problem in the incomplete market [83], which remains an ongoing topic, for instance, [29]. With the aid of the relevant techniques developed there, it has been known that FBSDEs play an essential role for formulating X-Value Adjustment (XVA), which is a collective term for various valuation adjustments for derivative instruments, for instance, credit risk and funding costs. Ever since the catastrophic wave of bankruptcies of big financial firms in the 2008 global financial crisis, the XVA pricing has been realized as one of most urgent problems in derivatives pricing. To be more precise, XVAs can be formulated in the form of a FBSDE with a Lipschitz continuous driver $f$, which is non-differentiable at some points, such as $f(t, x, y, z) = a(t) \max\{0, y\} + b(t)$. Ever since the pioneering work [267], a variety of extensions and refinements of the XVA formulation have been investigated, certainly in the framework of FBSDEs, from both theoretical and practical standpoints [32, 33, 75, 77, 215]. In particular, various numerical methods for the XVA pricing have also been proposed, such as a branching algorithm [167], a Fourier-based discretization method [38], higher-order discretization methods [247], a dual algorithm for the stochastic control problem [168] and the deep learning algorithm [136].

Despite being outside the scope of Assumption 1, the American option pricing problem [103, 104] can also be formulated in the framework of FBSDEs with a discontinuous driver of the form $f(t, x, y, z) = a(t)\mathbb{1}(y < \Phi(x)) + b(t)$, based on which various numerical methods have been proposed. Examples of such methods include the perturbative expansion and particle method [117] and a local polynomial approximation and branching processes method [40].

As such, there exist a variety of advanced features and relevant terminologies around FBSDEs and their applications in the literature, which we summarize here in brief. First, the FBSDE in the form (2.6)-(2.7) is called a decoupled FB-SDE, in contrast to "coupled" when the coefficients $\mu$ and $\sigma$ in the forward component (2.7) depend on the backward component (2.6) (Section 8.1). Moreover,

the term "fully coupled" is often attached when those coefficients depend on the outcome $\omega$. The backward component may be reflected at a given stochastic process (Section 8.2), while FBSDEs may contain jumps in the backward and/or forward component (Section 8.3). If the drift of the backward component (2.6) contains the second order derivative of the corresponding PDE (2.9), then such a coupled FBSDE is referred to as a second-order FBSDE (Section 8.6). If not only the coefficients $\mu$ and $\sigma$ depend on the law of the processes $(X, Y, Z)$ but also the driver $f$, then such a coupled FBSDE is called a McKean-Vlasov FBSDE (Section 8.7).

## 3. Backward numerical methods

With the basic background reviewed in Section 2, we begin with various numerical methods for BSDEs which work backwards in time under two categories with respect to the degree of discretization, namely, backward Euler methods (Section 3.1) and higher-order methods (Section 3.2). In each subsection, we present one or two representative numerical methods in some detail, while we skip lengthy technical explanations. In particular, when describing the algorithms in each subsection, we do not go into much technical detail in order to avoid digression from the main idea. We then give a brief overview of various other methods in the category.

In general, a discretization method is said to be of order $p$ if the discretization error has the convergence rate $\mathcal{O}(n^{-p})$ in a suitable norm where the given time interval is discretized into $n$ subintervals. However, this convergence rate does not necessarily represent the efficiency of the numerical algorithm under consideration. That is, in most instances, backward numerical methods require the additional computation of conditional expectations in each subinterval, whose efficiency depends largely on many factors, such as the dimension and the method applied. As we are only focusing on backward methods alone here, we will discuss various methods for computing conditional expectations collectively in Section 4, such as least-squares regression (Section 4.1), Malliavin calculus based methods (Section 4.2), quantization methods (Section 4.3), tree based methods (Section 4.4) and cubature methods (Section 4.5).

### *3.1. Backward Euler methods*

The first class of backward numerical methods we review are the so-called backward Euler schemes. The main references on the topic are [47, 317], although the ideas of the backward Euler scheme date back to [72]. The convergence of the backward Euler scheme is of order $1/2$ under standard Lipschitz assumptions and with Lipschitz terminal condition [47, 317]. However, the scheme has a convergence rate of 1 if the forward SDE can be simulated perfectly on the grid or if it is approximated by a higher order scheme under the conditions that the coefficients are sufficiently smooth [137]. We note that discretization methods of higher order are often collectively called higher-order discretization methods, but we do not discuss them here and instead devote Section 3.2 to them.

In the present context, there are certainly two general categories of explicit and implicit discretization schemes for the FBSDE (2.6)-(2.7) (see, also, [318, Section 5.3.2]), which can be summarized as follows:

---

**Algorithm 1:**

---

**Initialization**: Approximate the terminal condition $Y_{t_n}^n = \Phi(X_{t_n}^n)$ with the Euler-Maruyama scheme $X^n$.

**for** $k = (n-1)$ to $0$ **do**

$$Z_{t_k}^n = \frac{1}{t_{k+1} - t_k} \mathbb{E}\left[ Y_{t_{k+1}}^n (W_{t_{k+1}} - W_{t_k})^\top \Big| \mathcal{F}_{t_k} \right], \qquad (3.1)$$

$$Y_{t_k}^n = \begin{cases} \mathbb{E}\left[ Y_{t_{k+1}}^n + f(t_k, X_{t_k}^n, Y_{t_{k+1}}^n, Z_{t_k}^n)(t_{k+1} - t_k) \Big| \mathcal{F}_{t_k} \right], & \text{(explicit)} \\ \mathbb{E}\left[ Y_{t_{k+1}}^n \Big| \mathcal{F}_{t_k} \right] + f(t_k, X_{t_k}^n, Y_{t_k}^n, Z_{t_k}^n)(t_{k+1} - t_k). & \text{(implicit)} \end{cases}$$
$$(3.2)$$

**end**

---

As usual, the implicit scheme often provides better properties and performance relative to the explicit scheme, with these benefits coming in exchange for the additional computing effort for solving the defining equation for $Y_{t_k}^n$, which appears on both sides of the implicit scheme.

With $(Y^n, Z^n)_{t \in [0,T]}$ understood to be the pair of the step processes whose state at $t_k$ corresponds to (3.1)-(3.2), it was first derived [47, 317] that

$$\sup_{t \in [0,T]} \mathbb{E}\left[ \|Y_t - Y_t^n\|^2 \right] + \mathbb{E}\left[ \int_0^T \|Z_t - Z_t^n\|^2 \right] \leq \frac{C}{n}. \qquad (3.3)$$

The backward Euler scheme in the reflected case converges at order $1/2$, when $f$ is independent of $Z$ [11], but only at order $1/4$ in the general case [39, 229]. We note that convergence may depend on the regularity of the terminal condition in general through the $L^2$-regularity of $Z$ [127]. Stability analysis is conducted in [64] for the backward Euler method in a similar manner to the Euler method for ordinary differential equations.

Backward Euler methods have been successfully generalized to address a broader class of BSDEs with nonstandard features (Section 8), such as BSDEs with a general terminal condition and driver [177], BSDEs driven by an infinite activity Poisson random measure in addition to the Brownian motion [230], reflected BSDEs [39, 229, 281], BSDEs with jumps [42, 232] and quadratic BSDEs [272]. Backward Euler methods have also been tailored to, for instance, mean-field BSDEs [321], $G$-FBSDEs (FBSDEs driven by $G$-Brownian motion) [175], FBSDEs driven by càdlàg martingales [205], and backward stochastic Volterra integral equations of type I [24, 303] and of type II [160]. The convergence of the backward Euler method for BSDEs is discussed in [177] with general terminal condition and driver.

We close this section with [44, 106, 156], where some studies focus to construct numerical methods for nonlinear PDEs based on backward Euler methods for

FBSDEs. One may also look to [152, 322] for related studies on discretization, as well as [155] where a number of finite difference methods are reviewed for solving BSDEs, similar to backward Euler methods for the most part.

### *3.2. Higher-order discretization methods*

In the literature, a variety of discretization methods have been proposed to achieve a higher order of convergence than the backward Euler method of order 1 (Section 3.1). In principle, to achieve a higher order of convergence, the following three steps need to be planned and implemented carefully:

1. We first discretize the BSDE. A typical result on the discretization of a solution to a BSDE $(Y, Z)$ is given by

$$\sup_{k \in \{0, 1, \cdots, n\}} \mathbb{E} \left[ \left\| Y_{t_k} - Y_{t_k}^{\pi} \right\|^2 + \frac{1}{n} \left\| Z_{t_k} - Z_{t_k}^{\pi} \right\|^2 \right]^{1/2} \leq \frac{C}{n^2}, \qquad (3.4)$$

where discretization of the forward component $X$ has not been taken into account. Here, $Y^{\pi}$ and $Z^{\pi}$ denote suitable simple schemes involving conditional expectation representations for $Y$ and $Z$, respectively (see, for instance, [81]).

2. The corresponding forward component $X$ is to be discretized using a suitable high-order discretization scheme and in such a way that the overall error remains within the order of the backward component (like, the inequality (3.4)).

3. By then applying a computation method for the conditional expectation (Section 4), an efficient numerical scheme is obtained which has a faster convergence rate than a scheme with the backward Euler method (Section 3.1).

A variety of relevant numerical methods have been developed in conjunction with existing techniques in each of those three steps. We devote the present section to a review of such higher-order discretization methods.

For illustrative purposes, we begin with a simple numerical method of higher-order type based upon a random walk approximation of the Brownian motion. This method is given in the context of FBSDEs whose driver is independent of $X$, although the method can be easily extended to when one does not have this independence. The forward (SDE) component is approximated using the Euler-Maruyama method or Milstein method. The solution to the FBSDE is approximated by a discretization scheme that is derived by using the so-called theta method. For example, the numerical approximation for $Y$ is derived by first showing that

$$Y_{t_k} = \mathbb{E} \left[ Y_{t_{k+1}} \middle| \mathcal{F}_{t_k} \right] + \int_{t_k}^{t_{k+1}} \mathbb{E} \left[ f(s, Y_s, Z_s) \middle| \mathcal{F}_{t_k} \right] ds,$$

and then approximating the integral using a theta approximation: a convex combination of an explicit term (term at time $t_{k+1}$) and an implicit term (term at time $t_k$):

$$Y_{t_k} \approx \mathbb{E}\left[Y_{t_{k+1}}\middle|\mathcal{F}_{t_k}\right] + \Delta_n\theta_1 f(t_k, Y_{t_k}, Z_{t_k})$$
$$+ \Delta_n(1-\theta_1)\mathbb{E}\left[f(t_{k+1}, Y_{t_{k+1}}, Z_{t_{k+1}})\middle|\mathcal{F}_{t_k}\right], \quad (3.5)$$

for some $\theta_1 \in [0,1]$, where $\Delta_n := T/n(= t_{k+1} - t_k)$ is a fixed time step. In a similar manner, a numerical approximation for $Z$ is derived, and then a backward numerical scheme is presented using these approximations. Namely, the scheme for approximating $(Y, Z)$ by $(Y^\pi, Z^\pi)$ is given as $Y_{t_n}^\pi = \Phi(X_{t_n}^\pi)$, $Z_{t_n}^\pi = (\nabla\Phi(X_{t_n}^\pi))\sigma(t_n, X_{t_n}^\pi)$, and then

$$Z_{t_k}^\pi = -\frac{1-\theta_2}{\theta_2}\mathbb{E}\left[Z_{t_{k+1}}^\pi\middle|\mathcal{F}_{t_k}\right] + \frac{1}{\theta_2\Delta_n}\mathbb{E}\left[Y_{t_{k+1}}^\pi\Delta W_k\middle|\mathcal{F}_{t_k}\right]$$
$$+ \frac{1-\theta_2}{\theta_2}\mathbb{E}\left[f(t_{k+1}, Y_{t_{k+1}}^\pi, Z_{t_{k+1}}^\pi)\Delta W_k\middle|\mathcal{F}_{t_k}\right],$$
$$Y_{t_k}^\pi = \mathbb{E}\left[Y_{t_{k+1}}^\pi\middle|\mathcal{F}_{t_k}\right] + \theta_1\Delta_n f(t_k, Y_{t_k}^\pi, Z_{t_k}^\pi)$$
$$+ (1-\theta_1)\Delta_n\mathbb{E}\left[f(t_{k+1}, Y_{t_{k+1}}^\pi, Z_{t_{k+1}}^\pi)\middle|\mathcal{F}_{t_k}\right],$$

(3.6)

for $k \in \{n-1, \cdots, 0\}$ backwards, with $\theta_2 \in (0,1]$. The FBSDE and PDE equivalence (Theorem 2.3) is applied in computing the terminal condition for $Z$. Hence, Assumption 2 is essential here. The above scheme is proved in [328] to converge with order of convergence 2 when $\theta_1 = \theta_2 = 1/2$ and order of convergence 1 otherwise.

Since its first development [323], the $\theta$-scheme has been extensively generalized in various directions. For instance, as previously mentioned, for BSDEs whose driver is independent of the control process $Z$, the $\theta$-scheme is proved [328] to converge with order 2 when $\theta_1 = \theta_2 = 1/2$ and order 1 otherwise. In order to achieve the same orders (2 when $\theta_1 = \theta_2 = 1/2$ and order 1 otherwise) even when the driver depends on the pair $(Y, Z)$, the $\theta$-scheme is generalized [327] by introducing more parameters. The concept of the $\theta$-scheme has been applied to more general problem settings, for instance, mean-field BSDEs [283], BSDEs whose driver contains not only the present value $(Y_t, Z_t)$ but also the future values [173], and $G$-BSDEs [174]. In particular, the $\theta$-scheme with $\theta = 1/2$, termed the Crank-Nicolson scheme [301], is investigated along with error analysis [218, 326] and applications to FBSDEs [217], where higher-order discretization formulas are derived for the control process $Z$ via approximation of the derivative of the backward process $Y$ by the well-known Crank-Nicolson method, as well as to mean-field BSDEs [321]. Inspired by the $\theta$-scheme, a numerical method for one-dimensional FBSDEs is proposed [277] without using conditional expectations, along with a few analytical solutions of particular FBSDEs. From an implementation point of view, the $\theta$-scheme has been further accelerated via parallel computing [87].

To achieve even higher orders of convergence, the multistep scheme [329] is developed as an extension of the $\theta$-scheme with convergence at an arbitrary order when the driver is independent of the control process. In the multistep scheme, the pair $(Y_{t_k}^\pi, Z_{t_k}^\pi)$ is approximated based on the expectation of values at multiple future time points $\{(Y_{t_l}^\pi, Z_{t_l}^\pi)\}_{l \in \{k+1, \cdots, k+L\}}$, for which evidently more

extensive computing is required than the backward Euler method (Section 3.1) and the $\theta$-scheme. Error analyses are conducted in the cases where the driver depends on the control process [58] and with variable step sizes [164], backed up by convincing numerical results. Moreover, by exploiting the independence of the computation at each grid point, the multistep scheme is tailored [201] for parallelized computing on GPUs.

The concept of multistep methods has also been quite prevalent in the context of FBSDEs. A multistep scheme is developed for coupled FBSDEs [324] along with a crude Euler-Maruyama method employed for discretizing the forward component, with convergence analysis refined in [309]. The multistep method has been generalized to address more general BSDEs with nonstandard features (Section 8), such as BSDEs with jumps [112, 114], second-order FBSDEs [332] and fully nonlinear parabolic PDEs [207], as well as improved in combination with relevant techniques, such as the sparse-grid method [315], the spectral sparse grid approximation with fast Fourier transform [115], the Lagrange interpolation polynomial [221, 222], finite difference for approximating the derivatives of the solution of FBSDEs in multi-time levels [294], 3-point Gauss-Hermite quadrature rule and non-equidistant difference scheme [253], polynomial approximation [331], and predictor-corrector and least-squares Monte Carlo schemes [165]. Naturally, the multistep method for BSDEs and FBSDEs parallels with the relevant numerical methods for ODEs and PDEs. For instance, the Runge-Kutta method for ODEs is applied to BSDEs [61], where the order barrier is exhibited to be more restrictive for BSDEs than for ODEs. The defferred correction method for ODEs is tailored to coupled FBSDEs [288] and second-order FBSDEs [310]. It is proved [289] that a linear multistep method for FBSDEs is stable as long as the so-called root condition is met in a similar manner to that for ODEs. A general framework is constructed [311] to investigate the stability, consistency and convergence of discretization schemes for FBSDEs in a unified manner, including the backward Euler method, the $\theta$-scheme and various types of the multistep method. In [200, 201], the authors demonstrate an acceleration of the multistep method for FBSDEs, relying on parallel GPU computing using CUDA. It is natural to employ established higher-order discretization schemes for the forward component, such as the Milstein and stochastic Taylor methods, together with the multistep method for BSDEs [113, 325, 330]. Along this line, various first- and second-order discretization methods have been developed by combining the multistep method for the backward component (such as the trapezoidal rule) with a suitable discretization scheme for the forward component, and then compute the resulting conditional expectations efficiently by the Gauss-Hermite quadrature rule [330], a Lagrangian interpolation [113], and an approach with Malliavin derivative [325]. Error estimates are improved in [320] for the schemes developed in [330, 331]. Those methods are also applied to mean-field FBSDEs [282].

In [81], a second order discretization scheme is constructed for BSDEs with a nonsmooth boundary data using Brownian weights for the approximation of the $Z$ component under weaker conditions, where the gradient estimate of [78] is employed. In order to construct numerical methods for these FBSDEs, the

backward component is discretized by the trapezoidal or Simpson's rule, and the forward component by the cubature method on Wiener space (Section 4.5). Such methods result in the orders of convergence 1 [80], 2 [81], and up to 3 [247]. A similar approach is employed in [88] for discretizing McKean-Vlasov FBSDEs. When the forward and backward components are discretized, respectively, by the cubature and backward Euler methods, one can construct a discretization method on the basis of an explicit error expansion [66], resulting in an arbitrary order of convergence with the aid of the Richardson-Romberg extrapolation. Inspired by the development of [81], a second-order discretization method is proposed in [241], by introducing higher-order correction terms for the backward component in the form of a sum of polynomials of the Brownian motion through the Malliavin weights.

As previously mentioned, the computation of conditional expectations is undoubtedly a major step in higher-order discretization methods (Section 4). Here, Fourier analysis may play central roles for various purposes [68, 126, 180, 190, 251, 275, 302], for instance, via the characteristic function of the discretized forward component [274]. A tree-based regression is suggested in [291] for efficiently implementing higher-order discretization methods. In conjunction with the four-step scheme [226], higher-order discretization methods are investigated, for instance, on an approximation of spatial derivatives in the coefficients of the four-step scheme using finite difference [238], and with the Hermite-spectral method [227]. Finally, in [265], the authors explored the potential for parallel computing with FBSDEs when using the binomial tree type approximation. Due to the special structure of the numerical method proposed (which is similar to that in [179]), a block allocation algorithm is developed in parallelization, where large communication overhead is avoided, backed up by numerical illustration of encouraging speedups for the parallel implementation.

## 4. Computation of conditional expectations

In the present section, we survey major techniques for the computation of conditional expectations in the context of numerical methods (mostly, backward numerical methods (Section 3)) for BSDEs, namely, least-squares regression based methods (Section 4.1), Malliavin calculus based methods (Section 4.2), quantization methods (Section 4.3), tree based methods (Section 4.4) and cubature methods (Section 4.5). We remark that some deep learning based methods may also be interpreted as a mechanism for computing the conditional expectation in numerical methods for BSDEs, using nonlinear least-squares Monte Carlo with a deep neural network architecture. We do not review those methods here but review deep learning based methods collectively in Section 6, as this topic forms a rapidly growing field of research deserving of a separate section.

### *4.1. Least-squares regression based methods*

Methods that fit into this category are ones which use a form of least-squares regression to evaluate the conditional expectations appearing in a discretization

of the BSDE.

Here, we proceed with representative methods of [140, 213] to illustrate this type of method in a clear and concise manner. Let $\Delta_k = T/n$, that is, $t_k = kT/n$ and $\Delta W_k = W_{t_{k+1}} - W_{t_k}$ for $k \in \{0, 1, \cdots, n\}$. For each time point $t_k$, we take $\mathbb{R}^K$-valued deterministic function bases $(e_{i,k}^K)_{i \in \{0,1,\cdots,d\}}$, whose elements are given by, for instance, the sequence of Hermite polynomials or that of Laguerre polynomials of size $K$. For every $k \in \{1, \cdots, n\}$, let $\{\Delta W_k^m\}_{m \in \{1,\cdots,M\}}$ be independent copies of $\Delta W_k$, and let $\{X_k^{\pi,m}\}_{m \in \{1,\cdots,M\}}$ be corresponding copies of $X_k^\pi$. An algorithm for approximating the backward Euler scheme $(Y_k^\pi, Z_k^\pi)$ can be described as follows.

---

**Algorithm 2:**

---

Set $y_n^{n,M,K}(\cdot) = \Phi$

**for** $k = (n-1)$ to 1 **do**

Find

$$\alpha_{i,k}^{M,K} = \underset{\alpha}{\text{argmin}} \frac{1}{M} \sum_{m=1}^M \left| y_{k+1}^{n,M,K}(X_{k+1}^{\pi,m}) \frac{\Delta W_k^{i,m}}{\Delta_k} - \alpha \cdot e_{i,k}^K(X_k^{\pi,m}) \right|^2, \quad (4.1)$$

for $i \in \{1, \cdots, d\}$.
Put $z_k^{n,M,K} = (z_{1,k}^{n,M,K}, \cdots, z_{d,k}^{n,M,K})$ where $z_{i,k}^{n,M,K}(\cdot) = \alpha_{i,k}^{M,K} \cdot e_{i,k}^K(\cdot)$.

$$\alpha_{0,k}^{M,K} = \underset{\alpha}{\text{argmin}} \frac{1}{M} \sum_{m=1}^M \left| y_{k+1}^{n,M,K}(X_{k+1}^{\pi,m}) \right.$$

$$\left. + \Delta_k f(t_k, X_k^{\pi,m}, y_{k+1}^{n,M,K}(X_{k+1}^{\pi,m}), z_k^{n,M,K}(X_k^{\pi,m})) - \alpha \cdot e_{0,k}^K(X_k^{\pi,m}) \right|^2.$$
$$(4.2)$$

Put $y_k^{n,M,K}(\cdot) = \alpha_k^{M,K} \cdot e_{0,k}^K(\cdot)$.

**end**

Return

$$Y_0^{\pi,M,K} = \frac{1}{M} \sum_{m=1}^M (y_1^{n,M,K}(X_1^{\pi,m}) + \Delta_1 f(t_0, \mathbf{x}, y_1^{n,M,K}(X_1^{\pi,m}), z_1^{n,M,K}(X_1^{\pi,m}))).$$

---

Under the Lipschitz continuity in the state variables, the $(1/2)$-Hölder continuity in time of the coefficients of the Markovian BSDE, and the condition that for all measurable functions $\varphi$ such that $\varphi(X_k^\pi) \in L^2(\Omega)$, there is $(\beta_k^K)_k$ such that $\beta^K \cdot e_{i,k}^K(X_k^\pi) \to \varphi(X_k^\pi)$ in $L^2(\Omega, \mathscr{F}^{X_k^\pi})$ as $K \to \infty$. It is shown in [318] that $\lim_{K \to +\infty} y_k^{n,K}(X_k^\pi) = Y_k^\pi$ in $L^2(\Omega)$ for all $k \in \{0, 1, \cdots, n-1\}$, where $y_k^{n,K}$ and $z_k^{n,K}$ denote the theoretical means of the respective empirical ones $y_k^{n,M,K}$ and $z_k^{n,M,K}$ of (4.1) and (4.2) above, as well as

$$\lim_{M \to +\infty} Y_0^{\pi,M,K} = Y_0^{\pi,K},$$

almost surely, by the strong law of large numbers. While the convergence holds by taking sufficiently large $K$ and $M$, its rate has been found to be quite com-

plex. It is shown in [213], which extends [140], that the squared error of the time discretization is of the order $\mathcal{O}(n^{-1})$, and roughly speaking, the numbers of basis functions and of the paths are to be chosen as $K \approx n^d$ and $M \approx n^{d+3}$ in order to control the global squared error to be $\mathcal{O}(n^{-1})$, upon a suitable choice of basis functions. We refer the reader to, for instance, [26, 140, 213, 318] for detailed analyses.

In a similar spirit, a numerical scheme is developed in [147] for solving the multi-step forward dynamic programming equation arising from the discretization of the FBSDE. The resulting sequence of conditional expectations is computed using empirical least-squares regressions. Also, in [142], we see another algorithm based on least-squares Monte Carlo (LSMC). Here, the algorithm is designed to allow large scale parallelization of the computations on highly multicore processors, such as GPUs by stratifying sample paths to minimize the exposure to the memory requirements due to the storage of simulations. Possible discontinuities at the interfaces due to piecewise polynomial bases [142] are avoided [141, 146] by instead employing smooth orthonormal basis functions. Also, [26, 140] propose numerical schemes based on iterative least-square regressions on function bases, where the involved coefficients are evaluated using Monte Carlo simulations. We mention the so-called regression-later approach [134], which can be thought of as a variant of least-squares regression methods.

The primal-dual methodology is generalized in [25] to a backward dynamic programming equation associated with time discretization schemes of reflected BSDEs (Section 8.2). They suggest a pathwise approach to the dynamic programming equation, which avoids the evaluation of conditional expectations in backward recursion in time. This approach leads to a minimization problem and is thus thought of as a dual minimization problem. Under suitable assumptions, $Y$ can also be represented as the supremum over a class of classical optimal stopping problems, and this maximization problem can be seen as a primal problem. Using the representations for $Y$ as the value of a maximization and a minimization problem, confidence intervals are constructed for $Y_0$ using LSMC (even if only to generate an initial input approximation), along with a few numerical examples to test the performance of the algorithm for multi-dimensional reflected BSDEs in nonlinear pricing problems.

In [69], the authors apply the Stochastic Grid Bundling Method (SGBM) to develop a numerical scheme. The SGBM algorithm involves the approximation of conditional expectations by means of bundling Monte Carlo sample paths and a local regress-later technique within each bundle. By employing Hermite martingales, the problem of solving a FBSDE is formulated in [257] as the problem of solving a countably infinite-dimensional system of ODEs. On this basis, they develop a numerical scheme which involves the projection of the solution onto generalized Hermite polynomials. Similarly, a numerical scheme is proposed in [293] based on the projection of conditional expectations onto cubic spline polynomials. Finally, we refer to [241] for a second-order discretization method for FBSDEs based on an algorithm which utilizes polynomials of Brownian motions and is implemented by use of a least-squares Monte Carlo method.

### *4.2. Malliavin calculus based methods*

Malliavin calculus based methods are closely related to least-squares regression based methods (Section 4.1) in the sense that a lot use least-squares regression to solve the conditional expectations resulting from a discretization of a BSDE which incorporates Malliavin weights.

### *4.2.1. Representation of conditional expectations using integration-by-parts*

Malliavin calculus was first applied to BSDEs in [47] for the purpose of computing conditional expectations in line with [41, 111], for instance, the two terms $\mathbb{E}[Y^n_{t_{k+1}}(W_{t_{k+1}} - W_{t_k})^\top | \mathcal{F}_{t_k}]$ and $\mathbb{E}[Y^n_{t_{k+1}} | \mathcal{F}_{t_k}]$ appearing in the implicit discretization scheme BSDEs (3.1) and (3.2). With the aid of the Markovian property, those conditional expectations boil down to the form $\mathbb{E}[\phi(X_s)| X_t = \mathbf{x}]$ for $s \in (t, T]$, where $\phi : \mathbb{R}^q \to \mathbb{R}$ and $\{X_t : t \in [0, T]\}$ is the forward process (2.7). Here, we describe in brief how this conditional expectation can be reformulated without conditioning in such a way that Monte Carlo methods can play an effective role. In general, for two smooth Wiener functionals $F$ and $G$ in the sense of Malliavin, if $G$ is "non-degenerate", then the conditional expectation $\mathbb{E}[F|G = \mathbf{x}]$ can be written as fractions:

$$\mathbb{E}[F|G = \mathbf{x}] = \frac{\mathbb{E}[F\delta_\mathbf{x}(G)]}{\mathbb{E}[\delta_\mathbf{x}(G)]} = \frac{\mathbb{E}[H_\mathbf{x}(G)\delta^W(Fu)]}{\mathbb{E}[H_\mathbf{x}(G)\delta^W(u)]}, \qquad (4.3)$$

where $\delta_\mathbf{x}(G)$ is understood to be the composition of the delta function at $\mathbf{x}$ and $G$ as a Watanabe distribution on the Wiener space, $H_\mathbf{x}$ denotes a Heaviside function, that is, $H_\mathbf{x}(\mathbf{c}) := \prod_{k=1}^q \mathbb{1}(c_k \geq x_k)$, and $\delta^W$ denotes the Skorohod integral operator. The second equality is due to Malliavin integration by parts and the process $u$ satisfying $\int_0^T D_s Gu(s)ds = 1$. Under the further constraint $\int_0^T D_s Fu(s)ds = 0$, by applying the property of the Skorohod integral operator $\delta^W(Fu) = F\delta^W(u) - \int_0^T D_s Fu(s)ds$ and setting $F = \phi(X_s)$, $G = X_t$ and $u(s) = (D_s X_t)^{-1}(t^{-1}\mathbb{1}(s \in (0, t]) - (T - t)^{-1}\mathbb{1}(s \in (t, T]))$, the last expression in (4.3) can be rewritten in the following form, which lends itself to estimation of the conditional expectation $\mathbb{E}[\phi(X_s)| X_t = \mathbf{x}]$ by Monte Carlo methods:

$$\mathbb{E}[\phi(X_s)|X_t = \mathbf{x}] = \frac{\mathbb{E}[\phi(X_s)H_\mathbf{x}(X_t)\delta^W(u)]}{\mathbb{E}[H_\mathbf{x}(X_t)\delta^W(u)]}$$

$$\approx \frac{\frac{1}{M}\sum_{k=1}^M \phi(X_s^k)H_\mathbf{x}(X_t^k)\delta^{W^k}(u)}{\frac{1}{M}\sum_{k=1}^M H_\mathbf{x}(X_t^k)\delta^{W^k}(u)}, \qquad (4.4)$$

by generating independent copies with sufficiently large sample size $M$. This approach can also be applied to reflected BSDEs (Section 8.2) in [47].

