Three algorithms for solving high-dimensional fully-coupled FBSDEs through deep learning

Shaolin Ji[1], Shige Peng[2], Ying Peng[1], and Xichuan Zhang[2]

[1]Zhongtai Securities Institute for Financial Studies, Shandong University, 250100, China
[2]School of Mathematics, Shandong University, 250100, China

July 12, 2019

Abstract

Recently, the deep learning method has been used for solving forward backward stochastic differential equations (FBSDEs) and parabolic partial differential equations (PDEs). It has good accuracy and performance for high-dimensional problems. In this paper, we mainly solve fully coupled FBSDEs through deep learning and provide three algorithms. Several numerical results show remarkable performance especially for high-dimensional cases.

Keywords deep learning · fully-coupled FBSDEs · high-dimensional equation · stochastic control

1 Introduction

Bismut firstly introduced the linear style backward stochastic differential equation (BSDE) in 1973 [1]. In 1990, Pardoux and Peng proved the existence and uniqueness of the adapted solution for nonlinear BSDEs [2]. In 1997, E.Karoui, Peng and Quenez [3] found important applications of BSDEs in finance. When a BSDE is coupled with a (forward) stochastic differential equation (SDE), the system is usually called a forward backward stochastic differential equation (FBSDE). In recent years, the FBSDEs have shown important applications in many fields. For example, the FBSDEs could be used to model financial markets when a large investor influences the stock price [4]. The solution of a FBSDE is related to a second-order quasilinear partial differential equation (PDE) [5].

Generally speaking, it is difficult for us to obtain the explicit solution of a FBSDE. Therefore, it is necessary to find the approximate solution. In this paper, we aim to obtain the numerical solution of the following fully-coupled FBSDE through deep learning:

$$\begin{cases} X_t = X_0 + \int_0^t b(s, X_s, Y_s, Z_s)\mathrm{d}s + \int_0^t \sigma(s, X_s, Y_s, Z_s)\mathrm{d}W_s, \\ Y_t = g(X_T) + \int_t^T f(s, X_s, Y_s, Z_s)\,\mathrm{d}s - \int_t^T Z_s \mathrm{d}W_s. \end{cases} \tag{1.1}$$

There are several ways to find the numerical solution of FBSDE (1.1). Based on the relationship between FBSDEs and PDEs (see [5]), numerical methods for solving the PDEs, such as the finite element method, the finite difference method, or the sparse grid

method [6], can be applied to solve the FBSDEs. In [7, 8], Yong and Ma studied the solvability of coupled FBSDEs and proposed a four-step approach. Moreover, some probabilistic methods, which approximate the conditional expectation with numerical schemes, were developped to solve the FBSDEs. For example, [9] proposed a theta-scheme numerical method with high accuracy for coupled Markovian FBSDEs. [10] proposed a numerical scheme for coupled FBSDEs when the forward process X_t does not depend on Z_t. The BCOS method [11] and Fourier methods [12] are also proposed for solving the FBSDEs.

As is known, there is a significant difficulty for solving high dimensional BSDEs and FBSDEs, namely "curse of dimensionality" [13]. The computational complexity grows exponentially when the dimension increases, while the accuracy decline sharply. Therefore most of the aforementioned numerical methods can not deal with high-dimensional problems.

Recently, deep-learning method has achieved great success in many application areas [14], such as computer vision [15], natural language processing [16], gaming [17], etc. It provides a new point of view to approximate functions and shows optimistic performance in solving problems with high- dimension features. This poses a possible way to solve the "curse of dimensionality" although the reason why deep-learning has so remarkable performance has not been proven completely.

Weinan E and his collaborators [18, 19] constructed a neural network to approximate the conditional expectation. This method has shown superior performance and accuracy in solving high dimensional BSDEs on comparing with the traditional numerical methods. Han and Long [20] extended this method to solve the following coupled FBSDE, where the forward SDE does not depend on Z_t:

$$\begin{cases} X_t = X_0 + \int_0^t b(s, X_s, Y_s)\mathrm{d}s + \int_0^t \sigma(s, X_s, Y_s)\mathrm{d}W_s, \\ Y_t = g(X_T) + \int_t^T f(s, X_s, Y_s, Z_s)\,\mathrm{d}s - \int_t^T Z_s \mathrm{d}W_s. \end{cases}$$

They regard Z_t as a control and assume that

$$Z_t = \phi(X_t, Y_t),$$

where the function ϕ is simulated by neural network.

In this paper, we propose three algorithms to solve the fully-coupled FBSDEs (1.1) through deep learning. The first algorithm (Algorithm I) is inspired by the idea of the Picard iteration (see [5]). The term Z_t is regarded as the control and in doing iterations, we assume that Z_t depends on X_t, Y_t and Z_t according to their pathes. In more details, we set

$$\tilde{Z}_t^{k+1} = \phi(\tilde{X}_t^{k+1}, \tilde{Y}_t^k, \tilde{Z}_t^k),$$

where k is denoted as the iteration step. It should be noted that this iterative approach is path-to-path. The second algorithm (Algorithm II) is motivated by Han and Long [20]. We also regard Z_t as the control and suppose that Z_t depends on the state X_t of the forward SDE and the state Y_t of the BSDE, i.e.

$$\tilde{Z}_t^{k+1} = \phi(\tilde{X}_t^{k+1}, \tilde{Y}_t^{k+1}),$$

the above iterative relationship is adopted in [20] and we extend it to solve our fully-coupled FBSDE (1.1) in which b and σ depend on Z_t. In the third algorithm (Algorithm III), in

addition to the control Z_t, we regard Y_t in the forward SDE as a new control and denote it by u_t. Both u_t and Z_t are supposed to be dependent on the state X_t of the forward SDE:

$$u_t^{k+1} = \phi^1(\tilde{X}_t^{k+1}),$$
$$\tilde{Z}_t^{k+1} = \phi^2(\tilde{X}_t^{k+1}).$$

The price of doing this is that we need to add a penalty term in the cost function to punish the difference between the control u_t and the solution Y_t of the backward SDE.

As a result, all the three algorithms can approximate the solution of the FBSDE (1.1) and perform well in high-dimensional cases. As shown in the examples, the relative errors of these algorithms are less than 1%. Algorithm I takes only a few steps to achieve convergence results. But the iteration may take more time. Although Algorithms II and III are computationally fast, but they may require more steps to converge.

The remainder of this paper is organized as following. In Section 2, we firstly introduce the preliminaries on FBSDEs and give the existence and uniqueness conditions of fully-coupled FBSDEs. In Section 3, the relationship between FBSDEs and an optimal control problem are presented, which indicates that the FBSDEs can be solved from a control perspective. The theoretical proof is also given. According to different kinds of state feedback, we propose another two optimal control problems for solving FBSDE (1.1). In Section 4, we present our numerical schemes and the corresponding iterative algorithms. Section 5 gives some examples and shows the comparison among different algorithms for solving coupled FBSDEs.

2 Preliminaries on FBSDEs

In this section, we mainly introduce the form of FBSDEs and the existence and uniqueness conditions of fully-coupled FBSDEs [21].

Let $T > 0$ and $(\Omega, \mathcal{F}, \mathbb{P}, \mathbb{F})$ be a filtered probability space, where $W : [0, T] \times \Omega \to \mathbb{R}$ is a d-dimensional standard Brownian motion on $(\Omega, \mathcal{F}, \mathbb{P})$, $\mathbb{F} = \{\mathcal{F}_t\}_{0 \le t \le T}$ is the natural filtration generated by the Brownian motion W. $X_0 \in \mathcal{F}_0$ is the initial condition for the FBSDE.

Considering the following coupled FBSDE,

$$\begin{cases} X_t = X_0 + \displaystyle\int_0^t b(s, X_s, Y_s, Z_s)\mathrm{d}s + \int_0^t \sigma(s, X_s, Y_s, Z_s)\mathrm{d}W_s, \\ Y_t = g(X_T) + \displaystyle\int_t^T f(s, X_s, Y_s, Z_s)\,\mathrm{d}s - \int_t^T Z_s \mathrm{d}W_s, \end{cases} \tag{2.1}$$

where $\{X_t\}_{0 \le t \le T}, \{Y_t\}_{0 \le t \le T}, \{Z_t\}_{0 \le t \le T}$ are $\mathbb{F}$-adapted stochastic processes taking value in $\mathbb{R}^n, \mathbb{R}^m, \mathbb{R}^{m \times d}$ respectively. The functions

$$\begin{aligned} b &: \Omega \times [0, T] \times \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{m \times d} \to \mathbb{R}^n \\ \sigma &: \Omega \times [0, T] \times \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{m \times d} \to \mathbb{R}^{n \times d} \\ f &: \Omega \times [0, T] \times \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{m \times d} \to \mathbb{R}^m \\ g &: \Omega \times [0, T] \times \mathbb{R}^n \to \mathbb{R}^m \end{aligned}$$

are deterministic globally continuous functions. b and σ are the *drift* coefficient and *diffusion* coefficient of X respectively, and f is referred to as the *generator* of the coupled FBSDE. If there is a triple (X_t, Y_t, Z_t) satisfies the above FBSDE on $[0, T]$, $\mathbb{P}$-almost surely, square integrable and $\mathcal{F}_t$-adapted, the triple (X_t, Y_t, Z_t) are called the solutions of

FBSDE(2.1). When functions b and σ are independent of both Y and Z, FBSDE(2.1) is called a decoupled FBSDE.

Given a $m \times n$ full-rank matrix G, We define

$$u = \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \ A(t,u) = \begin{pmatrix} -G^T f \\ Gb \\ G\sigma \end{pmatrix}(t,u),$$

where $G\sigma = (G\sigma_1 \cdots G\sigma_d)$.

Firstly, we give two assumptions as the following,

Assumption 1. *(i) $A(t,u)$ is uniformly Lipschitz with respect to u;*

(ii) $A(\cdot,u)$ is in $M^2(0,T)$ for $\forall u$;

(iii) $g(x)$ is uniformly Lipschitz with respect to $x \in \mathbb{R}^n$;

(iv) $g(x)$ is in $L^2(\Omega, \mathcal{F}_T, \mathbb{P})$ for $\forall x$.

Assumption 2.

$$\langle A(t,u) - A(t,\bar{u}), u - \bar{u} \rangle \leq -\beta_1 |G\hat{x}|^2 - \beta_2(|G^T\hat{y}|^2 + |G^T\hat{z}|),$$
$$\langle g(x) - g(\bar{x}), G(x - \bar{x}) \rangle \geq \mu_1 |G\hat{x}|^2,$$

$$\forall u = (x,y,z), \bar{u} = (\bar{x},\bar{y},\bar{z}), \hat{x} = x - \bar{x}, \hat{y} = y - \bar{y}, \hat{z} = z - \bar{z},$$

where β_1, β_2 and μ_1 are given nonnegative constants with $\beta_1 + \beta_2 > 0, \mu_1 + \beta_2 > 0$.

Then, the following theorem is given:

Theorem 1. *Let Assumptions 1 and 2 hold, then there exists a unique adapted solution (X,Y,Z) of FBSDE (2.1).*

We omit the proof, see Theorem 2.6 of [21] in detail.

Remark. Assumptions 1 and 2 are necessary but not sufficient conditions for FBSDE (2.1). Many FBSDEs not satisfying the conditions A.1 and A.2 have also solutions, such as

$$\begin{cases} X_t = X_0 + \int_0^t Z_s \mathrm{d}W_s, \\ Y_t = X_T - \int_t^T Z_s \mathrm{d}W_s. \end{cases}$$

where we set $m = n$.

