

Probabilistic High Order Numerical Schemes for Fully Nonlinear Parabolic PDEs

Tao Kong¹, Weidong Zhao¹ and Tao Zhou^{2,*}

¹ School of Mathematics & Finance Institute, Shandong University, Jinan 250100, China.

² LSEC, Institute of Computational Mathematics, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China.

Received 24 May 2015; Accepted (in revised version) 28 August 2015

Abstract. In this paper, we are concerned with probabilistic high order numerical schemes for Cauchy problems of fully nonlinear parabolic PDEs. For such parabolic PDEs, it is shown by Cheridito, Soner, Touzi and Victoir [4] that the associated exact solutions admit probabilistic interpretations, i.e., the solution of a fully nonlinear parabolic PDE solves a corresponding second order forward backward stochastic differential equation (2FBSDEs). Our numerical schemes rely on solving those 2FBSDEs, by extending our previous results [W. Zhao, Y. Fu and T. Zhou, SIAM J. Sci. Comput., 36 (2014), pp. A1731-A1751.]. Moreover, in our numerical schemes, one has the flexibility to choose the associated forward SDE, and a suitable choice can significantly reduce the computational complexity. Various numerical examples including the HJB equations are presented to show the effectiveness and accuracy of the proposed numerical schemes.

AMS subject classifications: 60H35, 65H20, 65H30

Key words: Fully nonlinear parabolic PDEs, second order FBSDEs, probabilistic interpretations, probabilistic numerical schemes.

1 Introduction

The paper is concerned with probabilistic numerical schemes for solving nonlinear parabolic PDEs in the following form:

$$\begin{cases} u_t + F(t, x, u, Du, D^2u) = 0, & (t, x) \in [0, T) \times \mathbb{R}^m, \\ u(T, x) = g(x), & x \in \mathbb{R}^m, \end{cases} \quad (1.1)$$

*Corresponding author. Email addresses: vision.kt@gmail.com (T. Kong), wdzhao@sdu.edu.cn (W. Zhao), tzhou@lsec.cc.ac.cn (T. Zhou)

where $u(\cdot, \cdot)$ is a map from $[0, T] \times \mathbb{R}^m \rightarrow \mathbb{R}$; $Du(x)$ and $D^2u(x)$ stand for the gradient and the Hessian matrix of u with respect to x , respectively. The nonlinear operator F is a map $[0, T] \times \mathbb{R}^m \times \mathbb{R} \times \mathbb{R}^m \times \mathcal{S}^m \rightarrow \mathbb{R}$ and $g: \mathbb{R}^m \rightarrow \mathbb{R}$ is the terminal condition. The PDEs (1.1) are called fully nonlinear if the operator F is nonlinear with respect to the highest order derivatives D^2u .

Recently, there has been a great interest to derive probabilistic interpretations for solutions of PDEs. Pioneer work is due to Pardoux and Peng [17], where they show that the quasi-linear parabolic PDE is associated to a Markovian Backward SDE due to the nonlinear Feynman-Kac formula introduced by Pardoux and Peng [17]. Extensions to more general parabolic PDEs can be found in [2, 20]. To link fully nonlinear parabolic PDEs and backward SDEs, an recent work by Cheridito, Soner, Touzi and Victoir [4] introduced a notion of second order forward backward SDEs (2FBSDEs). They show that the solution of the fully nonlinear parabolic PDE solves a corresponding 2FBSDEs. We note that the G-expectation, a nonlinear expectation introduced by Peng [19] also deals with this issue.

Based on these probabilistic interpretations, one can derive the so called probabilistic numerical schemes for solving PDEs. In the quasi-linear case, the PDE is associated to a Markovian Backward SDE due to the nonlinear Feynman-Kac formula introduced by Pardoux and Peng [17]. One can refer to [2, 6, 7] and references therein for probabilistic numerical schemes, and to [1, 3, 5, 8, 11, 15, 16, 24–28] for numerical schemes for FBSDEs. There have also been numerous publications on the subject and the schemes have been extended to more general BSDEs, e.g. reflected BSDEs which is appropriate for pricing and hedging American options.

However, there are only a few work on 2FBSDEs [13, 22] and fully non-linear PDEs [9, 12, 23]. Moreover, existing work on fully non-linear PDEs aims at designing efficient schemes for high dimensional PDEs, however, the convergence rates are not satisfactory. In particular, we mention the work [12], where a numerical example for a 12-dimensional coupled FBSDE is reported, and it is shown by numerical test that the numerical method converges with order 1. Also, in [13], multistep schemes were proposed to solve 2FBSDEs, and high order convergence rates were obtained, however, only for low dimensional examples. We also note that people in the numerical PDEs community are paying more and more attention to the numerical approaches for fully nonlinear PDEs [10].

In this work, we aim at designing high order probabilistic numerical schemes for Cauchy problems of fully nonlinear parabolic PDEs. Our numerical schemes rely on solving those equivalent 2FBSDEs, by extending our previous results in [26], where the Euler-type method were used for the forward SDE, and highly accurate multistep method were used to approximate the derivatives derived from the backward stochastic differential equation in FBSDEs. The Euler method used to solve the forward SDE dramatically reduces the computational complexity. We show that in our framework one has the flexibility to choose the associated forward SDE, and a suitable choice can significantly reduce the computational complexity. Various numerical examples including the HJB equations are presented to show effectiveness and accuracy of the proposed numerical schemes.

The rest of the paper is organized as follows. In Section 2, we introduce some prelim-

inaries, which include the property of diffusion processes, the derivative approximation schemes, and the relationships between nonlinear parabolic PDEs and 2FBSDEs. In Section 3, we present our multi-step numerical schemes for solving the 2FBSDEs. Numerical experiments are reported in Section 4, and we finally draw some conclusions in Section 5.

2 Preliminaries

Let \mathcal{S}^d be the set of all $d \times d$ symmetric matrices. For $x \in \mathbb{R}^d$, and $B, C \in \mathcal{S}^d$, we denote

$$|x| = \sqrt{x_1^2 + \cdots + x_d^2}, \quad B:C = \sum_{i,j=1}^d B_{ij}C_{ij}, \quad |B| = \sqrt{B:B},$$

where the notation $:$ is called the Frobenius inner product for two matrices in \mathcal{S}^d . We denote by x^\top and B^\top the transposes of x and B , respectively. We denote by C_b^k the set of functions $\varphi(x): \mathbb{R}^m \rightarrow \mathbb{R}$ with uniformly bounded derivatives up to order k , and by C^{k_1, k_2} the set of functions $\varphi(t, x): [0, T] \times \mathbb{R}^m \rightarrow \mathbb{R}$ with continuous partial derivatives up to order k_1 with respect to $t \in \mathbb{R}$ and up to order k_2 with respect to $x \in \mathbb{R}^m$.

2.1 Properties of the generator of a diffusion process

The diffusion process is the driver of our probabilistic methods. Let (Ω, \mathcal{F}, P) be a probability space, $\{W_t\}_{t \in [0, T]}$ is a d -dimensional Brownian Motion defined on (Ω, \mathcal{F}, P) with the natural filtration $\mathbb{F} = \{\mathcal{F}_t\}_{0 \leq t \leq T}$ and all P -null sets in \mathcal{F}_0 . Let $\{X_t\}_{t \in [0, T]} \in \mathbb{R}^m$ be the diffusion process satisfying

$$X_t = x_0 + \int_{t_0}^t b(s, X_s) ds + \int_{t_0}^t \sigma(s, X_s) dW_s, \quad t \in [t_0, T], \quad (2.1)$$

with $x_0 \in \mathcal{F}_0$, $b: [0, T] \times \mathbb{R}^m \rightarrow \mathbb{R}^m$, $\sigma: [0, T] \times \mathbb{R}^m \rightarrow \mathbb{R}^{m \times d}$. Note that the diffusion process X_t is well defined under standard conditions on b and σ , such as,

$$\int_0^t |b(s, 0)| ds + \int_0^t \sigma^2(s, 0) ds < \infty, \\ |b(s, x) - b(s, y)| + |\sigma(s, x) - \sigma(s, y)| \leq L|x - y|.$$

We assume that the above conditions hold true in the sequel. We denote by $\mathbb{E}_t^x[\cdot]$ the conditional expectation operator on random variables under the condition $X_t = x$, that is, $\mathbb{E}_t^x[\cdot] = \mathbb{E}[\cdot | X_t = x]$.