As is clear, the effectiveness of this approach depends, next to the Heaviside function, on the Skorohod integral term $\delta^W(u)$. In principle, one needs a reasonably explicit expression for this term, without which simulation would be

essentially impossible. Even if one has an explicit form, its computation can be prohibitive when the problem dimension is high. To address this issue, a variant of the method above is proposed in [82] for a significant reduction of the numerical complexity by wisely modifying the Skorohod integral term $\delta^W(u)$ in such a way to reduce the terms involved and skip differentiation of the drift and diffusion coefficients of the forward component.

### 4.2.2. *Malliavin weights dynamic programming with regression*

We next describe a dynamic programming scheme with the regression methods, named the Malliavin weights dynamics programming, an application of Malliavin calculus in a different spirit from Section 4.2.1. For illustrative purposes, we proceed with the regression method [148], among many candidates, that solves a dynamic programming equation with Malliavin weights arising from the time-discretization of FBSDEs (which may also be fit for Section 3).

Now, the main idea of the dynamic programming scheme [148] is to approximate the solution $(Y, Z)$ of a FBSDE by the discrete time stochastic processes $(Y^\pi, Z^\pi)$, defined on the partition $\pi_n$, which are given as follows:

$$\begin{cases} Y_k^\pi = \mathbb{E}\left[\Phi(X_T) + \sum_{j=k}^{n-1} f_j(Y_{j+1}^\pi, Z_j^\pi)\Delta_j \,\Big|\, \mathcal{F}_{t_k}\right], \\ Z_k^\pi = \mathbb{E}\left[\Phi(X_T)H_n^{(k)} + \sum_{j=k+1}^{n-1} f_j(Y_{j+1}^\pi, Z_j^\pi)H_j^{(k)}\Delta_j \,\Big|\, \mathcal{F}_{t_k}\right], \end{cases} \tag{4.5}$$

for $k \in \{0, \cdots, n\}$ for the $Y$ component, and for $k \in \{0, \cdots, n-1\}$ for the $Z$ component, where $(\omega, y, \mathbf{z}) \to f_j(\omega, y, \mathbf{z})$ is $\mathcal{F}_{t_j} \otimes \mathcal{B}(\mathbb{R}) \otimes \mathcal{B}(\mathbb{R}^{1 \times d})$-measurable. This system is then solved backwards in the order $Y_n^\pi, Z_{n-1}^\pi, Y_{n-1}^\pi$ and so forth. It is called the Malliavin weights dynamic programming (MWDP) equation, as it literally takes the form of a dynamic programming equation with Malliavin weights. The MWDP equation (4.5) is inspired by [228], which gives the following representation of the control process $Z$:

$$Z_t = \mathbb{E}\left[\Phi(X_T)H_T^{(t)} + \int_t^T f(s, X_s, Y_s, Z_s)H_s^{(t)} ds \,\Big|\, \mathcal{F}_t\right], \tag{4.6}$$

where the processes $(H_s^{(t)})_{0 \le t < s \le T}$ are the Malliavin weights defined as follows:

$$H_s^{(t)} = \frac{1}{s-t}\left(\int_t^s \left(\sigma^{-1}(r, X_r)D_t X_r\right)^\top dW_r\right)^\top, \quad 0 \le t < s \le T, \tag{4.7}$$

with $(D_t X_r)_t$ denoting the Malliavin derivative of the marginal $X_r$. We note that this method adapts the least-squares multi-step forward dynamic programming algorithm to a scenario which incorporates Malliavin weights. Under suitable technical conditions presented in [147], it holds almost surely that $|Y_k^\pi| \le C(1 + (T - t_k)^{\theta_c}) =: C_{y,k}$ and

$$\|Z_k^\pi\| \le C\left(\frac{\mathrm{esssup}_\omega\, \mathbb{E}[|\Phi(X_T) - \mathbb{E}[\Phi(X_T)|\,\mathcal{F}_{t_k}]|^2|\,\mathcal{F}_{t_k}](\omega)}{(T-t_k)^{1/2}}\right.$$

$$\left. + \frac{1}{(T-t_k)^{1/2-\theta_c}} + (T-t_k)^{\theta_L/2}\right) =: C_{z,k},$$

for all $k \in \{0, \cdots, n-1\}$. Furthermore, there exists a constant $C_{y,z}$ such that $|Y_k^\pi| + \sqrt{T-t_k}|Z_k^\pi| \le C_{y,z}$ almost surely for all $k \in \{0, \cdots, n\}$.

The conditional expectations (and hence processes $Y^\pi$ and $Z^\pi$) appearing in the MWDP equation (4.5) are computed using a Monte Carlo least-squares regression scheme (Section 4.1). Due to the Markovian assumptions, there exist measurable and deterministic (but unknown) functions $y_k(\cdot) : \mathbb{R}^q \to \mathbb{R}$ and $z_k(\cdot) : \mathbb{R}^q \to \mathbb{R}^{1\times d}$ for all $k \in \{0, \cdots, n-1\}$ such that the solution $(Y_k^\pi, Z_k^\pi)_{k\in\{0,\cdots,n-1\}}$ of the MWDP equation (4.5) is given by $(Y_k^\pi, Z_k^\pi) := (y_k(X_k^\pi), z_k(X_k^\pi))$. The aim of the Monte Carlo regression scheme is thus to estimate these functions. Ordinary least-squares regression (OLS) is defined in such a way that easily allows path-dependence and joint laws. The authors reformulate the MWDP equation (4.5) in terms of the given definition of OLS. Specifically, take $\mathcal{K}_k^{(l)}$ to be any dense subset in the $\mathbb{R}^l$-valued functions belonging to $L_2(\mathcal{B}(\mathbb{R}^q), \mathbb{P} \circ (X_k^\pi)^{-1})$, and then for each $k \in \{0, \cdots, n-1\}$,

$$\begin{cases} y_k(\cdot) \text{ solves OLS}\Big(\Phi(x_n) + \sum_{j=k}^{n-1} f_j(y_{j+1}(x_{j+1}), z_j(x_j))\Delta_j, \\ \qquad\qquad \mathcal{K}_k^{(1)}, \mathbb{P} \circ (H_{k+1}^{(k)}, \cdots, H_n^{(k)}, X_k^\pi, \cdots, X_n^\pi)^{-1}\Big), \\ z_k(\cdot) \text{ solves OLS}\Big(\Phi(x_n)h_n + \sum_{j=k+1}^{n-1} f_j(y_{j+1}(x_{j+1}), z_j(x_j))h_j\Delta_j, \\ \qquad\qquad \mathcal{K}_k^{(d)}, \mathbb{P} \circ (H_{k+1}^{(k)}, \cdots, H_n^{(k)}, X_k^\pi, \cdots, X_n^\pi)^{-1}\Big), \end{cases} \quad (4.8)$$

for $(h_{k+1}, \cdots, h_n) \in (\mathbb{R}^{1\times d})^{n-k}$ and $(x_k, \cdots, x_n) \in (\mathbb{R}^q)^{n-k+1}$. In words, for instance, $y_k(\cdot)$ is set to be the least-squares approximation of the term $\Phi(x_n) + \sum_{j=k}^{n-1} f_j(y_{j+1}(x_{j+1}), z_j(x_j))\Delta_j$ in the space $\mathcal{K}_k^{(1)}$ with respect to the law $\mathbb{P} \circ (H_{k+1}^{(k)}, \cdots, H_n^{(k)}, X_k^\pi, \cdots, X_n^\pi)^{-1}$. However, the above least-squares regressions give rise to two computational problems, that is, $L_2(\mathcal{B}(\mathbb{R}^q), \mathbb{P} \circ (X_k^\pi)^{-1})$ is often infinite dimensional, and the integrals of the OLS in (4.8) are presumably computed using the untraceable law of $(H_{k+1}^{(k)}, \cdots, H_n^{(k)}, X_k^\pi, \cdots, X_n^\pi)$. These issues are addressed by approximating $y_k(\cdot)$ and $z_k(\cdot)$ on finite-dimensional function spaces $\mathcal{K}_{Y,k}$ and $\mathcal{K}_{Z,k}$, respectively, with respect to the empirical version $\nu_{k,M}$ of the law $\mathbb{P} \circ (H_{k+1}^{(k)}, \cdots, H_n^{(k)}, X_k^\pi, \cdots, X_n^\pi)^{-1}$ based on $M$ iid realizations.

The global error of the algorithm is a weighted time-average of three different errors. The first is the approximation error which relates to the error involved in approximating the functions $(y_k, z_k)$ in the finite dimensional approximation spaces. This accuracy is achieved asymptotically as the number of simulations goes to infinity. The second error term is the usual statistical error,

which improves with a larger number of simulations or a smaller dimension of the vector spaces. The third and final error term together is the interdependence error which is related to the inter dependencies between regressions at different times. This error is of the same magnitude as the statistical error terms, up to logarithmic factors. Thus, roughly speaking, the global error is of order of the best approximation errors plus the statistical errors.

A similar method can be found in [149], where the solution of the FBSDE is approximated by using a backward MWDP equation and LSMC regression (Section 4.1), along with importance sampling to minimize the conditional variance occurring in the LSMC algorithm and accelerate the convergence of Monte Carlo approximation in a similar manner to [23]. The Radon-Nikodym derivative is not given, as in [23], but rather computed adaptively within the LSMC procedure. However, it makes sense to apply importance sampling only if the driver is independent of $Z$. If the driver did depend on $Z$, there would be a propagation of "lack of variance reduction" on the $Y$ component due to the $Z$ component through the driver. This means that it would not be possible to keep track of the benefit of importance sampling for $Y$. If the Monte Carlo estimation of $Z$ is made with appropriate variance reduction (suited to $Z$ specifically), then this problem would be avoided, and would allow the driver to depend on $Z$, but this is left to future investigation.

### *4.3. Quantization methods*

The quantization method is a numerical scheme for computing the conditional expectation $\mathbb{E}[h(X_{k+1})|X_k = \mathbf{x}]$ for a $\mathbb{R}^q$-valued Markov chain $\{X_k\}_{k \in \{0,1,\cdots,n\}}$ by settling the chain onto a space grid $\Gamma_k := \{x_k^1, \ldots, x_k^{N_k}\}(\subset \mathbb{R}^q)$ of a suitable cardinality $N_k \in \mathbb{N}$. The quantization method can be regarded somewhere in the middle of deterministic and probabilistic methods, because it relies on space grids and weights just like deterministic methods, while the computation of weights often requires Monte Carlo methods. Such quantization methods are developed and examined in [11, 12, 13] for probabilistically solving multidimensional optimal stopping problems, and then applied to develop discretization schemes for reflected BSDEs (Section 8.2) based on an optimal discrete spatial quantization tree. Further improvements on quantized BSDE schemes are provided in [54, 109, 248, 252].

Here, we describe quantized BSDE schemes in brief in accordance with [109]. On the basis of the explicit backward Euler scheme (3.1) and (3.2), we denote by $\widehat{X}_{t_k}^n$ the so-called quantization of $X_{t_k}^n$ taking values in a finite grid $\Gamma_k(\subset \mathbb{R}^d)$, that is, $\widehat{X}_{t_k}^n := \mathrm{Proj}_{\Gamma_k}(X_{t_k}^n)$ where $\mathrm{Proj}_{\Gamma_k}$ is a Borel projection of $\mathbb{R}^d$ onto the grid $\Gamma_k$. For example, for a quantization level $N_k \in \mathbb{N}$, an optimal quantization grid $\Gamma_k$ is found according to $\|X_{t_k}^n - \mathrm{Proj}_{\Gamma_k}(X_{t_k}^n)\|_2 = \inf\{\|X_{t_k}^n - p(X_{t_k}^n)\|_2 :$ Borel measurable $p : \mathbb{R}^q \to \Gamma(\subset \mathbb{R}^q), \ \mathrm{card}(\Gamma) \leq N_k\}$, based on the nearest neighbor projection $\mathrm{Proj}_{\Gamma_k}$, given by $\mathrm{Proj}_{\Gamma_k}(X_{t_k}^n) = \sum_{i=1}^{N_k} x_k^i \mathbb{1}(X_{t_k}^n \in C_i^k(\Gamma_k))$,

where $N_k := \mathrm{card}(\Gamma_k)$ and $\{C_i^k(\Gamma_k)\}_{i \in \{1, \cdots, N_k\}}$ is a sequence of Voronoi partitions of $\mathbb{R}^q$, satisfying $C_i^k(\Gamma_k) \subset \{\xi \in \mathbb{R}^q : |x_k^i - \xi| = \min_{j \in \{1, \cdots, N_k\}} |x_k^j - \xi|\}$. A quantized BSDE can then be defined as follows: $\widehat{y}_n(x_i) = \Phi(x_i^n)$ for $i \in \{1, \cdots, N_n\}$, and then for $k \in \{n-1, \cdots, 1, 0\}$ backwards,

$$\widehat{y}_k(x_k^i) = \widehat{\alpha}_k(x_k^i) + f(t_k, x_k^i, \widehat{\alpha}_k(x_k^i), \widehat{\beta}_k(x_k^i))(t_{k+1} - t_k), \quad x_k^i \in \Gamma_k,$$

for $i \in \{1, \cdots, N_k\}$, where

$$\widehat{\alpha}_k(x_k^i) := \sum_{j=1}^{N_{k+1}} \widehat{y}_{k+1}(x_{k+1}^j) p_k^{ij}, \quad \widehat{\beta}_k(x_k^i) := \frac{1}{\sqrt{t_{k+1} - t_k}} \sum_{j=1}^{N_{k+1}} \widehat{y}_{k+1}(x_{k+1}^j) \Lambda_k^{ij},$$

with weights given in the forms of conditional probability and expectations:

$$p_k^{ij} := \mathbb{P}\left(\widehat{X}_{k+1} = x_{k+1}^j \,\middle|\, \widehat{X}_k = x_k^i\right),$$

$$\Lambda_k^{ij} := \mathbb{E}\left[(W_{t_{k+1}} - W_{t_k})\mathbb{1}(\widehat{X}_{k+1} = x_{k+1}^j) \,\middle|\, \widehat{X}_k = x_k^i\right],$$

which are to be estimated by Monte Carlo methods if analytic formulas are not available.

In principle, more accurate results are naturally expected by increasing the size of the quantizations, whereas from an implementation point of view, the size should be chosen carefully and reasonably, relative to the problem dimension, especially when the dynamics is multivariate with dependent components, since then the conditional distribution cannot be decomposed into a set of independent univariate distributions. The complexity of the quantization methods comes largely from the computation of the optimal quantizers and their transition probabilities, if Monte Carlo simulation needs to be employed for the estimation of the transition probabilities.

### 4.4. Tree based methods

Tree based methods have been found effective in the computation of conditional expectations, particularly in low-dimensional problems, with relevant analyses presented in [49, 225] (while its origin may date back to [72]). In order to illustrate the basics of tree based methods, we describe a numerical method proposed in [225] for a BSDE whose driver is independent of $Z$. The main idea is to approximate the Brownian motion $W$ by a simple random walk $B_t^n := \frac{1}{\sqrt{n}} \sum_{k=0}^{\lfloor nt \rfloor} \zeta_k^n$ for $t \in [0, 1]$, where $\{\zeta_k^n\}_{k \in \{1, \cdots, n\}}$ is a sequence of iid Rademacher random variables. Then, they use this approximation in a discretized version of the BSDE with a constant time step $\Delta := 1/n$, whose solution is denoted $(Y^n, Z^n)$. The solution to this discretized BSDE is then approximated by $(\widetilde{Y}^n, \widetilde{Z}^n)$, which satisfies the following algorithm.

---

**Algorithm 3:**

---

**Initialization**: Approximate the terminal conditions $\widetilde{Y}_{t_n}^n = \xi^n$ and $\widetilde{Z}_{t_n}^n = 0$.

**for** $k = (n-1)$ to $0$ **do**

$$\widetilde{X}_{t_k}^n = \mathbb{E}\left[\widetilde{Y}_{t_{k+1}}^n \,\Big|\, \mathcal{G}_k^n\right], \quad \widetilde{Y}_{t_k}^n = \widetilde{X}_{t_k}^n + \frac{1}{n}f(t_k, \widetilde{X}_{t_k}^n),$$

$$\widetilde{Z}_{t_k}^n = \mathbb{E}\left[\left(\widetilde{Y}_{t_{k+1}} + \frac{1}{n}f(t_k, \widetilde{Y}_{t_k}^n) - \widetilde{Y}_{t_k}^n\right)(\Delta B_{t_{k+1}}^n)^{-1}\,\Big|\, \mathcal{G}_k^n\right].$$

**end**

---

The involved conditional expectations are computed using a tree structure. We briefly look at the error involved in approximating $(Y, Z)$ by $(Y^n, Z^n)$ (and $(Y^n, Z^n)$ by $(\widetilde{Y}^n, \widetilde{Z}^n)$). It is assumed that $\xi = F(W)$ and thus $\xi^n = F(B^n)$, where $F : \Omega \to \mathbb{R}$ is a bounded Lipschitz function with respect to the uniform topology on $\Omega$, that is, $|F(\omega_1) - F(\omega_2)| \leq C\sup_{t\in[0,1]}|\omega_1(t) - \omega_2(t)|$ for all $\omega_1, \omega_2 \in \Omega$. The authors prove that it can then be assumed without loss of generality that the driver is bounded and furthermore that for all $k \in \{0, \cdots, n\}$ and for large $n$,

$$\sup_{\omega\in\Omega} |Y_{t_k}^n - \widetilde{Y}_{t_k}^n| \lesssim \frac{e^{2c}-1}{n}, \quad \sup_{\omega\in\Omega} |Z_{t_k}^n - \widetilde{Z}_{t_k}^n| \lesssim \frac{(e^{2c}-1)(2+c/n)}{\sqrt{n}},$$

where $c$ denotes the Lipschitz coefficient of the driver. Moreover, the sequence $(Y^n, U^n)$ is shown to converge weakly in the Skorohod topology to $(Y, \int ZdW)$, where

$$U_{t_k}^n = \sum_{j=0}^{k-1} Z_{t_j}^n \Delta B_{t_{j+1}}^n = Y^n - F(B^n) - \frac{1}{n}\sum_{j=1}^{k} f(t_j, Y_{t_j}^n).$$

We also mention Donsker's theorem which has been derived in the relevant context [49, 51]. In addition, stability and convergence of the discretized filtration have been analyzed in [9, 73, 74]. With the aid of those results, an approximation is investigated [50] in the sense that a sequence of solutions to a BSDE driven by a martingale, approximating the Brownian motion, converges to the solution to (2.1), and is further generalized to BSDEs with random terminal time [295].

Tree based methods can also be employed in conjunction with the theta approximation (3.5), which is a convex combination of explicit and implicit terms:

$$Y_{t_k} \approx \mathbb{E}\left[Y_{t_{k+1}} \,\big|\, \mathcal{F}_{t_k}\right] + \Delta_n\theta_1 f(t_k, Y_{t_k}, Z_{t_k})$$
$$+ \Delta_n(1-\theta_1)\mathbb{E}\left[f(t_{k+1}, Y_{t_{k+1}}, Z_{t_{k+1}})\,\big|\, \mathcal{F}_{t_k}\right],$$

for $\theta_1 \in [0, 1]$. Here, the resulting conditional expectations can be approximated on the basis of a recombining tree structure of a random walk approximation for the Brownian motion with the aid of the Markov property of the approximation of the forward process $X$. After all conditional expectations have been

approximated in this way, one obtains the following algorithm (in accordance with the $\theta$-scheme (3.6)) for computing $Y_{j,k}$ and $Z_{j,k}$ at node $(j,k)$ backward in time in the recombining tree structure (for instance, [179]).

---

**Algorithm 4:**

---

**Initialization**: Approximate the terminal conditions $Y_{j,n}^\pi = \Phi(X_{j,n}^\pi)$ and
$\quad Z_{j,n}^\pi = (\nabla \Phi(X_{j,n}^\pi))\sigma(t_n, X_{j,n}^\pi)$ for all $j \in \{0, \cdots, n\}$.
**for** $k = (n-1)$ to $0$ **do**
$\quad$ **for** $j = 0$ to $k$ **do**

$$Z_{j,k}^\pi = \frac{\theta_2 - 1}{2\theta_2} \left[ Z_{j+1,k+1}^\pi + Z_{j,k+1}^\pi \right] + \frac{1}{2\theta_2 \sqrt{\Delta_n}} \left[ Y_{j+1,k+1}^\pi - Y_{j,k+1}^\pi \right]$$
$$+ \frac{1 - \theta_2}{2\theta_2} \sqrt{\Delta_n} \left[ f(t_{k+1}, Y_{j+1,k+1}^\pi, Z_{j+1,k+1}^\pi) - f(t_{k+1}, Y_{j,k+1}^\pi, Z_{j,k+1}^\pi) \right],$$
$$Y_{j,k}^\pi = \frac{1}{2} \left[ Y_{j+1,k+1}^\pi + Y_{j,k+1}^\pi \right] + \theta_1 \Delta_n f(t_k, Y_{j,k}^\pi, Z_{j,k}^\pi)$$
$$+ (1 - \theta_1) \frac{\Delta_n}{2} \left[ f(t_{k+1}, Y_{j+1,k+1}^\pi, Z_{j+1,k+1}^\pi) + f(t_{k+1}, Y_{j,k+1}^\pi, Z_{j,k+1}^\pi) \right].$$

$\quad$ **end**
**end**

---

Since the order of convergence for the tree approximation is 1 and this error dominates the error coming from the $\theta$-discretization, the overall scheme has an order of convergence 1 for any values of $\theta_1 \in [0,1]$ and $\theta_2 \in (0,1]$. The expression for $Y_{j,k}^\pi$ above is given implicitly and thus needs approximation by Picard iterations.

We close this section by mentioning random walk approximations of BSDEs [71, 194, 231, 235, 243, 263] and an $L_2$-convergence of the random walk approximation of BSDEs derived in [129, 130] which makes use of [127]. Finally, we finish by noting that parallel computing on GPUs is found effective in accelerating tree-based methods [264, 265].

## *4.5. Cubature methods*

Finally, we review cubature methods on the Wiener space and its application in computing conditional expectations appearing in discretization methods for BSDEs. Consider the following Stratonovich forward SDE and backward SDE:

$$dX_s^{t,\mathbf{x}} = V_0(X_s^{t,\mathbf{x}})ds + \sum_{i=1}^d V_i(X_s^{t,\mathbf{x}}) \circ dW_s^i, \quad X_t^{\mathbf{x}} = \mathbf{x},$$
$$-dY_s^{t,\mathbf{x}} = f(X_s^{t,\mathbf{x}}, Y_s^{t,\mathbf{x}}, Z_s^{t,\mathbf{x}})ds - \sum_{i=1}^d Z_s^{i,t,\mathbf{x}}dW_s^i, \quad Y_T^{t,\mathbf{x}} = \Phi(X_T^{t,\mathbf{x}}),$$

where the coefficients satisfy suitable conditions and $\circ$ denotes the symbol for the Stratonovich integral. We say that the positive weights $\{\lambda_k\}_{k \in \{1,\cdots,N\}}$ and the paths of bounded variation $\omega_1, \cdots, \omega_N : [0,t] \to \mathbb{R}^d$ define a cubature formula of degree $m$ at time $t$ if the identity

$$\mathbb{E}\left[\int_{0<t_1<\cdots<t_k<t} \circ dW_{t_1}^{\alpha_1} \cdots \circ dW_{t_k}^{\alpha_k}\right]$$

$$= \sum_{\ell=1}^{N} \lambda_\ell \int_{0<t_1<\cdots<t_k<t} d\omega_{\ell,t_1}^{\alpha_1} \cdots d\omega_{\ell,t_k}^{\alpha_k} \quad (4.9)$$

holds true for any multi-index $\alpha \in \{0, 1, \cdots, d\}^k$ satisfying the condition $\#\{i : \alpha_i \in \{1, \cdots, d\}\} + 2\#\{i : \alpha_i = 0\} \leq m$, with $W_t^0 := t$. With the cubature measure $\mathbb{Q}_t^m := \sum_{\ell=1}^{N} \lambda_l \delta_{\omega_\ell}$ where $\delta_\omega$ denotes the Dirac measure mass at $\omega \in \mathcal{C}([0,t])$, the formula (4.9) can be rewritten as

$$\mathbb{E}\left[\int_{0<t_1<\cdots<t_k<t} \circ dW_{t_1}^{\alpha_1} \cdots \circ dW_{t_k}^{\alpha_k}\right]$$

$$= \mathbb{E}_{\mathbb{Q}_t^m}\left[\int_{0<t_1<\cdots<t_k<t} \circ dW_{t_1}^{\alpha_1} \cdots \circ dW_{t_k}^{\alpha_k}\right].$$

Let $x_\cdot^{0,\mathbf{x}}(\omega)$ be the solution of the ordinary differential equation obtained by replacing the $d$-dimensional Brownian motion $W$ with a path of bounded variation $\omega$ in $X_\cdot^{0,\mathbf{x}}$. Then, we have $\mathbb{E}_{\mathbb{Q}_t^m}[g(X_t^{0,\mathbf{x}})] = \sum_{\ell=1}^{N} \lambda_\ell g(x_t^{0,\mathbf{x}}(\omega_\ell))$ for a function $g : \mathbb{R}^q \to \mathbb{R}$. We note that implementable expressions of cubature measures are available for some specific degrees (such as $m = 3, 5$) and dimensions $d$. Broadly speaking, the cubature measure of degree $m$ yields the following short-time asymptotics:

$$\mathbb{E}\left[g(X_t^{0,\mathbf{x}})\right] - \mathbb{E}_{\mathbb{Q}_t^m}\left[g(X_t^{0,\mathbf{x}})\right] = \mathcal{O}(t^{(m+1)/2}), \quad (4.10)$$

for a smooth function $g$. We note that the error term depends on the bounds of the higher order derivatives of $g$. For instance, the following method provides an implicit computation scheme for the iterative conditional expectations in the backward Euler scheme:

$$\overline{R}_{i,n-1}\Phi(\mathbf{x}) = \mathbb{E}_{\mathbb{Q}_{\Delta_{i+1}}^m}\left[\overline{R}_{i+1,n-1}\Phi(X_{t_{i+1}}^{t_i,\mathbf{x}})\right]$$

$$+ \Delta_{i+1}f\left(\mathbf{x}, \overline{R}_{i,n-1}\Phi(\mathbf{x}), \Delta_{i+1}^{-1}\mathbb{E}_{\mathbb{Q}_{\Delta_{i+1}}^m}\left[\overline{R}_{i+1,n-1}\Phi(X_{t_{i+1}}^{t_i,\mathbf{x}})(W_{t_{i+1}} - W_{t_i})^\top\right]\right),$$

for $i \in \{n - 1, \cdots, 0\}$, with a suitable degree $m$ [80]. In order to reduce its computational cost, some additional techniques, such as the tree-based branching algorithm, are recommended in implementation. A non-uniform partition is found effective to treat possibly nonsmooth boundary data $\Phi$ based on the fact that the corresponding solution of the non-linear PDE $u(t, \cdot)$ is smooth under a suitable condition on $V_i$, even when $\Phi$ is not smooth enough, so that the cubature method (4.10) performs well. Similarly, cubature methods are employed in [81] for constructing a second order discretization scheme, in [247] for a third order scheme, and in [88] for discretizing McKean-Vlasov FBSDEs (Section 8.7). We refer the reader to [66] for an error expansion and the complexity control of the cubature method for solving BSDEs.