For convenience, in this article, we assume that L is the Lipschitz constant satisfying Assumption 1 , that is, for $\forall x, x', y, y', z, z'$

$$|l(t,x,y,z) - l(t,x',y',z')| \leq L(|x - x'| + |y - y'| + |z - z'|),$$
$$|g(x) - g(x')| \leq L(|x - x'|),$$

where l represents one of the functions among f, b and σ.

3 Solving FBSDEs from an optimal control perspective

Essentially, the deep neural network can be regarded as a control system, which is used to approximate the mapping from the input set to the label set. The parameters in the network can be seen as the control, and the cost function can be seen as the optimization objective. Thus, we first transform the FBSDE solving problem into an optimal control problem in order to apply the deep neural network.

3.1 Pichard iteration method

Before the transformation, we first give an existing conclusion. As we know, FBSDE (2.1) has a unique solution (X_t, Y_t, Z_t) under monotonicity conditions, and Pardoux and Tang's results [5] have shown that (2.1) can be constructed via Picard iteration

$$\begin{cases} X_t^{k+1} = X_0 + \int_0^t b(s, X_s^{k+1}, Y_s^k, Z_s^k)\mathrm{d}s + \int_0^t \sigma(s, X_s^{k+1}, Y_s^k, Z_s^k)\mathrm{d}W_s, \\ Y_t^{k+1} = g(X_T^{k+1}) + \int_t^T f(s, X_s^{k+1}, Y_s^{k+1}, Z_s^{k+1})\mathrm{d}s - \int_t^T Z_s^{k+1}\mathrm{d}W_s, \end{cases} \tag{3.1}$$

when Y^0, Z^0 are given, k is denoted as the iteration step. Therefore, the solution $(X_t^{k+1}, Y_t^{k+1}, Z_t^{k+1})$ of the decoupled FBSDE (3.1) converges to to the solution (X_t, Y_t, Z_t) of (2.1) when k tends to infinity, i.e.

$$\lim_{k\to\infty} \mathbb{E}\left[\sup_{0\le t\le T}(|X_t^{k+1} - X_t|^2 + |Y_t^{k+1} - Y_t|^2) + \int_0^T |Z_t^{k+1} - Z_t|^2\mathrm{d}t\right] = 0. \tag{3.2}$$

Regarding $\tilde{Y}_0^{k+1}$ and $\{\tilde{Z}_t^{k+1}\}_{0\le t\le T}$ as controls, we consider the following control problem

$$\inf_{\tilde{Y}_0^{k+1}, \{\tilde{Z}_t^{k+1}\}_{0\le t\le T}} \mathbb{E}[|g(\tilde{X}_T^{k+1}) - \tilde{Y}_T^{k+1}|^2], \tag{3.3}$$

$$s.t.\ \ \tilde{X}_t^{k+1} = X_0 + \int_0^t b(s, \tilde{X}_s^{k+1}, \tilde{Y}_s^k, \tilde{Z}_s^k)\mathrm{d}s + \int_0^t \sigma(s, \tilde{X}_s^{k+1}, \tilde{Y}_s^k, \tilde{Z}_s^k)\mathrm{d}W_s,$$

$$\tilde{Y}_t^{k+1} = \tilde{Y}_0^{k+1} - \int_0^t f(s, \tilde{X}_s^{k+1}, \tilde{Y}_s^{k+1}, \tilde{Z}_s^{k+1})\mathrm{d}s + \int_0^t \tilde{Z}_s^{k+1}\mathrm{d}W_s.$$

In the following, we will show that control problem (3.3) is equivalent to FBSDE (3.1).

When $\tilde{Y}^k, \tilde{Z}^k$ are known, we consider the following SDE

$$\begin{cases} \tilde{X}_t^{k+1} = X_0 + \int_0^t b(s, \tilde{X}_s^{k+1}, \tilde{Y}_s^k, \tilde{Z}_s^k)\mathrm{d}s + \int_0^t \sigma(s, \tilde{X}_s^{k+1}, \tilde{Y}_s^k, \tilde{Z}_s^k)\mathrm{d}W_s, \\ \tilde{Y}_t^{k+1} = \tilde{Y}_0^{k+1} - \int_0^t f(s, \tilde{X}_s^{k+1}, \tilde{Y}_s^{k+1}, \tilde{Z}_s^{k+1})\mathrm{d}s + \int_0^t \tilde{Z}_s^{k+1}\mathrm{d}W_s, \end{cases} \tag{3.4}$$

(3.4) has infinite number of solutions because both the initial value $\tilde{Y}_0^{k+1}$ and the process $\{\tilde{Z}_t^{k+1}\}_{0\le t\le T}$ are uncertain.

Given $\tilde{Y}_0^{k+1}$ and the process $\{\tilde{Z}_t^{k+1}\}_{0\le t\le T}$, assuming $\tilde{U}_t = (\tilde{X}_t^{k+1}, \tilde{Y}_t^{k+1})$, which takes value in $\mathbb{R}^{n+m}$ and depends on the initial condition $\tilde{Y}_0^{k+1}$ and process $\{\tilde{Z}_t^{k+1}\}_{0\le t\le T}$, then equation (3.4) can be written as

$$\tilde{U}_t = \tilde{U}_0 + \int_0^t \tilde{b}(s, \tilde{U}_s)\mathrm{d}s + \int_0^t \tilde{\sigma}(s, \tilde{U}_s)\mathrm{d}W_s, \tag{3.5}$$

where

$$\tilde{b}:\Omega\times[0,T]\times\mathbb{R}^{n+m}\to\mathbb{R}^{n+m}$$
$$\tilde{\sigma}:\Omega\times[0,T]\times\mathbb{R}^{n+m}\to\mathbb{R}^{(n+m)\times d}$$

are two functions denoted as

$$\tilde{b}(s,\tilde{U}_s)=(b(s,\tilde{X}_s^{k+1},\tilde{Y}_s^{k},\tilde{Z}_s^{k}),-f(s,\tilde{X}_s^{k+1},\tilde{Y}_s^{k+1},\tilde{Z}_s^{k+1}))$$
$$\tilde{\sigma}(s,\tilde{U}_s)=(\sigma(s,\tilde{X}_s^{k+1},\tilde{Y}_s^{k},\tilde{Z}_s^{k}),\tilde{Z}_s^{k+1}).$$

Lemma 1. *Assume that Assumptions 1 and 2 hold. Then equation (3.5) has a unique solution $\tilde{U}_t$ when $\tilde{Y}_0^{k+1}$ and process $\{\tilde{Z}_t^{k+1}\}_{0\le t\le T}$ are given, and equation (3.4) has a unique solution $(\tilde{X}_t^{k+1},\tilde{Y}_t^{k+1})$.*

Proof. According to Assumption 1 *(i)* and as the matrix G is full-rank, we can get that functions b,σ and f are all uniformly Lipschitz, satisfying

$$\begin{aligned}|\Delta b(t)|+|\Delta\sigma(t)|+|\Delta f(t)|&\le 3L(|x_1-x_2|+|y_1-y_2|),\\ \Delta b(t)&=b(t,x_1,y_1,\cdot)-b(t,x_2,y_2,\cdot)\\ \Delta\sigma(t)&=\sigma(t,x_1,y_1,\cdot)-\sigma(t,x_2,y_2,\cdot)\\ \Delta f(t)&=f(t,x_1,y_1,\cdot)-f(t,x_2,y_2,\cdot)\\ \forall\quad x_i\in\mathbb{R}^n,y_i&\in\mathbb{R}^m,t\in[0,T],i=1,2\end{aligned}$$

where L is the Lipschitz constant. Then we get the following inequation:

$$\begin{aligned}|\tilde{b}(t,u_1)-\tilde{b}(t,u_2)|+|\tilde{\sigma}(t,u_1)-\tilde{\sigma}(t,u_2)|&\le L'(|u_1-u_2|)\\ \forall\ u_i=(x_i,y_i)\in\mathbb{R}^{n+m},\ x_i\in\mathbb{R}^n,y_i&\in\mathbb{R}^m,t\in[0,T],i=1,2\end{aligned}$$

where L' only depends on L. Let $u=0$, then we get

$$\sup_t(|\tilde{b}(t,0)|+|\tilde{\sigma}(t,0)|)\le D<\infty$$

According to Assumption 1 *(ii)*, there exist a constant D, satisfying

$$\begin{aligned}|\tilde{b}(t,u)|+|\tilde{\sigma}(t,u)|&\le|\tilde{b}(t,u)-\tilde{b}(t,0)|+|\tilde{\sigma}(t,u)-\tilde{\sigma}(t,0)|+|\tilde{b}(t,0)|+|\tilde{\sigma}(t,0)|\\ &\le L'(|u|)+D\le C(1+|u|),\end{aligned}$$

where $C=\max\{L',D\}$. According to [22], the proof is completed. □

Lemma 1 shows that the solution of (3.4) is determined by $\tilde{Y}_0^{k+1}$ and $\{\tilde{Z}_t^{k+1}\}_{0\le t\le T}$. However, because of the uncertainty of $\tilde{Y}_0^{k+1}$ and $\{\tilde{Z}_t^{k+1}\}_{0\le t\le T}$, equation (3.4) has infinite number of solutions.

Now we denote $J(\tilde{Y}_0^{k+1},\tilde{Z}^{k+1})$ as

$$J(\tilde{Y}_0^{k+1},\tilde{Z}^{k+1})=\mathbb{E}[|g(\tilde{X}_T^{k+1})-\tilde{Y}_T^{k+1}|^2],$$

and then the control problem (3.3) becomes the problem of finding the minimum value of $J(\tilde{Y}_0^{k+1},\tilde{Z}^{k+1})$. In the following Theorem 2, we will show that the solution $(\tilde{X}_t^{k+1},\tilde{Y}_t^{k+1},\tilde{Z}_t^{k+1})$ of (3.4) converges to the solution (X_t,Y_t,Z_t) of FBSDE (2.1), when $J(\tilde{Y}_0^{k+1},\tilde{Z}^{k+1})$ goes to zero as k tends to infinity, which means that solving the control problem (3.3) is equivalent to solving the FBSDE (2.1).

Theorem 2. *Suppose Assumption 1 and 2 hold true and there exit* $C_5 \leq 1$*, where*

$$\begin{aligned}
C_1 &= e^{(4L+3L^2)T} \cdot (L + 3L^2) \\
C_2 &= C_1(L + L^2)e^{(2L+2L^2)T} \\
C_3 &= C_2 T \\
C_4 &= (3L^2 C_1 + 9L^2(C_1 + 2C_3) + 3C_3) \\
C_5 &= (C_4 + C_3)(T + 1).
\end{aligned}$$

If $J(\tilde{Y}_0^{k+1}, \tilde{Z}^{k+1})$ *satisfies*

$$\lim_{k\to\infty} J(\tilde{Y}_0^{k+1}, \tilde{Z}^{k+1}) = 0,$$

then the solution of SDE (3.4) $(\tilde{X}_t^{k+1}, \tilde{Y}_t^{k+1}, \tilde{Z}_t^{k+1})$ *satisfies*

$$\lim_{k\to\infty} \mathbb{E}[\sup_{0\leq t\leq T} |\tilde{Y}_t^{k+1} - Y_t|^2 + \int_0^T |\tilde{Z}_t^{k+1} - Z_t|^2 \mathrm{d}t] = 0 \tag{3.6}$$

where (X_t, Y_t, Z_t) *is the solution of FBSDE* (2.1).

Proof. The proof of this theorem is divided into two steps.