We introduce some basic properties:

Lemma 2.1. Let X_s be the diffusion process defined by the SDE (2.1). A_X is its generator (cf. [14, p. 121]). If $f \in C^{1,2}([0, T] \times \mathbb{R}^m)$, then we have

$$A_X f(t, x) = \mathcal{L}f(t, x), \quad A_X f(t, X_t) = \mathcal{L}f(t, X_t), \quad (2.2)$$

where operator \mathcal{L} is defined by

$$\mathcal{L}\varphi(t, x) := \varphi_t(t, x) + b(t, x)^\top D\varphi(t, x) + \frac{1}{2}\sigma(t, x)^\top \sigma(t, x) : D^2\varphi(t, x) \quad (2.3)$$

for any function $\varphi \in C^{1,2}$.

Note that $A_X f(t, X_t) \in \mathcal{F}_t$ is a stochastic process. Furthermore, we have [26]:

Theorem 2.1. Let $t_0 < t$ be a fixed time, and $x_0 \in \mathbb{R}^m$ be a fixed space point. If $f \in C_0^{1,2}([0, T] \times \mathbb{R}^m)$ and $\mathbb{E}_{t_0}^{x_0}[|\mathcal{L}f(t, X_t)|] < +\infty$, we have

$$\frac{d\mathbb{E}_{t_0}^{x_0}[f(t, X_t)]}{dt} = \mathbb{E}_{t_0}^{x_0}[A_X f(t, X_t)], \quad t \geq t_0. \quad (2.4)$$

Moreover, the following identity holds

$$\left. \frac{d\mathbb{E}_{t_0}^{x_0}[f(t, X_t)]}{dt} \right|_{t=t_0} = \left. \frac{d\mathbb{E}_{t_0}^{x_0}[f(t, \bar{X}_t)]}{dt} \right|_{t=t_0}, \quad (2.5)$$

where \bar{X}_t is a diffusion process satisfying

$$\bar{X}_t = x_0 + \int_{t_0}^t \bar{b}_s ds + \int_{t_0}^t \bar{\sigma}_s dW_s \quad (2.6)$$

with $\bar{b}: [0, T] \times \mathbb{R}^m \rightarrow \mathbb{R}^m, \bar{\sigma}: [0, T] \times \mathbb{R}^m \rightarrow \mathbb{R}^{m \times d}$ being smooth functions satisfying $\bar{b}(t_0, x_0) = b(t_0, x_0)$, and $\bar{\sigma}(t_0, x_0) = \sigma(t_0, x_0)$.

The above theorem indicates that the values of the derivatives of $\mathbb{E}_{t_0}^{x_0}[f(t, X_t)]$ at t_0 is independent of the future action of X_t . By choosing different \bar{b} and $\bar{\sigma}$, identity (2.5) gives different ways for approximating $d\mathbb{E}_{t_0}^{x_0}[f(t, X_t)]/dt|_{t=t_0}$. The computational complexity can be reduced significantly if appropriate choices of \bar{b} and $\bar{\sigma}$ are made.

2.2 Derivative approximation

Now we introduce the multi-step method for approximating function derivatives. Let $u(t) \in C_b^{k+1}$ with k being a positive integer, and $\{t_i\}_{i=0, \dots, k} \subset \mathbb{R}$ satisfying $t_0 < t_1 < \dots < t_k$. We denote $\Delta t_{0,i} = t_i - t_0$ for $i = 0, 1, \dots, k$. Then by Taylor's expansion, for each t_i , we have

$$u(t_i) = \sum_{j=0}^k \frac{(\Delta t_{0,i})^j}{j!} \frac{d^j u}{dt^j}(t_0) + \mathcal{O}(\Delta t_{0,i})^{k+1}.$$

By multiplying each $u(t_i)$ with an real parameter $\alpha_{k,i}$ and adding the products together, we get

$$\sum_{i=0}^k \alpha_{k,i} u(t_i) = \sum_{j=0}^k \frac{\sum_{i=0}^k \alpha_{k,i} (\Delta t_{0,i})^j}{j!} \frac{d^j u}{dt^j}(t_0) + \mathcal{O}\left(\sum_{i=0}^k \alpha_{k,i} (\Delta t_{0,i})^{k+1}\right).$$

By dropping the high order derivatives and choosing $\{\alpha_{k,i}\}_{i=0,\dots,k}$ satisfying

$$\frac{\sum_{i=0}^k \alpha_{k,i} (\Delta t_{0,i})^j}{j!} = \begin{cases} 1, & j=1, \\ 0, & j \neq 1, \end{cases} \quad (2.7)$$

one obtains

$$\frac{du}{dt}(t_0) = \sum_{i=0}^k \alpha_{k,i} u(t_i) + R_D, \quad (2.8)$$

where $R_D = \mathcal{O}(\sum_{i=0}^k \alpha_{k,i} (\Delta t_{0,i})^{k+1})$. In particular, when the points $\{t_i\}_{i=0,\dots,k}$ are equidistant, i.e. $\Delta t_{0,i} = i\Delta t$, to get Eq. (2.8), we have a specific linear system for $\alpha_{k,i}\Delta t$,

$$\sum_{i=1}^k i^j [\alpha_{k,i}\Delta t] = \begin{cases} 1, & j=1, \\ 0, & j \neq 1, \end{cases} \quad (2.9)$$

which can easily be solved. We list $\alpha_{k,i}\Delta t$ ($i=0,1,\dots,k$) of the system (2.9) for $k=1,2,\dots,6$ in Table 1.

Table 1: The value of $\alpha_{k,i}\Delta t$.

$\alpha_{k,i}\Delta t$	$i=0$	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$	$i=6$
$k=1$	-1	1					
$k=2$	$-\frac{3}{2}$	2	$-\frac{1}{2}$				
$k=3$	$-\frac{11}{6}$	3	$-\frac{3}{2}$	$\frac{1}{3}$			
$k=4$	$-\frac{25}{12}$	4	-3	$\frac{4}{3}$	$-\frac{1}{4}$		
$k=5$	$-\frac{137}{60}$	5	-5	$\frac{10}{3}$	$-\frac{5}{4}$	$\frac{1}{5}$	
$k=6$	$-\frac{49}{20}$	6	$-\frac{15}{2}$	$\frac{20}{3}$	$-\frac{15}{4}$	$\frac{6}{5}$	$-\frac{1}{6}$

Inspired by the classical stability theory of the multi-step schemes for solving ODEs, we know that the roots $\{\lambda_{k,j}\}_{j=1}^k$ of the characteristic equation

$$P(\lambda) = \alpha_{k,0}\lambda^k + \sum_{j=1}^k \alpha_{k,j}\lambda^{k-j} = 0 \quad (2.10)$$

should satisfy the root conditions, i.e., $|\lambda_{k,j}| \leq 1.0$, and if $|\lambda_{k,j}| = 1.0$, then $\lambda_{k,j}$ are simple roots (i.e., $P'(\lambda_{k,j}) \neq 0$).