## 5. Forward numerical methods

Unlike the backward numerical methods (Section 3), the methods that we review in the present section do not inherently work backwards in time, and thus (originally, at least) avoid the computation of conditional expectations (Section 4). We hence call this category "forward" numerical methods in a collective way. In a similar manner to Section 3, we examine one or two representative methods in some detail in each subsection, followed by a brief overview of various other methods in the category.

### 5.1. Picard iteration methods

In this section, we survey forward numerical methods based on Picard iterations. We start with a forward scheme proposed in [22] via Picard iterations on sample paths, which is aimed at numerically approximating sample paths of nonlinear FBSDEs, based upon the discretization of a Picard type iteration. It is assumed that $W$ and $X$ are multidimensional, while $Y$ is univariate. Also, we impose that $\mu$ and $\sigma$ are $1/2$-Hölder continuous with respect to the time variable, as well as $\Phi$ is Lipschitz. We note that it is not assumed that the matrix $\sigma$ is quadratic or that $\sigma^{\otimes 2}$ is invertible.

The limit of a Picard type iteration is used to approximate the processes $(Y, Z)$. Specifically, we set $(Y^{(0)}, Z^{(0)}) \equiv (0, 0)$, and $(Y^{(r)}, Z^{(r)})$ as the solution of the following FBSDE:

$$Y_t^{(r)} = \Phi(X_T) + \int_t^T f(s, X_s, Y_s^{(r-1)}, Z_s^{(r-1)})ds - \int_t^T Z_s^{(r)}dW_s.$$

By taking conditional expectation, the process $Y^{(r)}$ is given as follows:

$$Y_t^{(r)} = \mathbb{E}\left[\Phi(X_T) - \int_t^T f(s, X_s, Y_s^{(r-1)}, Z_s^{(r-1)})ds \middle| \mathcal{F}_t\right],$$

and $Z^{(r)}$ is given via the martingale representation theorem, meaning the above Picard iteration is implicit. The authors propose a time discretization of the above iteration which is explicit in time, but still requires the evaluation of conditional expectations. Given a partition $\pi_n$ of $[0, T]$ and an approximation $X^{(\pi)}$ of $X$, set $(Y^{(0,\pi)}, Z^{(0,\pi)}) \equiv (0, 0)$. Then, for $k \in \{0, 1, \cdots, n\}$ forward, define the conditional expectations

$$\begin{cases} Y_{t_k}^{(r,\pi)} := \mathbb{E}\left[\Phi(X_T^{(\pi)}) - \sum_{j=k}^{n-1} f(t_j, X_{t_j}^{(\pi)}, Y_{t_j}^{(r-1,\pi)}, Z_{t_j}^{(r-1,\pi)})\Delta_j \middle| \mathcal{F}_{t_k}\right], \\ Z_{l,t_k}^{(r,\pi)} := \mathbb{E}\left[\frac{W_{t_{k+1}}^l - W_{t_k}^l}{\Delta_k}\left(\Phi(X_T^{(\pi)})\right.\right. \\ \qquad\qquad \left.\left. - \sum_{j=k+1}^{n-1} f(t_j, X_{t_j}^{(\pi)}, Y_{t_j}^{(r-1,\pi)}, Z_{t_j}^{(r-1,\pi)})\Delta_j\right) \middle| \mathcal{F}_{t_k}\right], \end{cases}$$

for $l \in \{1, \cdots, d\}$, where $W^l$ denotes the $l$-th component of $W_.$. The processes $Y^{(r,\pi)}$ and $Z^{(r,\pi)}$ are extended to RCLL processes by using constant interpolation. By then approximating those conditional expectations, for instance, by using the LSMC regression method (Section 4.1), the convergence of the above discretized Picard type iteration is given in the form:

$$\sup_{t \in [0,T]} \mathbb{E}\left[|Y_t - Y_t^{(r,\pi)}|^2\right] + \mathbb{E}\left[\int_0^T |Z_s - Z_s^{(r,\pi)}|^2 ds\right]$$
$$\leq C\left(|\pi_n| + \left(\frac{1}{2} + C|\pi_n|\right)^r\right),$$

as $|\pi_n|$ is sufficiently small, provided that $\sup_{t \in [0,T]} \mathbb{E}[|X_t - X_t^{(\pi)}|^2] \leq C|\pi_n|$ and $\sup_{|\pi_n| \leq 1} \mathbb{E}[|\Phi(X_T^{(\pi)})|^2] \leq C$, for some positive constant $C > 0$.

The discretized Picard type iteration has no (high order) nestings of conditional expectations backwards in time, whereas it does have (lower order) nestings of conditional expectations forward in time in the number of Picard iterations. This turns out to be an advantage from a numerical point of view relative to backward methods. Overall, the error when approximating the involved conditional expectations (by a generic estimator) is significantly lower relative to backward methods. Specifically, it turns out the error grows moderately when the mesh of partition goes to zero and the number of Picard iterations tends to infinity. This is once again an advantage over backward methods, where the error often explodes when the mesh tends to zero. The algorithm that puts everything all together is tested on a hedging problem to demonstrate the proven theoretical convergence of the numerical method. We note that a variance reduced version of the algorithm is also mentioned and tested on a numerical example.

This work is extended in [23, 239], where the authors introduce a variance reduced version of the forward approximation scheme by means of importance sampling, or more specifically, by means of a measure transformation based on a general Radon-Nikodym derivative. The technique of importance sampling is generalized from simulating expectations to computing the initial value of a FBSDE. The convergence of this modified and fully implementable numerical method is proved and the success of the involved generalized importance sampling is illustrated by numerical examples in the context of financial option pricing.

We next look at forward numerical methods which focus on solving the equivalent PDE (Theorem 2.3) by Picard iterations. Here, we primarily examine one numerical method, which illustrates this category well. In [138], the authors combine two main ingredients to an algorithm which is aimed at solving the PDE equivalent to the general FBSDE (Theorem 2.3). Hence, Assumption 2 is essential. To describe the scheme, we prepare some notation. Let $v : [0,T] \times \mathbb{R}^q \to \mathbb{R}$ be $\mathcal{C}^1$ in space and define $f_v : [0,T] \times \mathbb{R}^q \to \mathbb{R}$ as the following function

$$f_v(t, \mathbf{x}) := f(t, \mathbf{x}, v(t, \mathbf{x}), (\nabla v(t, \mathbf{x}))^\top \sigma(t, \mathbf{x})), \tag{5.1}$$

where $f$ is the driver of the FBSDE (2.6) and $\sigma$ is the diffusion coefficient of the SDE (2.7). We define the following two random variables

$$
\begin{aligned}
\Psi(s, y, g_1, g_2) &:= \int_s^T g_1(r, X_r^{s,\mathbf{y}})dr + g_2(X_T^{s,\mathbf{y}}), \\
\Psi^N(s, y, g_1, g_2) &:= \int_s^T g_1(r, X_r^{N,s,\mathbf{y}})dr + g_2(X_T^{N,s,\mathbf{y}}),
\end{aligned}
\tag{5.2}
$$

where $X^{s,\mathbf{y}}$ (respectively, $X^{N,s,\mathbf{y}}$) denotes the diffusion process starting from $\mathbf{y}$ at time $s$ (respectively, its approximation with $N$-time steps) which solves the forward SDE (2.7). Finally, we define $c_{i,j}(\phi) := \sum_{k,l=0}^{i,j} |\nabla^k (\partial_t)^l \phi|_\infty$, with $c_0(\phi) := c_{0,0}(\phi)$ for $\phi \in \mathcal{C}_b^{i,j}$.

We first look at the case where the FBSDE has a driver which is independent of $Y$ and $Z$, that is, $f(t, \mathbf{x}, y, \mathbf{z}) = f(t, \mathbf{x})$, which means the FBSDE is linear. We demonstrate the adaptive control variate approach (one of the two main ingredients) in this simple setting. It is then extended to the general FBSDE via Picard iterations (the second of the two main ingredients). From Theorem 2.3, we see that in order to solve a linear FBSDE, one can equivalently solve the linear PDE (2.9), for which an adaptive control variate scheme is developed in [143]. Here, one aims to numerically solve the linear PDE:

$$
((\partial/\partial t) + \mathcal{L}_t)v + f = 0, \quad v(T, \cdot) = \Phi(\cdot). \tag{5.3}
$$

Using the Feynman-Kac formula, the probabilistic solution of this PDE can be expressed as the following conditional expectation in accordance with (5.2):

$$
v(t, \mathbf{x}) = \mathbb{E}_{t,\mathbf{x}} \left[ \Phi(X_T) + \int_t^T f(s, X_s)ds \right] = \mathbb{E}\left[ \Psi(t, \mathbf{x}, f, \Phi) \right], \tag{5.4}
$$

where $X^{t,\mathbf{x}}$ is the solution to the standard SDE (2.7) starting from $\mathbf{x}$ at time $t$. One wishes to compute a sequence of solutions $(v_r)_{r \in \{0,1,\cdots\}}$ by writing

$$
v_{r+1} = v_r + (\text{Monte Carlo evaluations of the error } (v - v_r)),
$$

as this sequence is proved to converge to the solution $v$ of (5.3). This approach is backed up by the probabilistic representation of the error term:

$$
v(t, \mathbf{x}) - v_r(t, \mathbf{x}) = \mathbb{E}\left[ \Psi\left(t, \mathbf{x}, f + ((\partial/\partial t) + \mathcal{L}_t)v_r, \Phi - v_r\right) \right] =: c_r(t, \mathbf{x}).
$$

We now proceed to the second ingredient, which is the most general form of Picard iterations [138] and will be used to approximate the solution of the nonlinear FBSDE by the solutions of a sequence of (simple) linear FBSDEs (ones with driver independent of $Y$ and $Z$), which converge geometrically to $(Y, Z)$. Define recursively the Picard iterative sequence $(Y^r, Z^r)_{r \in \{0,1,\cdots\}}$ with $(Y^0, Z^0) = (0, 0)$, as follows:

$$
-dY_t^{r+1} = f(t, X_t, Y_t^r, Z_t^r)dt - Z_t^{r+1}dW_t, \quad Y_t^{r+1} = \Phi(X_T), \quad t \in [0, T].
$$

for $r \in \{0, 1, \cdots\}$ forward. This sequence of linear FBSDEs converges to the unique solution $(Y, Z)$ of the original non-linear FBSDE $dt \otimes d\mathbb{P}$-*a.e.* By writing $Y_t^r = v_r(t, X_t)$ and $Z_t^r = (\nabla v_r(t, X_t))^\top \sigma(t, X_t)$ and by Theorem (2.3), each linear FBSDE can be equivalently written as a PDE:

$$((\partial/\partial t) + \mathcal{L}_t)v_{r+1} + f(\cdot, \cdot, v_r, (\nabla v_r)^\top \sigma) = 0, \quad v_{r+1}(T, \cdot) = \Phi(\cdot), \quad r \in \{0, 1, \cdots\}.$$

The sequence of solutions of linear PDEs $(v_r, \nabla v_r)_{r \in \{0, 1, \cdots\}}$ converges in a suitable $L_2$ norm to the solution $(v, \nabla v)$ of the semilinear PDE:

$$((\partial/\partial t) + \mathcal{L}_t)v + f(\cdot, \cdot, v, (\nabla v)^\top \sigma) = 0, \quad v(T, \cdot) = \Phi(\cdot),$$

which then yields the solution of the non-linear FBSDE by setting $(Y, Z) = (v, (\nabla v)^\top \sigma)$. The adaptive control variate method also converges geometrically, and so when combined, the two ingredients provide an algorithm with geometric convergence. To describe the algorithm, we employ the notation $\mathcal{L}_n u(s, \mathbf{x}) := \frac{1}{2}\mathrm{tr}[\sigma^{\otimes 2}(\phi_n(s), \mathbf{x})\mathrm{Hess}(u(s, \mathbf{x}))] + \langle \mu(\phi_n(s), \mathbf{x}), \nabla u(s, \mathbf{x}) \rangle$, and $\phi_n(s) := \max\{t_k \in \pi_n : t_k \leq s\}$ with respect to the partition $\pi_n$ on the interval $[0, T]$.

---

**Algorithm 5:**

---

**Initialization**: Set $v_0 \equiv 0$ and assume that an approximate solution $v_r$ of class $\mathcal{C}_b^{1,2}$ has been built at step $r$. Take $n$ points $(t_k^r, \mathbf{x}_k^r)_{k \in \{1, \cdots, n\}} \in [0, T] \times \mathbb{R}^q$ at each step of the iteration.

- Evaluate $c_r(t_k^r, \mathbf{x}_k^r)$ using $M$ iid realizations by

$$c_r^M(t_k^r, \mathbf{x}_k^r) = \frac{1}{M} \sum_{m=1}^{M} \Psi^N \left( t_k^r, \mathbf{x}_k^r, f_{v_r} + ((\partial/\partial t) + \mathcal{L}_n)v_r, \Phi - v_r \right)_m.$$

- Build the global solution $c_r^M(\cdot)$ based on the values $(c_r^M(t_k^r, \mathbf{x}_k^r))_{k \in \{1, \cdots, n\}}$ using a linear approximation operator $\mathcal{P}^r$:

$$\mathcal{P}^r c(\cdot) = \sum_{k=1}^{n} c(t_k^r, \mathbf{x}_k^r) w_k^r(\cdot),$$

where $(w_k^r)_{k \in \{1, \cdots, n\}}$ are suitable weight functions. The approximation of $v$ at step $r + 1$ is then computed as

$$v_{r+1}(t, \mathbf{x}) = \mathcal{P}^r(v_r + c_r^M)(t, \mathbf{x}).$$

---

We here address a few key questions regarding the algorithm. First, it is critical how to choose the grid points $(t_k^r, \mathbf{x}_k^r)_{k \in \{1, \cdots, n\}}$ at each iteration $r$. There, at each iteration, we take a new grid of $n$ points $(t_k^r, \mathbf{x}_k^r)_{k \in \{1, \cdots, n\}}$ that are independent and uniformly distributed on $[0, T] \times [-a, a]^q$ for a suitable positive constant $a$. As we want to solve the PDE on $[0, T] \times \mathbb{R}^q$, we must choose $a$ large enough. Next, an appropriate norm needs to be prepared for measuring the error. By combining results on BSDEs stated in a norm (leading to the integration with respect to the measure $e^{\beta s}ds$ [105]) and results on the bounds for solutions of linear PDEs in weighted Sobolev spaces (leading to the integration with respect to the measure $e^{-\lambda\|\mathbf{x}\|}d\mathbf{x}$ [30]), the convergence of the algorithm is derived in

the norm:

$$\|V\|_{\lambda,\beta}^2 := \mathbb{E}\left[\int_0^T \int_{\mathbb{R}^q} e^{\beta s}\|V_s(\mathbf{x})\|^2 e^{-\lambda\|\mathbf{x}\|} d\mathbf{x}ds\right] < +\infty,$$

for $\lambda > 0$, $\beta > 0$ and the set of processes $V : \Omega \times [0,T] \times \mathbb{R}^q \to \mathbb{R}^d$ that are $\mathcal{P}r \otimes \mathcal{B}(\mathbb{R}^q)$-measurable (where $\mathcal{P}r$ is the $\sigma$-field of predictable subsets of $\Omega \times [0,T]$). This norm is then employed to measure the error $(Y - Y^r, Z - Z^r)$, corresponding to the approximation error made at step $r$ of the algorithm. The expectations appearing in $\|Y - Y^r\|_{\lambda,\beta}^2$ and $\|Z - Z^r\|_{\lambda,\beta}^2$ are computed with respect to the law of $X$, $X^n$ and all the possible random variables used to compute $v_r$.

The sequence of approximation operators $(\mathcal{P}^r)_{r\in\{0,1,\cdots\}}$ is allowed to change at each step of the iteration, under a number of technical conditions: measurability, linearity, regularity, boundedness, able to approximate functions and spatial derivatives well and stability and centering property for random function. We refer the reader to [138] for explicit definitions of each property and an example of a sequence of operators $\mathcal{P}^r$ which satisfy the desired properties. The operators used are kernel based estimators and are based on the non-parametric technique local averaging. This scheme is further accelerated via parallel computing [210] by replacing the kernel operator with an extrapolating operator for approximating functions and their derivatives.

The geometric convergence results for the algorithm can be described as follows [138]. If Assumption 2 holds, $\Phi \in \mathcal{C}_b^{2+\alpha}$ and $f$ is bounded and Lipschitz, then there exists a constant $K(T)$ such that

$$\|Y - Y^r\|_{\lambda,\beta}^2 + \|Z - Z^r\|_{\lambda,\beta}^2 \leq S_k + K(T)\frac{c_{0,2}^2(v)}{n},$$

where $S_r \leq \eta S_{r-1} + \epsilon$ for suitable constants $\eta$ and $\epsilon$. Moreover, for $\beta$ and $\mathcal{P}$-parameters large enough so that $\eta < 1$, it holds that

$$\limsup_{r\to+\infty}\|Y - Y^r\|_{\lambda,\beta}^2 + \|Z - Z^r\|_{\lambda,\beta}^2 \leq \frac{\epsilon}{1-\eta} + K(T)\frac{c_{0,2}^2(v)}{n}.$$

This result is derived by splitting the error $\|Y - Y^r\|_{\lambda,\beta}^2 + \|Z - Z^r\|_{\lambda,\beta}^2$ into its difference sources, such as an Euler scheme, Picard iteration, approximation operator $\mathcal{P}$ and Monte Carlo simulations.

In [286], the authors propose an analytical approximation scheme which also centers on the representation of FBSDEs given in [228], where a Malliavin calculus method is applied to the forward SDE in conbination with the Picard iteration scheme for the backward component. Various numerical examples are presented to demonstrate the convergence of the proposed algorithm.

Before moving on, we mention that there is a line of research on deep learning based algorithms for solving FBSDEs and parabolic PDEs in high dimensions. The main idea is to make an analogy between the FBSDE and reinforcement learning, where PDEs play an important role from an implementation point

of view. As the class of deep learning based algorithms cannot be categorized simply as forward methods and has become a very active field of research on its own, we do not discuss it here but set up an individual section (Section 6) exclusively for this class of algorithms.

### *5.2. Branching diffusion system based methods*

We next look at the branching diffusion system based numerical methods for nonlinear PDEs and corresponding BSDEs [167, 170]. Consider the following semilinear PDE of KPP (Kolmogorov-Petrovskii-Piskunov) type:

$$((\partial/\partial t) + \Delta)u(t, \mathbf{x}) + \beta \left( \sum\nolimits_{k \in \mathbb{N}_0} p_k(u(t, \mathbf{x}))^k - u(t, \mathbf{x}) \right) = 0, \qquad (5.5)$$

with $u(T, \mathbf{x}) = \Phi(\mathbf{x})$, which can be written in the formulation (2.9) with $\mathcal{L}_t = \Delta$ and $f(t, \mathbf{x}, y, \mathbf{z}) = \beta(\sum_{k \in \mathbb{N}_0} p_k y^k - y)$, where $\Delta$ denotes the Laplacian and $\{p_k\}_{k \in \mathbb{N}_0}$ is a probability mass sequence satisfying $p_k \in [0, 1]$ and $\sum_{k \in \mathbb{N}_0} p_k = 1$. It is well known [233, 280, 306] that the solution to the PDE (5.5) admits a probabilistic representation based on the branching diffusion system in which every particle dies in an exponential time of parameter $\beta$ and creates $k$ iid descendants with probability $p_k$. Then, every descendant dies and reproduces iid descendants independently after independent exponential times in accordance with the same mechanism. In the other words, the solution to the PDE (5.5) can be represented as

$$u(t, \mathbf{x}) = \mathbb{E} \left[ \prod\nolimits_{k=1}^{N_T} \Phi(Z_T^k) \middle| (N_t, Z_t^1) = (1, \mathbf{x}) \right], \qquad (5.6)$$

where $N_T$ is the number of particles alive at time $T$, $Z_T^k$ denotes the position of the $k$th particle at time $T$ and the condition indicates that the system is initialized at time $t$ with one particle at position $\mathbf{x}$. For the semilinear PDE with the Laplacian $\Delta$ in (5.5) replaced by the Ito generator $\mathcal{L}_t$ and a general function $f(y)$:

$$((\partial/\partial t) + \mathcal{L}_t)u(t, \mathbf{x}) + f(u(t, \mathbf{x})) = 0, \quad u(T, \mathbf{x}) = \Phi(\mathbf{x}),$$

or the corresponding FBSDE (2.6) and (2.7) with driver $f(t, \mathbf{x}, y, \mathbf{z}) = f(y)$, the so-called "marked" branching diffusion method is proposed in [167] to evaluate $u(0, \mathbf{x})$ based on the equivalence between (5.5) and (5.6) after a polynomial approximation of the driver $f(y) \approx \beta(\sum_{k=0}^{m}(a_k/p_k)p_k y^k - y)$ of finite degree $m$, where $\{p_k\}_{k \in \{0, \cdots, m\}}$ here is such that $p_k \in [0, 1]$ and $\sum_{k \in \{0, \cdots, m\}} p_k = 1$ and the fraction $(a_k/p_k)$ represents the weight to count at vertices of type $k$. The corresponding FBSDE can thus be evaluated via a fully forward-looking simulation of particles owing to the representation (5.6).

The computation required for branching diffusion methods comes almost entirely from the construction of branching particles. Hence, in terms of the problem dimension, those methods are not as prohibitive as Picard iteration methods

(Section 5.1), let alone the computation of conditional expectations (Section 4), for which a few dimensions would be the best in practice. It is reported in [170] that eight-dimensional PDEs are successfully solved without an issue. We note that the marked branching diffusion method is extended further in [3] to elliptic semilinear PDEs by introducing absorption of particles and in [169] to address the case where the nonlinear driver depends on the $Z$ component. In addition, the branching diffusion system is rich enough to provide probabilistic representations of semilinear PDEs beyond the standard form (2.9). For instance, those up to any higher order derivative in the driver (not only up to the first order $\nabla u$ as of (2.9)) have been addressed in [245, 246] with numerical illustrations. Moreover, probabilistic representations and associated numerical methods have been developed in [258, 259] for parabolic and elliptic semilinear PDEs (2.9), but with the Ito generator $\mathcal{L}_t$ replaced by a fractional Laplacian that corresponds to subordination of the underlying Brownian motion by a stable subordinator.

Interestingly, the concept of the branching diffusion system can be employed for constructing backward numerical methods [45, 46], where the driver depends on $Z$ [45] or is independent of $Z$ [46]. Despite the backward nature lies outside the primary scope of Section 5, we illustrate such a method in brief, in accordance with the latter for the sake of simplicity. The other main assumptions made are that $\Phi : \mathbb{R}^d \to \mathbb{R}$ is measurable and bounded and that $f : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}$ is measurable (as usual), uniformly Lipschitz continuous in its first argument and satisfies linear growth and Lipschitz continuity in its second argument. As a consequence, there exists a constant $M \geq 1$ such that $|\Phi(X_T)| \leq M$ and $\|X\| + |Y| \leq M$ on $[0, T]$ almost surely. The driver $f$ of the FBSDE is approximated by $f_{l_0}$ which has a local polynomial structure and is given as $f_{l_0}(\mathbf{x}, y_1, y_2) := \sum_{j=1}^{j_0} \sum_{l=0}^{l_0} a_{j,l}(\mathbf{x}) y_1^l \phi_j(y_2)$, where $(a_{j,l}, \phi_j)_{(j,l) \in \{1, \cdots, j_0\} \times \{0, \cdots, l_0\}}$ is a family of continuous and bounded maps. For all $y_1, y_2 \in \mathbb{R}$, $j \in \{1, \cdots, j_0\}$, $l \in \{0, \cdots, l_0\}$ and a positive constant $C$, these maps satisfy $|a_{j,l}| \leq C$, $|\phi_j(y_1) - \phi_j(y_2)| \leq C|y_1 - y_2|$ and $|\phi_j| \leq 1$. If the driver was simply approximated by a polynomial, then typically, the approximating FBSDEs would explode in finite time and thus no convergence could be expected. The solution to the FBSDE $(\overline{Y}, \overline{Z})$ with driver $f$ replaced by $f_{l_0}$ can approximate the true solution $(Y, Z)$ well whenever $f_{l_0}$ is a good approximation of $f$:

$$\mathbb{E}\left[\sup_{t \in [0,T]} |Y_t - \overline{Y}_t|^2\right] + \mathbb{E}\left[\int_0^T \|Z_s - \overline{Z}_s\|^2 ds\right]$$
$$\leq C\mathbb{E}\left[\int_0^T |f - f_{l_0}|^2(X_s, Y_s, Y_s) ds\right],$$

where the positive constant $C$ is independent of $f_{l_0}$. Hence, for accurate approximation, one requires a driver that can be approximated well by polynomials.

The solution to the new FBSDE $(\overline{Y}, \overline{Z})$ is then approximated by means of a Picard type iteration scheme. To that end, define the process $\overline{Y}^m$ in a recursive way. That is, let $\overline{Y}_T^r := \Phi(X_T)$, and define, on each interval $[t_k, t_{k+1}]$, $(Y_\cdot^r, Z_\cdot^r)$

as the solution on $[t_k, t_{k+1}]$ of

$$Y_\cdot^r = \overline{Y}_{t_{k+1}}^r + \int_\cdot^{t_{k+1}} f_{l_0}(X_s, Y_s^r, \overline{Y}_s^{r-1})ds - \int_\cdot^{t_{k+1}} Z_s^r dW_s. \qquad (5.7)$$

Then, $\overline{Y}^r = Y^r$ on $(t_k, t_{k+1}]$, and $\overline{Y}_{t_k}^r = (-M) \vee Y_{t_k}^m \wedge M$. The aim is thus to solve the above Picard iteration backwards on each interval $[t_k, t_{k+1}]$, and then using the now completely solved iteration on $[0, T]$, begin solving the next iteration backwards across the intervals. Importantly, this algorithm requires the truncation of the approximation of $Y$ at some given time steps in order to reduce the approximating driver to a globally Lipschitz driver. The error due to the Picard iteration scheme is given in the form $|\overline{Y}_t^r - \overline{Y}_t| \leq C(T-t)^r$ for all $t \in [0, T]$ and $r \in \mathbb{N}$, where $|\overline{Y}_t^r|$ is uniformly bounded in $t$ and $r$.

Each step of the Picard iteration is conducted backwards on each interval $[t_k, t_{k+1})$ by using a representation of $\overline{Y}^r$ in terms of branching diffusion systems. A set of particles $(X^{(l)})_{l \in K}$ is constructed, where each particle is the solution to the forward SDE (but each with their own Brownian motion) with a killing time $T_l$. At this time, the particle splits into a random number of new particles which follow the same dynamics, but with their own killing times and Brownian motion. Define the set $\overline{\mathcal{K}}_t^l$ as the collection of particles in the $l$-th generation that were born before or at time $t$, and the set $\mathcal{K}_t^l$ as the collection of particles in $\overline{\mathcal{K}}_t^l$ that are still alive at time $t$. Also, given $v^{r-1}$ and $v^r(t_{k+1}, \cdot)$, define

$$V_{t,\mathbf{x}}^r := \left[ \prod_{l \in \mathcal{K}_{t_{k+1}-t}} \frac{v^r(t_{k+1}, X_{t_{k+1}-t}^{\mathbf{x},(l)})}{\overline{F}(t_{k+1} - t - T_{l-})} \right]$$

$$\times \left[ \prod_{l \in \overline{\mathcal{K}}_{t_{k+1}-t} \backslash \mathcal{K}_{t_{k+1}-t}} \frac{\sum_{j=1}^{j_0} a_{j,\xi_l}(X_{T_l}^{\mathbf{x},(l)}) \phi_j(v^{r-1}(t + T_l, X_{T_l}^{\mathbf{x},(l)}))}{p_{\xi_l} \rho(\delta_l)} \right],$$

for all $(t, \mathbf{x}) \in [t_k, t_{k+1}) \times \Theta$, where $\Theta$ is a compact subset of $\mathbb{R}^d$. Finally, set $v^r(t, \mathbf{x}) := \mathbb{E}[V_{t,\mathbf{x}}^r]$ for $(t, \mathbf{x}) \in (t_k, t_{k+1}) \times \Theta$ and $r \in \mathbb{N}$, and $v^r(t_k, \mathbf{x}) := (-M) \vee \mathbb{E}[V_{t_k,\mathbf{x}}^r] \wedge M$, for $\mathbf{x} \in \Theta$ and $r \in \mathbb{N}$. Then, it can be proven that $\overline{Y}_\cdot^r = v^r(\cdot, X)$ on $[0, T]$. By then approximating $\mathbb{E}[V]$ by $\widetilde{\mathbb{E}}[\widetilde{V}]$, a Monte Carlo approximation on a finite functional space, and putting everything together, one obtains a numerical method for approximating $(Y, Z)$. When implementing the algorithm in practice, it is best to modify the algorithm to avoid lots of expensive Picard iterations. Two modifications are suggested which use only one Picard iteration, but provide an accurate estimate nevertheless.