Step 1: Supposing that the following equation

$$\begin{cases}
\hat{X}_t^{k+1} = X_0 + \int_0^t b(s, \hat{X}_s^{k+1}, \tilde{Y}_s^k, \tilde{Z}_s^k)\mathrm{d}s + \int_0^t \sigma(s, \hat{X}_s^{k+1}, \hat{Y}_s^k, \hat{Z}_s^k)\mathrm{d}W_s, \\
\hat{Y}_t^{k+1} = g(\hat{X}_T^{k+1}) + \int_t^T f(s, \hat{X}_s^{k+1}, \hat{Y}_s^{k+1}, \hat{Z}_s^{k+1})\mathrm{d}s - \int_t^T \hat{Z}_s^{k+1}\mathrm{d}W_s,
\end{cases} \tag{3.7}$$

has a solution $(\hat{X}_t^{k+1}, \hat{Y}_t^{k+1}, \hat{Z}_t^{k+1})$. Let

$$\begin{aligned}
\delta X_t^{k+1} &= \hat{X}_t^{k+1} - X_t^{k+1}, \\
\delta Y_t^{k+1} &= \hat{Y}_t^{k+1} - Y_t^{k+1}, \\
\delta Z_t^{k+1} &= \hat{Z}_t^{k+1} - Z_t^{k+1}, \\
\delta Y_T^{k+1} &= g(\hat{X}_T^{k+1}) - g(X_T^{k+1}) \\
\delta b_t &= b(t, \hat{X}_t^{k+1}, \tilde{Y}_t^k, \tilde{Z}_t^k) - b(t, X_t^{k+1}, Y_t^k, Z_t^k) \\
\delta \sigma_t &= \sigma(t, \hat{X}_t^{k+1}, \tilde{Y}_t^k, \tilde{Z}_t^k) - \sigma(t, X_t^{k+1}, Y_t^k, Z_t^k) \\
\delta f_t &= f(t, \hat{X}_t^{k+1}, \hat{Y}_t^{k+1}, \hat{Z}_t^{k+1}) - f(t, X_t^{k+1}, Y_t^{k+1}, Z_t^{k+1}).
\end{aligned}$$

From (3.1) and (3.7), we get

$$\begin{aligned}
\delta X_t^{k+1} &= \int_0^t \delta b_s \mathrm{d}s - \int_0^t \delta \sigma_s \mathrm{d}W_s, \\
\delta Y_t^{k+1} &= \delta Y_T^{k+1} + \int_t^T \delta f_s \mathrm{d}s - \int_t^T \delta Z_s^{k+1}\mathrm{d}W_s,
\end{aligned}$$

whose differential form is

$$\begin{aligned}
\mathrm{d}\delta X_t^{k+1} &= \delta b_t \mathrm{d}t + \delta \sigma_t^{k+1}\mathrm{d}W_t, \\
-\mathrm{d}\delta Y_t^{k+1} &= \delta f_t \mathrm{d}t - \delta Z_t^{k+1}\mathrm{d}W_t,
\end{aligned}$$

plugging Ito's formula into $|\delta X_t^{k+1}|^2$,

$$\begin{aligned}\mathrm{d}|\delta X_t^{k+1}|^2 &= 2\delta X_t^{k+1}\cdot \mathrm{d}\delta X_t^{k+1} + \mathrm{d}\delta X_t^{k+1}\cdot \mathrm{d}\delta X_t^{k+1}\\ &= 2\delta X_t^{k+1}(\delta b_t \mathrm{d}t + \delta\sigma_t^{k+1}\mathrm{d}W_t) - |\delta\sigma_t^{k+1}|^2\mathrm{d}t,\end{aligned}$$

integrate from 0 to t,

$$|\delta X_t^{k+1}|^2 = 2\int_0^t \delta X_s^{k+1}(\delta b_s \mathrm{d}t + \delta\sigma_s^{k+1}\mathrm{d}W_s) + \int_0^t |\delta\sigma_s^{k+1}|^2\mathrm{d}s,$$

and take the expectation

$$\begin{aligned}\mathbb{E}[|\delta X_t^{k+1}|^2 &= \mathbb{E}[\int_0^t (2\delta X_s^{k+1}\delta b_s + |\delta\sigma_s^{k+1}|^2)\mathrm{d}s]\\ &\le 2\mathbb{E}[\int_0^t |\delta X_s^{k+1}|L(|\delta X_s^{k+1}| + |\tilde{Y}_t^k - Y_t^k| + |\tilde{Z}_t^k - Z_t^k|)\mathrm{d}s]\\ &\quad + \mathbb{E}[\int_0^t L^2(|\delta X_s^{k+1}| + |\tilde{Y}_t^k - Y_t^k| + |\tilde{Z}_t^k - Z_t^k|)^2\mathrm{d}s]\\ &\le \mathbb{E}[\int_0^t ((2L+L+L)|\delta X_s^{k+1}|^2 + L|\tilde{Y}_t^k - Y_t^k|^2 + L|\tilde{Z}_t^k - Z_t^k|^2)\mathrm{d}s]\\ &\quad + \mathbb{E}[\int_0^t 3L^2(|\delta X_s^{k+1}|^2 + |\tilde{Y}_t^k - Y_t^k|^2 + |\tilde{Z}_t^k - Z_t^k|^2)\mathrm{d}s]\\ &= (4L+3L^2)\mathbb{E}[\int_0^t |\delta X_s^{k+1}|^2] + (L+3L^2)\mathbb{E}[\int_0^t |\tilde{Y}_t^k - Y_t^k|^2 + |\tilde{Z}_t^k - Z_t^k|^2\mathrm{d}s]\\ &\le (4L+3L^2)\mathbb{E}[\int_0^t |\delta X_s^{k+1}|^2] + (L+3L^2)\mathbb{E}[\int_0^T |\tilde{Y}_t^k - Y_t^k|^2 + |\tilde{Z}_t^k - Z_t^k|^2\mathrm{d}s]\end{aligned}$$

based on the Gronwall inequality, we get

$$\begin{aligned}\mathbb{E}[|\delta X_t^{k+1}|^2] &\le (L+3L^2)\mathbb{E}[\int_0^T |\tilde{Y}_t^k - Y_t^k|^2 + |\tilde{Z}_t^k - Z_t^k|^2\mathrm{d}s]\cdot e^{(4L+3L^2)T}\\ &= C_1\mathbb{E}[\int_0^T |\tilde{Y}_t^k - Y_t^k|^2 + |\tilde{Z}_t^k - Z_t^k|^2\mathrm{d}s],\end{aligned} \tag{3.8}$$

Similarly, we have

$$\begin{aligned}-\mathrm{d}|\delta Y_t^{k+1}|^2 &= -2\delta Y_t^{k+1}\cdot \mathrm{d}\delta Y_t^{k+1} - \mathrm{d}\delta Y_t^{k+1}\cdot \mathrm{d}\delta Y_t^{k+1}\\ &= 2\delta Y_t^{k+1}(\delta f_t \mathrm{d}t - \delta Z_t^{k+1}\mathrm{d}W_t) - |\delta Z_t^{k+1}|^2\mathrm{d}t,\end{aligned}$$

integrate from t to T,

$$|\delta Y_t^{k+1}|^2 + \int_t^T |\delta Z_s^{k+1}|^2\mathrm{d}s = |\delta Y_T^{k+1}|^2 + 2\int_t^T \delta Y_s^{k+1}(\delta f_s \mathrm{d}t - \delta Z_s^{k+1}\mathrm{d}W_s).$$

and take the expectation

$$
\begin{aligned}
&\mathbb{E}[|\delta Y_t^{k+1}|^2+\int_t^T|\delta Z_s^{k+1}|^2\mathrm{d}s]\\
=&\mathbb{E}[|\delta Y_T^{k+1}|^2]+2\mathbb{E}[\int_t^T\delta Y_s^{k+1}\delta f_s\mathrm{d}s]\\
\leq&\mathbb{E}[|\delta Y_T^{k+1}|^2]+2\mathbb{E}[\int_t^T|\delta Y_s^{k+1}|L(|\delta X_s^{k+1}|+|\delta Y_s^{k+1}|+|\delta Z_s^{k+1}|)\mathrm{d}s]\\
\leq&\mathbb{E}[|\delta Y_T^{k+1}|^2]+\mathbb{E}[\int_t^T L(|\delta X_s^{k+1}|^2+|\delta Y_s^{k+1}|^2)\mathrm{d}s]+\mathbb{E}[\int_t^T 2L|\delta Y_s^{k+1}|^2\mathrm{d}s]\\
&+\mathbb{E}[\int_t^T 2L^2|\delta Y_s^{k+1}|^2+\frac{1}{2}|Z_s^{k+1}|^2\mathrm{d}s]\\
\leq&\mathbb{E}[|\delta Y_T^{k+1}|^2]+L\cdot C_1\mathbb{E}[\int_0^T|\tilde{Y}_t^k-Y_t^k|^2+|\tilde{Z}_t^k-Z_t^k|^2\mathrm{d}s]\\
&+(3L+2L^2)\mathbb{E}[\int_t^T|\delta Y_s^{k+1}|^2\mathrm{d}s]+\mathbb{E}[\int_t^T\frac{1}{2}|Z_s^{k+1}|^2\mathrm{d}s]
\end{aligned}
$$

Then we have

$$
\begin{aligned}
&\mathbb{E}[|\delta Y_t^{k+1}|^2+\frac{1}{2}\int_t^T|\delta Z_s^{k+1}|^2\mathrm{d}s]\\
\leq&\mathbb{E}[|\delta Y_T^{k+1}|^2]+L\cdot C_1\mathbb{E}[\int_0^T|\tilde{Y}_t^k-Y_t^k|^2+|\tilde{Z}_t^k-Z_t^k|^2\mathrm{d}s]\\
&+(3L+2L^2)\mathbb{E}[\int_t^T|\delta Y_s^{k+1}|^2\mathrm{d}s],
\end{aligned}
\tag{3.9}
$$

thus

$$
\begin{aligned}
\mathbb{E}[|\delta Y_t^{k+1}|^2\leq&\mathbb{E}[L^2|\delta X_T^{k+1}|^2]+L\cdot C_1\mathbb{E}[\int_0^T|\tilde{Y}_t^k-Y_t^k|^2+|\tilde{Z}_t^k-Z_t^k|^2\mathrm{d}s]\\
&+(3L+2L^2)\mathbb{E}[\int_t^T|\delta Y_s^{k+1}|^2\mathrm{d}s]\\
\leq&(L+L^2)C_1\mathbb{E}[\int_0^T|\tilde{Y}_t^k-Y_t^k|^2+|\tilde{Z}_t^k-Z_t^k|^2\mathrm{d}s]\\
&+(3L+2L^2)\mathbb{E}[\int_t^T|\delta Y_s^{k+1}|^2\mathrm{d}s],
\end{aligned}
$$

based on the Gronwall inequality, we get

$$
\begin{aligned}
\mathbb{E}[|\delta Y_t^{k+1}|^2]&\leq C_1(L+L^2)\mathbb{E}[\int_0^T|\tilde{Y}_t^k-Y_t^k|^2+|\tilde{Z}_t^k-Z_t^k|^2\mathrm{d}s]\cdot e^{(2L^2+2L)T}\\
&=C_2\mathbb{E}[\int_0^T|\tilde{Y}_t^k-Y_t^k|^2+|\tilde{Z}_t^k-Z_t^k|^2\mathrm{d}s],
\end{aligned}
\tag{3.10}
$$

then

$$\begin{aligned}\mathbb{E}[\int_0^T |\delta Y_t^{k+1}|^2 \mathrm{d}t] &\leq \int_0^T \mathbb{E}|\delta Y_t^{k+1}|^2 \mathrm{d}t \\ &\leq \int_0^T C_3 \mathbb{E}[\int_0^T |\tilde{Y}_t^k - Y_t^k|^2 + |\tilde{Z}_t^k - Z_t^k|^2 \mathrm{d}s] \mathrm{d}s \\ &= TC_2 \mathbb{E}[\int_0^T |\tilde{Y}_t^k - Y_t^k|^2 + |\tilde{Z}_t^k - Z_t^k|^2 \mathrm{d}s] \\ &= C_3 \mathbb{E}[\int_0^T |\tilde{Y}_t^k - Y_t^k|^2 + |\tilde{Z}_t^k - Z_t^k|^2 \mathrm{d}s]. \end{aligned} \tag{3.11}$$