For $\alpha_{k,j}$ defined in (2.9), we list the maximum absolute values of the roots for $k=2,3,\dots,8$ except the simple root 1.0. We learn that the multi-step schemes with coefficients $\alpha_{k,j}$ are unstable for $k \geq 7$, that is why we have only listed the $\alpha_{k,i}\Delta t$'s for $1 \leq k \leq 6$ in Table 1. For more details, one can refer to [26].

Table 2: The maximum absolute root of (2.10) except 1.0

k	2	3	4	5	6	7	8
$\max(\lambda_{k,j})$	0.3333	0.4264	0.5608	0.7087	0.8633	1.0222	1.1839

2.3 The probabilistic representation of fully nonlinear parabolic PDEs

In this section we will derive the probabilistic representation of a second-order fully nonlinear parabolic PDE. Based on the representations (i.e., the 2FBSDEs), we shall deduce three reference equations, which play an important role in the design of our multistep schemes. We are interested in the probabilistic scheme for approximating the viscosity solution $u \in \mathcal{A} \subset C^0([t_0, T] \times \mathbb{R}^m)$ of the following second-order fully nonlinear parabolic PDE

$$\begin{cases} u_t + F(t, x, u, Du, D^2u) = 0, & (t, x) \in [t_0, T] \times \mathbb{R}^m, \\ u(T, x) = g(x), & x \in \mathbb{R}^m, \end{cases} \quad (2.11)$$

where the operator F is a continuous, nonlinear operator with F elliptic. The operator F is called fully nonlinear if it is nonlinear with respect to the Hessian matrix D^2u . We recall the following standard definition.

Definition 2.1. The operator F is elliptic, if for all $(t, x, \lambda, p) \in [t_0, T] \times \mathbb{R}^m \times \mathbb{R} \times \mathbb{R}^m$ there holds

$$F(t, x, \lambda, p, \gamma_1) \geq F(t, x, \lambda, p, \gamma_2), \quad \forall \gamma_1, \gamma_2 \in \mathcal{S}^d, \quad \gamma_1 \geq \gamma_2,$$

where $\gamma_1 \geq \gamma_2$ means that $\gamma_1 - \gamma_2$ is a nonnegative definite matrix.

Let $u \in C_b^{1,3}([0, T] \times \mathbb{R}^m)$ be the solution of (2.11). Assume $\sigma \in C_b^{1,2}([0, T] \times \mathbb{R}^m)$. Let

$$\begin{aligned} Y_t &:= u(t, X_t), & Z_t &:= (\sigma^\top Du)(t, X_t), \\ \Gamma_t &:= (\sigma^\top D(\sigma^\top Du))(t, X_t), & A_t &:= \mathcal{L}(\sigma^\top Du)(t, X_t), \end{aligned} \quad (2.12)$$

and define the function \hat{f} by

$$\hat{f} = \hat{f}(t, X_t, Y_t, Z_t, \Gamma_t) = f(t, X_t, u(t, X_t), Du(t, X_t), D^2u(t, X_t)),$$

where the function f is defined in

$$F(t, x, u, Du, D^2u) = f(t, x, u, Du, D^2u) + b^\top(t, x)Du + \frac{1}{2}\sigma^\top\sigma(t, x):D^2u(t, x). \quad (2.13)$$

Now we make the following 2FBSDEs.

$$X_t = x + \int_{t_0}^t b(s, X_s)ds + \int_{t_0}^t \sigma(s, X_s)dW_s, \quad (2.14a)$$

$$Y_t = g(X_T) + \int_t^T \hat{f}(s, X_s, Y_s, Z_s, \Gamma_s)ds - \int_t^T Z_s^\top dW_s, \quad t \in [t_0, T], \quad (2.14b)$$

$$Z_t = Z_{t_0} + \int_{t_0}^t A_s ds + \int_{t_0}^t \Gamma_s dW_s. \quad (2.14c)$$

Remark 2.1. The above 2FBSDEs admit a slightly different (yet equivalent) form from the ones in [4], and one can easily show their equivalence. The 2FBSDEs was first introduced by the authors in [4] in order to study fully nonlinear PDEs, and further investigated by the authors in [21].

The connection between solutions of 2PDEs (2.11) and second-order forward backward SDEs is given in the following proposition [4, 21].

Proposition 2.1. Assume $u(t, x)$ is the only viscosity solution of the PDE (2.11), and the 2FBSDE (2.14) admits a unique solution $(Y_t, Z_t, \Gamma_t, A_t)$. Then the following identity holds

$$u(t, x) = Y_t^{t, x}, \quad (2.15)$$

where $Y_t^{t, x}$ is the value of Y_t corresponds to a diffusion process X_t starting at (t, x) .

Furthermore, if $\sigma \in C_b^{1,2}([0, T] \times \mathbb{R}^m)$ and the 2PDE (2.11) has a solution $u \in C_b^{1,3}([0, T] \times \mathbb{R}^m)$, then the 2FBSDEs have the solution $(X_t, Y_t, Z_t, \Gamma_t, A_t)$ with $(Y_t, Z_t, \Gamma_t, A_t)$ defined in (2.12).

3 Numerical schemes for 2FBSDEs

3.1 The reference equations

Let N be a positive integer. For the time interval $[t_0, T]$, we introduce a regular time partition \mathcal{T} :

$$t_0 < t_1 < \cdots < t_N = T.$$

We denote $\Delta t_{t_n, k} = t_{n+k} - t_n$ for $n \in \{1, 2, \dots, N\}$ and $k \in \mathbb{N}$ satisfying $n+k \leq N$. Moreover, for $t \geq t_n$, we denote $\Delta W_{t_n, k} = W_{t_{n+k}} - W_{t_n}$, $\Delta t_{t_n, t} = t - t_n$ and $\Delta W_{t_n, t} = W_t - W_{t_n}$. We also denote $\mathbb{E}_{t_n}^x[\cdot] = \mathbb{E}[\cdot | \mathcal{F}_{t_n}^{t_n, x}]$. Let $\Theta_t = (X_t, Y_t, Z_t, \Gamma_t)$ be the solution of the second-order forward backward SDEs (2.14).

By taking conditional expectation $\mathbb{E}_{t_n}^x[\cdot]$ on both sides of Eq. (2.14b), we obtain the following integral equation

$$\mathbb{E}_{t_n}^x[Y_t] = \mathbb{E}_{t_n}^x[g(X_T)] + \int_t^T \mathbb{E}_{t_n}^x[\hat{f}(s, \Theta_s)] ds, \quad t \in [t_n, T]. \quad (3.1)$$

By taking derivative with respect to t in (3.1) we obtain the following reference ordinary differential equations (ODEs):

$$\frac{d\mathbb{E}_{t_n}^x[Y_t]}{dt} = -\mathbb{E}_{t_n}^x[\hat{f}(t, \Theta_t)], \quad t \in [t_n, T], \quad (3.2)$$

if the integrand $\mathbb{E}_{t_n}^x[\hat{f}(s, \Theta_s)]$ is continuous at $s = t$.