### 5.3. Asymptotic expansion

The method of asymptotic expansion provides a tailor-made approximation for BSDEs by expanding a nonlinear BSDE into by a sequence of linear BSDEs in a flexible way, yet with a rigorous error analysis. By taking advantage of simple

computation involved, the method offers a fast computing scheme for BSDEs. Asymptotic expansion is categorized here as a forward methods since the resulting linear BSDEs here can be computed by forward Monte Carlo schemes alone, without the need for backward or regression schemes. We discuss the method of asymptotic expansion here separately from Picard based forward methods, as PDEs are certainly useful here (as we describe shortly) but not fully essential.

We here describe the method in accordance with [285]. On a filtered probability space $(\Omega, \mathcal{F}, \mathbb{F}, \mathbb{P})$, consider the following FBSDE with a small perturbation $\varepsilon \in [0, 1]$:

$$dX_t = \mu(t, X_t)dt + \sigma(t, X_t)dW_t, \tag{5.8}$$

$$-dY_t^\varepsilon = \varepsilon f(t, X_t, Y_t^\varepsilon, Z_t^\varepsilon)dt - Z_t^\varepsilon dW_t, \quad Y_T^\varepsilon = \Phi(X_T), \tag{5.9}$$

where the driver $f$ is sufficiently smooth, with corresponding PDE given by

$$((\partial/\partial t) + \mathcal{L}_t)v^\varepsilon(t, \mathbf{x}) + \varepsilon f(t, \mathbf{x}, v^\varepsilon(t, \mathbf{x}), (\nabla v^\varepsilon(t, \mathbf{x}))^\top \sigma(t, \mathbf{x})) = 0,$$

and $v^\varepsilon(T, \mathbf{x}) = \Phi(\mathbf{x})$, due to Theorem 2.3. Recall that $Y_t^\varepsilon = v^\varepsilon(t, X_t)$ and $Z_t^\varepsilon = (\nabla v^\varepsilon(t, X_t))^\top \sigma(t, X_t) =: \nabla v^\varepsilon \sigma(t, X_t)$ hold. Now, we expand the process $(Y^\varepsilon, Z^\varepsilon)$ around the solution of null driver (linear) BSDE $(Y^0, Z^0)$:

$$-dY_t^0 = -Z_t^0 dW_t, \quad Y_T^0 = \Phi(X_T).$$

Note that $(Y^0, Z^0)$ is explicitly solvable as $Y_t^0 = \mathbb{E}[\Phi(X_T)|\mathcal{F}_t] = v^0(t, X_t)$ and $Z_t^0 = (\nabla v^0(t, X_t))^\top \sigma(t, X_t) = \nabla v^0 \sigma(t, X_t)$, where $v^0$ is the solution of the corresponding linear PDE $((\partial/\partial t) + \mathcal{L}_t)v^0(t, \mathbf{x}) = 0$ with $v^0(T, \mathbf{x}) = \Phi(\mathbf{x})$. The following expansion is then proposed [118, 285]: for every $m \in \mathbb{N}$,

$$(Y^\varepsilon, Z^\varepsilon) \approx (Y^0, Z^0) + \sum_{k=1}^m \frac{\varepsilon^k}{k!}(Y^{(k)}, Z^{(k)}), \tag{5.10}$$

where each process $(Y^{(k)}, Z^{(k)})$ evidently satisfies

$$(Y^{(k)}, Z^{(k)}) = \left((\partial^k/\partial \varepsilon^k)Y^\varepsilon, (\partial^k/\partial \varepsilon^k)Z^\varepsilon\right)|_{\varepsilon=0},$$

and, moreover, can be characterized by the linear BSDE:

$$-dY_t^{(k)} = k \frac{d^{k-1}}{d\varepsilon^{k-1}} f(t, X_t, Y_t^\varepsilon, Z_t^\varepsilon)|_{\varepsilon=0}dt - Z_t^{(k)}dW_t, \quad Y_T^{(k)} = 0.$$

It is essential that every $(Y^{(k)}, Z^{(k)})$ is explicitly solvable since each solves a linear BSDE. The expansion (5.10) is justified in [285] as follows: for each $\beta > 0$, $\lambda > 0$ and $m \in \mathbb{N}$, there exists $C > 0$ such that

$$\left\| v^\varepsilon - \left( v^0 + \sum_{k=1}^m \frac{\varepsilon^k}{k!}v^{(k)} \right) \right\|_{\beta,\lambda}^2 + \left\| \nabla v^\varepsilon \sigma - \left( \nabla v^0 \sigma + \sum_{k=1}^m \frac{\varepsilon^k}{k!}w^{(k)} \right) \right\|_{\beta,\lambda}^2$$
$$\leq C\varepsilon^{2(m+1)},$$

for all $\varepsilon \in (0, 1]$, where

$$v^{(k)}(t, \mathbf{x}) := \frac{\partial^k}{\partial \varepsilon^k} v^\varepsilon(t, \mathbf{x})|_{\varepsilon=0}, \quad w^{(k)}(t, \mathbf{x}) := \frac{\partial^k}{\partial \varepsilon^k} \nabla v^\varepsilon \sigma(t, \mathbf{x})|_{\varepsilon=0},$$

for $k \in \mathbb{N}$, with the norm $\|h\|_{\beta,\lambda} := \int_0^T \int_{\mathbb{R}^q} e^{\beta s} \|h(s, \mathbf{x})\| e^{-\lambda|\mathbf{x}|} d\mathbf{x} ds$ for $\beta > 0$ and $\lambda > 0$. This error estimate is consistent with the expansion (5.10) in the sense of $Y_t^{(k)} = v^{(k)}(t, X_t)$ and $Z_t^{(k)} = w^{(k)}(t, X_t)$ for $k \in \mathbb{N}$.

The development in [285] is generalized in [145] to nonsmooth drivers. The development in [145] can be summarized as follows: for a driver $f$ which is Lipschitz in $(t, x)$ and is Gateaux-differentiable in $(y, z)$ in the sense that for any square-integrable predictable processes $\phi$ and $\psi$, there exist $\kappa \in (0, 1]$ and a square-integrable predictable process $Df.(\phi, \psi)$ such that $\mathbb{E}[\int_0^T |(f(t, X_t, Y_t^0 + \varepsilon\phi_t, Z_t^0 + \varepsilon\psi_t) - f(t, X_t, Y_t^0, Z_t^0) - \varepsilon Df_t(\phi, \psi))/\varepsilon|^2 dt] = o(\varepsilon^{2\kappa})$, it holds that

$$\sup_{t \in [0,T]} \mathbb{E}\left[\left|Y_t^\varepsilon - \left(Y_t^0 + \varepsilon Y_t^{(1)} + \frac{\varepsilon^2}{2} Y_t^{(2)}\right)\right|^2\right]$$
$$+ \mathbb{E}\left[\int_0^T \left|Z_t^\varepsilon - \left(Z_t^0 + \varepsilon Z_t^{(1)} + \frac{\varepsilon^2}{2} Z_t^{(2)}\right)\right|^2 dt\right] = o(\varepsilon^{4+2\kappa}).$$

The effectiveness and practical accuracy of the schemes has been well demonstrated via numerical experiments in [118, 145]. In practice, the expansion up to the first or second order often provides a sufficient accuracy even for non-smooth drivers. An efficient implementation with interacting particles is developed in [120]. We also refer to [76] for nonlinear Monte Carlo scheme using asymptotic expansion method with interacting particles.

The method of asymptotic expansion has found its application to a variety of models in mathematical finance. For instance, it is employed in [119] for a complete market on BSDEs. Numerical analyses are conducted for American options in [117] based on asymptotic expansion method with interacting particles. Asymptotic expansion is applied in [77] for numerical experiments in a counterparty risk model. In [10], a dynamic framework is introduced for analyzing XVAs with asymptotic expansion employed in numerical experiments. A polynomial expansion method is developed in [116] by approximating target BSDEs via a recursive system of linear ODEs. Using asymptotic expansion, numerical approximations are derived in [4] for McKean-type anticipative BSDEs arising in initial margin requirement models. We close this section with even different lines of research, such as [276] for quadratic BSDEs, [121] for BSDEs with jumps, [123, 242, 284] for deep learning schemes inspired by asymptotic expansion, and [107, 122, 286] for similar but different types of expansions in nonlinear PDE and BSDEs.

Based on asymptotic expansion, though not quite in the same spirit as above, statistical inference has been investigated for BSDEs, as from a practical point of view, it may well happen that some problem coefficients cannot be fully determined upon implementation. For instance, in the case where the drift coefficient $\mu(t, \mathbf{x}; \theta)$ of the forward process (2.7) remains partially unspecified in its

parameter $\theta$ and the diffusion coefficient $\epsilon\sigma(t,\mathbf{x})$ is kept small with $\epsilon \approx 0$, the approximation problem for the solution to FBSDEs is investigated in [208, 209] with maximum likelihood estimation of the unknown parameter $\theta$ concurrently conducted. The case where the diffusion coefficient is parameterized instead is investigated in [125].

## *5.4. Multilevel Picard approximation*

Multilevel Picard approximation is an emerging class of numerical methods, mainly consisting of the following three steps [98]:

1. reformulation of the PDE as a stochastic fixed point problem,
2. approximation of the unique fixed point by Picard iterations,
3. and then approximation of the iterations by multilevel Monte Carlo methods.

The method is said to be full history recursive in the sense that the realizations in the $m$-th iteration require those in the 1st, 2nd, $\cdots$, $(m-1)$-th iteration. Literally, its name originates from the two major components, that is, Picard iterations and multilevel Monte Carlo methods. As only a few Picard iterations are required in practice and expectations are nested along the iterations, the conditional expectations may be approximated by the standard Monte Carlo method, leading to nested simulations in the number of iterations. The cost can be kept tractable by using only a small number of samples in each nested layer of simulations, for which a very efficient variance reduction is needed, typically handled by the multilevel Monte Carlo approach.

A major motivation for introducing multilevel Picard approximation is to solve high dimensional nonlinear PDEs in a similar spirit to deep learning based methods (Section 6). The category is initiated in the work [99], in which the applicability of approximation methods based on Picard iterations and multilevel Monte Carlo methods is investigated for high dimensional nonlinear PDEs arising in physics and finance. It is subsequently shown [183] that the complexity grows polynomially both in the dimension and in the reciprocal of the required accuracy in the case of semilinear heat equations with gradient-independent and globally Lipschitz continuous nonlinearities. For instance, its complexity is shown to be $\mathcal{O}(d\epsilon^{-4})$ for such semilinear heat equations, where $d$ is the dimension of the problem and $\varepsilon$ is the required precision.

Multilevel Picard approximation methods have been developed further in many papers. A class of multilevel Picard approximation is proposed in [20] for computing iterated nested expectations. For very high-dimensional problems, there is another nested Monte Carlo algorithm for the PDE formulation, where the nesting is along a random time grid [305] Multilevel Picard approximation algorithms are shown to overcome the curse of dimensionality for high dimensional nonlinear heat equations with general time horizons and gradient-dependent nonlinearities [182] and also overcome the curse of dimensionality in the $L^p$ sense for high-dimensional semilinear PDEs [186]. It is shown in [185]

that a Monte Carlo type numerical method approximates the solution path of BSDEs with complexity which is at most polynomial in the model dimension and at most quadratic in the reciprocal of the prescribed approximation accuracy. It is proved in [187] that semilinear heat equations with gradient-dependent non-linearities can be approximated under suitable technical conditions with poly-nomial complexity both in the dimension and the reciprocal of the accuracy. In [188], Picard iterations for backward stochastic differential equations with glob-ally Lipschitz continuous nonlinearity are shown to converge, at least, at the rate of square-root of factorial. An extension of multilevel Picard approxima-tion is applied to general forward diffusion in [189]. Other numerical algorithms can be found in [100] for approximating solutions of general high-dimensional semilinear parabolic partial differential equations at single space-time points, and in [21] for parametric approximation problems by combining Monte Carlo algorithms with machine learning techniques to learn the random variables in Monte Carlo simulations including the multilevel Picard approximation. Yet an-other type can be found in [60] based on a Picard iteration scheme in which a sequence of linear-quadratic optimization problems are solved by means of the stochastic gradient descent algorithm.

### 5.5. Further on forward numerical methods

We also find a fully forward-looking numerical method in [117], where the non-linear driver of the FBSDE is treated as a perturbation and consequently the FBSDE is converted into a series of decoupled linear FBSDE. The method proposed for approximating these linear FBSDEs uses an interacting particle method in order to perform the involved integration steps, as opposed to us-ing direct Monte Carlo simulation. The homotopy analysis method has been reported effective [333] for FBSDEs. The coupled FBSDEs are transformed into a control problem [86] by discretizing the backward component in the forward direction and are simulated by combining the standard backward discretiza-tion with Picard iterations [28]. Finally, a forward method can be found in [52] based on a Wiener chaos expansion, Picard iterations and Malliavin calculus techniques.

## 6. Deep learning

The method of deep learning is one of the most active emerging categories in the context of numerical methods for BSDEs. These methods have attracted a great deal of attention for their unique feature of solving high dimensional nonlinear BSDEs and corresponding nonlinear PDEs, owning to neural network approx-imation by which the estimation of nested high-dimensional expectations and gradients is significantly eased. So far, a wide variety of numerical results have been developed for approximating the solution of nonlinear PDEs by a neural network from physics and finance and FBSDEs related to a nonlinear pricing

model for financial derivatives. Those numerical methods demonstrate the efficiency and accuracy of the algorithm for several 100-dimensional nonlinear PDEs and FBSDEs. This effectiveness in high-dimensional problems, in addition to the intrinsic structure involving both aspects of backward and forward methods, makes the method of deep learning quite distinctive in sharp contrast to backward and forward methods (Sections 3 and 5). Despite the complexity of deep learning methods not being able to be investigated in a general form at present, there exist a few theoretical results in the literature that support deep neural networks in overcoming the curse of dimensionality in numerical approximations of some linear and nonlinear PDEs, where the complexity is shown to grow at most polynomially in both the PDE dimension and the reciprocal of the approximation accuracy (for instance, [19, 31, 153, 154, 184, 195, 271, 287]).

In what follows, we describe and summarize such deep learning based methods. To this end, we start with the basics on multilayer neural networks. We reserve $L$ for the number of layers. For $k \in \{0, 1, \cdots, L-1\}$, let $\ell_k \in \mathbb{N}$ denote the size of input of the $k$-th layer and let $\ell_L$ be the size of the output layer. For $k \in \{1, \cdots, L\}$, define $W_k := (w_{i,j}^k)_{i,j}$ in $\mathbb{R}^{\ell_k \times \ell_{k-1}}$ with $w_{i,j}^k := \theta_{j+\ell_k(i-1)}^k$ called weight and $b_k := (\theta_{\ell_{k-1} \times \ell_k + 1}^k, \cdots, \theta_{\ell_{k-1} \times \ell_k + k}^k)$ in $\mathbb{R}^{\ell_k}$ called bias. Let $\mathbb{A}_k : \mathbf{x}(\in \mathbb{R}^{\ell_{k-1}}) \mapsto W_k \mathbf{x} + b_k(\in \mathbb{R}^{\ell_k})$ and define $A_e(\mathbf{x}) := (a(x_1), \cdots, a(x_e))$ for $\mathbf{x} \in \mathbb{R}^e$, where $a$ denotes a real-valued nonlinear function on $\mathbb{R}$, called the activation function. With all those, we define the neural network $\mathcal{N}_L^\Theta :$ $\mathbb{R}^{\ell_0} \to \mathbb{R}^{\ell_L}$ by $\mathcal{N}_L^\Theta(\mathbf{x}) := \mathbb{A}_L \circ \cdots \circ A_{\ell_2} \circ \mathbb{A}_2 \circ A_{\ell_1} \circ \mathbb{A}_1(\mathbf{x})$ for $\mathbf{x} \in \mathbb{R}^{\ell_0}$, with $\Theta := (\theta_1^1, \cdots, \theta_{\ell_{L-1} \times \ell_L + L}^L)$. Note that every continuous function on a compact set can be approximated by a neural network to any desired precision.

## 6.1. Deep BSDE

We first summarize a numerical scheme called the deep BSDE method [97, 162]. The essence of deep BSDE methods is, broadly speaking, to make use of the gradient of the solution (with respect to the control process $Z$) as the policy function, and approximate it through a neural network as is done in the standard deep reinforcement learning. For the reader's convenience, we recall the FBSDE (2.6) and (2.7):

$$dX_t = \mu(t, X_t)dt + \sigma(t, X_t)dW_t,$$
$$-dY_t = f(t, X_t, Y_t, Z_t)dt - Z_t dW_t,$$

with $X_0 = \mathbf{x}$ and $Y_T = \Phi(X_T)$, and Theorem 2.3, which asserts $(Y_t, Z_t) = (v(t, X_t), (\nabla v(t, X_t))^\top \sigma(t, X_t))$, with $v$ satisfying the semilinear parabolic PDE under Assumption 2:

$$((\partial/\partial t) + \mathcal{L}_t)v(t, \mathbf{x}) + f(t, \mathbf{x}, v(t, \mathbf{x}), (\nabla v(t, \mathbf{x}))^\top \sigma(t, \mathbf{x})) = 0,$$

with $v(T, \mathbf{x}) = \Phi(\mathbf{x})$.

In general, the deep learning-based method aims to approximate $Y_0 = v(0, \mathbf{x})$. We start with the following stochastic optimization problem:

$$\inf_{y,Z} \mathbb{E}\left[|\Phi(X_T) - Y_T^{y,Z}|^2\right]$$

$$\text{s.t. } Y_t^{y,Z} := y - \int_0^t f(s, X_s, Y_s^{y,Z}, Z_s)ds + \int_0^t Z_s dW_s, \quad t \in [0, T].$$

Evidently, to solve this minimization problem, the expectation $\mathbb{E}[|\Phi(X_T) - Y_T^{y,Z}|^2]$ needs to be computable or approximated. For this purpose, define the sequence $\{\overline{X}_{t_k}^n\}_{k \in \{0,1,\cdots,n\}}$ as the Euler-Maruyama discretization of the forward process:

$$\overline{X}_{t_{k+1}}^n = \overline{X}_{t_k}^n + \mu(t_k, \overline{X}_{t_k}^n)(t_{k+1} - t_k) + \sigma(t_k, \overline{X}_{t_k}^n)\left(W_{t_{k+1}} - W_{t_k}\right), \quad (6.1)$$

for $k \in \{0, 1, \cdots, n-1\}$, with initial state $\overline{X}_{t_0}^n = \mathbf{x}$, and define the *forward discretization* $\{\overline{Y}_{t_k}^n\}_{k \in \{0,\cdots,n-1\}}$ of the backward process by

$$\overline{Y}_{t_{k+1}}^n = \overline{Y}_{t_k}^n - f(t_k, \overline{X}_{t_k}^n, \overline{Y}_{t_k}^n, z_k(\overline{X}_{t_k}^n))(t_{k+1} - t_k)$$
$$+ z_k(\overline{X}_{t_k}^n)\left(W_{t_{k+1}} - W_{t_k}\right), \quad k \in \{0, 1, \cdots, n-1\}, \quad (6.2)$$

with initial state $Y_{t_0}^n = y$, in order to approximate the minimization problem, as follows:

$$\inf_{y,Z} \mathbb{E}\left[|\Phi(X_T) - Y_T^{y,Z}|^2\right] \approx \inf_{y,\{z_k\}_k} \mathbb{E}\left[|\Phi(\overline{X}_T^n) - \overline{Y}_T^n|^2\right]$$

$$\approx \inf_{y,\{z_k\}_k} \frac{1}{M}\sum_{l=1}^M |\Phi(\overline{X}_{T,l}^n) - \overline{Y}_{T,l}^n|^2, \quad (6.3)$$

for sufficiently large $M$ for decent estimation quality, where $\{\overline{X}_{T,l}^n\}_{l \in \mathbb{N}}$ and $\{\overline{Y}_{T,l}^n\}_{l \in \mathbb{N}}$ are independent copies of $\overline{X}_T^n$ and $\overline{Y}_T^n$, respectively. At the end, the obtained solutions, say, $y^*$ and $\{z_k^*\}_{k \in \{0,1,\cdots,n-1\}}$, provide the approximation $Y_0 \approx y^*$ and $Z_{t_k} \approx z_k^*(\overline{X}_{t_k}^n)$ for $k \in \{0, 1, \cdots, n-1\}$.

Deep learning comes into play when solving the approximate minimization problem (6.3). For $k \in \{1, \cdots, n-1\}$, approximate the target continuous function $z_k$ by a neural network as $\mathbf{x} \mapsto z_k^{\theta_k}(\mathbf{x}) \approx \mathcal{N}_L^{\theta_k}(\mathbf{x})$ with parameter $\theta_k \in \mathbb{R}^{q(L)}$ for a suitable dimension $q(L) \in \mathbb{N}$, depending on the number of layers $L$. For $\Theta = (\theta^1, \cdots, \theta^{d+1}, \theta_1, \cdots, \theta_{n-1})$ with $\theta^k \in \mathbb{R}$ for $k \in \{1, \cdots, d+1\}$, the forward discretization of the backward process (6.2) can be parameterized with $\Theta$ as

$$\overline{Y}_{t_{k+1}}^n(\Theta) = \overline{Y}_{t_k}^n(\Theta) - f(t_k, \overline{X}_{t_k}^n, \overline{Y}_{t_k}^n(\Theta), z_k^{\theta_k}(\overline{X}_{t_k}^n))(t_{k+1} - t_k)$$
$$+ z_k^{\theta_k}(\overline{X}_{t_k}^n)\left(W_{t_{k+1}} - W_{t_k}\right), \quad k \in \{0, 1, \cdots, n-1\},$$

with initial states $\overline{Y}_0^n(\Theta) = \theta^1$ and $z_0^{\theta_0}(\mathbf{x}) = (\theta^2, \cdots, \theta^{1+d})$. We seek the parameter vector $\Theta$ via the following minimization problem:

$$\inf_{\Theta} \frac{1}{M}\sum_{l=1}^M \left|\Phi(\overline{X}_{T,l}^n) - \overline{Y}_{T,l}^n(\Theta)\right|^2 =: \inf_{\Theta} \phi(\Theta),$$

by stochastic gradient descent, that is, $\Theta_m = \Theta_{m-1} - \gamma \nabla \phi(\Theta_{m-1})$ recursively for $m \in \mathbb{N}$, where $\gamma$ is a suitable learning rate. After a sufficiently large number of recursions, say $K$, we obtain the parameter vector $\Theta_K$ with which the approximation of the target is given by $Y_0 = u(0, \mathbf{x}) \approx \overline{Y}_0^n(\Theta_K)$.

As described in brief in the beginning of the present section, the significance of deep BSDE methods is to adopt deep neural networks for computing gradients of the solution and approximating the backward component forward in time so that high-dimensional PDEs and BSDEs can be solved in a realistic runtime. An error analysis is conducted for deep BSDE methods in [163], where a posteriori estimates are derived for coupled FBSDEs which relate the quadratic terminal loss to the approximation error for the numerical solution of the FBSDE (a posteriori estimates for BSDEs can also be found in [27]). We note that the approximation of the backward component $Y$ parametrically forward in time as above and then the minimization of an error criterion has been found valid even outside the realm of deep learning [8].

The deep BSDE method is modified in [304] by measuring the loss at the forward initial time (rather than the terminal time, as above), called Deep BSDE-ML method, for approximating linear decoupled FBSDEs, as well as is extended in [135] for BSDEs with jumps. The backward deep solvers are developed in [299, 313] in order to apply the deep BSDE solver-based method to financial problems, such as pricing Bermudan swaption and nonlinear pricing in high-dimensional settings. High-dimensional coupled FBSDEs with non-Lipschitz diffusion coefficients are numerically solved in [198] using the deep BSDE method. The deep BSDE method is also developed in [6] for strongly coupled FBSDEs stemming from stochastic control.

### 6.2. *Deep backward dynamic programming*

Deep backward dynamic programming is proposed in [181] and then further improved in [132], literally on the basis of backward dynamic programming arising from discretization methods of BSDEs for high dimensional nonlinear PDEs via the minimization of loss functions at each step, defined recursively by backward induction. In contrast to deep BSDE methods (Section 6.1), deep backward dynamic programming is built in reference to the backward resolution technique upon an implicit backward Euler scheme. It employs machine learning techniques for estimating the solution and its gradient by minimizing a loss function on each time step, where such local problems are then solved recursively with a stochastic gradient algorithm backward in time. Two schemes are proposed in [181] for dealing with those local problems: **(DB1)** approximating both the solution and its gradient by a neural network, and **(DB2)** approximating the solution alone by a neural network, with its gradient estimated directly with automatic differentiation. It is worth mentioning that from a viewpoint of the computation of conditional expectations, deep backward dynamic programming may also be regarded as a backward Euler scheme where nonlinear regression is employed with a deep learning technique.

Here, we describe deep backward dynamic programming in brief. The solution $(X, Y, Z)$ of the FBSDE (2.6) and (2.7) is discretized, first through the Euler-Maruyama discretization of the forward component $\{\overline{X}_{t_k}^n\}_{k\in\{0,1,\cdots,n\}}$ with $\overline{X}_{t_0}^n = \mathbf{x}$ on the time grid $(0 =)t_0 < t_1 < \cdots < t_n(= T)$ in a similar manner to (6.1) and the backward component, by starting with $\overline{Y}_{t_n}^n = \Phi(\overline{X}_{t_n}^n)$ and then defining

$$\overline{Y}_{t_k}^n = \overline{Y}_{t_{k+1}}^n + f(t_k, \overline{X}_{t_k}^n, \overline{Y}_{t_k}^n, \overline{Z}_{t_k}^n)(t_{k+1} - t_k) - \overline{Z}_{t_k}^n(W_{t_{k+1}} - W_{t_k}),$$

for $k \in \{n-1, \cdots, 1, 0\}$ backwards, where the component $\{\overline{Z}_{t_k}^n\}_{k\in\{0,\cdots,n\}}$ is not explicit as of yet and to be found during the following procedure. Along the discretized pair $\{(\overline{X}_{t_k}^n, \overline{Y}_{t_k}^n)\}_{k\in\{0,\cdots,n\}}$, we describe the aforementioned two schemes of deep backward dynamic programming:

**(DB1)** With the terminal condition $\mathcal{U}_n = \Phi$, iterate

$$(\mathcal{U}_k, \mathcal{Z}_k) = \underset{(u_k, z_k)}{\operatorname{argmin}} \mathbb{E}\left[\left|\mathcal{U}_{k+1}(\overline{X}_{t_{k+1}}^n) - \left(u_k(\overline{X}_{t_k}^n)\right.\right.\right.$$
$$\left.\left.\left. + f(t_k, \overline{X}_{t_k}^n, u_k(\overline{X}_{t_k}^n), z_k(\overline{X}_{t_k}^n))(t_{k+1} - t_k) - z_k(\overline{X}_{t_k}^n)(W_{t_{k+1}} - W_{t_k})\right)\right|^2\right],$$

for $k \in \{n-1, \cdots, 1, 0\}$ backwards.

**(DB2)** With the terminal condition $\mathcal{V}_n = \Phi$, iterate

$$\mathcal{V}_k = \underset{u_k}{\operatorname{argmin}} \mathbb{E}\left[\left|\mathcal{V}_{k+1}(\overline{X}_{t_{k+1}}^n) - \left(u_k(\overline{X}_{t_k}^n)\right.\right.\right.$$
$$+ f(t_k, \overline{X}_{t_k}^n, u_k(\overline{X}_{t_k}^n), (\nabla u_k(\overline{X}_{t_k}^n))^\top \sigma(t_k, \overline{X}_{t_k}^n))(t_{k+1} - t_k)$$
$$\left.\left.\left. - (\nabla u_k(\overline{X}_{t_k}^n))^\top \sigma(t_k, \overline{X}_{t_k}^n)(W_{t_{k+1}} - W_{t_k})\right)\right|^2\right],$$

for $k \in \{n-1, \cdots, 1, 0\}$ backwards, where $\nabla u_k$ is the automatic differentiation of the network function $u_k$. At each step $k \in \{n-1, \cdots, 1, 0\}$, the network functions $(\mathcal{U}_k, \mathcal{Z}_k)$ in **(DB1)** and $\mathcal{V}_k$ in **(DB2)** can be found literally by employing deep neural networks. The effectiveness of deep backward dynamic programming has been supported by numerical results on nonlinear PDEs up to 50-dimension [181] as well as error analysis [132].