Similarly, we get

$$\mathbb{E}[\int_0^T |\delta Z_t^{k+1}|^2 \mathrm{d}t] \leq C_3 \mathbb{E}[\int_0^T |\tilde{Y}_t^k - Y_t^k|^2 + |\tilde{Z}_t^k - Z_t^k|^2 \mathrm{d}s],$$

and

$$\mathbb{E}[\int_0^T (|\delta Y_t^{k+1}|^2 + |\delta Z_t^{k+1}|^2) \mathrm{d}t] \leq 2C_3 \mathbb{E}[|\delta Y_T^{k+1}|^2].$$

Note that

$$\delta Y_t^{k+1} = \delta Y_T^{k+1} + \int_t^T \delta f_s \mathrm{d}s - \int_t^T \delta Z_s^{k+1} \mathrm{d}W_s,$$

then

$$\sup_{0\leq t\leq T} |\delta Y_t^{k+1}|^2 \leq 3|\delta Y_T^{k+1}|^2 + 3 \sup_{0\leq t\leq T} |\int_t^T \delta f_s \mathrm{d}s|^2 + 3 \sup_{0\leq t\leq T} |\int_t^T \delta Z_s^{k+1} \mathrm{d}W_s|^2,$$

as

$$\begin{aligned}\sup_{0\leq t\leq T} |\int_t^T \delta f_s \mathrm{d}s|^2 \leq (\int_0^T |\delta f_s| \mathrm{d}s)^2 &\leq (\int_0^T L^2(|\delta X_s^{k+1}| + |\delta Y_s^{k+1}| + |\delta Z_s^{k+1}|) \mathrm{d}s)^2 \\ &\leq 3L^2 \int_0^T (|\delta X_s^{k+1}|^2 + |\delta Y_s^{k+1}|^2 + |\delta Z_s^{k+1}|^2) \mathrm{d}s,\end{aligned}$$

we have

$$\begin{aligned}\mathbb{E}[\sup_{0\leq t\leq T} |\delta Y_t^{k+1}|^2] &\leq 3\mathbb{E}[|\delta Y_T^{k+1}|^2] + 9L^2 \mathbb{E}[\int_0^T (|\delta X_s^{k+1}|^2 + |\delta Y_s^{k+1}|^2 + |\delta Z_s^{k+1}|^2) \mathrm{d}s] \\ &\quad + 3\mathbb{E}[\sup_{0\leq t\leq T} |\int_t^T \delta Z_s^{k+1} \mathrm{d}W_s|^2] \\ &\leq (3L^2 C_1 + 9L^2(C_1 + 2C_3) + 3C_3) \mathbb{E}[\int_0^T |\tilde{Y}_t^k - Y_t^k|^2 + |\tilde{Z}_t^k - Z_t^k|^2 \mathrm{d}s] \\ &= C_4 \mathbb{E}[\int_0^T |\tilde{Y}_t^k - Y_t^k|^2 + |\tilde{Z}_t^k - Z_t^k|^2 \mathrm{d}s],\end{aligned}$$

then we get

$$\begin{aligned}
&\mathbb{E}[\sup_{0\le t\le T}|\hat{Y}_t^{k+1}-Y_t^{k+1}|^2+\int_0^T|\hat{Z}_t^{k+1}-Z_t^{k+1}|^2\mathrm{d}t]\\
\le&(C_4+C_3)\mathbb{E}[\int_0^T|\tilde{Y}_t^{k}-Y_t^{k}|^2+|\tilde{Z}_t^{k}-Z_t^{k}|^2\mathrm{d}s]\\
\le&(C_4+C_3)(T+1)\mathbb{E}[\sup_{0\le t\le T}|\tilde{Y}_t^{k}-Y_t^{k}|^2+\int_0^T|\tilde{Z}_t^{k}-Z_t^{k}|^2\mathrm{d}t]\\
=&C_5\mathbb{E}[\sup_{0\le t\le T}|\tilde{Y}_t^{k}-Y_t^{k}|^2+\int_0^T|\tilde{Z}_t^{k}-Z_t^{k}|^2\mathrm{d}t].
\end{aligned}$$

Step 2: According to Lemma 1, the SDE (3.4) has a unique solution $(\tilde{X}_t^{k+1},\tilde{Y}_t^{k+1},\tilde{Z}_t^{k+1})$ while $\tilde{Y}_0^{k+1}$ and $\{\tilde{Z}_t^{k+1}\}_{0\le t\le T}$ are given, then we have

$$\begin{aligned}
\tilde{Y}_t^{k+1}&=\tilde{Y}_0^{k+1}-\int_0^t f(s,\tilde{X}_s^{k+1},\tilde{Y}_s^{k+1},\tilde{Z}_s^{k+1})\mathrm{d}s+\int_0^t\tilde{Z}_s^{k+1}\mathrm{d}W_s\\
&=\tilde{Y}_T^{k+1}+\int_t^T f(s,\tilde{X}_s^{k+1},\tilde{Y}_s^{k+1},\tilde{Z}_s^{k+1})\mathrm{d}s-\int_t^T\tilde{Z}_s^{k+1}\mathrm{d}W_s. \qquad (3.12)
\end{aligned}$$

By using the similar method of proof with Step 1, we get

$$\mathbb{E}[\sup_{0\le t\le T}|\tilde{Y}_t^{k+1}-\hat{Y}_t^{k+1}|^2+\int_0^T|\tilde{Z}_t^{k+1}-\hat{Z}_t^{k+1}|^2\mathrm{d}t]\le C\mathbb{E}[|\tilde{Y}_T^{k+1}-g(\tilde{X}_T^{k+1})|^2], \quad (3.13)$$

for a constant C. Then

$$\begin{aligned}
&\mathbb{E}[\sup_{0\le t\le T}|\tilde{Y}_t^{k+1}-Y_t^{k+1}|^2+\int_0^T|\tilde{Z}_t^{k+1}-Z_t^{k+1}|^2\mathrm{d}t]\\
\le&\mathbb{E}[\sup_{0\le t\le T}|\tilde{Y}_t^{k+1}-\hat{Y}_t^{k+1}|^2+\int_0^T|\tilde{Z}_t^{k+1}-\hat{Z}_t^{k+1}|^2\mathrm{d}t]\\
&+\mathbb{E}[\sup_{0\le t\le T}|\hat{Y}_t^{k+1}-Y_t^{k+1}|^2+\int_0^T|\hat{Z}_t^{k+1}-Z_t^{k+1}|^2\mathrm{d}t]\\
\le&C\mathbb{E}[|\tilde{Y}_T^{k+1}-g(\tilde{X}_T^{k+1})|^2]+C_5\mathbb{E}[\sup_{0\le t\le T}|\tilde{Y}_t^{k}-Y_t^{k}|^2+\int_0^T|\tilde{Z}_t^{k}-Z_t^{k}|^2\mathrm{d}t].
\end{aligned}$$

Denote that

$$\begin{aligned}
a_{k+1}&=\mathbb{E}[\sup_{0\le t\le T}|\tilde{Y}_t^{k+1}-Y_t^{k+1}|^2+\int_0^T|\tilde{Z}_t^{k+1}-Z_t^{k+1}|^2\mathrm{d}t]\\
b_{k+1}&=C\mathbb{E}[|\tilde{Y}_T^{k+1}-g(\tilde{X}_T^{k+1})|^2],
\end{aligned}$$

we have the following iterative relationships

$$\begin{aligned}
a_{k+1}&\le b_{k+1}+C_5a_k\\
&\le b_{k+1}+C_5b_k+C_5^2a_{k-1}\\
&\le b_{k+1}+C_5b_k+C_5^2b_{k-1}+\cdots+C_5^{k+1}b_0
\end{aligned}$$

where $b_0=a_0$. Note that $C_5<1$ and $\lim_{k\to\infty}b_{k+1}=0$, we can easily proof that

$$\lim_{k\to\infty}a_{k+1}=\lim_{k\to\infty}\mathbb{E}[\sup_{0\le t\le T}|\tilde{Y}_t^{k+1}-Y_t^{k+1}|^2+\int_0^T|\tilde{Z}_t^{k+1}-Z_t^{k+1}|^2\mathrm{d}t]=0,$$

combine it with (3.2), then equation (3.6) holds. The proof is done.

□

Theorem 2 shows that we only need to find a sequence of $(\tilde{Y}_0^{k+1}, \tilde{Z}^{k+1})(k = 0, 1, 2, \cdots)$ to make $J(\tilde{Y}_0^{k+1}, \tilde{Z}^{k+1})$ tends to zero. Based on this theorem, we propose a Picard iteration method which approximate the solution of FBSDEs by neural networks, the detailed algorithm will be shown in Section 4.

3.2 State feedback control methods

In subsection 3.1, we transform the solution of FBSDE (2.1) to the control problem (3.3). It provides another point of view for solving FBSDEs besides the traditional numerical methods. Similar ideas have been proposed in [18, 19] and [20]. As the solution of FBSDE (2.1) can be converted into different control problems, different methods could be proposed according to the corresponding control problems. In this subsection, besides the Picard iteration method, we propose another two control methods for solving FBSDE (2.1) according to different kinds of state feedback. Detailed explanation will be found in Section 4.

3.2.1 Case 1: Single control method

In the first case, we extend the method in [18, 19, 20] for solving FBSDE (2.1), and we take a single process as the control. Consider the following variational problem:

$$\inf_{Y_0, \{Z_t\}_{0\le t\le T}} \mathbb{E}[|g(X_T^{Y_0,Z}) - Y_T^{Y_0,Z}|^2], \tag{3.14}$$

$$\begin{aligned} s.t.\ \ X_t^{Y_0,Z} &= X_0 + \int_0^t b(s, X_s^{Y_0,Z}, Y_s^{Y_0,Z}, Z_s)\,\mathrm{d}s + \int_0^t \sigma(s, X_s^{Y_0,Z}, Y_s^{Y_0,Z}, Z_s)\,\mathrm{d}W_s, \\ Y_t^{Y_0,Z} &= Y_0 - \int_0^t f(s, X_s^{Y_0,Z}, Y_s^{Y_0,Z}, Z_s)\,\mathrm{d}s + \int_0^t Z_s \mathrm{d}W_s, \end{aligned}$$

where Y_0 is $\mathcal{F}_0$-measurable random variable valued in $\mathbb{R}^m$ and Z_t is a $\mathcal{F}$-adapted and square-integrable process. The couple $(Y_0, \{Z_t\}_{0\le t\le T})$ is regarded as the control of variational problem (3.14). As discussed in Section 3.1, this control problem is equivalent to the original FBSDE. We give this following lemma without proof. Similar proofs can be referred to in Section 3.1.

Lemma 2. *Suppose that the control problem* (3.14) *satisfies that*

$$\inf_{Y_0, \{Z_t\}_{0\le t\le T}} \mathbb{E}[|g(X_T^{Y_0,Z}) - Y_T^{Y_0,Z}|^2] = 0,$$

then the corresponding triple $(X_t^{Y_0,Z}, Y_t^{Y_0,Z}, Z_t)$ *is the solution of FBSDE* (2.1).