Note that, by Eqs. (2.14b) and (2.14c), we also have

$$\begin{aligned} Y_{t_n} &= Y_t + \int_{t_n}^t \hat{f}(s, \Theta_s) ds - \int_{t_n}^t Z_s dW_s, \\ Z_{t_n} &= Z_t - \int_{t_n}^t A_s ds - \int_{t_n}^t \Gamma_s dW_s, \end{aligned} \quad t \in [t_n, T]. \quad (3.3)$$

By Multiplying $\Delta W_{t_n, t}^\top$ on both sides of Eq. (3.3), and taking the conditional expectation $\mathbb{E}_{t_n}^x[\cdot]$, we obtain, for $t \in [t_n, T]$,

$$0 = \mathbb{E}_{t_n}^x[Y_t \Delta W_{t_n, t}^\top] + \int_{t_n}^t \mathbb{E}_{t_n}^x[\hat{f}(s, \Theta_s) \Delta W_{t_n, s}^\top] ds - \int_{t_n}^t \mathbb{E}_{t_n}^x[Z_s] ds, \quad (3.4)$$

$$0 = \mathbb{E}_{t_n}^x[Z_t \Delta W_{t_n, t}^\top] - \int_{t_n}^t \mathbb{E}_{t_n}^x[A_s \Delta W_{t_n, s}^\top] ds - \int_{t_n}^t \mathbb{E}_{t_n}^x[\Gamma_s] ds. \quad (3.5)$$

Assume that the two integrands in (3.4) and (3.5) are continuous at $s = t$, by taking derivative with respect to $t \in [t_n, T)$ on both sides, one gets the following reference equations:

$$\frac{d\mathbb{E}_{t_n}^x[Y_t \Delta W_{t_n, t}^\top]}{dt} = -\mathbb{E}_{t_n}^x[\hat{f}(t, \Theta_t) \Delta W_{t_n, t}^\top] + \mathbb{E}_{t_n}^x[Z_t], \quad t \in [t_n, T], \quad (3.6)$$

$$\frac{d\mathbb{E}_{t_n}^x[Z_t \Delta W_{t_n, t}^\top]}{dt} = \mathbb{E}_{t_n}^x[A_t \Delta W_{t_n, t}^\top] + \mathbb{E}_{t_n}^x[\Gamma_t], \quad t \in [t_n, T]. \quad (3.7)$$

Eqs. (3.2), (3.6) and (3.7) are reference ODEs for the 2FBSDEs (2.14). Our numerical schemes will be derived by approximating the derivatives and the conditional expectations in (3.2), (3.6) and (3.7).

3.2 Fully discrete schemes

To introduce our fully discrete schemes, we first introduce the time-space partition \mathcal{D}_h as

$$\mathcal{D}_h := \{\mathcal{D}_{h_n}^n\}_{n=0,1,\dots,N},$$

where $\mathcal{D}_{h_n}^n := \{x_j | x_j \in \mathbb{R}^m, j \in \mathbf{N}\}$ is the space partition on the time level $t = t_n \in \mathcal{T}$. Here h_n denotes the density of the partition $\mathcal{D}_{h_n}^n$, defined by $h_n = \max_{x \in \mathbb{R}^m} d(x, \mathcal{D}_{h_n}^n)$, where $d(A, B)$ is the distance between two sets A and B in \mathbb{R}^m . The elements of $\mathcal{D}_{h_n}^n$ is called grid points. We use \mathcal{D}_h to denote the unified space partition of \mathbb{R}^m if the space partitions $\mathcal{D}_{h_n}^n$ do not depend on n . Furthermore, denote by $U_{n,x}^\mathcal{D}$ the finite grid set of $\mathcal{D}_{h_n}^n$, satisfying $d(x, U_{n,x}^\mathcal{D}) < d(x, \mathcal{D}_{h_n}^n \setminus U_{n,x}^\mathcal{D})$. We call $U_{n,x}^\mathcal{D}$ the finite neighbor grid set of $x \in \mathcal{D}_{h_n}^n$.

We shall first propose the semi-discrete scheme for solving u on the time partition \mathcal{T} , and then propose the fully discrete scheme for solving the u on the time-space partition $\mathcal{T} \times \mathcal{D}_h$.

Let $\{\bar{X}_t^{t_n,x}\}_{t \in [t_n, T]}$ be a new diffusion process defined by the SDE

$$\bar{X}_t^{t_n,x} = x + \int_{t_n}^t \bar{b}(s, \bar{X}_s^{t_n,x}) ds + \int_{t_n}^t \bar{\sigma}(s, \bar{X}_s^{t_n,x}) dW_s, \quad t \in [t_n, T], \quad (3.8)$$

where $\bar{b}(t, x)$ and $\bar{\sigma}(t, x)$ are chosen smooth functions satisfying $\bar{b}(t_n, x) = b(t_n, x)$ and $\bar{\sigma}(t_n, x) = \sigma(t_n, x)$.

Note that the processes Y_t, Z_t and Γ_t are functions of (t, X_t) . We denote by $(\bar{Y}_t^{t_n,x}, \bar{Z}_t^{t_n,x}, \bar{\Gamma}_t^{t_n,x})$ the value of (Y, Z, Γ) at the time-space point $(t, \bar{X}_t^{t_n,x})$, that is,

$$\bar{Y}_t^{t_n,x} = Y(t, \bar{X}_t^{t_n,x}), \quad \bar{Z}_t^{t_n,x} = Z(t, \bar{X}_t^{t_n,x}), \quad \bar{\Gamma}_t^{t_n,x} = \Gamma(t, \bar{X}_t^{t_n,x}). \quad (3.9)$$

By Theorem 2.1 and identity (2.8), we deduce

$$\begin{cases} \left. \frac{d\mathbb{E}_{t_n}^x[Y_t]}{dt} \right|_{t=t_n} = \sum_{i=0}^k \alpha_{k,i} \mathbb{E}_{t_n}^x[\bar{Y}_{t_{n+i}}^{t_n,x}] + R_{y,n}^k, \\ \left. \frac{d\mathbb{E}_{t_n}^x[Y_t \Delta W_{t_n,t}^\top]}{dt} \right|_{t=t_n} = \sum_{i=1}^k \alpha_{k,i} \mathbb{E}_{t_n}^x[\bar{Y}_{t_{n+i}}^{t_n,x} \Delta W_{n,i}^\top] + R_{z,n}^k, \\ \left. \frac{d\mathbb{E}_{t_n}^x[Z_t \Delta W_{t_n,t}^\top]}{dt} \right|_{t=t_n} = \sum_{i=1}^k \alpha_{k,i} \mathbb{E}_{t_n}^x[\bar{Z}_{t_{n+i}}^{t_n,x} \Delta W_{n,i}^\top] + R_{\Gamma,n}^k, \end{cases} \quad (3.10)$$

where $\alpha_{k,i}$ are defined by (2.7), and $R_{y,n}^k$, $R_{z,n}^k$, and $R_{\Gamma,n}^k$ are truncation errors defined as following

$$\begin{aligned} R_{y,n}^k &= \left. \frac{d\mathbb{E}_{t_n}^x[Y_t]}{dt} \right|_{t=t_n} - \sum_{i=0}^k \alpha_{k,i} \mathbb{E}_{t_n}^x[\bar{Y}_{t_{n+i}}^{t_n,x}], \\ R_{z,n}^k &= \left. \frac{d\mathbb{E}_{t_n}^x[Y_t \Delta W_{t_n,t}^\top]}{dt} \right|_{t=t_n} - \sum_{i=1}^k \alpha_{k,i} \mathbb{E}_{t_n}^x[\bar{Y}_{t_{n+i}}^{t_n,x} \Delta W_{n,i}^\top], \\ R_{\Gamma,n}^k &= \left. \frac{d\mathbb{E}_{t_n}^x[Z_t \Delta W_{t_n,t}^\top]}{dt} \right|_{t=t_n} - \sum_{i=1}^k \alpha_{k,i} \mathbb{E}_{t_n}^x[\bar{Z}_{t_{n+i}}^{t_n,x} \Delta W_{n,i}^\top]. \end{aligned} \quad (3.11)$$

By plugging (3.10) into (3.2), (3.6) and (3.7), we get

$$\begin{cases} Z_{t_n} = \sum_{i=1}^k \alpha_{k,i} \mathbb{E}_{t_n}^x[\bar{Y}_{t_{n+i}}^{t_n,x} \Delta W_{n,i}^\top] + R_{z,n}^k, \\ \Gamma_{t_n} = \sum_{i=1}^k \alpha_{k,i} \mathbb{E}_{t_n}^x[\bar{Z}_{t_{n+i}}^{t_n,x} \Delta W_{n,i}^\top] + R_{\Gamma,n}^k, \\ -\alpha_{k,0} Y_{t_n} = \sum_{i=1}^k \alpha_{k,i} \mathbb{E}_{t_n}^x[\bar{Y}_{t_{n+i}}^{t_n,x}] + \hat{f}(t_n, x, Y_{t_n}, Z_{t_n}, \Gamma_{t_n}) + R_{y,n}^k, \end{cases} \quad (3.12)$$

where $Y_{t_n} = \bar{Y}_{t_n}^{t_n, x}$, $Z_{t_n} = \bar{Z}_{t_n}^{t_n, x}$.