### 6.3. Deep splitting

In a similar yet different line from the previous two methods, the so-called deep splitting method is proposed in [17] on the basis of the splitting principle with deep neural networks. Consider the nonlinear parabolic PDE:

$$((\partial/\partial t) + \mathcal{L}_t)u(t, \mathbf{x}) + f(t, \mathbf{x}, u(t, \mathbf{x}), (\nabla u^\top \sigma)(t, \mathbf{x})) = 0, \quad (t, \mathbf{x}) \in [0, T) \times \mathbb{R}^q,$$

with $u(T, \mathbf{x}) = \Phi(\mathbf{x})$, which can be represented, by the nonlinear Feynman-Kac formula, as

$$u(t, X_t) = \mathbb{E}\left[\Psi(X_T) + \int_t^T f(s, X_s, u(s, X_s), (\nabla u^\top \sigma)(s, X_s))ds \,\middle|\, \mathcal{F}_t\right],$$

where the forward component $\{X_t : t \in [0, T]\}$ is as given in Section 6.1. In light of this nonlinear Feynman-Kac representation, one wishes to construct the recursive approximation $V_k(\cdot) \approx u(t_k, \cdot)$ for $k \in \{n-1, \cdots, 1, 0\}$ backwards at discrete time points $(0 =)t_0 < t_1 < \cdots < t_n(= T)$, given by

$$V_k(\overline{X}_{t_k}^n) = \mathbb{E}\Big[V_{k+1}(\overline{X}_{t_{k+1}}^n)$$
$$+ f(t_{k+1}, \overline{X}_{t_{k+1}}^n, V_{k+1}(\overline{X}_{t_{k+1}}^n), (\nabla V_{k+1}(\overline{X}_{t_{k+1}}^n))^\top \sigma(t_{k+1}, \overline{X}_{t_{k+1}}^n))(t_{k+1} - t_k)\,\Big|\, \overline{X}_{t_k}^n\Big],$$

where $\{\overline{X}_{t_k}^n\}_{k \in \{0,1,\cdots,n\}}$ is a sequence of discrete observations of the forward component by the Euler-Maruyama scheme with $\overline{X}_{t_0}^n = \mathbf{x}$ in a similar manner to (6.1). In the deep splitting method, this recursion is approximated by a continuous function on the support of the marginal via the minimization:

$$V_k \approx \underset{v \in \mathcal{C}(\text{supp}(\overline{X}_{t_k}^n))}{\text{argmin}} \mathbb{E}\left[\left|v(\overline{X}_{t_k}^n) - \left(V_{k+1}(\overline{X}_{t_{k+1}}^n)\right.\right.\right.$$
$$\left.\left.\left. + f(t_{k+1}, \overline{X}_{t_{k+1}}^n, V_{k+1}(\overline{X}_{t_{k+1}}^n), (\nabla V_{k+1}(\overline{X}_{t_{k+1}}^n))^\top \sigma(t_{k+1}, \overline{X}_{t_{k+1}}^n))(t_{k+1} - t_k)\right)\right|^2\right],$$

so as to approximate the map $\mathbf{x} \mapsto V_k(\mathbf{x})$ by a neural network $V_k(\cdot) \approx \mathcal{N}_L^\Theta(\cdot)$ with respect to parameter set $\Theta$. It is reported in [17] that the deep splitting method succeeds to deal with as high-dimensional nonlinear PDEs as 10000 dimensions.

### 6.4. Deep Galerkin method and physics-informed neural networks

Deep Galerkin Method (DGM) [279] and Physics-Informed Neural Networks (PINN) [240] are proposed for solving high-dimensional nonlinear problems. Despite both methods being based largely on deterministic PDEs, we summarize their essence here for the reason that those may provide solvers for nonlinear BSDEs as well. Hereafter, we refer to DGM and PINN collectively as the deep learning based PDE solver and give a brief summary in accordance with [279].

Let $D \subset \mathbb{R}^d$ and let $u : [0, T] \times D \to \mathbb{R}$ solve the following (nonlinear) PDE:

$$\begin{cases} (\partial/\partial t)u(t, \mathbf{x}) + f(u(t, \mathbf{x}), \nabla u(t, \mathbf{x}), \text{Hess}(u(t, \mathbf{x}))) = 0, & (t, \mathbf{x}) \in [0, T] \times D, \\ u(0, \mathbf{x}) = \Psi(\mathbf{x}), & \mathbf{x} \in D, \\ u(t, \mathbf{x}) = g(t, \mathbf{x}), & (t, \mathbf{x}) \in [0, T] \times \partial D, \end{cases}$$

where $f$ is a nonlinear differential operator, and $\Psi : \mathbb{R}^d \to \mathbb{R}$ and $g : [0,T] \times \mathbb{R}^d \to \mathbb{R}$ represent the initial and boundary conditions, respectively. In the deep learning based PDE solver, the solution $u$ is approximated by a function $u^\Theta = \mathcal{N}_L^\Theta$ through a deep neural network with respect to parameter set $\Theta$, by minimizing the error between both sides of the nonlinear PDE above, evaluated at sampled points in the space-time domain and in its boundary. An optimal parameter set $\Theta^*$ is here searched for in such a way to ideally minimize the loss function:

$$\ell(\Theta) := \left\| (\partial/\partial t)u^\Theta + f(u^\Theta, \nabla u^\Theta, \mathrm{Hess}(u^\Theta)) \right\|_{[0,T] \times D, \nu_1}$$
$$+ \left\| u^\Theta(0, \cdot) - \Psi(\cdot) \right\|_{D, \nu_2} + \left\| u^\Theta - g \right\|_{[0,T] \times \partial D, \nu_3}, \quad (6.4)$$

where $\|\varphi\|_{S,\nu} := \int_S |\varphi(\mathbf{x})|^2 \nu(d\mathbf{x})$ with the probability measure $\nu$ and its support $S$. For minimization, one employs stochastic gradient decent on an approximate loss function:

$$\Theta_{n+1} = \Theta_n - \gamma_n \nabla G_n(\Theta_n),$$

where $\{\gamma_n\}_{n \in \mathbb{N}}$ is a sequence of learning rates. Here, $G_n$ is an approximation of the loss function (6.4), defined by

$$G_n(\Theta) := \left| (\partial/\partial t)u^\Theta(t_n, x_n) + f(u^\Theta(t_n, x_n), \nabla u^\Theta(t_n, x_n), \mathrm{Hess}(u^\Theta(t_n, x_n))) \right|^2$$
$$+ \left| u^\Theta(0, w_n) - \Psi(w_n) \right|^2 + \left| u^\Theta(\tau_n, z_n) - g(\tau_n, z_n) \right|^2,$$

where $(t_n, x_n)$, $w_n$ and $(\tau_n, z_n)$ are random elements, respectively, taking values in $[0,T] \times D$, in $D$ and in $[0,T] \times \partial D$, according to the probability measures $\nu_1$, $\nu_2$ and $\nu_3$. The iterative procedure above is to be repeated until a suitable convergence criterion is met.

In addition, more generalized frameworks using Monte Carlo methods and an efficient implementation of the neural network are discussed in [279], along with various numerical results on high-dimensional American options, high-dimensional HJB equations and Burger's equations. We do not go into further details and applications on the deep learning based PDE solver, since those would lie way outside the scope of the present survey on BSDEs. Instead, we refer the reader to [202, 224] for more applications and to [249, 269] for its close connection with BSDEs.

### 6.5. *Further on deep learning based methods*

A wide variety of deep learning based schemes have been developed for solving high-dimensional BSDEs, some of which take advantage of the aforementioned backward and forward methods (Sections 3 and 5). For instance, in [123], a deep learning scheme is introduced in combination with asymptotic expansion (Section 5.3), which is further extended in [242, 284, 297]. The deep learning technique of [97] is extended in [124, 312] to address both terminal and boundary

conditions of PDEs. As a computational framework for portfolio risk management problems, the so-called Deep xVA solver is proposed [136] by recursively using the deep BSDE method for a coupled system of BSDEs. A discretization scheme is employed in [244] for solving BSDEs based on deep learning regressions (Section 4.1). A deep signature/log-signature FBSDE algorithm is developed in [108] for approximating FBSDEs with state and path dependent features. A deep Runge-Kutta method is proposed in [59]. In the presence of constraints on the gains process, an approximation of BSDEs is obtained in [204] by neural network approximation. Iterative diffusion optimization techniques are studied using deep learning techniques [250] for applications such as importance sampling and rare event simulation. The Long Short Term Memory networks is applied in [199] to improve the Deep BSDE method [97]. A new algorithm is proposed in [292] based on a $\theta$-discretization of the time-integrands with eXtreme Gradient Boosting (XGBoost) regression for efficiently computing conditional expectations. A deep learning-based stochastic branching algorithm is developed in [245] for numerically solving fully nonlinear PDEs. High-dimensional fully-coupled FBSDEs are solved in [196] with three algorithms based on deep learning. A deep learning based method is proposed in [290] for solving forward-backward doubly stochastic differential equations. Finally, we refer the reader to [19, 35, 98, 133] for more recent developments and surveys on deep learning based methods, to [67] for singular BSDEs, to [18] for second-order BSDEs (Section 8.6), to [56, 110, 131, 161] for McKean-Vlasov BSDEs (Section 8.7), and to [197, 266] for stochastic control problems.

## 7. Discussion

In the preceding sections (Sections 3, 4, 5 and 6), we have presented a systematic survey and categorization of various numerical methods for BSDEs. However, these methods have thus far been presented in isolation and so not in an appropriate manner for drawing comparisons. In the present section, we thus present those categories all on the table, with a brief description of each, so that a relevant contrast can be made against a few key factors regarding their implementation. Such factors include the convergence property, the dimensionality, and the complexity, which directly influence, individually and/or in combination, what method would be chosen for the BSDE in question. For instance, if a very accurate approximation is needed, on the one hand, then it is desirable to select a method with a strong theoretical convergence guarantee even if the method is more difficult to code and has a longer running time. On the other hand, if a less rigorous approximation is enough for the time being (such as, for preliminary testing purposes), then it would be more reasonable to choose a method that is easier to code and runs faster, even if it may be less theoretically accurate.

In order to make comprehensive, yet, to-the-point contrasts, we must, out of sheer necessity, restrict ourselves to a representative method or two in displaying each category and do not claim that the resulting conclusions entirely hold

for every numerical method in a certain category. Moreover, throughout, we focus on the approximation of the backward component $Y$ of each BSDE model $(X, Y, Z)$ and the corresponding PDE solution (either the point $u(t, \mathbf{x})$ or the function $u(t, \cdot)$). We reserve $d$ for the dimension of the forward component $X$ and assume $Y$ is one-dimensional, that is, $\mathbf{x}(\in \mathbb{R}^d) \mapsto u(t, \mathbf{x})(\in \mathbb{R})$, and write $C$ for constant multiples whose values change depending on the context.

## *7.1.  Backward numerical methods along with computation of conditional expectations*

As backward discretization (Section 3) is intrinsically built upon the computation of conditional expectations (Section 4), we discuss representative methods consisting of those two methodological components in combination. Here, we let $n$ represent the number of discretization points in the given time interval, that is, $(t_0, t_1, \cdots, t_n)$ and let $M$ indicate the number of iid replications for Monte Carlo methods involved in each subinterval.

- LSMC (Section 4.1) + Backward Euler (Section 3.1): Let $Y_{t_k}^{n,M,K}$ denote an approximation at time $t_k$ in accordance with Section 4.1, and let $n$ be the number of time discretization, instead of the mesh $\pi$. It is shown under suitable conditions and choice of basis functions [213] that

$$\max_{k \in \{0, \cdots, n\}} \left\| Y_{t_k} - Y_{t_k}^{n,M,K} \right\|_2 = \mathcal{O}\left( n^{-1/2} \right), \tag{7.1}$$

  provided that $K \approx n^d$ and $M \approx n^{d+3}$. Despite the availability of theoretical analyses [139, 140, 213], it is still an open question as to the choice of basis functions for a given BSDE. It is worth mentioning that LSMC methods are effective in approximating the function $u(t, \cdot)$ (rather than a single point $u(t, \mathbf{x})$ alone).

- Malliavin (Section 4.2) + Backward Euler (Section 3.1): Let $Y_{t_k}^{n,M}$ denote an approximation at time $t_k$ in accordance with [47]. It is shown under suitable conditions that

$$\max_{k \in \{0, \cdots, n\}} \left\| Y_{t_k} - Y_{t_k}^{n,M} \right\|_2 \le C n^{-1/2} + C \frac{n^{d/4}}{M^{1/2}}, \tag{7.2}$$

  which suggests $M \approx n^{d/2+1}$ to bound the overall rate by $\mathcal{O}(n^{-1/2})$. In general, the required computation can be heavy for dealing with terms involving Skorohod integrals (such as (4.4)), for which an improved algorithm is proposed in [82], while as many as $2^d$ iid standard normal random variables need to be generated.

- Quantization (Section 4.3) + Backward Euler (Section 3.1): Let $Y_{t_k}^{n,N}$ denote an approximation at time $t_k$ in accordance with [11], where $N = 1 + N_1 + \cdots + N_n$ with each $N_k$ the number of points in $\mathbb{R}^d$ used to make up the space grid at the $k$-th discretization step. It is shown under suitable

conditions that

$$\max_{k \in \{0, \cdots, n\}} \left\| Y_{t_k} - Y_{t_k}^{n,N} \right\|_2 \leq C n^{-1/2} + C \frac{n^{1+1/d}}{N^{1/d}}, \qquad (7.3)$$

which suggests $N \approx n^{(3/2)d+1}$ to bound the overall rate by $\mathcal{O}(n^{-1/2})$. As described in the above and Section 4.3, Monte Carlo simulation is required for completing the quantization method with care on the number of iid replications $M$. Quantization methods work in relatively low dimensions, while generally being effective for the approximation of the function $u(t, \cdot)$.

- Tree (Section 4.4) + Backward Euler (Section 3.1): Let $Y_{t_k}^n$ be an approximation at time $t_k$ by the standard (deterministic) binomial tree method [49, 225]. In order to satisfy the order $1/2$, that is,

$$\max_{k \in \{0, \cdots, n\}} \left\| Y_{t_k} - Y_{t_k}^n \right\|_2 = \mathcal{O}\left( n^{-1/2} \right), \qquad (7.4)$$

the required complexity is approximately $2^{dn}$, which is very large, whereas the complexity drops down to $(n+1)$ with a recombination tree in one-dimensional problems.

- Cubature (Section 4.5) + Second-order discretization (Section 3.2): Let $Y_0^{n,N}$ be an approximation of the initial state $Y_0$ by a cubature method and the tree-based branching algorithm (TBBA) measure with the cubature degree of $7$ and $N$ particles at every time point $t_k$, while $n$ here denotes the number of time discretization by the second order method [81]. It is known under suitable conditions that

$$\mathbb{E}\left[ \left| Y_0 - Y_0^{n,N} \right| \right] \leq C n^{-2} + C \frac{n}{N^{1/2}}.$$

We refer the reader to [66] for relevant complexity analysis.

With numerical methods for conditional expectations (Section 4) now collectively aligned in conjunction with backward discretization methods (Section 3), we make a few relevant remarks before moving on to forward numerical methods (Section 7.2). First, recall that the backward Euler method $Y^n$ with forward Euler-Maruyama scheme $X^n$ is shown to be of order $1/2$ for the backward process $Y$ of a Markovian FBSDE $(X, Y, Z)$ under minimal Lipschitz conditions [317], that is, $\max_k \|Y_{t_k} - Y_{t_k}^n\|_2 \leq C n^{-1/2}$, whereas the better rate $\|Y_0 - Y_0^n\| \leq C n^{-1}$ at the initial time has also been reported elsewhere [72, 137]. This is not an essential disagreement but a simple difference in the evaluation point, due to the error on the forward component of order $1/2$ for all time points but the initial time (that is, $\|X_{t_k} - X_{t_k}^n\|_2 = \mathcal{O}(n^{-1/2})$ for all $k \in \{1, \cdots, n\}$) and the deterministic equality at the initial time (that is, $X_0 = X_0^n = \mathbf{x}$). Thus, the rates described in (7.1), (7.2), (7.3) and (7.4) can be conservative if the approximation is focused on the initial state $Y_0$ alone.

Next, we make a note on the complexity and dimensionality, which are undoubtedly important upon implementation individually, as well as inextricably bound up together. LSMC can work in up to 10 dimensions and tends to

be more efficient than Malliavin calculus based methods. For example, it is known [213] that the complexity of LSMC is $\mathcal{O}(\epsilon^{-d-4})$ to achieve the squared error $\epsilon^2$ in some instances, while the Malliavin calculus based method bears the complexity $\mathcal{O}(\epsilon^{-d-13})$, provided that the forward component is a geometric Brownian motion [47]. In general, the application of quantization or tree based methods is limited to considerably low-dimensional problems due to the use of the safety grid. Cubature methods with multi-linear interpolation result in the complexities $\mathcal{O}(\epsilon^{-3d/2+1/2})$ if the backward Euler discretization is employed and $\mathcal{O}(\epsilon^{-d+1/2})$ with the second-order discretization with the Richardson-Romberg extrapolation [66]. As such, in most cases, the complexity of backward methods (in combination with computation for conditional expectations) tends to explode exponentially as the dimension of the problem increases.

We also remark on the standing assumption of each category. Least-squares regression based methods generally work under minimal assumptions on BS-DEs (and SDEs), while a few extra regularity assumptions on the coefficients are required to allow the derivation of robust estimates for the involved error using various regression tools and also to ensure the stability of the algorithm (for instance, [61, 147, 257, 293]). Malliavin calculus based methods tend to require lots of extra assumptions, including various regularity assumptions and in particular, assumptions regarding the Malliavin weights (as in [148, 149, 241]). Quantization and tree based methods work under the standard Lipschiz conditions on forward and backward SDEs, while cubature methods require some extra smoothness on coefficients of the forward SDE and a smooth driver with Lipschitz terminal condition in most instances.

### 7.2.  *Forward numerical methods*

Next, we make a contrast of the four categories of forward numerical methods (Section 5), again, by providing a brief summary of the convergence property, and key features.

- Picard iteration (Section 5.1): For an approximate solution $(Y_t^{r,n})_{t\in[0,T]}$ in accordance with the Picard iteration method [22], it is shown that

$$\sup_{t\in[0,T]} \mathbb{E}\left[|Y_t - Y_t^{r,n}|^2\right] \le Cn^{-1} + C\left(\frac{1}{2} + Cn^{-1}\right)^r,$$

  for sufficiently large $n$. This scheme needs to be combined with a numerical method for computing conditional expectations (Section 4) and, by and large, similar limitations apply as described in Section 7.1. As such, in its implementation, variance reduction techniques play an importance role, for instance, importance sampling [23, 239] and control variates [138], in solving high-dimensional problems.
- Branching diffusion system (Section 5.2): For a BSDE $(Y, Z)$ with driver $f$ (independent of $\mathbf{z}$), it is shown under suitable conditions [170] that

$$\left|Y_0 - Y_0^{n,\ell}\right| \le C\left\|f - f^{n,\ell}\right\| + Cn^{-1/2},$$

where $Y^{n,\ell}$ denotes a branching diffusion approximation to the BSDE with a discretized $\ell$-th order polynomial driver $f^{n,\ell}$ [45, 46]. After the branching diffusion approximation has been applied, it is the usual practice to employ Monte Carlo simulation for implementation, such as a nonlinear Monte Carlo simulation [76], which is known to be effective in high-dimensional nonlinear pricing problems.

- Asymptotic expansion (Section 5.3): For a BSDE $(Y^\varepsilon, Z^\varepsilon)$ with a perturbed driver $\varepsilon f$, it holds under suitable conditions [145, 285] that

$$\sup_{t\in[0,T]} \mathbb{E}\left[\left|Y_t^\epsilon - \left(Y_t^{(0)} + \epsilon Y_t^{(1)} + \cdots + (\epsilon^m/m!)Y_t^{(m)}\right)\right|^2\right]^{1/2} = \mathcal{O}(\epsilon^{m+1}),$$

where $Y_t^{(k)} = ((d^k/d\epsilon^k)Y_t^\epsilon)|_{\epsilon=0}$ for $k \in \{0,1,\cdots\}$. After the expansion has been performed, similarly to the aforementioned branching diffusion system, often Monte Carlo simulation is used for implementation [76].

- Multilevel Picard approximation (Section 5.4): Consider a solution $u$ of a $d$-dimensional semilinear PDE corresponding to a BSDE model, and let $u_{n,M}^\theta : [0,T] \times \mathbb{R}^d \times \Omega \to \mathbb{R}$ denote an approximate solution to the PDE $u(\cdot,\cdot)$ by the multilevel Picard approximation in accordance with [98, 183]. For each $\mathbf{x} \in \mathbb{R}^d$, there exist $K : (0,1] \to \mathbb{N}$ and $c > 0$ such that for all $d \in \mathbb{N}$ and $\epsilon \in (0,1]$,

$$\mathbb{E}\left[\left|u(T,\mathbf{x}) - u_{K(\varepsilon),K(\varepsilon)}^0(T,\mathbf{x})\right|^2\right]^{1/2} \le \epsilon,$$

with the complexity $\mathcal{O}(\varepsilon^{-c}d^c)$, which suggests that for each initial state of the forward component, the initial state $Y_0$ of the backward component can be approximated without suffering from the curse of dimensionality.

To conclude the categories of forward numerical methods, we make a short remark on the standing assumption, as we have done towards the end of Section 7.1. As mentioned throughout the survey, forward numerical methods do not inherently work backwards to avoid the computation of conditional expectations, often with the aid of the FBSDE and PDE equivalence (Theorem 2.3). In return, it is rather evident that numerical methods based on solving the equivalent PDE (Section 5) must impose Assumption 2 so that the FBSDE and PDE equivalence holds. In addition, a few extra regularity assumptions are often made. For instance, the branching diffusion system based method (as of [45, 46, 170]) often requires the key assumption that the driver can be represented as the sum of a power series to be approximated by polynomials, which obviously does not hold for every BSDE. The asymptotic expansion method (as of [77, 118, 120, 285]) requires some smoothness on the driver to expand a nonlinear BSDE by linear BSDEs, while the condition can be relaxed in some instances [145]. Finally, to employ the multilevel Picard iteration method, one needs to carefully verify the relevant conditions on the driver as well as check the structure of the forward component (see, for example, [182, 183, 187, 188, 189]).

### *7.3. Summary*

To conclude the comparisons and discussions made above, and by considering the enormous numerical examples on BSDEs in the literature, we summarize in Table 1 key aspects of each category upon implementation, that is, what is approximated and the problem dimension. In particular, the dimension of the problem at hand is one of the more restrictive conditions for which method can be chosen, in particular, most methods are not efficient in high-dimensional problems. Note that we did not make a comparative discussion on deep learning based methods (Section 6) in the form of a separate subsection (like Sections 7.1 and 7.2), as their error and complexity analyses are still evolving, some in infancy, and awaiting major advances, as opposed to the effectiveness in very high-dimensional spatial approximation problems. Still, for the sake of completeness, we align the category of deep learning based methods at the bottom of Table 1 with a very broad perspective.

TABLE 1
*Two key aspects upon implementation*

| methods | target | dimension |
|---|---|---|
| LSMC + Backward Euler | function $u(t, \cdot)$ | $d \approx 10$ |
| Malliavin + Backward Euler | function $u(t, \cdot)$ | a few |
| Quantization + Backward Euler | function $u(t, \cdot)$ | a few |
| Tree + Backward Euler | function $u(t, \cdot)$ | one or two |
| Cubature + Second-order discretization | function $u(t, \cdot)$ | a few |
| Picard iteration | point $u(t, \mathbf{x})$ | higher than a few |
| Branching diffusion system | point $u(t, \mathbf{x})$ | higher than a few |
| Asymptotic expansion | point $u(t, \mathbf{x})$ | higher than a few |
| Multilevel Picard approximation | point $u(t, \mathbf{x})$ | $d > 100$ |
| Deep learning | function $u(t, \cdot)$ | $d > 100$ |

Without a doubt, computing time is a vital factor to consider when selecting a method, as there may be a need for very quick calculations. However, if no time pressure exists, then the choice of method is less restrictive. We remark that various attempts have been made so far in the literature by wisely splitting the required computation for massive parallelization on highly multicore GPUs. Parallelization here has naturally proven effective because the primary issue does not lie in the computational time but memory consumption requirements, for instance, by algorithms that require many sample paths at once on memory, such as binomial lattice based methods [87, 264, 265], the multistep method [201], LSMC [141, 142, 146], the four-step scheme [296], and the forward Picard iteration [210].

## 8. Numerical methods for BSDEs with nonstandard features

To address problems in stochastic control, finance, and partial differential equations, BSDEs often need to be equipped with extra features for better capturing relevant properties under consideration. There has been an increasing interest

in those classes of BSDEs and in developing numerical methods exclusively for each or some combinations of those features. Despite most of what follows having already appeared in their respective subsections, we again categorize and summarize those numerical methods here in terms of the class of BSDEs in brief, without going into much detail in order to avoid overloading the paper with lengthy technical intricacies.

## 8.1. Coupled FBSDEs

Consider the following FBSDE, where the coefficients of the forward SDE can depend on the backward components $(Y, Z)$:

$$\begin{cases} X_t = \mathbf{x} + \int_0^t \mu(s, X_s, Y_s, Z_s)ds + \int_0^t \sigma(s, X_s, Y_s)dW_s, \\ Y_t = \Phi(X_T) + \int_t^T f(s, X_s, Y_s, Z_s)ds - \int_t^T Z_s dW_s, \end{cases}$$

for $t \in [0, T]$. The FBSDE in this form is called a coupled FBSDE in the sense that the backward components are allowed to couple in the drift and the diffusion of the forward component, in sharp contrast to the decoupled formulation (2.6)-(2.7). Hence, the standard Ito generator $\mathcal{L}_t$ and the diffusion coefficient $\sigma$ in the PDE (2.9) for decoupled FBSDEs needs to replaced accordingly in order to carry the backward components, resulting in the following PDE under suitable technical conditions:

$$\begin{cases} (\partial/\partial t)v(t, \mathbf{x}) + \langle \mu(t, \mathbf{x}, v(t, \mathbf{x}), (\nabla v(t, \mathbf{x}))^\top \sigma(t, \mathbf{x}, v(t, \mathbf{x}))), \nabla v(t, \mathbf{x}) \rangle \\ \qquad + \frac{1}{2}\mathrm{tr}[\sigma^{\otimes 2}(t, \mathbf{x}, v(t, \mathbf{x}))\mathrm{Hess}(v(t, \mathbf{x}))] \\ \qquad + f(t, \mathbf{x}, v(t, \mathbf{x}), (\nabla v(t, \mathbf{x}))^\top \sigma(t, \mathbf{x}, v(t, \mathbf{x}))) = 0, \quad (t, \mathbf{x}) \in [0, T) \times \mathbb{R}^q, \\ v(T, \mathbf{x}) = \Phi(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^q. \end{cases}$$

As such, numerical approximation of a coupled FBSDE or the equivalent PDE is evidently far more intricate than that of a decouple counterpart, of which the forward component can be treated separately from the backward components. In particular, if the forward component depends on $Z$ as well, then the gradient of the solution $\nabla u(t, \mathbf{x})$ needs to be addressed inside the forward component. Such coupled FBSDEs naturally appear in the utility maximization problem with general utility function in economics [172].

As for numerical methods, the four-step scheme [226] opened the door for coupled FBSDEs as early as in the 1990s, followed by various studies through the corresponding semi- or quasi-linear parabolic PDE. Based on this four-step scheme, the authors in [92] develop an implementable numerical method. Specifically, for the PDE part, they use the combined characteristics and finite difference method to approximate its solution. Almost sure uniform convergence and weak convergence of the method are proven, with the rate of convergence being comparable to that of the approximation of the forward SDE (done using an Euler type scheme). In [89] and moreover [90], we find improved alternatives of the method described in [92], which weakens regularity assumptions

required. The authors in [236, 237, 238] also give a numerical method based on the four-step scheme, but propose a different method for solving the involved PDE. They approximate the solution of the PDE by using layer methods, which are constructed by means of a probabilistic approach. The derivatives of the solution to the PDE are found by using finite differences. The four-step scheme once again appears and is reconstructed in [296] with new conditions. It is then associated with the idea of domain decomposition methods (associated with the Schawrz waveform relaxation method). This approach is used in order to parallelize the related equations. Finally, in [151], we find a numerical method based on the four-step scheme which focuses on approximating the solution to coupled FBSDEs.

We add that various numerical methods have also been applied in tailored ways to coupled FBSDEs, such as Picard iterations [28], a transform into a control problem via a fully forward discretization [86], multistep schemes [221, 324], a defferred correction method for ODEs [288], Fourier methods for computing conditional expectations [180], and more recently, deep learning based methods [6, 163, 196, 197].

### 8.2. *Reflected BSDEs*

We first review reflected backward stochastic differential equations (RBSDEs) [103], which form an important class of BSDEs in the sense that they provide a deep insight into many practical problems, such as optimal stopping problems, American option pricing, stochastic optimal controls and differential games. Their solutions are reflected at a given stochastic process, called the obstacle, and provide a probabilistic representation for the unique viscosity solution of an obstacle problem for a nonlinear parabolic partial differential equation.