The detailed iteration algorithm will be given in Section 4.

3.2.2 Case 2: Double control method

In (3.3) and (3.14), the initial value of process Y is regarded as a control. Now we regard the whole process Y as a control, then we have the following control problem

$$\inf_{\{u_t,Z_t\}_{0\le t\le T}} \mathbb{E}[|g(X_T^{u,Z}) - Y_T^{u,Z}|^2 + \int_0^T |Y_t^{u,Z} - u_t|^2 \mathrm{d}t], \tag{3.15}$$
$$s.t.\ \ X_t^{u,Z} = X_0 + \int_0^t b(s, X_s^{u,Z}, u_s, Z_s)\,\mathrm{d}s + \int_0^t \sigma(s, X_s^{u,Z}, u_s, Z_s)\,\mathrm{d}W_s,$$
$$Y_t^{u,Z} = u_0 - \int_0^t f(s, X_s^{u,Z}, Y_s^{u,Z}, Z_s)\,\mathrm{d}s + \int_0^t Z_s \mathrm{d}W_s,$$

where u_t and Z_t are two $\mathcal{F}$-adapted and square-integrable processes. The couple $(\{u_t, Z_t\})_{0\le t\le T}$ are the controls of the variational problem (3.15). We give the following lemma without proof. Similar proofs can be referred to in Section 3.1.

Lemma 3. *Suppose that the control problem (3.15) satisfies that*

$$\inf_{\{u_t,Z_t\}_{0\le t\le T}} \mathbb{E}[|g(X_T^{u,Z}) - Y_T^{u,Z}|^2 + \int_0^T |Y_t^{u,Z} - u_t|^2 \mathrm{d}t] = 0,$$

then the corresponding triple $(X_t^{u,Z}, Y_t^{u,Z}, Z_t)$ is the solution of FBSDE (2.1).

4 Numerical scheme for fully-coupled FBSDEs and the iterative algorithms

In Section 3, the FBSDE (2.1) is transformed to forward SDE (3.4). Then existing discretization methods could be applied in solving forward SDEs. In this paper, the Euler scheme is used for the time discretization. Kloeden [23] has shown that the classical Euler scheme has strong order of convergence which equals to $\frac{1}{2}$ for SDEs. Based on the Euler scheme, [10] has proposed a numerical scheme for coupled FBSDEs when the forward SDE does not depend on Z_t.

In this section, three iterative algorithms are proposed. In the first algorithm, the Picard iteration method is used and the three items (X, Y, Z) are taken as the input of the network. The other two algorithms are state feedback algorithms and two cases of state feedback are considered. In the first case, the inputs of the network are states X and Y, and the outputs are Z. In the second case, double control processes are employed, the state X is taken as the input of the network, the controls u and Z are taken as the output of the network.

4.1 Numerical scheme for fully-coupled FBSDEs

Let π be a partition of the time interval $[0, T]$, where $0 = t_0 < t_1 < t_2 < \cdots < t_{N-1} < t_N = T$. We define $\Delta t_i = t_{i+1} - t_i$ and $\Delta W_{t_i} = W_{t_{i+1}} - W_{t_i}$, where $W_{t_i} \sim \mathcal{N}(0, t_i)$, for $i = 0, 1, 2, \cdots, N-1$. We also denote

$$\delta = \sup_{0\le i\le N-1} \Delta t_i,$$

then the Euler scheme of forward SDE (3.1) can be written as

$$\begin{cases} X_{t_{i+1}}^{k+1,\pi} = X_{t_i}^{k+1,\pi} + b(t_i, X_{t_i}^{k+1,\pi}, Y_{t_i}^{k,\pi}, Z_{t_i}^{k,\pi})\Delta t_i + \sigma(t_i, X_{t_i}^{k+1,\pi}, Y_{t_i}^{k,\pi}, Z_{t_i}^{k,\pi})\Delta W_{t_i}, \\ Y_{t_{i+1}}^{k+1,\pi} = Y_{t_i}^{k+1,\pi} - f(t_i, X_{t_i}^{k+1,\pi}, Y_{t_i}^{k+1,\pi}, Z_{t_i}^{k+1,\pi})\Delta t_i + Z_{t_i}^{k+1,\pi}\Delta W_{t_i}, \\ X_0^{k+1,\pi} = X_0, Y_0^{k+1,\pi} = \tilde{Y}_0^{k+1}, \end{cases} \tag{4.1}$$

where $Y_{t_i}^{k,\pi}, Z_{t_i}^{k,\pi}$ and $\{Z_j^{k+1,\pi}\}$ are time discretization schemes of $Y_t^k, Z_t^k, \{Z_t^{k+1}\}_{0\le t\le T}$ respectively, and the values of $\tilde{Y}_0^{k+1,\pi}$ and $\{Z_j^{k+1,\pi}\}$ for $j = t_0, t_1, \cdots, t_{N-1}$ are given. According to [23], (4.1) converges at a rate of $\frac{1}{2}$, that is to say,

$$\mathbb{E}(|v_T^{k+1} - v_T^{k+1,\pi}|) \le K\delta^{\frac{1}{2}} \tag{4.2}$$

for a certain constant K, where

$$v_T^{k+1} = (\tilde{X}_T^{k+1}, \tilde{Y}_T^{k+1}), v_T^{k+1,\pi} = (X_T^{k+1,\pi}, Y_T^{k+1,\pi}),$$

and $(\tilde{X}_t^{k+1}, \tilde{Y}_t^{k+1})_{0\le t\le T}$ is the solution of forward SDE (3.4).

As the couple process $(\tilde{Y}_t^k, \tilde{Z}_t^k)$ is $\mathcal{F}_t$-adapted, we define $Z_{t_i}^{k+1,\pi}$ as a function of the triple $(X_{t_i}^{k+1,\pi}, Y_{t_i}^{k,\pi}, Z_{t_i}^{k,\pi})$:

$$Z_{t_i}^{k+1,\pi} = \phi_i(X_{t_i}^{k+1,\pi}, Y_{t_i}^{k,\pi}, Z_{t_i}^{k,\pi}), \tag{4.3}$$

where ϕ_i is the mapping from $(X_{t_i}^{k+1,\pi}, Y_{t_i}^{k,\pi}, Z_{t_i}^{k,\pi})$ to $Z_{t_i}^{k+1}$.

Combining (4.3) with (4.1), we have

$$\begin{cases} X_{t_{i+1}}^{k+1,\pi} = X_{t_i}^{k+1,\pi} + b(t_i, X_{t_i}^{k+1,\pi}, Y_{t_i}^{k,\pi}, Z_{t_i}^{k,\pi})\Delta t_i + \sigma(t_i, X_{t_i}^{k+1,\pi}, Y_{t_i}^{k,\pi}, Z_{t_i}^{k,\pi})\Delta W_{t_i}, \\ Y_{t_{i+1}}^{k+1,\pi} = Y_{t_i}^{k+1,\pi} - f(t_i, X_{t_i}^{k+1,\pi}, Y_{t_i}^{k+1,\pi}, Z_{t_i}^{k+1,\pi})\Delta t_i + Z_{t_i}^{k+1,\pi}\Delta W_{t_i}, \\ X_0^{k+1,\pi} = X_0, Y_0^{k+1,\pi} = \tilde{Y}_0^{k+1}, \\ Z_{t_i}^{k+1,\pi} = \phi_i(X_{t_i}^{k+1,\pi}, Y_{t_i}^{k,\pi}, Z_{t_i}^{k,\pi}), \end{cases} \tag{4.4}$$

When the mapping functions $\{\phi_i\}_{0\le i\le N-1}$ are known, we can approximate the solution of (3.4) with (4.4), and let

$$J(Y_0^{k+1,\pi}, Z^{k+1,\pi}) \to 0,$$

where k tends to infinity. According to Theorem 2, we can get the numerical solution of (2.1).

4.2 Algorithm 1: Picard iteration algorithm using neural network

Adopting the similar idea of [18] and [20], we use forward neural networks to simulate the functions $\{\phi_i\}_{0\le i\le N-1}$ mentioned in (4.4). The universal approximation theorem [24] has shown that a feed-forward network containing a finite number of neurons can approximate continuous functions for a given accuracy.

In order to apply the neural network method in the solution of FBSDEs, we rewrite (4.3) and (4.4) as

$$Z_{t_i}^{k+1,\pi} = \phi(X_{t_i}^{k+1,\pi}, Y_{t_i}^{k,\pi}, Z_{t_i}^{k,\pi}; \theta_i), \tag{4.5}$$

and

$$\begin{cases} X_0^{k+1,\pi} = X_0, Y_0^{k+1,\pi} = \tilde{Y}_0^{k+1}, \\ X_{t_{i+1}}^{k+1,\pi} = X_{t_i}^{k+1,\pi} + b(t_i, X_{t_i}^{k+1,\pi}, Y_{t_i}^{k,\pi}, Z_{t_i}^{k,\pi})\Delta t_i + \sigma(t_i, X_{t_i}^{k+1,\pi}, Y_{t_i}^{k,\pi}, Z_{t_i}^{k,\pi})\Delta W_{t_i}, \\ Y_{t_{i+1}}^{k+1,\pi} = Y_{t_i}^{k+1,\pi} - f(t_i, X_{t_i}^{k+1,\pi}, Y_{t_i}^{k+1,\pi}, Z_{t_i}^{k+1,\pi})\Delta t_i + Z_{t_i}^{k+1,\pi}\Delta W_{t_i}, \\ Z_{t_i}^{k+1,\pi} = \phi(X_{t_i}^{k+1,\pi}, Y_{t_i}^{k,\pi}, Z_{t_i}^{k,\pi}; \theta_i), \end{cases}$$

where θ_i represents the parameters of the feed-forward neural network at time t_i. We denote $\theta = \{\theta_i\}_{0\le i\le N-1}$ and define $Y_0^{k+1,\pi}$ as parameter. The network at each time point

t_i consists of four layers including one $m \times d$-dim input layer, two hidden $m \times d + 10$-dim layers and a $m \times d$-dim output layer. We adopt the ReLU activation function and Adam stochastic gradient descent-type algorithm in the network. The loss function is defined as

$$loss = \frac{1}{2M} \sum_{i=0}^{M} |Y_T^{k+1,\pi} - g(X_T^{k+1,\pi})|^2,$$

where M is the number of samples.

For convenience, the time interval $[0, T]$ is partitioned evenly, i.e. $\Delta t_i = t_{i+1} - t_i = T/N$ for all $i = 0, 1, 2, \cdots, N-1$. We define $\Delta W_i = W_{i+1} - W_i$ and denote the iteration step by k which is marked by superscript in the algorithm.

The detailed algorithm based on the conclusions of section 3 is given as following.

Algorithm 1 Picard iteration algorithm for implicit functions

Input: The Brownian motion ΔW_{t_i}, initial parameters $(\theta^0, Y_0^{0,\pi})$, learning rate η and the couple precess $(Y_{t_i}^{0,\pi}, Z_{t_i}^{0,\pi})$;

Output: The couple precess $(Y_{t_i}^{k+1,\pi}, Z_{t_i}^{k+1,\pi})$.