Let u^n (Du^n resp.) be the numerical approximation of u (Du resp.) at the time level $t = t_n$, where u is the solution of the PDE (2.11), and let Y^n , Z^n and Γ^n be the numerical approximations of the solution process Y_t , Z_t and Γ_t of the auxiliary second-order forward backward SDEs (2.14) at time t_n , respectively. In addition, we denote by $\{\bar{Y}^{n+i}\}_{i=0, \dots, k}$, and $\{\bar{Z}^{n+i}\}_{i=0, \dots, k}$ the values of $\{Y^{n+i}\}_{i=0, \dots, k}$, and $\{Z^{n+i}\}_{i=0, \dots, k}$ at $\{\bar{X}_{t_{n+i}}^{t_n, x}\}_{i=1, \dots, k}$, respectively.

Now by removing the truncations error $R_{y,n}^k$, $R_{z,n}^k$, and $R_{\Gamma,n}^k$ from (3.12), we propose the semi-discrete scheme for solving the 2PDE as follows.

Scheme 1. Given $Y^n = u^n$, $Z^n = \bar{\sigma}(t_n, X^n) \nabla u^n$ for $n = N, \dots, N-k+1$, for $n = N-k, \dots, 0$, solve $u^n = u^n(X^n)$ by

$$\begin{cases} Z^n = \sum_{i=1}^k \alpha_{k,i} \mathbb{E}_{t_n}^{X^n} [\bar{Y}^{n+i} \Delta W_{n,i}^\top], \\ \Gamma^n = \sum_{i=1}^k \alpha_{k,i} \mathbb{E}_{t_n}^{X^n} [\bar{Z}^{n+i} \Delta W_{n,i}^\top], \\ -\alpha_{k,0} Y^n = \sum_{i=1}^k \alpha_{k,i} \mathbb{E}_{t_n}^{X^n} [\bar{Y}^{n+i}] + \hat{f}(t_n, x, Y^n, Z^n, \Gamma^n), \\ u^n = Y^n, \end{cases} \quad (3.13)$$

with $\bar{X}_t^{t_n, X^n}$ the solution of the SDE (3.8).

Though there are many ways to choose the drift coefficient \bar{b} and the diffusion coefficient $\bar{\sigma}$ in (3.8), in the sequel, we propose the following choice

$$\bar{b}(s, X_s^{t_n, x}) = b(t_n, x), \quad \bar{\sigma}(s, X_s^{t_n, x}) = \sigma(t_n, x), \quad \forall s \in [t_n, T]. \quad (3.14)$$

In this case, the diffusion process $\{\bar{X}_t^{t_n, x}\}$ at time levels $t = t_{n+i}$ (denote by $\bar{X}^{n,i}$) can be solved exactly by

$$\bar{X}^{n,i} = x + b(t_n, x) \Delta t_{n,i} + \sigma(t_n, x) \Delta W_{n,i}, \quad i = 1, \dots, k, \quad (3.15)$$

which yield the simplest explicit Euler scheme for solving (2.1). Then we have

$$\begin{aligned} \mathbb{E}_{t_n}^x [\bar{Y}_{t_{n+i}} \Delta W_{n,i}^\top] &= \mathbb{E}_{t_n}^x [Y_{t_{n+i}} (x + b(t_n, x) \Delta t_{n,i} + \sigma(t_n, x) \Delta W_{n,i}) \Delta W_{n,i}^\top], \\ \mathbb{E}_{t_n}^x [\bar{Z}_{t_{n+i}} \Delta W_{n,i}^\top] &= \mathbb{E}_{t_n}^x [Z_{t_{n+i}} (x + b(t_n, x) \Delta t_{n,i} + \sigma(t_n, x) \Delta W_{n,i}) \Delta W_{n,i}^\top], \\ \mathbb{E}_{t_n}^x [\bar{Y}_{t_{n+i}}] &= \mathbb{E}_{t_n}^x [Y_{t_{n+i}} (x + b(t_n, x) \Delta t_{n,i} + \sigma(t_n, x) \Delta W_{n,i})]. \end{aligned} \quad (3.16)$$

In order to use Scheme 1 to solve nonlinear PDEs, two more approximations must be done. The first one is the approximation of conditional expectations, and the second one

is the approximation of the values of functions at the space point $\bar{X}^{n,i}$, which generally does not hit the grid points in $\mathcal{D}_{h_{n+i}}^{n+i}$.

Before we give the fully-discrete scheme for solving nonlinear 2PDEs, we introduce two operators $\hat{\mathbb{E}}^{n,x}[\cdot]$ and $\mathbb{I}_{\mathcal{D}}^n$. For random variable X , we use $\hat{\mathbb{E}}^{n,x}[X]$ to denote the approximation of the conditional expectation $\mathbb{E}_{t_n}^x[X]$. And we use $\mathbb{I}_{\mathcal{D}}^n$ to denote the local interpolation operator, such that for any function g , $\mathbb{I}_{\mathcal{D}}^n g$ is the continuous function interpolated by the values of g on the grid points in \mathcal{D} on the time level $t = t_n$. For simplicity, we denote $\mathbb{I}_{\mathcal{U}_{n,x}^{\mathcal{D}}}^n$ by $\mathbb{I}_{\mathcal{D},x}^n$.

Now we rewrite (3.12) in the following equivalent form.

$$\begin{cases} Z_{t_n} = \sum_{i=1}^k \alpha_{k,i} \hat{\mathbb{E}}^{n,x} \left[\mathbb{I}_{\mathcal{D},\bar{X}^{n,i}}^{n+i} Y_{t_{n+i}} \Delta W_{n,i}^\top \right] - R_{z,n}^k + R_{z,n}^{k,\mathbb{E}} + R_{z,n}^{k,\mathbb{I}}, \\ \Gamma_{t_n} = \sum_{i=1}^k \alpha_{k,i} \hat{\mathbb{E}}^{n,x} \left[\mathbb{I}_{\mathcal{D},\bar{X}^{n,i}}^{n+i} Z_{t_{n+i}} \Delta W_{n,i}^\top \right] - R_{\Gamma,n}^k + R_{\Gamma,n}^{k,\mathbb{E}} + R_{\Gamma,n}^{k,\mathbb{I}}, \\ -\alpha_{k,0} Y_{t_n} = \sum_{i=1}^k \alpha_{k,i} \hat{\mathbb{E}}^{n,x} \left[\mathbb{I}_{\mathcal{D},\bar{X}^{n,i}}^{n+i} Y_{t_{n+i}} \right] + \hat{f}(t_n, x, Y_{t_n}, Z_{t_n}, \Gamma_{t_n}) \\ + R_{y,n}^k + R_{y,n}^{k,\mathbb{E}} + R_{y,n}^{k,\mathbb{I}}, \end{cases} \quad (3.17)$$