Now, consider the following BSDE, where the process $(Y_t)_{t\in[0,T]}$ is forced to stay above the process $(h(t, X_t))_{t\in[0,T]}$:

$$\begin{cases} Y_t = h(T, X_T) + \int_t^T f(s, X_s, Y_s, Z_s)ds + K_T - K_t - \int_t^T Z_s dW_s, \\ Y_t \geq h(t, X_t), \\ \int_0^T (Y_s - h(s, X_s))dK_s = 0, \\ X_t = \mathbf{x} + \int_0^t \mu(X_s)ds + \int_0^t \sigma(X_s)dW_s, \end{cases}$$

for $t \in [0, T]$, where the stochastic process $(K_t)_{t\in[0,T]}$ is continuous and non-decreasing with $K_0 = 0$, and grows only when $Y_t = h(t, X_t)$. The process $(Y_t)_{t\in[0,T]}$ can then be written in the form of the solution to an optimal stopping problem, as

$$Y_t = \operatorname*{esssup}_{\tau \in \mathcal{T}_t} \mathbb{E}\left[\int_t^\tau f(s, X_s, Y_s)ds + h(\tau, X_\tau)\Big| \mathcal{F}_t\right], \quad t \in [0, T],$$

where $\mathcal{T}_t$ denotes the set of $(\mathcal{F}_s)_{s\in[t,T]}$-stopping times taking values in $[t, T]$, and provides a probabilistic interpretation $Y_t = v(t, X_t)$ where $v$ is a solution to the

obstacle problem

$$
\begin{cases}
\max\left\{((\partial/\partial t) + \mathcal{L}_t)v(t, \mathbf{x}) + f(t, \mathbf{x}, v(t, \mathbf{x})),\, h(t, \mathbf{x}) - v(t, \mathbf{x})\right\} = 0, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (t, \mathbf{x}) \in [0, T) \times \mathbb{R}^q, \\
v(T, \mathbf{x}) = h(T, \mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^q,
\end{cases}
$$

in a similar yet different manner from Theorem 2.3. For theoretical details on RBSDEs, we refer the reader to, for instance, the monograph [91, Chapter 14].

The most popular application of RBSDEs is, perhaps, the pricing of American options [103, 104], due to its structure with lower obstacles. In fact, the RBSDE framework is not strictly necessary for the pricing of American options, as the classical binomial pricing model for example works well for its low-dimensional problems, while least-square Monte-Carlo methods [223] remain efficient for high-dimensional problems, both outside the RBSDE framework. Nonetheless, the RBSDE framework is useful for the pricing of American options, for instance, through detailed error analysis by approximating RBSDEs in quantization methods [11, 12] and its stability results for approximating its payoff function [48].

As for discretization methods, it is proved in [11] that the backward Euler type discretization attains the order $1/2$ for BSDEs with Lipchitz coefficients and the order 1 for those with semi-convex obstacles. Other backward Euler methods are developed for RBSDEs in [194, 231] based on the random walk approximation, in [263] based on the lattice tree and in [47] in combination with LSMC regression and the Malliavin approach (Sections 4.1 and 4.2) for approximating conditional expectations. The stability analysis for backward Euler methods for RBSDEs is conducted in [39, 229]. Most recently, the stability and convergence analysis is refined in [281] with a focus on quadratic RBSDEs.

After discretizing RBSDEs, more complex equations need to be solved concurrently with approximation of conditional expectations in the presence of the obstacle term. This difficulty has been addressed by discretizing the state space, such as the so-called quantization, before employing the dynamic programming principle. Quantization algorithms for RBSDEs, initially developed in [13], have been investigated intensively in [11, 12] to deal with multi-dimensional RBSDEs with an application to the pricing of an American option, and further extended, for example, in combination with the higher-order discretization [234] and for those whose driver depends on the control process [248]. In addition, the primal-dual method is generalized in [25] to a backward dynamic programming equation associated with time discretization schemes for RBSDEs.

In the literature, other advanced types of RBSDEs have also attracted attention, such as discrete RBSDEs, RBSDEs with multiple obstacles and RBSDEs with jumps. To deal with those classes, the concept of penalization, which was originally employed for proving the existence and uniqueness of RBSDEs [103], proves useful for numerical discretization [235] as well, and can be combined with Fourier transform techniques [190]. It has been applied to optimal switching problems in real option pricing [158], followed by RBSDEs with oblique reflections [63, 178], doubly RBSDEs [308], and doubly RBSDEs with jumps [95].

A numerical method is developed in [300] for doubly RBSDEs whose generator depends on the future values of the solution with application to the default risk modeling.

### *8.3. BSDEs with jumps*

BSDEs may contain jumps in their backward component as

$$-dY_t = f(t, X_t, Y_t, Z_t, U_t) - Z_t dW_t - U_t dN_t,$$

with a suitable jump process $(N_t)_{t \in [0,T]}$ and an control process $(U_t)_{t \in [0,T]}$, and often in their forward component as well. On top of non-trivial issues on existence, uniqueness and stability [254], the presence of jumps increases complexity of numerical implementation, for instance, by paying a great deal of attention to the jump size and timing. For instance, discretization schemes are developed, based on Malliavin calculus on the Gaussian space [42], and the so-called shot noise representation of the Poisson random measure [314] in [232]. A forward scheme for BSDEs [52] is extended [128] to BSDEs with jumps by Wiener chaos expansion and Picard iterations. A numerical algorithm is developed in [1] to approximate the solution to a decoupled FBSDE driven by a pure Lévy process with its small jumps approximated by a suitable Brownian motion. An explicit prediction-correction scheme is developed for solving decoupled FBSDEs with jumps [112]. In [212], Brownian and Poissonian components are independently approximated by random walks. More complex structures are also of interest. A Fourier-based method [38] and asymptotic expansion [121] are proposed, by which non-Poissonian local jump measures can be taken care of. Numerical methods for doubly reflected BSDEs with jumps are developed, with [94] and without [95] penalization. A single jump (not systematic jumps) is considered in the context of BSDEs in [203], where a single jump can represent a default time in credit risk or counterparty risk. A deep learning based method is developed in [135] and applied to BSDEs with jumps. We refer the reader to the monograph [91] for details as well as applications in insurance and finance.

### *8.4. BSDEs with non-global Lipschitz conditions*

The majority of key properties of numerical methods for BSDEs are derived upon the global Lipschitz conditions on the drivers and terminal condition, whereas such restrictive conditions do not hold all the time in important applications, such as the Fisher-KPP and FitzHugh-Nagumo equations. In the literature, considerable attempts have been made to develop viable numerical methods where those Lipschitz conditions are relaxed. The $L^2$-time regularity of the $Z$-component is studied in [144] for an irregular terminal condition (such as an indicator function), as an extension of the $L^2$-time regularity result [317] for a Lipschitz terminal, where the order of convergence is explicitly connected to the rate of decrease of the expected conditional variance of the terminal. Two

discretization methods are studied in [298] for a class of locally Lipschitz Markovian BSDEs: One is a backward Euler scheme (Section 3.1) by approximating a projection of the $Z$ process, while the other is the Malliavin weight scheme (Section 4.2, especially (4.6) and (4.7)) by directly approximating the marginals of the $Z$ process, where advanced a priori estimates and stability results for this class of BSDEs are derived by extending the representation theorem [228] and employed for obtaining competitive convergence rates. It is proved [71] that a backward stochastic difference equation (BS$\Delta$E) admits a solution and its sequence is convergent to a BSDE as the time-grid gets finer even when their drivers are not Lipschitz in the $Z$ component. An error analysis of a time discretization method is performed in [219] for systems of BSDEs with drivers of polynomial growth and monotone in the state variable, where a tamed version of the explicit Euler scheme is shown to converge despite that the standard version may diverge unlike with the canonical Lipschitz driver. A general framework is developed in [220] for explicit numerical schemes for BSDEs with drivers of polynomial growth, where the convergence of some modified explicit scheme is of the same rate as implicit schemes and has comparable computing cost to the standard explicit scheme. Finally, for drivers not Lipschitz in the backward component, the Picard iteration is applied to discretized FBSDEs [37] and is employed for deriving the existence and uniqueness for $G$-BSDEs [319].

## 8.5.  Quadratic BSDEs

BSDEs with generators of quadratic growth (with respect to the variable $\mathbf{z}$ as in $|f(t, \mathbf{x}, \mathbf{y}, \mathbf{z})| \leq C(1 + \|\mathbf{y}\| + \|\mathbf{z}\|^2)$) have been studied actively under the name of quadratic BSDEs. Quadratic BSDEs play important roles in mathematical finance, such as utility optimization with exponential utility functions and risk minimization for the entropic risk measure. Their numerical aspects have thus been investigated ever since the theoretical formalization [206]. In contrast to standard BSDEs, the treatment of the quadratic driver is not trivial, particularly in placing the upper bound estimate of the control process $Z$. In the development of numerical methods, this issue is typically addressed by truncating the quadratic driver.

The well-known path regularity theorem [317] is extended in [191] to the quadratic-growth setting along with convergence rate of the distance between the solution of a quadratic BSDE and its approximation by truncation. In [192], the Cole-Hopf exponential transformation is applied to the approximate solution. Time-discretization schemes are developed in [272] for quadratic BSDEs based on truncation and non-uniform time partition, and in [65] with the error of order less than $1/2$. For quadratic BSDEs with reflection, a truncated discrete-time numerical scheme is proposed in [281] along with some practical examples. In [119], an asymptotic expansion technique (Section 5.3) is developed for a quadratic-growth FBSDE appearing in an incomplete market with stochastic volatility. An approximation method is constructed in [122] for quadratic BSDEs using semi-analytic asymptotic expansion. With a view towards utility

maximization problems, a special class of backward stochastic partial differential equations (Section 8.8) is investigated in [193], along with numerical simulation for relevant portfolio optimization problems.

### 8.6. Second-order BSDEs

Second-order BSDEs (2BSDEs) are a type of BSDE whose nonlinear drift contains the second order derivative of the corresponding PDE, widely applied in financial modeling such as the uncertain volatility model, transaction cost model, illiquid market model and the pricing for passport options. In general, a process $(Y_t, Z_t, \Gamma_t, \alpha_t)_{t \geq 0}$ is called a 2BSDE if it solves the system

$$\begin{cases} -dY_t = \left[ f(X_t, Y_t, Z_t, \Gamma_t) - \frac{1}{2}\mathrm{tr}\left[\sigma^{\otimes 2}(t, X_t)\Gamma_t\right]\right] dt - Z_t \sigma(t, X_t) dW_t, \\ dZ_t = \alpha_t dt + \Gamma_t \sigma(t, X_t) dW_t, \end{cases}$$

for $t \in [0, T]$, with $Y_T = g(X_T)$ and a suitable forward process $X$ in the form of (2.7). This system can be expressed as $Y_t = u(t, X_t)$, $Z_t = \nabla u(t, X_t)$, $\Gamma_t = \mathrm{Hess}(u(t, X_t))$ and $\alpha_t = ((\partial/\partial t) + \mathcal{L})\nabla u(t, X_t)$ for $t \in [0, T]$ on the basis of the parabolic PDE:

$$\frac{\partial}{\partial t}u(t, x) + f(t, x, u(t, x), \nabla u(t, x), \mathrm{Hess}(u(t, x))) = 0, \quad u(T, x) = g(x),$$

whose full nonlinearity makes the class of 2BSDE distinct from the standard BSDEs. Its existence and the uniqueness are proved in [70] under Lipschitz continuity of the driver and suitable conditions, along with the numerical scheme: starting with $\overline{Y}_T = g(\overline{X}_T)$ and $\overline{Z}_T = \nabla g(\overline{X}_T)$, proceed backwards for $k \in \{1, \cdots, n\}$,

$$\begin{cases} \overline{Y}_{t_{k-1}} = \mathbb{E}[\overline{Y}_{t_k}|\mathscr{F}_{t_{k-1}}] + \left[ f(t_{k-1}, \overline{X}_{t_{k-1}}, \overline{Y}_{t_{k-1}}, \overline{Z}_{t_{k-1}}, \overline{\Gamma}_{t_{k-1}}) \right. \\ \left. \hspace{4cm} -\frac{1}{2}\mathrm{tr}\left[\sigma^{\otimes 2}(t_{k-1}, \overline{X}_{t_{k-1}})\overline{\Gamma}_{t_{k-1}}\right] \right](t_k - t_{k-1}), \\ \overline{Z}_{t_{k-1}} = \frac{1}{t_k - t_{k-1}}(\sigma^{-1}(t_{k-1}, \overline{X}_{t_{k-1}}))^\top \mathbb{E}\left[\overline{Y}_{t_k}(W_{t_k} - W_{t_{k-1}})|\mathscr{F}_{t_{k-1}}\right], \\ \overline{\Gamma}_{t_{k-1}} = \frac{1}{t_k - t_{k-1}}\mathbb{E}\left[\overline{Z}_{t_k}(W_{t_k} - W_{t_{k-1}})^\top|\mathscr{F}_{t_{k-1}}\right]\sigma^{-1}(t_{k-1}, \overline{X}_{t_{k-1}}), \end{cases}$$

based on the aforementioned nonlinear Feynman-Kac representation. In [106], an implementable explicit method is developed to approximate the conditional expectations appearing in the above scheme. A variance reduction method is proposed in [5] for the computation of involving conditional expectations. 2BSDEs can also be effectively solved numerically with a monotone scheme [156]. We mention that deep learning based approximation algorithms (Section 6) are developed in [18] for 2BSDEs.

### 8.7. McKean-Vlasov FBSDEs

When dealing with a stochastic differential game with mean field interactions, also known as a mean field game, one often encounters the following optimal

control problem:

$$\inf_{\alpha \in \mathcal{A}} \mathbb{E} \left[ \int_t^T f(s, X_s, \mu_s, \alpha_s) ds + g(X_T, \mu_T) \Big| X_t = \mathbf{x} \right] =: u(t, \mathbf{x}), \qquad (8.1)$$

subject to the so-called McKean-Vlasov stochastic differential equation

$$dX_s = b(s, X_s, \mu_s, \alpha_s) ds + \sigma(s, X_s, \mu_s, \alpha_s) dW_s, \qquad (8.2)$$

where $(\mu_t)_{t \in [0,T]}$ is a deterministic flow of probability measures and the infimum in (8.1) is taken over the set $\mathcal{A}$ of all progressively measurable processes. The McKean-Vlasov SDE (8.2) can be interpreted as a limit of the empirical measure of increasing individual particles, known as the propagation of chaos [57] where the particles tend to be independent of each other in this limit as the impact of each decreases. Under suitable technical conditions, the solution $u$ to the optimization problem (8.1) corresponds to that of a suitable HJB equation, as well as providing an insightful probabilistic interpretation through the representation $u(t, X_t) = Y_t$ in accordance with the McKean-Vlasov FBSDE, in general: for $s \in [t, T]$,

$$dX_s = b(s, X_s, \mathcal{L}(X_s), \mathcal{L}(Y_s), \mathcal{L}(Z_s)) ds + \sigma(s, X_s, \mathcal{L}(X_s), \mathcal{L}(Y_s), \mathcal{L}(Z_s)) dW_s,$$
$$-dY_s = f(s, X_s, \mathcal{L}(X_s), \mathcal{L}(Y_s), \mathcal{L}(Z_s)) ds - Z_s dW_s,$$

with $X_t = \xi$ and $Y_T = g(X_T, \mathcal{L}(X_T))$, where $\mathcal{L}(F)$ denotes the law of the random element $F$. We note that two approaches (of Pontryagin and weak types) result in distinct McKean-Vlasov FBSDEs and thus numerical methods [7].

As for relevant numerical methods, a variety of deterministic ones can be found in the literature for mean field games [2, 211], for instance, based on finite differences or variational approaches, whereas McKean-Vlasov FBSDEs above pave the way towards probabilistic numerical methods and analysis [7], such as a tree method [62], first-order and Crank-Nicolson schemes [321], a higher order discretization method for decoupled cases [88], McKean-Vlasov anticipative FBSDEs arising in initial margin requirements [4] and a posteriori error estimates for fully coupled McKean-Vlasov FBSDEs [270], and recently very actively, deep learning based methods [55, 56, 110, 131, 161].

### *8.8.  BSPDEs*

Backward stochastic partial differential equations are of emerging interest from an implementation point view in its intimate relation to stochastic optimal and utility maximization problems. A few existing developments of numerical methods are as follows. For forward-backward stochastic heat equations from stochastic optimal control, discretization methods are developed via spatial discretization [96], space-time discretization [268], and the local discontinuous Galerkin method [216]. A special class of BSPDEs is investigated in [193] via their reduction to BSDEs from both theoretical and numerical perspectives in the context

of utility maximization and portfolio optimization problems. A deep learning based method is developed in [290] for forward-backward doubly stochastic differential equations, where the additional diffusion terms in its BSDE component yields an equivalence to semilinear parabolic SPDEs as well as optimal control problems.

## 9. Concluding remarks

In this survey, we have focused on examining a wide array of numerical methods for BSDEs along with the main assumptions made, key convergence properties, as well as summaries of the key advantages and disadvantages. In particular, we have characterized broadly into three categories; backward (Section 3), forward (Section 5) and deep learning based methods (Section 6), along with the computation of conditional expectations (Section 4) to complete the categories of backward numerical methods (Section 3). Within those categories, we have further categorized the involved methods. For backward methods, we have sub-categorized the methods as backward Euler methods (Section 3.1) and higher-order methods (Section 3.2). For the computation of conditional expectations to complement backward methods (Section 3), we have sub-categorized the methods as least-squares regression based methods (Section 4.1), Malliavin calculus based methods (Section 4.2), Malliavin weights dynamic programming with regression methods (Section 4.2.2), quantization methods (Section 4.3), tree based methods (Section 4.4), and cubature methods (Section 4.5). For forward methods, we have sub-categorized the methods as Picard iteration methods (Section 5.1), branching diffusion system based methods (Section 5.2), asymptotic expansion (Section 5.3), and multilevel Picard iterations (Section 5.4). For deep learning based methods, we have sub-categorized the methods as deep BSDE (Section 6.1), deep backward dynamic programming (Section 6.2), deep splitting (Section 6.3), and deep Galerkin methods and physics-informed neural networks (Section 6.4).

As is clear from the present survey, numerical methods for BSDEs are intrinsically involved, and further, have been presented in a scattered manner in the literature. Hence, for the sake of collective comparison, we have devoted Section 7 to displaying those surveyed categories against a few important practical aspects, such as convergence properties and implementation aspects, each of which is crucial for understanding the power and limitations of numerical methods from a practical viewpoint.

In addition, BSDEs of complex structure, and thus the numerical methods employed for them, have become of growing interest in the years to come. To enhance this line of research, we have also surveyed existing numerical methods in terms of BSDEs with nonstandard features in Section 8, such as coupled (Section 8.1), reflection (Section 8.2), jumps (Section 8.3), non-Lipschtiz (Section 8.4), quadratic (Section 8.5), second-order (Section 8.6) and McKean-Vlasov BSDEs (Section 8.7), and BSPDEs (Section 8.8). To the best of our knowledge, numerical aspects have not been explored for other types as of yet,

such as ergodic BSDEs, delayed BSDEs, and BSDEs with regime-switching. As such, we hope this survey can serve as an initial yet insightful guideline when selecting and developing appropriate numerical methods for the interested BSDE, now and in the future.

With the advent of deep learning based methods (Section 6), advances in numerical methods for BSDEs appear to have entered a new era. In sharp contrast to the conventional approaches (Sections 3, 4, and 5), deep learning based methods have opened the door wide for very high-dimensional BSDEs and nonlinear PDEs by employing the policy functions for readily approximating gradient dependent nonlinearities, to which the existing methods have devoted substantial effort. Further lines of numerical methods are still expected to come, which will benefit from being built upon the distinctive developments and advantageous features of the existing numerical methods for BSDEs and PDEs, to say nothing of expanding fields of application in relation to high-dimensional and nonlinear problems. However, it also remains to address a variety of unexplored theoretical aspects of the existing numerical methods, for instance, ones for BSDEs with nonstandard features (Section 8), not only from the traditional perspective of applied mathematics and numerical analysis, but also from the viewpoint of pure mathematics. Synergy and complementarity of those distinctive expertise would be beneficial and indispensable for making further advances on numerical methods for BSDEs.

## Acknowledgments

## References

[1] S. Aazizi. Discrete-time approximation of decoupled forward-backward stochastic differential equations driven by pure jump Lévy processes. *Advances in Applied Probability*, 45(3):791–821, 2013. MR3102472

[2] Y. Achdou and M. Laurière. Mean field games and applications: Numerical aspects. In P. Cardaliaguet and A. Porretta, editors, *Mean Field Games: Cetraro, Italy 2019*, pages 249–307. Springer International Publishing, Cham, 2020. MR4214777

[3] A. Agarwal and J. Claisse. Branching diffusion representation of semi-linear elliptic PDEs and estimation using Monte Carlo method. *Stochastic Processes and their Applications*, 130(8):5006–5036, 2020. MR4108480

[4] A. Agarwal, S. D. Marco, E. Gobet, J. G. López-Salas, F. Noubiagain, and A. Zhou. Numerical approximations of McKean anticipative backward stochastic differential equations arising in initial margin requirements. *ESAIM: ProcS*, 65:1–26, 2019. MR3968534

[5] S. Alanko and M. Avellaneda. Reducing variance in the numerical solution of BSDEs. *Comptes Rendus Mathematique*, 351(3):135–138, 2013. MR3038003

[6] K. Andersson, A. Andersson, and C. W. Oosterlee. Convergence of a robust deep FBSDE method for stochastic control. *SIAM Journal on Scientific Computing*, 45(1):A226–A255, 2023. MR4551869

[7] A. Angiuli, C. V. Graves, H. Li, J.-F. Chassagneux, F. Delarue, and R. Carmona. Cemracs 2017: numerical probabilistic approach to MFG. *ESAIM: ProcS*, 65:84–113, 2019. MR3968538

[8] S. Ankirchner and A. Fromm. Solving fully coupled FBSDEs by minimizing a directly calculable error functional. hal-01915772, 2018.

[9] F. Antonelli and A. Kohatsu-Higa. Filtration stability of backward sde's. *Stochastic Analysis and Applications*, 18(1):11–37, 2000. MR1739298

[10] Y. Armenti and S. Crépey. Central clearing valuation adjustment. *SIAM Journal on Financial Mathematics*, 8(1):274–313, 2017. MR3634986

[11] V. Bally and G. Pagès. Error analysis of the optimal quantization algorithm for obstacle problems. *Stochastic Processes and their Applications*, 106(1):1–40, 2003. MR1983041

[12] V. Bally and G. Pagès. A quantization algorithm for solving multi-dimensional discrete-time optimal stopping problems. *Bernoulli*, 9(6):1003–1049, 2003. MR2046816

[13] V. Bally, G. Pagès, and J. Printems. A stochastic quantization method for nonlinear problems. *Monte Carlo Methods and Applications*, 7(1-2):21–34, 2001. MR1828193

[14] F. Bao, R. Archibald, and P. Maksymovych. Lévy backward SDE filter for jump diffusion processes and its applications in material sciences. *Communications in Computational Physics*, 27(2):589–618, 2019. MR4040956

[15] F. Bao, Y. Cao, and H. Chi. Adjoint forward backward stochastic differential equations driven by jump diffusion processes and its application to nonlinear filtering problems. *International Journal for Uncertainty Quantification*, 9(2):143–159, 2019. MR3977063

[16] F. Bao, Y. Cao, and J. Yong. Data informed solution estimation for forward-backward stochastic differential equations. *Analysis and Applications*, 19(03):439–464, 2021. MR4233707

[17] C. Beck, S. Becker, P. Cheridito, A. Jentzen, and A. Neufeld. Deep splitting method for parabolic PDEs. *SIAM Journal on Scientific Computing*, 43(5):A3135–A3154, 2021. MR4310910

[18] C. Beck, W. E, and A. Jentzen. Machine learning approximation algorithms for high-dimensional fully nonlinear partial differential equations and second-order backward stochastic differential equations. *Journal of Nonlinear Science*, 29(4):1563–1619, 2019. MR3993178

[19] C. Beck, M. Hutzenthaler, A. Jentzen, and B. Kuckuck. An overview on deep learning-based approximation methods for partial differential equations. *Discrete and Continuous Dynamical Systems – B*, 28(6):3697–3746, 2023. MR4547087

[20] C. Beck, A. Jentzen, and T. Kruse. Nonlinear Monte Carlo methods

with polynomial runtime for high-dimensional iterated nested expectations. arXiv:2009.13989, 2020.

[21] S. Becker, A. Jentzen, M. S. Müller, and P. von Wurstemberger. Learning the random variables in Monte Carlo simulations with stochastic gradient descent: Machine learning for parametric PDEs and financial derivative pricing. arXiv:2202.02717, 2022.

[22] C. Bender and R. Denk. A forward scheme for backward SDEs. *Stochastic Processes and their Applications*, 117:1793–1812, 2007. MR2437729

[23] C. Bender and T. Moseler. Importance sampling for backward SDEs. *Stochastic Analysis and Applications*, 28(2):226–253, 2010. MR2739557

[24] C. Bender and S. Pokalyuk. Discretization of backward stochastic Volterra integral equations. In *Recent Developments in Computational Finance*, pages 245–278. World Scientific, 2013. MR3288637

[25] C. Bender, N. Schweizer, and J. Zhuo. A primal-dual algorithm for BSDEs. *Mathematical Finance*, 27(3):866–901, 2017. MR3668160

[26] C. Bender and J. Steiner. Least-squares Monte Carlo for backward SDEs. In R. A. Carmona, P. Del Moral, P. Hu, and N. Oudjane, editors, *Numerical Methods in Finance*, pages 257–289, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. MR3287774

[27] C. Bender and J. Steiner. A posteriori estimates for backward SDEs. *SIAM/ASA Journal on Uncertainty Quantification*, 1(1):139–163, 2013. MR3283881

[28] C. Bender and J. Zhang. Time discretization and Markovian iteration for coupled FBSDEs. *The Annals of Applied Probability*, 18(1):143 – 177, 2008. MR2380895

[29] C. Bénézet, J.-F. Chassagneux, and C. Reisinger. A numerical scheme for the quantile hedging problem. *SIAM Journal on Financial Mathematics*, 12(1):110–157, 2021. MR4207173

[30] A. Bensoussan and J.-L. Lions. *Applications des Inequations variationnelles en controle stochastique*, volume 6 of *Methodes mathematicques de l'informatique*. Dunod, 1978. MR0513618

[31] J. Berner, P. Grohs, and A. Jentzen. Analysis of the generalization error: Empirical risk minimization over deep artificial neural networks overcomes the curse of dimensionality in the numerical approximation of Black–Scholes partial differential equations. *SIAM Journal on Mathematics of Data Science*, 2(3):631–657, 2020. MR4127967

[32] M. Bichuch, A. Capponi, and S. Sturm. Arbitrage-free pricing of XVA – Part II: PDE representation and numerical analysis. arXiv:1502.06106, 2016.

[33] M. Bichuch, A. Capponi, and S. Sturm. Arbitrage-free XVA. *Mathematical Finance*, 28(2):582–620, 2018. MR3780968

[34] J. Bismut. Conjugate convex functions in optimal stochastic control. *Journal of Mathematical Analysis and Applications*, 44(2):384–404, 1973. MR0329726

[35] J. Blechschmidt and O. G. Ernst. Three ways to solve partial differential equations with neural networks — a review. *GAMM-Mitteilungen*,

44(2):e202100006, 2021. MR4270459

[36] D. Borkowski and K. Jańczak-Borkowska. Reduction of JPEG artifacts using BSDEs. In *WSCG 2021 Proceedings CSRN 3101*, 2021.