1: **for** $k = 0$ to $maxstep$ **do**
2: $\quad X_0^{k+1,\pi} = X_0, Y_0^{k+1,\pi} = Y_0^{k,\pi}$;
3: $\quad$ **for** $i = 0$ to $N-1$ **do**
4: $\quad\quad X_{t_{i+1}}^{k+1,\pi} = X_{t_i}^{k+1,\pi} + b(t_i, X_{t_i}^{k+1,\pi}, Y_{t_i}^{k,\pi}, Z_{t_i}^{k,\pi})\Delta t_i + \sigma(t_i, X_{t_i}^{k+1,\pi}, Y_{t_i}^{k,\pi}, Z_{t_i}^{k,\pi})\Delta W_{t_i}$;
5: $\quad\quad Z_{t_i}^{k+1,\pi} = \phi(X_{t_i}^{k+1,\pi}, Y_{t_i}^{k,\pi}, Z_{t_i}^{k,\pi}; \theta_i^k)$;
6: $\quad\quad Y_{t_{i+1}}^{k+1,\pi} = Y_{t_i}^{k+1,\pi} - f(t_i, X_{t_i}^{k+1,\pi}, Y_{t_i}^{k+1,\pi}, Z_{t_i}^{k+1,\pi})\Delta t_i + Z_{t_i}^{k+1,\pi}\Delta W_{t_i}$;
7: $\quad$ **end for**
8: $\quad (\theta^{k+1}, Y_0^{k+1,\pi}) = (\theta^k, Y_0^{k,\pi}) - \eta\nabla(\frac{1}{2M}\sum_{i=0}^{M} |Y_T^{k+1,\pi} - g(X_T^{k+1,\pi})|^2)$;
9: **end for**

Remark. In this algorithm, the Brownian motion is denoted as W. The initial paths $Y_{t_i}^{0,\pi}$ and $Z_{t_i}^{0,\pi}$ are generated randomly, and they do not influence the convergence of $Y_0^{k,\pi}$ and the process $Z^{k,\pi}$. As the aim of the algorithm is to find the optimal parameters θ^* to approximate the map ϕ in equation (4.5), The gradient descent method is used in line 8 of the algorithm which makes θ^k approximate to θ^* after k times of iterations. The numerical results shown in Section 5 confirm the convergence of the algorithm. Besides, when the function (4.5) only depends on X_{t_i}, the numerical results also demonstrate good convergence for the neural network approximation.

4.3 Algorithm 2: state feedback algorithm case 1

In [20], coupled FBSDEs where the forward SDE does not depend on Z_t have been studied. The values $(X_{t_i}, Y_{t_i}, Z_{t_i})_{0 \le i \le N-1}$ are calculated time-step by time-step, i.e. the triple $(X_{t_i}, Y_{t_i}, Z_{t_i})$ of the current time-step is used to calculate the triple $(X_{t_{i+1}}, Y_{t_{i+1}}, Z_{t_{i+1}})$ of the next time-step. We extend this idea to solve FBSDE (2.1).

For the control problem (3.14), the corresponding Euler scheme is

$$\begin{cases} X_{t_{i+1}}^{\pi} = X_{t_i}^{\pi} + b(t_i, X_{t_i}^{\pi}, Y_{t_i}^{\pi}, Z_{t_i}^{\pi})\Delta t_i + \sigma(t_i, X_{t_i}^{\pi}, Y_{t_i}^{\pi}, Z_{t_i}^{\pi})\Delta W_{t_i}, \\ Y_{t_{i+1}}^{\pi} = Y_{t_i}^{\pi} - f(t_i, X_{t_i}^{\pi}, Y_{t_i}^{\pi}, Z_{t_i}^{\pi})\Delta t_i + Z_{t_i}^{\pi}\Delta W_{t_i}, \\ X_0^{\pi} = X_0, Y_0^{\pi} = Y_0, \end{cases} \tag{4.6}$$

because the process $\{X_{t_i}^\pi, Y_{t_i}^\pi, Z_{t_i}^\pi\}_{0\le i\le N-1}$ is Markovian, $Z_{t_i}^\pi$ should be represented as a function of $(X_{t_i}^\pi, Y_{t_i}^\pi, Z_{t_i}^\pi)$

$$Z_{t_i}^\pi = \phi_i(X_{t_i}^\pi, Y_{t_i}^\pi, Z_{t_i}^\pi). \tag{4.7}$$

However, when simulating the function ϕ_i by the neural network, , $Z_{t_i}^\pi$ cannot be used as both input and output of the network. Notice that (4.7) is an implicit function, assuming that its explicit form is

$$Z_{t_i}^\pi = \phi_i'(X_{t_i}^\pi, Y_{t_i}^\pi), \tag{4.8}$$

which only depends on $X_{t_i}^\pi$ and $Y_{t_i}^\pi$. Though both of the functions ϕ_i and ϕ_i' are unknown , the objective of network estimation changes from approximating ϕ_i to approximating ϕ_i'. The network contains four layers including one $(n+m)$-dim input layer, two hidden $(n+m)+10$-dim layers and a $(m\times d)$-dim output layer. The cost function is the same as that for Algorithm 1. The corresponding algorithm is presented in the following.

Algorithm 2 State feedback algorithm case 1

Input: The Brownian motion ΔW_{t_i}, initial parameters $(\theta^0, Y_0^{0,\pi})$, learning rate η;
Output: The couple precess (X_T^π, Y_T^π).

for $k = 0$ to $maxstep$ **do**
  $X_0^{k,\pi} = X_0$, $Y_0^{k,\pi} = Y_0^{k,\pi}$;
  for $i = 0$ to $N-1$ **do**
    $Z_{t_i}^{k,\pi} = \phi(X_{t_i}^{k,\pi}, Y_{t_i}^{k,\pi}; \theta_i^k)$;
    $X_{t_{i+1}}^{k,\pi} = X_{t_i}^{k,\pi} + b(t_i, X_{t_i}^{k,\pi}, Y_{t_i}^{k,\pi}, Z_{t_i}^{k,\pi})\Delta t_i + \sigma(t_i, X_{t_i}^{k,\pi}, Y_{t_i}^{k,\pi}, Z_{t_i}^{k,\pi})\Delta W_{t_i}$;
    $Y_{t_{i+1}}^{k,\pi} = Y_{t_i}^{k,\pi} - f(t_i, X_{t_i}^{k,\pi}, Y_{t_i}^{k,\pi}, Z_{t_i}^{k,\pi})\Delta t_i + Z_{t_i}^{k,\pi}\Delta W_{t_i}$;
  end for
  $(\theta^{k+1}, Y_0^{k+1,\pi}) = (\theta^k, Y_0^{k,\pi}) - \eta\nabla(\frac{1}{2M}\sum_{i=0}^{M}|Y_T^{k,\pi} - g(X_T^{k,\pi})|^2)$;
end for

4.4 Algorithm 3: state feedback algorithm case 2

As discussed in subsection 3.2, we consider u_t and Z_t as controls. The corresponding Euler scheme is

$$\begin{cases} X_{t_{i+1}}^\pi = X_{t_i}^\pi + b(t_i, X_{t_i}^\pi, u_{t_i}^\pi, Z_{t_i}^\pi)\Delta t_i + \sigma(t_i, X_{t_i}^\pi, u_{t_i}^\pi, Z_{t_i}^\pi)\Delta W_{t_i}, \\ Y_{t_{i+1}}^\pi = Y_{t_i}^\pi - f(t_i, X_{t_i}^\pi, Y_{t_i}^\pi, Z_{t_i}^\pi)\Delta t_i + Z_{t_i}^\pi\Delta W_{t_i}, \\ X_0^\pi = X_0, Y_0^\pi = u_0, \\ u_{t_i}^\pi = \phi_i^1(X_{t_i}^\pi, u_{t_i}^\pi, Z_{t_i}^\pi), \\ Z_{t_i}^\pi = \phi_i^2(X_{t_i}^\pi, u_{t_i}^\pi, Z_{t_i}^\pi), \end{cases} \tag{4.9}$$

solving functions ϕ_i^1 and ϕ_i^2, we can get

$$\begin{cases} u_{t_i}^\pi = \phi_i'^1(X_{t_i}^\pi) = \phi^1(X_{t_i}^\pi; \theta_i^1), \\ Z_{t_i}^\pi = \phi_i'^2(X_{t_i}^\pi) = \phi^2(X_{t_i}^\pi; \theta_i^2). \end{cases} \tag{4.10}$$

Thus we need to construct two networks at the same time, one for simulating u and another for simulating Z. The network simulating u at each time point consists of four layers including one n-dim input layer, two hidden $n+10$-dim layers and a m-dim output layer. The network layers simulating Z is the same as that of u except that the output layer is $m\times d$-dim. All parameters of the two networks are represented as θ.

The lost function is denoted as

$$loss = \frac{1}{2M}\sum_{i=0}^{M}[|Y_T^\pi - g(X_T^\pi)|^2 + \frac{T}{N}\sum_{j=0}^{N-1}|Y_{t_j}^\pi - u_{t_j}^\pi|^2],$$

The detailed algorithm is shown as following:

Algorithm 3 State feedback algorithm case 2

Input: The Brownian motion ΔW_{t_i}, initial parameters θ^0, learning rate η;
Output: $X_T^{k,\pi}$ and precess $(Y_{t_i}^{k,\pi})_{0\le i\le N}$.

for $k = 1$ to $maxstep$ **do**
  $L = 0$;
  $X_0^{k,\pi} = X_0$;
  $Y_0^{k,\pi} = \phi^1(X_0; \theta_0^{k-1})$;
  for $i = 0$ to $N-1$ **do**
    $u_{t_i}^{k,\pi} = \phi^1(X_{t_i}^{k,\pi}; \theta_i^{1,k-1})$;
    $Z_{t_i}^{k,\pi} = \phi^2(X_{t_i}^{k,\pi}; \theta_i^{2,k-1})$;
    $X_{t_{i+1}}^{k,\pi} = X_{t_i}^{k,\pi} + b(t_i, X_{t_i}^{k,\pi}, u_{t_i}^{k,\pi}, Z_{t_i}^{k,\pi})\Delta t_i + \sigma(t_i, X_{t_i}^{k,\pi}, u_{t_i}^{k,\pi}, Z_{t_i}^{k,\pi})\Delta W_{t_i}$;
    $Y_{t_{i+1}}^{k,\pi} = Y_{t_i}^{k,\pi} - f(t_i, X_{t_i}^{k,\pi}, Y_{t_i}^{k,\pi}, Z_{t_i}^{k,\pi})\Delta t_i + Z_{t_i}^{k,\pi}\Delta W_{t_i}$;
    $L = L + \frac{T}{N}|Y_{t_{i+1}}^{k,\pi} - u_{t_{i+1}}^{k,\pi}|^2$;
  end for
  $Loss = \frac{1}{2M}\sum_{i=0}^{M}(|Y_T^{k,\pi} - g(X_T^{k,\pi})|^2 + L)$;
  $\theta^k = \theta^{k-1} - \eta\nabla Loss$;
end for

5 Numerical results

In this section, we present the numerical results of our algorithms for different cases including partially-coupled cases and fully-coupled cases. All the examples of this section are implemented with 256 sample-paths, learning rate $\eta = 5 \times 10^{-3}$, the number of time-points $N = 25$ if not specifically noted. The numerical experiments are performed in PYTHON on a LENOVO computer with a 2.40 Gigahertz(GHz) Inter Core i7 processor and 8 gigabytes(GB) random-access memory(RAM).

5.1 Example 1. Partially-coupled case(BSDE)

We consider an example in [25, 26] for the solution of partially-coupled FBSDEs.