where

$$\begin{aligned} R_{z,n}^{k,\mathbb{E}} &= \sum_{j=1}^k \alpha_{k,j} (\mathbb{E}_{t_n}^x - \hat{\mathbb{E}}^{n,x}) \left[\bar{Y}_{t_{n+j}} \Delta W_{n,j}^\top \right], \\ R_{z,n}^{k,\mathbb{I}} &= \sum_{j=1}^k \alpha_{k,j} \mathbb{E}_{t_n}^{x,h} \left[(\bar{Y}_{t_{n+j}} - \mathbb{I}_{\mathcal{D},\bar{X}^{n,x}}^{n+j} Y_{t_{n+j}}) \Delta W_{n,j}^\top \right], \\ R_{\Gamma,n}^{k,\mathbb{E}} &= \sum_{j=1}^k \alpha_{k,j} (\mathbb{E}_{t_n}^x - \hat{\mathbb{E}}^{n,x}) \left[\bar{Z}_{t_{n+j}} \Delta W_{n,j}^\top \right], \\ R_{\Gamma,n}^{k,\mathbb{I}} &= \sum_{j=1}^k \alpha_{k,j} \mathbb{E}_{t_n}^{x,h} \left[(\bar{Z}_{t_{n+j}} - \mathbb{I}_{\mathcal{D},\bar{X}^{n,x}}^{n+j} Z_{t_{n+j}}) \Delta W_{n,j}^\top \right], \\ R_{y,n}^{k,\mathbb{E}} &= -\sum_{j=1}^k \alpha_{k,j} (\mathbb{E}_{t_n}^x - \hat{\mathbb{E}}^{n,x}) \left[\bar{Y}^{n+j} \right], \\ R_{y,n}^{k,\mathbb{I}} &= -\sum_{j=1}^k \alpha_{k,j} \mathbb{E}_{t_n}^{x,h} \left[\bar{Y}_{t_{n+j}} - \mathbb{I}_{\mathcal{D},\bar{X}^{n,x}}^{n+j} Y_{t_{n+j}} \right]. \end{aligned} \quad (3.18)$$

The three terms $R_{y,n}^{k,\mathbb{E}}$, $R_{z,n}^{k,\mathbb{E}}$, and $R_{\Gamma,n}^{k,\mathbb{E}}$ are approximation errors resulted from approximating conditional expectations, and the other three terms $R_{y,n}^{k,\mathbb{I}}$, $R_{z,n}^{k,\mathbb{I}}$, and $R_{\Gamma,n}^{k,\mathbb{I}}$ are errors introduced by numerical interpolations.

Let u_j^n denote the numerical value of u for $x_j \in \mathcal{D}_h^n$ at time level n . By removing the nine error terms $R_{y,n}^k$, $R_{y,n}^{k,\mathbb{E}}$, $R_{y,n}^{k,\mathbb{I}}$, $R_{z,n}^k$, $R_{z,n}^{k,\mathbb{E}}$, $R_{z,n}^{k,\mathbb{I}}$, $R_{\Gamma,n}^k$, $R_{\Gamma,n}^{k,\mathbb{E}}$ and $R_{\Gamma,n}^{k,\mathbb{I}}$ from (3.17), we propose our probabilistic scheme for solving 2PDEs as follows:

Scheme 2. Given $\{u^n(x)$ and $Z^n(x) = \bar{\sigma}(t_n, x) \nabla u^n(x)$ for $n = N, \dots, N-k+1$ and $x \in \cup_{n=N-k+1}^N \mathcal{D}_{h_n}^n$, for $n = N-k, \dots, 0$, and for $x \in \mathcal{D}_{h_n}^n$, solve $u^n = u^n(x)$ by

$$u^n = Y^n, \quad (3.19)$$

where Y^n is solved by the following procedure

$$\bar{X}^{n,i} = x + b(t_n, x) \Delta t_{n,i} + \sigma(t_n, x) \Delta W_{n,i}, \quad i = 1, \dots, k, \quad (3.20a)$$

$$Z^n = \sum_{i=1}^k \alpha_{k,i} \hat{\mathbb{E}}^{n,x} \left[\mathbb{I}_{\mathcal{D}, \bar{X}^{n,i}}^{n+i} Y^{n+i} \Delta W_{n,i}^\top \right], \quad (3.20b)$$

$$\Gamma^n = \sum_{i=1}^k \alpha_{k,i} \hat{\mathbb{E}}^{n,x} \left[\mathbb{I}_{\mathcal{D}, \bar{X}^{n,i}}^{n+i} Z^{n+i} \Delta W_{n,i}^\top \right], \quad (3.20c)$$

$$-\alpha_{k,0} Y^n = \sum_{i=1}^k \alpha_{k,i} \hat{\mathbb{E}}^{n,x} \left[\mathbb{I}_{\mathcal{D}, \bar{X}^{n,i}}^{n+i} Y^{n+i} \right] + \hat{f}(t_n, x, Y_j^n, Z_j^n, \Gamma_j^n). \quad (3.20d)$$

For a fixed $x \in \mathcal{D}_h^n$, in Scheme 2, we first solve $\bar{X}^{n,i}$ by the Euler scheme (3.20a) for $i = 1, \dots, k$; then we solve Z^n and Γ^n by (3.20b) and (3.20c) explicitly; and finally, we solve Y^n by (3.20d) implicitly, in which some iteration methods may be needed for solving Y^n . Here, we shall use the following iteration procedure to iteratively solve Y^n

$$-\alpha_{k,0} Y^{n,l+1} = \sum_{i=1}^k \alpha_{k,i} \hat{\mathbb{E}}^{n,x} \left[\mathbb{I}_{\mathcal{D}, \bar{X}^{n,i}}^{n+i} Y^{n,l} \right] + \hat{f}(t_n, x, Y^{n,l}, Z^n, \Gamma^n) \quad (3.21)$$

with a prior given iteration stop condition, such as $|Y^{n,l+1} - Y^{n,l}| \leq \epsilon_0$, where $\epsilon_0 > 0$ is a prescribed tolerance. For small time partition step size Δt_n , the solution $Y^{n,l}$ converges to Y^n as $l \rightarrow \infty$, provided that $\hat{f}(t_n, x, y, z, \gamma)$ is Lipschitz continuous with respect to y . It is worth to note that the procedures for solving u^n at different grids in \mathcal{D}_h^n are totally independent, which means that high performance parallel computing technique can be easily used to solve u^n efficiently.

Different from the classical finite difference methods, Scheme 2 also solve the first- and second-order derivatives Du and D^2u directly, by the relationship (2.12). For example, if the diffusion is just the Brownian motion, then $Du^n = Z^n$ and $D^2u^n = \Gamma^n$.

Remark 3.1. The quadrature methods in Scheme 2 could be any quadrature rules such as the Monte-Carlo methods, the quasi-Monte-Carlo methods, and the Gaussian quadrature methods and so on. However, when quadrature method is applied, non-grid points may be used. That's to say, for $x \in \mathcal{D}_h^n$, points $X^{n,j}$ (defined by (3.15) e.t.c.) not in \mathcal{D}_h^{n+j} may be used when approximating the conditional expectation at time level t_{n+j} . Thus, interpolation methods are needed. Note that any interpolation methods can be used here, however, care should be made if one wants to guarantee the stability and accuracy.

Remark 3.2. The local truncation errors of Scheme 2 are given by (3.11) and (3.18). For the errors caused by approximations of the derivative $R_{\cdot,n}^k$ in (3.11) and by the numerical interpolation $R_{\cdot,n}^{k,\mathbb{I}}$ in (3.18), when the data b, σ, f and g are smooth enough, the following estimates hold (if the r degree polynomials interpolation is used)

$$R_{y,n}^k, R_{z,n}^k, R_{\Gamma,n}^k = \mathcal{O}((\Delta t)^k), \quad R_{z,n}^{k,\mathbb{I}}, R_{y,n}^{k,\mathbb{I}}, R_{\Gamma,n}^{k,\mathbb{I}} = \mathcal{O}(h^{r+1}). \quad (3.22)$$

The other three terms $R_{y,n}^{k,\mathbb{E}}$, $R_{z,n}^{k,\mathbb{E}}$, and $R_{\Gamma,n}^{k,\mathbb{E}}$ are the local truncation errors resulted from the approximations of the conditional mathematical expectations in (3.16). It is noticed that these conditional expectations are functions of Gaussian random variables, which may be approximated by Hermite-Gauss Quadrature with high order accurately.