[37] D. Borkowski and K. Jańczak-Borkowska. Approximation of BSDE with non Lipschitz coefficient. *Stochastic Analysis and Applications*, 40(1):63–77, 2022. MR4357214

[38] A. Borovykh, A. Pascucci, and C. W. Oosterlee. Efficient computation of various valuation adjustments under local Lévy models. *SIAM Journal on Financial Mathematics*, 9(1):251–273, 2018. MR3759867

[39] B. Bouchard and J.-F. Chassagneux. Discrete-time approximation for continuously and discretely reflected BSDEs. *Stochastic Processes and their Applications*, 118(12):2269–2293, 2008. MR2474351

[40] B. Bouchard, K. W. Chau, A. Manai, and A. Sid-Ali. Monte-Carlo methods for the pricing of American options: a semilinear BSDE point of view. *ESAIM: Proceedings and Surveys*, 65:294–308x, 2019. MR3968545

[41] B. Bouchard, I. Ekeland, and N. Touzi. On the Malliavin approach to Monte Carlo approximation of conditional expectations. *Finance and Stochastics*, 8(1):45–71, 2004. MR2022978

[42] B. Bouchard and R. Elie. Discrete-time approximation of decoupled forward-backward SDE with jumps. *Stochastic Processes and their Applications*, 118(1):53–75, 2008. MR2376252

[43] B. Bouchard, R. Elie, and L. Moreau. Regularity of BSDEs with a convex constraint on the gains-process. *Bernoulli*, 24(3):1613–1635, 2018. MR3757510

[44] B. Bouchard, R. Elie, and N. Touzi. Discrete-time approximation of BSDEs and probabilistic schemes for fully nonlinear PDEs. In H. Albrecher, W. J. Runggaldier, and W. Schachermayer, editors, *Advanced Financial Modelling*, pages 91–124. De Gruyter, 2009. MR2648459

[45] B. Bouchard, X. Tan, and X. Warin. Numerical approximation of general Lipschitz BSDEs with branching processes. *ESAIM: Proceedings and Surveys*, 65, 2017. MR3968546

[46] B. Bouchard, X. Tan, X. Warin, and Y. Zou. Numerical approximation of BSDEs using local polynomial drivers and branching processes. *Monte Carlo Methods and Applications*, 23(4), 2017. MR3745458

[47] B. Bouchard and N. Touzi. Discrete-time approximation and Monte-Carlo simulation of backward stochastic differential equations. *Stochastic Processes and their Applications*, 111(2):175–206, 2004. MR2056536

[48] B. Bouchard and X. Warin. Monte-Carlo valuation of American options: Facts and new algorithms to improve existing methods. In R. A. Carmona, P. Del Moral, P. Hu, and N. Oudjane, editors, *Numerical Methods in Finance*, pages 215–255, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. MR3287773

[49] P. Briand, B. Delyon, and J. Mémin. Donsker-type theorem for BSDEs. *Electronic Communications in Probability*, 6(1):1–14, 2001. MR1817885

[50] P. Briand, B. Delyon, and J. Mémin. On the robustness of backward stochastic differential equations. *Stochastic Processes and their Applica-*

*tions*, 97(2):229–253, 2002. MR1875334

[51] P. Briand, C. Geiss, S. Geiss, and C. Labart. Donsker-type theorem for BSDEs: Rate of convergence. *Bernoulli*, 27(2):899–929, 2021. MR4255220

[52] P. Briand and C. Labart. Simulation of BSDEs by Wiener chaos expansion. *The Annals of Applied Probability*, 24(3):1129–1171, 2014. MR3199982

[53] M. Broadie, J. Cvitaniç, and H. Soner. Optimal replication of contingent claims under portfolio constraints. *Review of Financial Studies*, 11(1):59–79, 1998.

[54] G. Callegaro, A. Gnoatto, and M. Grasselli. A fully quantization-based scheme for FBSDEs. *Applied Mathematics and Computation*, 441:127666, 2023. MR4510021

[55] S. Campbell, Y. Chen, A. Shrivats, and S. Jaimungal. Deep learning for principal-agent mean field games. arXiv:2110.01127, 2021.

[56] R. Carmona and M. Laurière. Convergence analysis of machine learning algorithms for the numerical solution of mean field control and games: II— the finite horizon case. *The Annals of Applied Probability*, 32(6):4065 – 4105, 2022. MR4522347

[57] L.-P. Chaintron and A. Diez. Propagation of chaos: A review of models, methods and applications. II. applications. *Kinetic and Related Models*, 15(6):1017–1173, 2022. MR4489769

[58] J.-F. Chassagneux. Linear multistep schemes for BSDEs. *SIAM Journal on Numerical Analysis*, 52(6):2815–2836, 2014. MR3284573

[59] J.-F. Chassagneux, J. Chen, and N. Frikha. Deep Runge-Kutta schemes for BSDEs. arXiv:2212.14372, 2022.

[60] J.-F. Chassagneux, J. Chen, N. Frikha, and C. Zhou. A learning scheme by sparse grids and Picard approximations for semilinear parabolic PDEs. *IMA Journal of Numerical Analysis*, 11 2022. drac066.

[61] J.-F. Chassagneux and D. Crisan. Runge-Kutta schemes for backward stochastic differential equations. *The Annals of Applied Probability*, 24(2):679–720, 2014. MR3178495

[62] J.-F. Chassagneux, D. Crisan, and F. Delarue. Numerical method for FBSDEs of McKean–Vlasov type. *The Annals of Applied Probability*, 29(3):1640 – 1684, 2019. MR3914553

[63] J.-F. Chassagneux, R. Elie, and I. Kharroubi. Discrete-time approximation of multidimensional BSDEs with oblique reflections. *The Annals of Applied Probability*, 22(3):971–1007, 2012. MR2977983

[64] J.-F. Chassagneux and A. Richou. Numerical stability analysis of the Euler scheme for BSDEs. *SIAM Journal on Numerical Analysis*, 53(2):1172–1193, 2015. MR3338675

[65] J.-F. Chassagneux and A. Richou. Numerical simulation of quadratic BSDEs. *The Annals of Applied Probability*, 26(1):262–304, 2016. MR3449318

[66] J.-F. Chassagneux and C. A. G. Trillos. Cubature method to solve BSDEs: Error expansion and complexity control. *Mathematics of Computation*, 89:1895–1932, 2020. MR4081922

[67] J.-F. Chassagneux and M. Yang. Numerical approximation of singular forward-backward SDEs. *Journal of Computational Physics*, 468:111459,

2022. MR4458991

[68] K. W. Chau and C. W. Oosterlee. On the wavelet-based SWIFT method for backward stochastic differential equations. *IMA Journal of Numerical Analysis*, 38(2):1051–1083, 2018. MR3800047

[69] K. W. Chau and C. W. Oosterlee. Stochastic grid bundling method for backward stochastic differential equations. *International Journal of Computer Mathematics*, 96(11):2272–2301, 2019. MR4008125

[70] P. Cheridito, H. M. Soner, N. Touzi, and N. Victoir. Second-order backward stochastic differential equations and fully nonlinear parabolic PDEs. *Communications on Pure and Applied Mathematics*, 60(7):1081–1110, 2007. MR2319056

[71] P. Cheridito and M. Stadje. BS$\Delta$Es and BSDEs with non-Lipschitz drivers: Comparison, convergence and robustness. *Bernoulli*, 19(3):1047–1085, 2013. MR3079306

[72] D. Chevance. Numerical methods for backward stochastic differential equations. In L. C. G. Rogers and D. Talay, editors, *Numerical Methods in Finance*, pages 232–244. Cambridge University Press, 1997. MR1470517

[73] F. Coquet, V. Mackevičius, and J. Mémin. Stability in D of martingales and backward equations under discretization of filtration. *Stochastic Processes and their Applications*, 75(2):235–248, 1998. MR1632205

[74] F. Coquet, J. Mémin, and L. Slominski. On weak convergence of filtrations. *Séminaire de probabilités de Strasbourg*, 35:306–328, 2001. MR1837294

[75] S. Crépey, R. Gerboud, Z. Grbac, and N. Ngor. Counterparty risk and funding: The four wings of the TVA. *International Journal of Theoretical and Applied Finance*, 16(2):1350006, 2013. MR3056818

[76] S. Crépey and T. M. Nguyen. Nonlinear Monte Carlo schemes for counterparty risk on credit derivatives. In K. Glau, Z. Grbac, M. Scherer, and R. Zagst, editors, *Innovations in Derivatives Markets*, pages 53–82, Cham, 2016. Springer International Publishing.

[77] S. Crépey and S. Song. Counterparty risk and funding: immersion and beyond. *Finance and Stochastics*, 20(4):901–930, 2016. MR3551856

[78] D. Crisan and F. Delarue. Sharp derivative bounds for solutions of degenerate semi-linear partial differential equations. *Journal of Functional Analysis*, 263(3024-3101), 2012. MR2973334

[79] D. Crisan and K. Manolarakis. Probabilistic methods for semilinear partial differential equations. Applications to finance. *ESAIM: M2AN*, 44(5):1107–1133, 2010. MR2731405

[80] D. Crisan and K. Manolarakis. Solving backward stochastic differential equations using the cubature method: Application to nonlinear pricing. *SIAM Journal on Financial Mathematics*, 3(1):534–571, 2012. MR2968045

[81] D. Crisan and K. Manolarakis. Second order discretization of backward SDEs and simulation with the cubature method. *Annals of Applied Probability*, 24(2):652–678, 2014. MR3178494

[82] D. Crisan, K. Manolarakis, and N. Touzi. On the Monte Carlo simulation of BSDEs: An improvement on the Malliavin weights. *Stochastic Processes and their Applications*, 120(7):1133 – 1158, 2010. MR2639741

[83] J. Cvitaniç and I. Karatzas. Hedging contingent claims with constrained profolios. *The Annals of Applied Probaility*, 3(3):652–681, 1993. MR1233619

[84] J. Cvitaniç and I. Karatzas. Backward stochastic differential equations with reflection and Dynkin games. *The Annals of Probability*, 24(4):2024–2056, 1996. MR1415239

[85] J. Cvitanić, I. Karatzas, and H. Soner. Backward stochastic differential equations with constraints on the gains-process. *The Annals of Probability*, 26(4):1522–1551, 1998. MR1675035

[86] J. Cvitanic and J. Zhang. The steepest descent method for forward-backward SDEs. *Electronic Journal of Probability*, 10:1468 – 1495, 2005. MR2191636

[87] B. Dai, Y. Peng, and B. Gong. Parallel option pricing with BSDE method on GPU. In *2010 Ninth International Conference on Grid and Cloud Computing*, pages 191–195, 2010.

[88] P. C. de Raynal and C. G. Trillos. A cubature based algorithm to solve decoupled McKean-Vlasov forward–backward stochastic differential equations. *Stochastic Processes and their Applications*, 125(6):2206–2255, 2015. MR3322862

[89] F. Delarue and S. Menozzi. A forward-backward stochastic algorithm for quasi-linear PDEs. *The Annals of Applied Probability*, 16(1):140–184, 2006. MR2209339

[90] F. Delarue and S. Menozzi. An interpolated stochastic algorithm for quasilinear PDEs. *Mathematics of Computation*, 77:125–158, 2008. MR2353946

[91] L. Delong. *Backward Stochastic Differential Equations with Jumps and Their Actuarial and Financial Applications*. Springer, London, 2013. MR3089193

[92] J. Douglas, J. Ma, and P. Protter. Numerical methods for forward-backward stochastic differential equations. *The Annals of Applied Probability*, 6(3):940–968, 1996. MR1410123

[93] P. D. Drummond. Forward, backward, and weighted stochastic bridges. *Physical Review E*, 96:042123, 2017.

[94] R. Dumitrescu and C. Labart. Numerical approximation of doubly reflected BSDEs with jumps and RCLL obstacles. *Journal of Mathematical Analysis and Applications*, 442(1):206–243, 2016. MR3498326

[95] R. Dumitrescu and C. Labart. Reflected scheme for doubly reflected BSDEs with jumps and RCLL obstacles. *Journal of Computational and Applied Mathematics*, 296:827–839, 2016. MR3430176

[96] T. Dunst and A. Prohl. The forward-backward stochastic heat equation: Numerical analysis and simulation. *SIAM Journal on Scientific Computing*, 38(5):A2725–A2755, 2016. MR3543162

[97] W. E., J. Han, and A. Jentzen. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Communications in Mathematics and Statistics*, 5(4):349–380, 2017. MR3736669

[98] W. E, J. Han, and A. Jentzen. Algorithms for solving high dimensional PDEs: from nonlinear Monte Carlo to machine learning. *Nonlinearity*,

35(1):278–310, 2021. MR4356985

[99] 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 equation. *Journal of Scientific Computing*, 79(3):1534–1571, 2019. MR3946468

[100] W. E, M. Hutzenthaler, A. Jentzen, and T. Kruse. Multilevel Picard iterations for solving smooth semilinear parabolic heat equations. *Partial Differential Equations and Applications*, 2(6):80, 2021. MR4338044

[101] N. El-Karoui and S. Hamadène. BSDEs and risk-sensitive control, zero-sum and nonzero-sum game problems of stochastic functional differential equations. *Stochastic Processes and their Applications*, 107(1):145–169, 2003. MR1995925

[102] N. El-Karoui, S. Hamadène, and A. Matoussi. BSDEs and applications. In R. Carmona, editor, *Indifference Pricing: Theory and Applications*, volume 8, pages 267–320. Princeton University Press, 2008.

[103] N. El-Karoui, C. Kapoudjian, E. Paradoux, S. Peng, and M. Quenez. Reflected solutions of backward SDE's, and related obstacle problems for PDE's. *The Annals of Probaility*, 25(2):702–737, 1997. MR1434123

[104] N. El-Karoui, E. Pardoux, and M. Quenez. Reflected backward SDEs and American options. In L. C. G. Rogers and D. Talay, editors, *Numerical Methods in Finance*, pages 215–231. Cambridge University Press, 1997. MR1470516

[105] N. El-Karoui, S. Peng, and M. Quenez. Backward stochastic differential equations in finance. *Mathematical Finance*, 7(1):1–71, 1997. MR1434407

[106] A. Fahim, N. Touzi, and X. Warin. A probabilistic numerical method for fully nonlinear parabolic PDEs. *The Annals of Applied Probability*, 21(4):1322 – 1364, 2011. MR2857450

[107] M. Fahrenwaldt. Short-time asymptotic expansions of semilinear evolution equations. *Proceedings of the Royal Society of Edinburgh*, 146A(141-167), 2016. MR3457634

[108] Q. Feng, M. Luo, and Z. Zhang. Deep signature FBSDE algorithm. *Numerical Algebra, Control and Optimization*, 2022.

[109] L. Fiorin, G. Pagès, and A. Sagna. Product Markovian quantization of a diffusion process with applications to finance. *Methodology and Computing in Applied Probability*, 21(4):1087–1118, 2019. MR4029233

[110] J.-P. Fouque and Z. Zhang. Deep learning methods for mean field control problems with delay. *Frontiers in Applied Mathematics and Statistics*, 6:11, 2020.

[111] E. Fournié, J.-M. Lasry, J. Lebuchoux, and P.-L. Lions. Applications of Malliavin calculus to Monte Carlo methods in finance. II. *Finance and Stochastics*, 5(2):201–236, 2001. MR1841717

[112] Y. Fu, J. Yang, and W. Zhao. Prediction-correction scheme for decoupled forward backward stochastic differential equations with jumps. *East Asian Journal on Applied Mathematics*, 6(3):253–277, 2016. MR3539338

[113] Y. Fu and W. Zhao. An explicit second-order numerical scheme to solve

decoupled forward backward stochastic equations. *East Asian Journal on Applied Mathematics*, 4(4):368–385, 2014. MR3281570

[114] Y. Fu, W. Zhao, and T. Zhou. Multistep schemes for forward backward stochastic differential equations with jumps. *Journal of Scientific Computing*, 69(2):651–672, 2016. MR3551340

[115] Y. Fu, W. Zhao, and T. Zhou. Efficient spectral sparse grid approximations for solving multi-dimensional forward backward SDEs. *Discrete & Continuous Dynamical Systems – B*, 22(9):3439–3458, 2017. MR3674067

[116] M. Fujii. A polynomial scheme of asymptotic expansion for backward SDEs and option pricing. *Quantitative Finance*, 16(3):427–445, 2016. MR3473963

[117] M. Fujii, S. Sato, and A. Takahashi. An FBSDE approach to American option pricing with an interacting particle method. *Asia-Pacific Financial Markets*, 22(3):239–260, 2015.

[118] M. Fujii and A. Takahashi. Analytical approximation for non-linear FBSDEs with perturbation scheme. *International Journal of Theoretical and Applied Finance*, 15(5):1250034, 2012. MR2964927

[119] M. Fujii and A. Takahashi. Perturbative expansion of FBSDE in an incomplete market with stochastic volatility. *Quarterly Journal of Finance*, 2(3):1250015, 2012.

[120] M. Fujii and A. Takahashi. Perturbative expansion technique for nonlinear FBSDEs with interacting particle method. *Asia-Pacific Financial Market*, 22(3):283–304, 2015.

[121] M. Fujii and A. Takahashi. Asymptotic expansion for forward-backward SDEs with jumps. *Stochastics*, 91(2):175–214, 2019. MR3895863

[122] M. Fujii and A. Takahashi. Solving backward stochastic differential equations with quadratic-growth drivers by connecting the short-term expansions. *Stochastic Processes and their Applications*, 129(5):1492–1532, 2019. MR3944774

[123] M. Fujii, A. Takahashi, and M. Takahashi. Asymptotic expansion as prior knowledge in deep learning method for high dimensional BSDEs. *Asia-Pacific Financial Markets*, 26(3):391–408, 2019.

[124] N. Ganesan, Y. Yu, and B. Hientzsch. Pricing barrier options with deep backward stochastic differential equation methods. *Journal of Computational Finance*, 25(4):1–25, March 2022.

[125] S. Gasparyan and Y. Kutoyants. On approximation of the BSDE with unknown volatility in forward equation. *Armenian Journal of Mathematics*, 7(1):59–79, May 2015. MR3357567

[126] Y. Ge, L. Li, and G. Zhang. A Fourier transform method for solving backward stochastic differential equations. *Methodology and Computing in Applied Probability*, 24(1):385–412, 2022. MR4379495

[127] C. Geiss, S. Geiss, and E. Gobet. Generalized fractional smoothness and $L_p$-variation of BSDEs with non-lipschitz terminal condition. *Stochastic Processes and their Applications*, 122(5):2078–2116, 2012. MR2921973

[128] C. Geiss and C. Labart. Simulation of BSDEs with jumps by Wiener chaos expansion. *Stochastic Processes and their Applications*, 126(7):2123–2162,

2016. MR3483749

[129] C. Geiss, C. Labart, and A. Luoto. Mean square rate of convergence for random walk approximation of forward-backward SDEs. *Advances in Applied Probability*, 52(3):735–771, 2020. MR4153587

[130] C. Geiss, C. Labart, and A. Luoto. Random walk approximation of BSDEs with Hölder continuous terminal condition. *Bernoulli*, 26(1):159–190, 2020. MR4036031

[131] M. Germain, J. Mikael, and X. Warin. Numerical resolution of McKean-Vlasov FBSDEs using neural networks. *Methodology and Computing in Applied Probability*, 24(4):2557–2586, 2022. MR4528393

[132] M. Germain, H. Pham, and X. Warin. Approximation error analysis of some deep backward schemes for nonlinear PDEs. *SIAM Journal on Scientific Computing*, 44(1):A28–A56, 2022. MR4358471

[133] M. Germain, H. Pham, and X. Warin. Neural networks-based algorithms for stochastic control and PDEs in finance. In C.-A. Lehalle and A. Capponi, editors, *Machine Learning And Data Sciences For Financial Markets: A Guide To Contemporary Practices*. Cambridge University Press, 2022.

[134] K. Gnameho, M. Stadje, and A. Pelsser. A regress-later algorithm for backward stochastic differential equations. arXiv:1706.07986, 2017.

[135] A. Gnoatto, M. Patacca, and A. Picarelli. A deep solver for BSDEs with jumps. arXiv:2211.04349, 2022.

[136] A. Gnoatto, A. Picarelli, and C. Reisinger. Deep xVA solver: A neural network-based counterparty credit risk management framework. *SIAM Journal on Financial Mathematics*, 14(1):314–352, 2023. MR4567407

[137] E. Gobet and C. Labart. Error expansion for the discretization of backward stochastic differential equations. *Stochastic Processes and their Applications*, 117(7):803–829, 2007. MR2330720

[138] E. Gobet and C. Labart. Solving BSDE with adaptive control variate. *SIAM Journal on Numerical Analysis*, 48(1):257–277, 2010. MR2608369

[139] E. Gobet and J.-P. Lemor. Numerical simulation of BSDEs using empirical regression methods: theory and practice. Proceedings of the "Fifth Colloquium on BSDEs" (29th May – 1st June 2005, Shanghai) available at arXiv:0806.4447, 2008.

[140] E. Gobet, J.-P. Lemor, and X. Warin. A regression-based Monte Carlo method to solve backward stochastic differential equations. *Annals of Applied Probability*, 15(3):2172–2202, 2005. MR2152657

[141] E. Gobet, J. López-Salas, and C. Vázquez. Quasi-regression Monte-Carlo scheme for semi-linear PDEs and BSDEs with large scale parallelization on GPUs. *Archives of Computational Methods in Engineering*, 27(3):889–921, 2020. MR4103948

[142] E. Gobet, J. G. López-Salas, P. Turkedjiev, and C. Vásquez. Stratified regression Monte-Carlo scheme for semilinear PDEs and BSDEs with large scale parallelization on GPUs. *SIAM Journal on Scientific Computing*, 38(6):652–677, 2015. MR3573315

[143] E. Gobet and S. Maire. Sequential control variates for functionals of Markov processes. *SIAM Journal on Numerical Analysis*, 43(3):1256–1275,

2005. MR2177804

[144] E. Gobet and A. Makhlouf. $\mathbf{L}_2$-time regularity of BSDEs with irregular terminal functions. *Stochastic Processes and their Applications*, 120(7):1105–1132, 2010. MR2639740

[145] E. Gobet and S. Pagliarani. Analytical approximations of BSDEs with nonsmooth driver. *SIAM Journal on Financial Mathematics*, 6(1):919–958, 2015. MR3396392

[146] E. Gobet, J. G. L. Salas, and C. Vázquez. Quasi-regression Monte-Carlo method for semi-linear PDEs and BSDEs. *Proceedings*, 21(1), 2019.

[147] E. Gobet and P. Turkedjiev. Linear regression MDP scheme for discrete backward stochastic differential equations under general conditions. *Mathematics of Computation*, 86(299):1359–1391, 2014. MR3454368

[148] E. Gobet and P. Turkedjiev. Approximation of backward stochastic differential equations using Malliavin weights and least-squares regression. *Bernoulli*, 22(1):530–562, 2016. MR3449792

[149] E. Gobet and P. Turkedjiev. Adaptive importance sampling in least-squares Monte Carlo algorithms for backward stochastic differential equations. *Stochastic Processes and their Applications*, 127(4):1171–1203, 2017. MR3619267

[150] B. Gong, W. Liu, T. Tang, W. Zhao, and T. Zhou. An efficient gradient projection method for stochastic optimal control problems. *SIAM Journal on Numerical Analysis*, 55(6):2982–3005, 2017. MR3730543

[151] B. Gong and H. Rui. One order numerical scheme for forward-backward stochastic differential equations. *Applied Mathematics and Computation*, 271:220 – 231, 2015. MR3414799

[152] B. Gong and W. Zhao. Optimal error estimates for a fully discrete Euler scheme for decoupled forward backward stochastic differential equations. *East Asian Journal on Applied Mathematics*, 7(3):548–565, 2017. MR3708990

[153] L. Gonon, P. Grohs, A. Jentzen, D. Kofler, and D. Šiška. Uniform error estimates for artificial neural network approximations for heat equations. *IMA Journal of Numerical Analysis*, 42(3):1991–2054, 08 2021. MR4454915

[154] P. Grohs, F. Hornung, A. Jentzen, and P. von Wurstemberger. A proof that artificial neural networks overcome the curse of dimensionality in the numerical approximation of Black-Scholes partial differential equations. arXiv:1809.02362, 2018.

[155] G. Guo. Finite difference methods for the BSDEs in finance. *International Journal of Financial Studies*, 6(1):1–15, 2018.

[156] W. Guo, J. Zhang, and J. Zhuo. A monotone scheme for high-dimensional fully nonlinear PDEs. *The Annals of Applied Probability*, 25(3):1540 – 1580, 2015. MR3325281

[157] S. Hamadène. Reflected BSDE's with discontinuous barrier and applications. *Stochastics and Stochastics Reports*, 74(3):571–596, 2002. MR1943580

[158] S. Hamadène and M. Jeanblanc. On the starting and stopping problem: application in reversible investments. *Mathematics of Operations Research*,

32(1):182–192, 2007. MR2292506

[159] S. Hamadène and J. Lepeltier. Reflected BSDEs and mixed game problem. *Stochastic Processes and their Applications*, 85(2):177–188, 2000. MR1731020

[160] Y. Hamaguchi and D. Taguchi. Approximations for adapted M-solutions of Type-II backward stochastic Volterra integral equations. *ESAIM: PS*, 27:19–79, 2023. MR4529444

[161] J. Han, R. Hu, and J. Long. Learning high-dimensional McKean-Vlasov forward-backward stochastic differential equations with general distribution dependence. arXiv:2204.11924, 2022.

[162] J. Han, A. Jentzen, and W. E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, 2018. MR3847747

[163] J. Han and J. Long. Convergence of the deep BSDE method for coupled FBSDEs. *Probability, Uncertainty and Quantitative Risk*, 5(1), Jul 2020. MR4122227

[164] Q. Han. Variable step size Adams methods for BSDEs. *Journal of Mathematics*, 2021:9799627, 2021. MR4354176

[165] Q. Han and S. Ji. A multi-step algorithm for BSDEs based on a predictor-corrector scheme and least-squares Monte Carlo. *Methodology and Computing in Applied Probability*, 24(4):2403–2426, 2022. MR4528387

[166] K. P. Hawkins, A. Pakniyat, and P. Tsiotras. Value function estimators for Feynman-Kac forward-backward SDEs in stochastic optimal control. arXiv:2103.14246, 2021.

[167] P. Henry-Labordère. Cutting CVA's complexity. *Risk*, 25(7):67–73, 2012.

[168] P. Henry-Labordère, C. Litterer, and Z. Ren. A dual algorithm for stochastic control problems: Applications to uncertain volatility models and CVA. *SIAM Journal on Financial Mathematics*, 7(1):159–182, 2016. MR3488163

[169] P. Henry-Labordère, N. Oudjane, X. Tan, N. Touzi, and X. Warin. Branching diffusion representation of semilinear PDEs and Monte Carlo approximation. *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, 55(1):184 – 210, 2019. MR3901645

[170] P. Henry-Labordère, X. Tan, and N. Touzi. A numerical algorithm for a class of BSDEs via the branching process. *Stochastic Processes and their Applications*, 124(2):1112–1140, 2014. MR3138609

[171] B. Hientzsch. Introduction to solving quant finance problems with time-stepped FBSDE and deep learning. arXiv:1911.12231, 2019.

[172] U. Horst, Y. Hu, P. Imkeller, A. Réveillac, and J. Zhang. Forward-backward systems for expected utility maximization. *Stochastic Processes and their Applications*, 124(5):1813–1848, 2014. MR3170226

[173] M. Hu and L. Jiang. Explicit $\theta$-schemes for solving anticipated backward stochastic differential equations. arXiv:1906.01793, 2019.

[174] M. Hu and L. Jiang. Numerical schemes for backward stochastic differential equations driven by *G*-Brownian motion. arXiv:1911.13070, 2019.

[175] M. Hu and L. Jiang. An efficient numerical method for forward-backward stochastic differential equations driven by G-Brownian motion. *Applied Nu-*

*merical Mathematics*, 165:578–597, 2021. MR4236332

[176] Y. Hu, P. Imkeller, and M. Müller. Utility maximization in incomplete markets. *The Annals of Applied Probaility*, 15(3):1691–1712, 2005. MR2152241

[177] Y. Hu, D. Nualart, and X. Song. Malliavin calculus for backward stochastic differential equations and application to numerical solutions. *The Annals of Applied Probability*, 21(6):2379 – 2423, 2011. MR2895419

[178] Y. Hu and S. Tang. Multi-dimensional BSDE with oblique reflection and optimal switching. *Probability Theory and Related Fields*, 147(1):89–121, 2010. MR2594348

[179] T. Huijskens. *A Numerical Method for Backward Stochastic Differential Equations with Applications in Finance*. PhD thesis, Technische Universiteit Delft, 2013.

[180] T. Huijskens, M. Ruijter, and C. Oosterlee. Efficient numerical Fourier methods for coupled forward-backward SDEs. *Journal of Computational and Applied Mathematics*, 296:593 – 612, 2016. MR3430160

[181] C. Huré, H. Pham, and X. Warin. Deep backward schemes for high-dimensional nonlinear PDEs. *Mathematics of Computation*, 89:1547–1579, 2020. MR4081911

[182] M. Hutzenthaler, A. Jentzen, and T. Kruse. Overcoming the curse of dimensionality in the numerical approximation of parabolic partial differential equations with gradient-dependent nonlinearities. *Foundations of Computational Mathematics*, 22(4):905–966, 2022. MR4462404

[183] M. Hutzenthaler, A. Jentzen, T. Kruse, T. Anh Nguyen, and P. von Wurstemberger. Overcoming the curse of dimensionality in the numerical approximation of semilinear parabolic partial differential equations. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 476(2244):20190630, 2020. MR4203091

[184] M. Hutzenthaler, A. Jentzen, T. Kruse, and T. A. Nguyen. A proof that rectified deep neural networks overcome the curse of dimensionality in the numerical approximation of semilinear heat equations. *SN Partial Differential Equations and Applications*, 1(2):10, 2020. MR4292849

[185] M. Hutzenthaler, A. Jentzen, T. Kruse, and T. A. Nguyen. Overcoming the curse of dimensionality in the numerical approximation of backward stochastic differential equations. *Journal of Numerical Mathematics*, 31(1):1–28, 2023. MR4557620

[186] M. Hutzenthaler, A. Jentzen, B. Kuckuck, and J. L. Padgett. Strong $L^p$-error analysis of nonlinear Monte Carlo approximations for high-dimensional semilinear partial differential equations. arXiv:2110.08297, 2021.