Assume $t \in [0,T], x = (x_1, \cdots, x_d) \in \mathbb{R}^d, y \in \mathbb{R}, z \in \mathbb{R}^d$ and the functions b, σ, f, g in (2.1) satisfy

$$\begin{aligned} b(t,x,y,z) =& 0 \\ \sigma(t,x,y,z) =& 0.25 diag(x) \\ f(t,x,y,z) =& 0.25 \times (y - \frac{2 + 0.25^2 \times d}{2 \times 0.25^2 \times d})(\sum_{i=1}^{d} z_i) \\ g(x) =& \frac{exp(T + \sum_{i=1}^{d} x_i)}{1 + exp(T + \sum_{i=1}^{d} x_i)} \end{aligned}$$

where $diag(x)$ represents a diagonal matrix where the value of the ith diagonal element is x_i , and the explicit solution of this FBSDE is

$$Y(t,x) = \frac{exp(t + \sum_{i=1}^{d} x_i)}{1 + exp(t + \sum_{i=1}^{d} x_i)}.$$

We set $T = 0.5, X_0 = 0$ for different dimensions, and the explicit solution of Y_0 is 0.5.

Figure 1 shows that in the case of $d = 1$, the network solution is closer to the explicit solutions when the number of iteration steps increases. After 10000 steps, the value of Y_0 is 0.50496 and has a relative error of 0.99% comparing to the explicit solution.

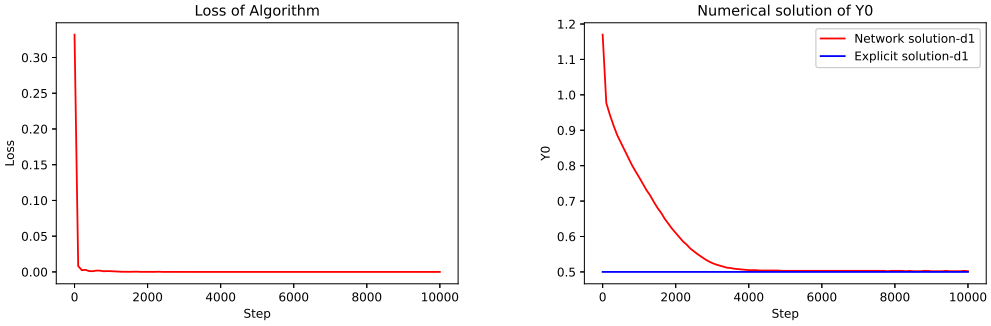

Figure 1: Case d=1. The left figure represents the curve of loss when the iteration step increases, and the loss achieves 4.52×10^{-7} after 10000 iteration steps. The right figure shows the comparison between the network solution and the explicit solution which are represented with the red curve and the blue curve respectively.

For $d = 100$ case, the value of Y_0 is 0.50413 after 15000 steps. The network has a rather optimistic performance with a relative error of 0.83% to the explicit solution, see figure 2 in detail.

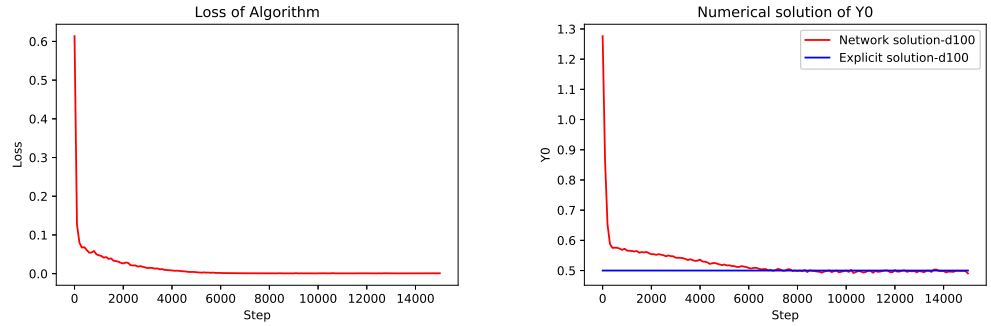

Figure 2: Case d=100. The curve of loss in the left figure shows that after 15000 steps, the loss value is 3.30×10^{-3}. As mentioned above, the red and blue curve represent respectively the network solution and the explicit solution.

The initial value of Y_0 is selected randomly in the interval $[1.0, 2.0]$ and the numerical algorithm is performed 10 times independently. Table 1 shows the detailed numerical results.

Table 1: Numerical results comparing with the explicit solutions

Step	Mean of Y_0	Variance of Y_0	Relative error of Y_0	Mean of runtime(s)
3000	0.54501	3.540E-02	0.0900	4715.6
6000	0.50970	6.803E-04	0.0194	8751.3
9000	0.49882	5.049E-06	0.0024	12693.2
12000	0.49658	3.752E-07	0.0068	16615.4
15000	0.49602	4.369E-07	0.0080	20528.8

5.2 Example 2. The forward SDE not containing Z term

We adopt the example of [27] which does not contain Z term in the forward SDE. Consider the following FBSDE,

$$\begin{cases} X_t = x + \int_0^t b(s, X_s, Y_s)ds + \int_0^t \sigma(s, X_s, Y_s)dW_s \\ Y_t = g(X_T) + \int_t^T f(s, X_s, Y_s, Z_s)ds - \int_t^T \langle Z_s, dW_s \rangle_{\mathbb{R}^d} \end{cases}$$

where

$$\begin{aligned} b_i(t, x, y) &= \frac{t}{2}\cos^2(y + x_i) \\ \sigma_{i,i}(t, x, y) &= \frac{t}{2}\sin^2(y + x_i) \\ g(x) &= \frac{1}{d}(\sum_{i=1}^{d-1} x_i^2(x_{i+1} + T) + x_d^2(x_1 + T)) \end{aligned}$$

and

$$\begin{aligned} f(t, x, y, z) = &\sum_{i=1}^{d} z_i - \frac{1}{d}(1 + \frac{t}{2})\sum_{i=1}^{d} x_i^2 - \frac{t}{d}(\sum_{i=1}^{d-1} x_i(x_{i+1} + t) + x_q(x_1 + t)) \\ &- \frac{t^2}{2d^2}(\sum_{i=1}^{d-1}(x_{i+1} + t)\sin^4(y + x_i) + (x_1 + t)\sin^4(y + x_d)) \end{aligned}$$

The explicit solution of this FBSDE is

$$Y_t = \frac{1}{d}(\sum_{i=1}^{d-1} X_{t,i}^2(X_{t,i+1} + t) + X_{t,d}^2(X_{t,1} + t)).$$

[27] has shown the results of different dimensions for $d = 2, 3, 4, 5$. In the case $d = 5, x = 1.0$, [27] has achieved a relative error of 5.489×10^{-4} in 134 seconds, and we get an approximated result of 1.0002 for Y_0 with a relative error of 0.02% comparing with the explicit solution of 1.0. Figure 3 shows the details.

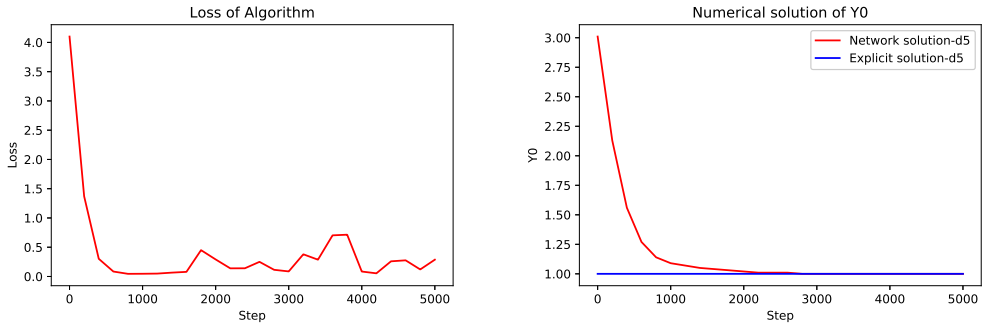

Figure 3: Case d=5. The curve of loss for 5000 iteration steps is shown in the left and the comparison with the explicit solution is shown in the right figure. As mentioned above, the red and blue curve represent the results of the network solution and the explicit solution respectively.

Similarly, we perform 10 independent runs for the case $d = 5$. Detailed results are shown in Table 2:

Table 2: Numerical results comparing with the explicit solutions

Step	Mean of Y_0	Variance of Y_0	Relative error of Y_0	Mean of runtime(s)
1000	1.09393	6.445E-02	9.393E-02	279.9
2000	1.02127	1.046E-04	2.127E-02	461.4
3000	1.00247	7.194E-06	2.470E-03	658.2
4000	1.00025	8.857E-06	2.538E-04	880.9
5000	1.00020	8.000E-06	1.996E-04	1077.4

For high dimensional cases, the method of [27] is not applicable while our neural network method shows satisfactory results. For the case $d = 100$, our network demonstrates remarkable performance and the relative error of Y_0 is 0.1%. See Figure 4 in detail.

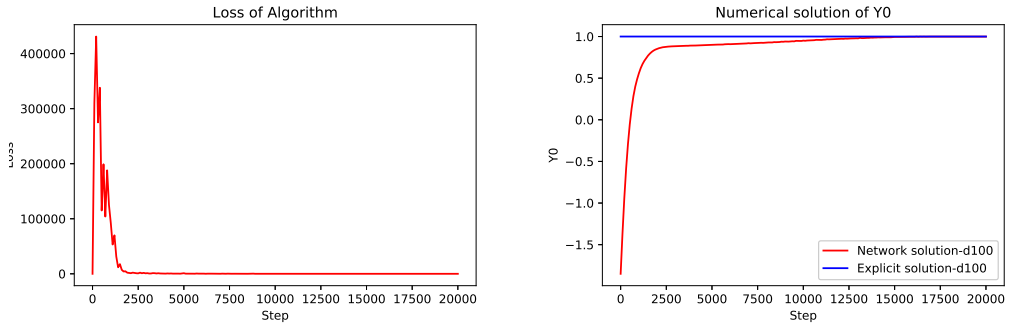

Figure 4: Case d=100. the left figure shows the curve of loss when the number of iteration steps increases. The right figure shows the comparison between the network solution and the explicit solution, the relative error have a downward trends when the number of iteration steps increases and tends to be stable at about 0.1%.

5.3 Example 3. 1-dim fully-coupled case

We adopt the example of [9] for the fully-coupled case. Consider the following FBSDE,

$$\begin{cases} X_t = X_0 - \int_0^t \frac{1}{2}\sin(s+X_s)\cos(s+X_s)(Y_s^2+Z_s)ds \\ \qquad + \int_0^t \frac{1}{2}\cos(s+X_s)(Y_s\sin(s+X_s)+Z_s+1)dW_s \\ Y_t = \sin(T+X_T) + \int_t^T Y_s Z_s - \cos(s+X_s)ds - \int_t^T Z_s dW_s \end{cases}$$

which satisfies Assumption 1 and Assumption 2, the solutions are $Y(t, X_t) = \sin(t + X_t)$ and $Z(t, X_t) = \cos^2(t + X_t)$. The numerical solution for this kind of FBSDEs is much more difficult as its diffusion coefficient is dependent on Z. [28] has also used this example in a Fourier methods.

We set $X_0 = 1.0, T = 0.1$, and the explicit solution $Y_0 \approx 0.84147$. The numerical results are shown in Figure 5 which are close to the results of [9], and the relative error is 7.49×10^{-4}.

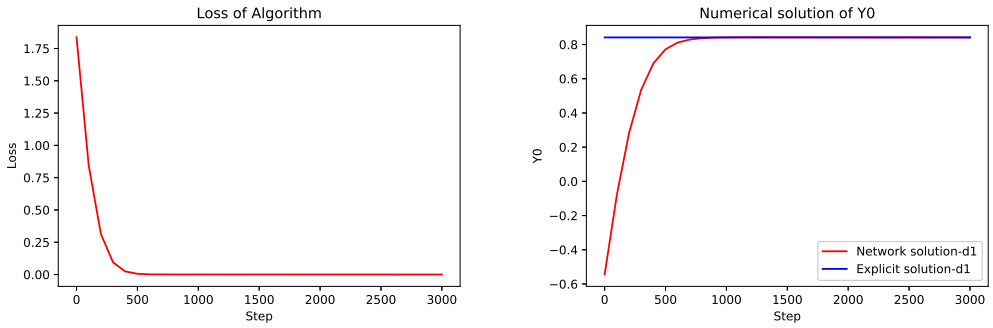

Figure 5: Case d=1. After 3000 iteration steps, the loss is reduced to 2.49×10^{-4}, and the relative error is reduced to 7.49×10^{-4}.