We also remark that Scheme 2 also covers the quasi-linear PDE, which corresponds to a coupled forward backward SDE. The operator for quasi-linear PDE is of the form

$$F(t, x, u, Du, D^2u) = a_2(x, u, Du)D^2u + a_0(x, u, Du). \quad (3.23)$$

However, the routine described in Section 2.3 and Section 3.2 can still be applied, and then we can deduce the decoupled 2nd order forward backward SDE, thus Scheme 2 is still valid to this case. One can easily show that our framework can also be used to solve semi-linear parabolic PDEs.

4 Numerical experiments

In this section, we shall present several constructive numerical examples to show the efficiency of the proposed multistep schemes in the last sections. For all the tests, we shall use the uniform partitions, i.e., we divide the space $[0, T] \times \mathbb{R}^m$ equidistantly by the grid points $(t, x) \in \mathcal{T} \times \mathcal{D}_h$, where $\mathcal{T}, \mathcal{D}_h$ are defined by

$$\begin{aligned} \mathcal{T} &:= \left\{ t_n \mid t_n = n\Delta t, \ n = 0, 1, \dots, N, \ \Delta t = \frac{T}{N} \right\}, \\ \mathcal{D}_h &:= \{ x_j \mid x_j = \mathbf{j} \cdot h, \ \mathbf{j} := (j_1, j_2, \dots, j_m)^\top, \text{ for each } j_i \in \mathbf{Z} \}, \end{aligned}$$

and $N > 0, h > 0$. Particularly, for the one dimensional case, we have $\mathcal{D}_h = \{x_j \mid x_j = jh + x_0, j = 0, \pm 1, \pm 2, \dots, x_0 \in \mathbb{R}\}$.

In each numerical example, we aim to solving $u(t, x)$ at every grid point $(t_n, x_j) \in \mathcal{T} \times \mathcal{D}_h$, by using scheme 2.

In our numerical experiments, the interpolation operator $\mathbb{I}_{\mathcal{D}}$ in Scheme 2 is taken as the local Lagrange interpolation method such that the interpolation error estimates in (3.22) hold, and the quadrature operator $\hat{\mathbb{E}}^{n,x}[\cdot]$ is chosen to be the Hermite-Gaussian quadrature. We shall use 10-point Hermite-Gauss quadrature rule so that the numerical quadrature error can be negligible. For more details of the Gaussian quadrature rule, one can refer to [26]. Hence for a standard d -dimensional normal random variable $\xi \sim N(0, \mathbf{1})$, it holds that

$$\hat{\mathbb{E}}[g(\xi)] = \sum_{\mathbf{j}} g(a_{\mathbf{j}}) w_{\mathbf{j}},$$

where $\hat{\mathbb{E}}$ stands for the numerical quadrature, g is an integrable function, $\mathbf{j} = (j_1, \dots, j_d)$ with each $j_i \in \{1, 2, \dots, 10\}$, $a_{\mathbf{j}} = (a_{j_1}, \dots, a_{j_d})$, $w_{\mathbf{j}} = \prod_{i=1}^d w_{j_i}$, and a_i and w_i are Gaussian points and weights, respectively. For more details of the Gaussian quadrature rule, one can refer

to [26]. To balance the errors resulting from the time discrete truncation and the spacial truncation, we choose $h = (\Delta t)^{\frac{k+1}{r+1}}$ where r is the degree of the Lagrange interpolation polynomial. Note that, given k and Δt , the space mesh size h depends on the r , and that the bigger the r is, the larger the h is, which implies that large space step size h can be used by choosing high-order Lagrangian interpolations. $\{Y^{N-j}, Z^{N-j}\}_{j=1}^k$ are given for fixed k in such a way that their effects for the convergence rate are also negligible.

All our numerical test are done by designing the code in Fortran 95 together with OpenMP 3.0. The numerical results are obtained by running the code on a workstation with one Intel Xeon E5-2620 v2 (12 cores, 2.10 GHz) CPU. *Long double* type (real(16)) for the float variables are used when programming to guarantee the computing precision. However, the time spent increases dramatically compared with the time spent when all variables are defined as double (real(8)). Even so, our code only takes a very short time to run.

In what follows, we will denote by CR the convergence rate and T_r the running time respectively.

Example 4.1. We first test an quasi-linear example. Consider the following HJB equation

$$\begin{cases} u_t + \inf_{\alpha \in \mathbb{R}} \left\{ \frac{\sigma^2}{2} u_{xx} + \beta \alpha u_x + p x^2 + q \alpha^2 \right\} = 0, & (t, x) \in [0, T] \times \mathbb{R}, \\ u(T, x) = 0, & x \in \mathbb{R}, \end{cases} \quad (4.1)$$

where $\sigma = 0.5, \beta = 0.5, p = 0.5, q = 1.5, T = 1.0$ are given constants. It can be shown that the true solution is

$$u(t, x) = a(t)x^2 + b(t),$$

with $a(t) = \frac{\sqrt{pq}}{\beta} \tanh(\beta \sqrt{\frac{p}{q}}(T-t))$ and $b(t) = \frac{q\sigma^2}{\beta^2} \log(\cosh(\beta \sqrt{\frac{p}{q}}(T-t)))$.

The optimal α is given by $\alpha^* = -\frac{\beta}{2q} u_x$, by putting this into Eq. (4.1), we deduce

$$\begin{cases} u_t + \frac{\sigma^2}{2} u_{xx} - \frac{\beta^2}{4q} u_x^2 + p x^2 = 0, & (t, x) \in [0, T] \times \mathbb{R}, \\ u(T, x) = 0, & x \in \mathbb{R}. \end{cases} \quad (4.2)$$

To derive the associated 2FBSDEs, we propose the following diffusion process

$$dX_t = c_1 \beta dt + c_2 \sigma dW_t, \quad t \in [0, 1], \quad (4.3)$$

with c_1, c_2 being constants that can be changed. Then, the associated 2FBSDEs yield

$$\begin{aligned} dX_t &= c_1 \beta dt + c_2 \sigma dW_t, & t \in (0, 1], \\ -dY_t &= \left(\frac{1}{2} \Gamma_t \frac{1-c_2^2}{c_2^2} - \frac{\beta^2}{4q\sigma^2 c_2^2} Z_t^2 - \frac{c_1 \beta}{c_2 \sigma} Z_t + p X_t^2 \right) dt - Z_t dW_t, & t \in [0, 1), \\ dZ_t &= A_t dt + \Gamma_t dW_t, & t \in [0, 1], \\ Y_T &= 0. \end{aligned}$$

Table 3: $|u_{t_0} - u^0|$ of Example 4.1: $c_1 = 1.5$, $c_2 = 1$.

$N \setminus k$	1	2	3	4	5	6	7	8
32	5.65E-03	1.44E-03	3.44E-06	4.58E-07	4.30E-09	3.89E-10	8.01E-12	2.95E-13
64	2.79E-03	3.66E-04	4.76E-07	2.99E-08	1.51E-10	6.47E-12	7.23E-14	1.19E-13
128	1.39E-03	9.23E-05	6.24E-08	1.90E-09	5.00E-12	1.04E-13	6.02E-16	1.01E-11
256	6.90E-04	2.32E-05	7.99E-09	1.20E-10	1.60E-13	1.64E-15	4.84E-18	NaN
512	3.45E-04	5.81E-06	1.01E-09	7.54E-12	5.08E-15	2.58E-17	1.56E-20	NaN
CR	1.01	1.99	2.94	3.97	4.93	5.96	7.17	NaN
T_r	2.01s	6.71s	19.56s	38.89s	69.89s	140.84s	165.12s	329.44s
r	8	8	8	12	12	12	22	28

Table 4: $|u_{t_0} - u^0|$ of Example 4.1: $c_1 = 1.5$, $c_2 = 1.0$ with larger N .