[187] M. Hutzenthaler and T. Kruse. Multilevel Picard approximations of high-dimensional semilinear parabolic differential equations with gradient-dependent nonlinearities. *SIAM Journal on Numerical Analysis*, 58(2):929–961, 2020. MR4075337

[188] M. Hutzenthaler, T. Kruse, and T. A. Nguyen. On the speed of convergence of Picard iterations of backward stochastic differential equa-

tions. *Probability, Uncertainty and Quantitative Risk*, 7(2):133–150, 2022. MR4488351

[189] M. Hutzenthaler and T. A. Nguyen. Multilevel Picard approximations for high-dimensional decoupled forward-backward stochastic differential equations. arXiv:2204.08511, 2022.

[190] C. B. Hyndman and P. O. Ngou. A convolution method for numerical solution of backward stochastic differential equations. *Methodology and Computing in Applied Probability*, 19(1):1–29, 2017. MR3611534

[191] P. Imkeller and G. Dos Reis. Path regularity and explicit convergence rate for BSDE with truncated quadratic growth. *Stochastic Processes and their Applications*, 120(3):348–379, 2010. MR2584898

[192] P. Imkeller, G. Dos Reis, and J. Zhang. Results on numerics for FB-SDE with drivers of quadratic growth. In C. Chiarella and A. Novikov, editors, *Contemporary Quantitative Finance: Essays in Honour of Eckhard Platen*, pages 159–182. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. MR2732845

[193] P. Imkeller, A. Réveillac, and J. Zhang. Solvability and numerical simulation of BSDEs related to BSPDEs with applications to utility maximization. *International Journal of Theoretical and Applied Finance*, 14(05):635–667, 2011. MR2826789

[194] K. Jańczak-Borkowska. Discrete approximations of generalized RBSDE with random terminal time. *Discussiones Mathematicae. Probability and Statistics*, 32, 01 2012. MR3039965

[195] A. Jentzen, D. Salimova, and T. Welti. A proof that deep artificial neural networks overcome the curse of dimensionality in the numerical approximation of Kolmogorov partial differential equations with constant diffusion and nonlinear drift coefficients. *Communications in Mathematical Sciences*, 19(5):1167–1205, 2021. MR4283528

[196] S. Ji, S. Peng, Y. Peng, and X. Zhang. Three algorithms for solving high-dimensional fully coupled FBSDEs through deep learning. *IEEE Intelligent Systems*, 35(3):71–84, 2020.

[197] S. Ji, S. Peng, Y. Peng, and X. Zhang. A deep learning method for solving stochastic optimal control problems driven by fully-coupled FBSDEs. arXiv:2204.05796, 2022.

[198] Y. Jiang and J. Li. Convergence of the Deep BSDE method for FBSDEs with non-Lipschitz coefficients. *Probability, Uncertainty and Quantitative Risk*, 6(4):391–408, 2021. MR4399896

[199] L. Kapllani and L. Teng. Deep learning algorithms for solving high dimensional nonlinear backward stochastic differential equations. arXiv:2010.01319, 2021.

[200] L. Kapllani and L. Teng. Multistep schemes for solving backward stochastic differential equations on GPU. *Journal of Mathematics in Industry*, 12(1):5, 2022. MR4373450

[201] L. Kapllani, L. Teng, and M. Ehrhardt. A multistep scheme to solve backward stochastic differential equations for option pricing on GPUs. In I. Dimov and S. Fidanova, editors, *Advances in High Performance Computing*,

pages 196–208, Cham, 2021. Springer International Publishing. MR4373450

[202] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(422-440), 2021.

[203] I. Kharroubi and T. Lim. A decomposition approach for the discrete-time approximation of FBSDEs with a jump. *Random Operators and Stochastic Equations*, 23(2):81–109, 2015. MR3353626

[204] I. Kharroubi, T. Lim, and X. Warin. Discretization and machine learning approximation of BSDEs with a constraint on the Gains-process. *Monte Carlo Methods and Applications*, 27(1):27–55, 2021. MR4221281

[205] A. Khedher and M. Vanmaele. Discretisation of FBSDEs driven by càdlàg martingales. *Journal of Mathematical Analysis and Applications*, 435(1):508–531, 2016. MR3423411

[206] M. Kobylanski. Backward stochastic differential equations and partial differential equations with quadratic growth. *The Annals of Probability*, 28(2):558 – 602, 2000. MR1782267

[207] T. Kong, W. Zhao, and T. Zhou. Probabilistic high order numerical schemes for fully nonlinear parabolic PDEs. *Communications in Computational Physics*, 18(5):1482–1503, 2015. MR3427591

[208] Y. A. Kutoyants. On approximation of BSDE and multi-step MLE-processes. *Probability, Uncertainty and Quantitative Risk*, 1(1):4, 2016. MR3583186

[209] Y. A. Kutoyants and L. Zhou. On approximation of the backward stochastic differential equation. *Journal of Statistical Planning and Inference*, 150:111–123, 2014. MR3206723

[210] C. Labart and J. Lelong. A parallel algorithm for solving BSDEs. *Monte Carlo Methods and Applications*, 19(1):11–39, 2013. MR3039401

[211] M. Laurière. Numerical methods for mean field games and mean field type control. In *Mean Field Games*, pages 221–282. American Mathematical Society, 01 2021. MR4368188

[212] A. Lejay, E. Mordecki, and S. Torres. Numerical approximation of backward stochastic differential equations with jumps. Research Report RR-8595, INRIA, Sept. 2014. MR4178791

[213] 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(5):889–916, 2006. MR2265667

[214] J. P. Lepeltier, S. Hamadène, and A. Matoussi. Double barrier backward SDEs with continuous coefficients. In *Pitman Research Notes in Mathematics Series 364*, pages 115–128. Cambridge University Press, 1997. MR1752681

[215] A. Lesniewski and A. Richter. Managing counterparty credit risk via BSDEs. arXiv:1608.03237, 2016.

[216] Y. Li. A high-order numerical method for BSPDEs with applications to mathematical finance. *SIAM Journal on Financial Mathematics*, 13(1):147–178, 2022. MR4379624

[217] Y. Li, J. Yang, and W. Zhao. Convergence error estimates of the Crank-

Nicolson scheme for solving decoupled FBSDEs. *Science China Mathematics*, 60(5):923–948, 2017. MR3640065

[218] Y. Li and W. Zhao. $L^p$-error estimates for numerical schemes for solving certain kinds of backward stochastic differential equations. *Statistics & Probability Letters*, 80(21):1612–1617, 2010. MR2684007

[219] A. Lionnet, G. dos Reis, and L. Szpruch. Time discretization of FBSDE with polynomial growth drivers and reaction–diffusion PDEs. *The Annals of Applied Probability*, 25(5):2563–2625, 2015. MR3375884

[220] A. Lionnet, G. dos Reis, and L. Szpruch. Convergence and qualitative properties of modified explicit schemes for BSDEs with polynomial growth. *The Annals of Applied Probability*, 28(4):2544–2591, 2018. MR3843836

[221] Y. Liu, Y. Sun, and W. Zhao. A fully discrete explicit multistep scheme for solving coupled forward backward stochastic differential equations. *Advances in Applied Mathematics and Mechanics*, 12:643–663, 2020. MR4087938

[222] Y. Liu, Y. Sun, and W. Zhao. Explicit multistep stochastic characteristic approximation methods for forward backward stochastic differential equations. *Discrete and Continuous Dynamical Systems – S*, 15(4):773–795, 2022. MR4396380

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

[224] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis. DeepXDE: A deep learning library for solving differential equations. *SIAM Review*, 63(1):208–228, 2021. MR4209661

[225] J. Ma, P. Protter, J. Martín, and S. Torres. Numerical method for backward stochastic differential equations. *The Annals of Applied Probaility*, 12(1):302–316, 2002. MR1890066

[226] J. Ma, P. Protter, and J. Yong. Solving forward-backward stochastic differential equations explicitly – a four step scheme. *Probability Theory and Related Fields*, 98(3):339–359, 1994. MR1262970

[227] J. Ma, J. Shen, and Y. Zhao. On numerical approximations of forward-backward stochastic differential equations. *SIAM Journal on Numerical Analysis*, 46(5):2636–2661, 2008. MR2421051

[228] J. Ma and J. Zhang. Representation theorems for backward stochastic differential equations. *The Annals of Applied Probability*, 12(4):1390–1418, 2002. MR1936598

[229] J. Ma and J. Zhang. Representations and regularities for solutions to BSDEs with reflections. *Stochastic Processes and their Applications*, 115(4):539–569, 2005. MR2128629

[230] D. Madan, M. Pistorius, and M. Stadje. Convergence of BS$\Delta$Es driven by random walks to BSDEs: The case of (in)finite activity jumps with general driver. *Stochastic Processes and their Applications*, 126(5):1553–1584, 2016. MR3473105

[231] M. Martínez, J. S. Martín, and S. Torres. Numerical method for reflected backward stochastic differential equations. *Stochastic Analysis and Appli-*

*cations*, 29(6):1008–1032, 2011. MR2847333

[232] T. Massing. Approximation and error analysis of forward-backward SDEs driven by general Lévy processes using shot noise series representations. arXiv:2108.04777, 2021.

[233] H. P. McKean. Application of Brownian motion to the equation of Kolmogorov-Petrovskii-Piskunov. *Communications on Pure and Applied Mathematics*, 28(3):323–331, 1975. MR0400428

[234] T. A. McWalter, R. Rudd, J. Kienitz, and E. Platen. Recursive marginal quantization of higher-order schemes. *Quantitative Finance*, 18(4):693–706, 2018. MR3773172

[235] J. Mémin, S. Peng, and M. Xu. Convergence of solutions of discrete reflected backward SDE's and simulations. *Acta Mathematicae Applicatae Sinica, English Series*, 24(1):1–18, 2008. MR2385005

[236] G. Milstein and M. Tretyakov. Numerical algorithms for forward-backward stochastic differential equations. *SIAM Journal on Scientific Computing*, 28(2):561–582, 2006. MR2231721

[237] G. N. Milstein and M. V. Tretyakov. *Stochastic Numerics for Mathematical Physics*. Springer, Berlin, Heidelberg, 2004. MR2069903

[238] G. N. Milstein and M. V. Tretyakov. Discretization of forward–backward stochastic differential equations and related quasi-linear parabolic equations. *IMA Journal of Numerical Analysis*, 27(1):24–44, 01 2007. MR2289270

[239] T. Moseler. *A Picard-type Iteration for Backward Stochastic Differential Equations: Convergence and Importance Sampling*. PhD thesis, Univserity of Konstanz, 2010.

[240] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. MR3881695

[241] R. Naito and T. Yamada. A second-order discretization for forward-backward SDEs using local approximations with Malliavin calculus. *Monte Carlo Methods and Applications*, 25(4):341–361, 2019. MR4036636

[242] R. Naito and T. Yamada. An acceleration scheme for deep learning-based BSDE solver using weak expansions. *International Journal of Financial Engineering*, 07(02):2050012, 2020. MR4124457

[243] T. Nakayama. Approximation of BSDE's by stochastic difference equation's. *Journal of Mathematical Sciences, The University of Tokyo*, 9:257–277, 2002. MR1904932

[244] B. Negyesi, K. Andersson, and C. W. Oosterlee. The One Step Malliavin scheme: new discretization of BSDEs implemented with deep learning regressions. arXiv:2110.05421, 2021.

[245] J. Y. Nguwi, G. Penent, and N. Privault. A deep branching solver for fully nonlinear partial differential equations. arXiv:2203.03234, 2022.

[246] G. Penent and N. Privault. A fully nonlinear Feynman-Kac formula with derivatives of arbitrary orders. *Journal of Evolution Equations*, 23(1):22, 2023. MR4554042

[247] S. Ninomiya and Y. Shinozaki. Higher-order discretization methods of forward-backward SDEs using KLNV-scheme and their applications to XVA pricing. *Applied Mathematical Finance*, 26(3):257–292, 2019. MR3992509

[248] R. E. Nmeir and G. Pagès. Quantization-based approximation of reflected BSDEs with extended upper bounds for recursive quantization. arXiv:2105.07684, 2021.

[249] N. Nüsken and L. Richter. Interpolating between BSDEs and PINNs – deep learning for elliptic and parabolic boundary value problems. arXiv:2112.03749, 2021.

[250] N. Nüsken and L. Richter. Solving high-dimensional Hamilton–Jacobi–Bellman PDEs using neural networks: perspectives from the theory of controlled diffusions and measures on path space. *Partial Differential Equations and Applications*, 2(4):48, 2021. MR4338293

[251] P. Oyono Ngou and C. Hyndman. A Fourier interpolation method for numerical solution of FBSDEs: Global convergence, stability, and higher order discretizations. *Journal of Risk and Financial Management*, 15(9):388, 2022.

[252] G. Pagès and A. Sagna. Improved error bounds for quantization based numerical schemes for BSDE and nonlinear filtering. *Stochastic Processes and their Applications*, 128(3):847–883, 2018. MR3758340

[253] C.-K. Pak, M.-C. Kim, and C.-H. Rim. An efficient third-order scheme for BSDEs based on nonequidistant difference scheme. *Numerical Algorithms*, 85(2):467–483, 2020. MR4146520

[254] A. Papapantoleon, D. Possamaï, and A. Saplaouras. Stability of backward stochastic differential equations: the general case. arXiv:2107.11048, 2021.

[255] E. Paradoux and S. Peng. Adapted solution of a backward stochastic differential equation. *Systems and Control Letters*, 14:55–61, 1990. MR1037747

[256] E. Pardoux and A. Răşcanu. *Stochastic Differential Equations, Backward SDEs, Partial Differential Equations*. Springer International Publishing, Cham, 2014. MR3308895

[257] A. Pelsser and K. Gnameho. A Monte Carlo method for backward stochastic differential equations with Hermite martingales. *Monte Carlo Methods and Applications*, 25(1):37–60, 2019. MR3918488

[258] G. Penent and N. Privault. Existence of solutions for nonlinear elliptic PDEs with fractional Laplacians on open balls. arXiv:2110.09941, 2021.

[259] G. Penent and N. Privault. Existence and probabilistic representation of the solutions of semilinear parabolic PDEs with fractional Laplacians. *Stochastics and Partial Differential Equations: Analysis and Computations*, 10(2):446–474, 2022. MR4439989

[260] S. Peng. A general stochastic maximum principle for optimal control problems. *SIAM Journal of Control and Optimization*, 28(4):966–979, 1990. MR1051633

[261] S. Peng. Backward stochastic differential equations and applications to optimal control. *Applied Mathematics and Optimization*, 27(2):125–144, 1993.

MR1202528

[262] S. Peng. Backward stochastic differential equation, nonlinear expectation and their applications. *Proceedings of the International Congress of Mathematicians*, 1:393–432, 2011. MR2827899

[263] S. Peng and M. Xu. Numerical algorithms for backward stochastic differential equations with 1-d Brownian motion: Convergence and simulations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 45(2):335–360, 2011. MR2804642

[264] Y. Peng, B. Gong, H. Liu, and B. Dai. Option pricing on the GPU with backward stochastic differential equation. In *2011 Fourth International Symposium on Parallel Architectures, Algorithms and Programming*, pages 19–23, 2011. MR2764297

[265] Y. Peng, B. Gong, H. Lui, and Y. Zhang. Parallel computing for option pricing based on the backward stochastic differential equation. In W. Zhang, Z. Chen, C. C. Douglas, and W. Tong, editors, *High Performance Computing and Applications*, volume 5938, pages 325–330. Springer, Berlin, Heidelberg, 2010.

[266] M. A. Pereira, A. D. Saravanos, O. So, and E. A. Theodorou. Decentralized safe multi-agent stochastic optimal control using deep FBSDEs and ADMM. In *Robotics: Science and Systems (RSS)*, 2022.

[267] V. Piterbarg. Funding beyond discounting: collateral agreements and derivatives pricing. *Risk*, 23(2):97, 2010.

[268] A. Prohl and Y. Wang. Strong rates of convergence for a space-time discretization of the backward stochastic heat equation, and of a linear-quadratic control problem for the stochastic heat equation. *ESAIM: COCV*, 27:54, 2021. MR4268037

[269] M. Raissi. Forward-backward stochastic neural networks: Deep learning of high-dimensional partial differential equations. arXiv:1804.07010, 2018.

[270] C. Reisinger, W. Stockinger, and Y. Zhang. A posteriori error estimates for fully coupled McKean-Vlasov forward-backward SDEs. arXiv:2007.07731, 2020.

[271] C. Reisinger and Y. Zhang. Rectified deep neural networks overcome the curse of dimensionality for nonsmooth value functions in zero-sum games of nonlinear stiff systems. *Analysis and Applications*, 18(06):951–999, 2020. MR4154658

[272] A. Richou. Numerical simulation of BSDEs with drivers of quadratic growth. *The Annals of Applied Probability*, 21(5):1933–1964, 2011. MR2884055

[273] R. Rouge and N. El-Karoui. Pricing via utility maximization and entropy. *Mathematical Finance*, 10(2):259–276, 2000. MR1802922

[274] M. Ruijter and C. Oosterlee. Numerical Fourier method and second-order Taylor scheme for backward SDEs in finance. *Applied Numerical Mathematics*, 103:1–26, 2016. MR3458021

[275] M. J. Ruijter and C. W. Oosterlee. A Fourier cosine method for an efficient computation of solutions to BSDEs. *SIAM Journal on Scientific Computing*, 37(2):A859–A889, 2015. MR3336306

[276] J. Sekine. An approximation for exponential hedging. *Advanced Studies in Pure Mathematics, Stochastic Analysis and Related Topics in Kyoto: In honour of Kiyosi Itô*, 41:279–299, 2004. MR2083715

[277] A. Sghir and S. Hadiri. A new numerical method for 1-D backward stochastic differential equations without using conditional expectations. *Random Operators and Stochastic Equations*, 28(2):79–91, 2020. MR4104365

[278] E. Shamarova, R. Chertovskih, A. F. Ramos, and P. Aguiar. Backward-stochastic-differential-equation approach to modeling of gene expression. *Physical Review E*, 95:032418, 2017. MR3782489

[279] J. Sirignano and K. Spiliopoulos. DGM: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375(15):1339–1364, 2018. MR3874585

[280] A. Skorokhod. Branching diffusion processes. *Theory of Probability and Its Applications*, 9(3):445–449, 1964. MR0168030

[281] D. Sun, G. Liang, and S. Tang. Quantitative stability and numerical analysis of Markovian quadratic BSDEs with reflection. *Probability, Uncertainty and Quantitative Risk*, 7(1):13–30, 2022. MR4405348

[282] Y. Sun and W. Zhao. An explicit second-order numerical scheme for mean-field forward backward stochastic differential equations. *Numerical Algorithms*, 84(1):253–283, 2020. MR4089699

[283] Y. Sun, W. Zhao, and T. Zhou. Explicit $\theta$-schemes for mean-field backward stochastic differential equations. *SIAM Journal on Numerical Analysis*, 56(4):2672–2697, 2018. MR3849120

[284] A. Takahashi, Y. Tsuchida, and T. Yamada. A new efficient approximation scheme for solving high-dimensional semilinear PDEs: Control variate method for Deep BSDE solver. *Journal of Computational Physics*, 454:110956, 2022. MR4371130

[285] A. Takahashi and T. Yamada. An asymptotic expansion of forward-backward SDEs with a perturbed driver. *International Journal of Financial Engineering*, 2(2):1550020, 2015. MR3454652

[286] A. Takahashi and T. Yamada. An asymptotic expansion for forward-backward SDEs: A Malliavin calculus approach. *Asia-Pacific Financial Markets*, 23(4):337–373, 2016.

[287] A. Takahashi and T. Yamda. Asymptotic expansion and deep neural networks overcome the curse of dimensionality in the numerical approximation of Kolmogorov partial differential equations with nonlinear coefficients, 2022. preprint available at https://www.carf.e.u-tokyo.ac.jp/research/f547/.

[288] T. Tang, W. Zhao, and T. Zhou. Deferred correction methods for forward backward stochastic differential equations. *Numerical Mathematics: Theory, Methods and Applications*, 10(2):222–242, 2017. MR3647975

[289] X. Tang and J. Xiong. Stability analysis of general multistep methods for Markovian backward stochastic differential equations. *IMA Journal of Numerical Analysis*, 42(2):1789–1805, 05 2022. MR4410760

[290] B. Teng, Y. Shi, and Q. Zhu. Solving high-dimensional forward-backward doubly SDEs and their related SPDEs through deep learning. *Personal and*

*Ubiquitous Computing*, 26(4):925–932, 2022.

[291] L. Teng. A review of tree-based approaches to solve forward-backward stochastic differential equations. *Journal of Computational Finance*, 25(3), Decemeber 2021. MR4276015

[292] L. Teng. Gradient boosting-based numerical methods for high-dimensional backward stochastic differential equations. *Applied Mathematics and Computation*, 426:127119, 2022. MR4404037

[293] L. Teng, A. Lapitckii, and M. Günther. A multi-step scheme based on cubic spline for solving backward stochastic differential equations. *Applied Numerical Mathematics*, 150:117 – 138, 2020. MR4046444

[294] L. Teng and W. Zhao. High-order combined multi-step scheme for solving forward backward stochastic differential equations. *Journal of Scientific Computing*, 87(3):81, 2021. MR4251693

[295] S. Toldo. Stability of solutions of BSDEs with random terminal time. *ESAIM: Probability and Statistics*, 10:141–163, 2006. MR2218406

[296] M.-B. Tran. A parallel four step domain decomposition scheme for coupled forward-backward stochastic differential equations. *Journal de Mathematiques Pures et Appliquees*, 96(4):337–394, 2011. MR2832640

[297] Y. Tsuchida. Control variate method for deep BSDE solver using weak approximation. *Asia-Pacific Financial Markets*, 2022.

[298] P. Turkedjiev. Two algorithms for the discrete time approximation of Markovian backward stochastic differential equations under local conditions. *Electronic Journal of Probability*, 20:49pp., 2015. MR3347919

[299] H. Wang, H. Chen, A. Sudjianto, R. Liu, and Q. Shen. Deep learning-based BSDE solver for Libor market model with application to Bermudan swaption pricing and hedging. arXiv:1807.06622, 2018.

[300] J. Wang and R. Korn. Numerical algorithms for reflected anticipated backward stochastic differential equations with two obstacles and default risk. *Risks*, 8(3), 2020.

[301] J.-l. Wang, C.-x. Luo, and W.-D. Zhao. Crank-Nicolson scheme and its error estimates for backward stochastic differential equations. *Acta Mathematicae Applicatae Sinica, English Series*, 2009.

[302] X. Wang, W. Zhao, and T. Zhou. Sinc-$\theta$ schemes for backward stochastic differential equations. *SIAM Journal on Numerical Analysis*, 60(4):1799–1823, 2022. MR4456705

[303] Y. Wang. A numerical scheme for BSVIEs. arXiv:1605.04865, 2016.

[304] Y. Wang and Y.-H. Ni. Deep BSDE-ML learning and its application to model-free optimal control. arXiv:2201.01318, 2022.

[305] X. Warin. Nesting Monte Carlo for high-dimensional non-linear PDEs. *Monte Carlo Methods and Applications*, 24(4):225–247, 2018. MR3891748

[306] S. Watanabe. On the branching process for Brownian particles with an absorbing boundary. *Journal of Mathematics of Kyoto University*, 4(2):385–398, 1965. MR0178505

[307] P. Whittle. *Risk-Sensitive Optimal Control*. Wiley-Interscience Series in Systems and Optimization (Book 2). Wiley, 1st edition, 1990. MR1093001

[308] M. Xu. Numerical algorithms and simulations for reflected backward

stochastic differential equations with two continuous barriers. *Journal of Computational and Applied Mathematics*, 236(6):1137–1154, 2011. MR2854041

[309] J. Yang and W. Zhao. Convergence of recent multistep schemes for a forward-backward stochastic differential equation. *East Asian Journal on Applied Mathematics*, 5(4):387–404, 2015. MR3421810

[310] J. Yang, W. Zhao, and T. Zhou. Explicit deferred correction methods for second-order forward backward stochastic differential equations. *Journal of Scientific Computing*, 79(3):1409–1432, 2019. MR3946463

[311] J. Yang, W. Zhao, and T. Zhou. A unified probabilistic discretization scheme for FBSDEs: Stability, consistency, and convergence analysis. *SIAM Journal on Numerical Analysis*, 58(4):2351–2375, 2020. MR4135389

[312] B. Yu, X. Xing, and A. Sudjianto. Deep-learning based numerical BSDE method for barrier options. arXiv:1904.05921, 2019.

[313] Y. Yu, N. Ganesan, and B. Hientzsch. Backward deep BSDE methods and applications to nonlinear problems. *Risks*, 11(3), 2023.

[314] S. Yuan and R. Kawai. Numerical aspects of shot noise representation of infinitely divisible laws and related processes. *Probability Surveys*, 18:201–271, 2021. MR4298927

[315] G. Zhang, M. Gunzburger, and W. Zhao. A sparse-grid method for multidimensional backward stochastic differential equations. *Journal of Computational Mathematics*, 31(3):221–248, 2013. MR3063734

[316] J. Zhang. *Some Fine Properties of Backward Stochastic Differential Equations*. PhD thesis, Purdue University, 2001. MR2703162

[317] J. Zhang. A numerical scheme for BSDEs. *The Annals of Applied Probability*, 14(1):459–488, 2004. MR2023027

[318] J. Zhang. *Backward Stochastic Differential Equations: From Linear to Fully Nonlinear Theory*. Springer, New York, NY, 2017. MR3699487

[319] W. Zhang and L. Jiang. Solutions of BSDEs with a kind of non-Lipschitz coefficients driven by G-Brownian motion. *Statistics & Probability Letters*, 171:109024, 2021. MR4198706

[320] W. Zhang and H. Min. Weak convergence analysis and improved error estimates for decoupled forward-backward stochastic differential equations. *Mathematics*, 9(8):848, 2021.

[321] W. Zhang and H. Min. $L^p$-error estimates for numerical schemes for solving certain kinds of mean-field backward stochastic differential equations. *Journal of Theoretical Probability*, 2022.

[322] Y. Zhang and W. Zheng. Discretizing a backward stochastic differential equation. *International Journal of Mathematics and Mathematical Sciences*, 32:357319, 2002. MR1937828

[323] W. Zhao, L. Chen, and S. Peng. A new kind of accurate numerical method for backward stochastic differential equations. *SIAM Journal on Scientific Computing*, 28(4):1563–1581, 2006. MR2255470

[324] W. Zhao, Y. Fu, and T. Zhou. New kinds of high-order multistep schemes for coupled forward backward stochastic differential equations. *SIAM Journal on Scientific Computing*, 36(4):A1731–A1751, 2014. MR3245875

[325] W. Zhao, Y. Li, and Y. Fu. Second-order schemes for solving decoupled forward backward stochastic differential equations. *Science China Mathematics*, 57(4):665–686, 2014. MR3178016

[326] W. Zhao, Y. Li, and L. Ju. Error estimates of the Crank-Nicolson scheme for solving backward stochastic differential equations. *International Journal of Numerical Analysis and Modeling*, 10(4):876–898, 2013. MR3125863

[327] W. Zhao, Y. Li, and G. Zhang. A generalized $\theta$-scheme for solving backward stochastic differential equations. *Discrete & Continuous Dynamical Systems – B*, 17(5):1585–1603, 2012. MR2903768

[328] W. Zhao, J. Wang, and S. Peng. Error estimates of the $\theta$-scheme for backward stochastic differential equations. *Discrete and Continuous Dynamical Systems – Series B*, 12(4):905–924, 2008. MR2552079

[329] W. Zhao, G. Zhang, and L. Ju. A stable multistep scheme for solving backward stochastic differential equations. *SIAM Journal on Numerical Analysis*, 48(4):1369–1394, 2010. MR2684339

[330] W. Zhao, W. Zhang, and L. Ju. A numerical method and its error estimates for the decoupled forward-backward stochastic differential equations. *Communications in Computational Physics*, 15(3):618–646, 2014. MR3126886

[331] W. Zhao, W. Zhang, and L. Ju. A multistep scheme for decoupled forward-backward stochastic differential equations. *Numerical Mathematics: Theory, Methods and Applications*, 9(2):262–288, 2016. MR3503961

[332] W. Zhao, T. Zhou, and T. Kong. High order numerical schemes for second-order FBSDEs with applications to stochastic optimal control. *Communications in Computational Physics*, 21(3):808–834, 2017. MR3639253

[333] X. Zhong and S. Liao. On the homotopy analysis method for backward/forward-backward stochastic differential equations. *Numerical Algorithms*, 76(2):487–519, 2017. MR3704879