We also calculate the solution for Algorithm 2 and Algorithm 3 with this example. We use the same time-point, learning rate, number of iteration steps and samples as in Algorithm 1. In order to compare the results of the three algorithms more throughly, we calculate the variance of 1000 iteration steps before the current iteration. When the variance is less than 1×10^{-7} or the number of iteration steps achieves the upper limit 10000, the loop iteration is terminated. We record the number of iteration steps and the running time when the algorithm terminate. As expected, all the results are convergent. The comparison of the three algorithms are shown in Table 3. We run each algorithm 10 times independently.

Table 3: Numerical results comparison of the 3 algorithms

Method	Mean of Y_0	Variance of Y_0	Relative error of Y_0	Terminal step	Time(s)
Algorithm 1	0.84210	5.7423E-08	4.241E-04	2030.4	1076.4
Algorithm 2	0.83811	1.2623E-05	4.317E-03	2588.7	390.9
Algorithm 3	0.83344	5.3671E-06	9.864E-03	10000	2160.0

From Table 3, we can see that Algorithm 1 is more accurate and stable for this example. It requires the least number of iteration steps to achieve a smooth convergence result but it takes the longest time for each iteration step. Algorithm 2 and Algorithm 3 takes less

running time for each iteration step, but Algorithm 2 can not get a stable convergence result up to the maximum iteration step.

5.4 Example 4. 100-dim nonlinear generator for the FBSDE

Assume $t \in [0, T], x = (x_1, \cdots, x_n) \in \mathbb{R}^n, y \in \mathbb{R}, z \in \mathbb{R}^n$, consider this following FBSDE:

$$\begin{cases} \mathrm{d}X_{i,t} = n \exp(-\frac{1}{n}\sum_{i=0}^{n} X_{i,t}) Z_{i,t} \mathrm{d}W_{i,t} \\ -\mathrm{d}Y_t = -\exp(-\frac{1}{n}\sum_{i=0}^{n} X_{i,t})(\sum_{i=1}^{n} Z_{i,t}^2)\mathrm{d}t - Z_t^{\mathrm{T}} \mathrm{d}W_t \\ X_0 = x, Y_T = \exp(\frac{1}{n}\sum_{i=0}^{n} X_{i,T}) \end{cases} \tag{5.1}$$

where $\mathrm{d}W$ takes value in $\mathbb{R}^n$. We can check that the explicit solution of this FBSDE is

$$Y(t, x) = \exp(\frac{1}{n}\sum_{i=0}^{n} X_{i,t})$$

by using Ito's formula.

For example, we set $T = 0.1, X_0 = 1.0$ for $d = 100$, the explicit solution of Y_0 is $e \approx 2.7183$, and the number of time points is $N = 25$. The loss curve and the numerical solution of Y_0 are shown in Figure 6.We can see from Figure 6 that in the case of $d = 100$, the result of the network solution is closer to the explicit solution when the number of iteration steps increases. After 4000 steps, the value of Y_0 is 2.71662 and has a relative error of 0.061%.

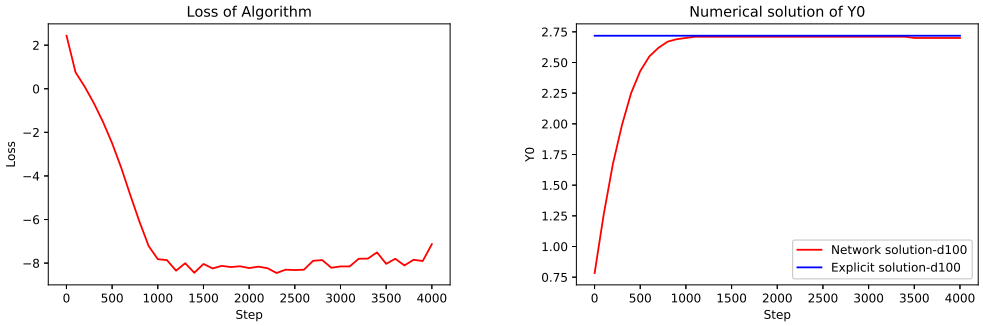

Figure 6: Case d=100. The curve of loss in the left figure shows that after 4000 steps, the loss value is 1.53×10^{-5}. The red and blue curves in the right figure represent respectively the network solution and the explicit solution.

We show the results with different initial values for $d = 100$ in Table 4. The neural network demonstrates satisfactory results. For the case of $x = 0.5$ and $x = 1.0$, we compare the three algorithms with the same stopping condition as mentioned in Example 3. The comparison results of the three algorithms are shown in Table 5 for $x = 0.5$ and Table 6 for $x = 1.0$ respectively.

Table 4: Numerical results for $d = 100$ with different initial values

	$x = 0.0$	$x = 0.1$	$x = 0.2$	$x = 0.5$	$x = 1.0$	$x = 2.0$
Explicit solution	1.00000	1.10517	1.22140	1.64872	2.71828	7.38906
Simulation	0.99777	1.10390	1.22103	1.64780	2.71662	7.38735
Absolute error	2.23E-03	1.27E-03	3.71E-04	9.21E-04	1.66E-03	1.71E-03
Relative error	2.23E-03	1.15E-03	3.03E-04	5.58E-04	6.11E-04	2.31E-03

Table 5: Numerical results comparison of the 3 algorithms in case $x = 0.5$

Method	Mean of Y_0	Variance of Y_0	Relative error of Y_0	Terminal step	Time(s)
Algorithm 1	1.64780	9.3351E-08	5.568E-04	1087.8	1957.8
Algorithm 2	1.64807	4.8513E-07	3.930E-04	1518.2	1000.8
Algorithm 3	1.64869	1.2622E-09	1.941E-05	2049.0	423.6

Table 6: Numerical results comparison of the 3 algorithms in case $x = 1.0$

Method	Mean of Y_0	Variance of Y_0	Relative error of Y_0	Terminal step	Time(s)
Algorithm 1	2.71644	1.2824E-07	6.762E-04	1589.2	2768.4
Algorithm 2	2.71537	3.5622E-07	1.071E-03	3516.8	2205.3
Algorithm 3	2.71825	4.4556E-10	1.214E-05	2557.7	503.4

From the running results of Example 3 and 4, we can get the following phenomenon. Algorithm 1 needs least number of iteration steps to get a stable convergence rate, but takes the longest time for a given step. Algorithm 3 computes as fast or faster than Algorithm 2, probably because it has fewer network parameters. In terms of accuracy, the three algorithms show different performance results for different problems. For example, Algorithm 3 has the best variance for Example 4, but for Example 3 it can not meet the requirement of stable convergence.

References

[1] J. M. Bismut, "Conjugate convex functions in optimal stochastic control," *Journal of Mathematical Analysis and Applications*, vol. 44, no. 2, pp. 384–404, 1973.

[2] E. Pardoux and S. Peng, "Adapted solution of a backward stochastic differential equation," *Systems and Control Letters*, vol. 14, no. 1, pp. 55–61, 1990.

[3] N. E. Karoui, S. Peng, and M. C. Quenez, "Backward stochastic differential equations in finance," *Mathematical Finance*, vol. 7, no. 1, pp. 1–71, 1997.

[4] J. Cvitanić and J. Ma, "Hedging options for a large investor and forward-backward sdes," *Annals of Applied Probability*, vol. 6, no. 2, pp. 370–398, 1996.

[5] E. Pardoux and S. Tang, "Forward-backward stochastic differential equations and quasilinear parabolic pdes," *Probability Theory and Related Fields*, vol. 114, no. 2, pp. 123–150, 1999.

[6] E. Tadmor, "A review of numerical methods for nonlinear partial differential equations," *Bulletin of the American Mathematical Society*, vol. 49, no. 4, pp. 507–554, 2012.

[7] Y. Jiongmin and M. Jin, *Forward-Backward stochastic differential equations and their applications*. Springer, 2007.

[8] J. Ma, P. Protter, and J. Yong, "Solving forward-backward stochastic differential equations explicitly — a four step scheme," *Probability Theory and Related Fields*, vol. 98, no. 3, pp. 339–359, 1994.

[9] F. Yu, W. Zhao, and T. Zhou, "Multistep schemes for forward backward stochastic differential equations with jumps," *Journal of Scientific Computing*, vol. 69, no. 2, pp. 1–22, 2016.

[10] C. Bender and J. Zhang, "Time discretization and markovian iteration for coupled fbsdes," *Annals of Applied Probability*, vol. 18, no. 1, pp. 143–177, 2008.

[11] M. Ruijter and C. W. Oosterlee, "A fourier-cosine method for an efficient computation of solutions to bsdes," *Social Science Electronic Publishing*, vol. 37, no. 2, pp. A829–A889, 2015.

[12] T. P. Huijskens, M. Ruijter, and C. W. Oosterlee, "Efficient numerical fourier methods for coupled forward-backward sdes," *Journal of Computational and Applied Mathematics*, vol. 296, pp. 593–612, 2016.

[13] B. Richard and K. Robert, "Dynamic programming and statistical communication theory," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 43, no. 8, pp. 749–751, 1957.

[14] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning." *Nature*, vol. 521, no. 7553, p. 436, 2015.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *International Conference on Neural Information Processing Systems*, 2012.

[16] G. E. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. W. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[17] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. V. Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[18] E. Weinan, 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*, vol. 5, no. 4, pp. 349–380, 2017.

[19] J. Han, A. Jentzen, and E. Weinan, "Solving high-dimensional partial differential equations using deep learning," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 115, no. 34, pp. 8505–8510, 2018.

[20] J. Han and J. Long, "Convergence of the deep bsde method for coupled fbsdes," *arXiv:1811.01165*, 2018.

[21] S. Peng and Z. Wu, "Fully coupled forward-backward stochastic differential equations and applications to optimal control," *Siam Journal on Control and Optimization*, vol. 37, no. 3, pp. 825–843, 1999.

[22] B. Oksendal, "Stochastic differential equations," *The Mathematical Gazette*, vol. 77, no. 480, pp. 65–84, 1985.

[23] P. E. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations*, 1992.

[24] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems*, vol. 2, no. 4, pp. 303–314, 1989.

[25] J.-F. Chassagneux, "Linear multi-step schemes for bsdes," *SIAM Journal on Numerical Analysis*, vol. 52, no. 6, pp. 2815–2836, 2014.

[26] E. Weinan, M. Hutzenthaler, A. Jentzen, and T. Kruse, "On multilevel picard numerical approximations for high-dimensional nonlinear parabolic partial differential equations and high-dimensional nonlinear backward stochastic differential equations," *arXiv: Numerical Analysis*, 2017.

[27] Y. Fu, W. Zhao, and T. Zhou, "Efficient spectral sparse grid approximations for solving multi-dimensional forward backward sdes," *Discrete and Continuous Dynamical Systems - Series B*, vol. 22, no. 9, 2016.

[28] T. P. Huijskens, M. Ruijter, and C. W. Oosterlee, "Efficient numerical fourier methods for coupled forward-backward sdes," *Journal of Computational and Applied Mathematics*, vol. 296, pp. 593–612, 2016.