$N \setminus K$	3	5	6	$N \setminus K$	7
1024	8.927E-11	1.598E-16	4.047E-19	550	1.272E-20
2048	1.120E-11	5.011E-18	6.332E-21	600	5.925E-20
4096	1.402E-12	1.568E-19	9.901E-23	650	1.527E-19
CR	3.00	5.00	6.00	CR	-14.9
T_r	301.54s	1243.59s	2580.70s	T_r	388.03s

We solve this example by Scheme 2 with parameters $c_1 = 1.5$ and $c_2 = 1$ and the numerical results are listed in Table 3.

Table 3 shows that, the scheme converges for $1 \leq k \leq 7$, and diverges for $k > 7$. However, to check the stability of our scheme, we test relatively larger N and the corresponding results for $k = 3, 5, 6$ and 7 are listed in Table 4. It is learned from Table 4 that Scheme 2 is stable for $1 \leq k \leq 6$, and is unstable with $k = 7$, which coincides with the classical numerical ODE theory.

Furthermore, the multistep schemes admit high order convergence rates, which lead to very accurate approximation for a given time partition, however, it seems that a small time step should be used to guarantee the stability. To show the efficiency of the high order convergence rates, we further test this numerical example by different k with a prescribed accuracy. The results are shown in Table 5.

Table 5 indicates that to obtain the same precision, smaller partition number can be used for larger step k . The high order multistep schemes are more efficient. To see the effect of the choice of the diffusion process on numerical results, we have done several tests with different values of the parameters c_1 and c_2 . Fix $c_2 = 1$, numerical results with $c_1 = 0, 0.5, 1.0, 1.5, 2.5$ are almost the same. At this time, we can choose the diffusion process as the simplest Brownian Motion ($c_1 = 0, c_2 = 1$). However for different values of the parameter c_2 , the results differ from each other. Numerical results with $c_2 = 0.5$ and 1.5 are reported in Table 6 and Table 7, respectively. We list the results for k up to 5.

Table 5: Partition size and running time of Example 4.1: $c_1 = 1.5$, $c_2 = 1$ with fixed precision.

k	1	2	3	3	4	5	6
$ u_{t_0} - u^0 $	2.15E-05	2.32E-05	2.20E-05	1.01E-09	1.90E-09	4.30E-09	3.99E-09
N	8192	256	16	512	128	32	12
T_r	38.29s	1.68s	0.07s	13.12s	3.23s	0.50s	0.09s

Table 6: $|u_{t_0} - u^0|$ of Example 4.1: $c_1 = 1.0$, $c_2 = 0.5$.

N \ k	1	2	3	4	5
32	6.21E-03	1.05E-03	2.40E-06	3.46E-07	3.13E-09
64	3.08E-03	2.68E-04	3.37E-07	2.26E-08	1.11E-10
128	1.53E-03	6.76E-05	4.45E-08	1.44E-09	3.69E-12
256	7.64E-04	1.70E-05	5.72E-09	9.08E-11	1.19E-13
512	3.81E-04	4.25E-06	7.25E-10	5.70E-12	3.77E-15
CR	1.01	1.99	2.93	3.97	4.92
T_r	2.01s	6.47s	15.04s	36.58s	199.89s
r	8	8	8	15	25

Table 7: $|u_{t_0} - u^0|$ of Example 4.1: $c_1 = 1.0$, $c_2 = 1.5$.

N \ k	1	2	3	4	5
32	1.36E-02	1.07E-03	1.34E-06	3.57E-07	2.55E-09
64	6.82E-03	2.73E-04	2.00E-07	2.31E-08	9.19E-11
128	3.42E-03	6.88E-05	2.70E-08	1.48E-09	3.07E-12
256	1.71E-03	1.73E-05	3.51E-09	9.37E-11	9.91E-14
512	8.56E-04	4.33E-06	4.48E-10	5.88E-12	3.15E-15
CR	1	1.99	2.89	3.97	4.91
T_r	2.03s	7.3s	20.99s	47.36s	280.15s
r	8	8	8	20	30

By comparing the numerical results in Table 3, Table 6 and Table 7, we figure out that for small $k \leq 3$, although the same interpolate order is used, the running time T_r which indicates the computational effort is different. For large $k > 3$ the computational effort differs much more obviously, besides the different interpolation order used. Thus, different choices of the diffusion process lead to different complexity of computation. In the future work, We shall focus on how to choose the "best" b and σ .

Example 4.2. We now turn to the following fully nonlinear PDE

$$\begin{cases} u_t + \frac{u_{xx}}{2+u_{xx}} - u_x, \\ -\frac{e^{t+x}(1+e^{t+x})^3(1-e^{t+x})}{2(1+e^{t+x})^6 + e^{2t+2x}(1-e^{t+x})^2} = 0, & (t,x) \in [0,T] \times \mathbb{R}, \\ u(T,x) = \frac{e^{T+x}}{1+e^{T+x}}, & x \in \mathbb{R}, \end{cases} \quad (4.4)$$

with $T = 1.0$. It can be checked that the true solution is $u(t,x) = \frac{e^{t+x}}{1+e^{t+x}}$.

We propose the diffusion process as follows

$$dX_t = c_a dt + c_b dW_t, t \in (0,T), \quad (4.5)$$

where c_a and c_b are two constants. According to Section 2.3, since $f = F - \frac{1}{2}c_b^2 u_{xx} - c_a u_x$, we deduce that

$$\hat{f}(t,x,y,z,\gamma) = \frac{c_b^2 \gamma}{2c_b^4 + \gamma^2} - \frac{1}{2}\gamma - \frac{c_a + 1}{c_b}z - \frac{e^{t+x}(1+e^{t+x})^3(1-e^{t+x})}{2(1+e^{t+x})^6 + e^{2t+2x}(1-e^{t+x})^2},$$

Hence, the PDE (4.4) can be represented probabilistically as follows

$$\begin{aligned} dX_t &= c_a dt + c_b dW_t, & t \in (0,T], \\ -dY_t &= \frac{c_b^2 \gamma}{2c_b^4 + \gamma^2} - \frac{1}{2}\gamma - \frac{c_a + 1}{c_b}z \\ &\quad - \frac{e^{t+x}(1+e^{t+x})^3(1-e^{t+x})}{2(1+e^{t+x})^6 + e^{2t+2x}(1-e^{t+x})^2} dt - Z_t dW_t, & t \in [0,T), \\ dZ_t &= A_t dt + \Gamma_t dW_t, & t \in [0,T], \\ Y_T &= \frac{e^{T+X_T}}{1+e^{T+X_T}}. \end{aligned}$$

The numerical results by using Scheme 2 are reported in Table 8 with $c_a, c_b = 1$.

As stated in Example 4.1, numerical results with $1 \leq k \leq 7$ are listed which show that the schemes converge for $k \leq 6$. Meanwhile, it is shown that for a fixed partition number(N), more accurate approximations are obtained by applying scheme with larger step k . To check the stability, we provide more numerical results for $k = 2, 3, 4, 5, 6$ with $N = 1024, 2048, 4096$ in Table 9.

Table 8, together with Table 9, show that the k -step scheme 2 is a k th-order, stable scheme for $1 \leq k \leq 6$, and moreover, the scheme with larger k admits more accurate numerical approximations. However, for $c_b = 0.5$, the numerical result is bad compared with that for $c_b = 1.0$, as listed in Table 10. Table 10 indicates that we fail to get the expected convergence rate for $k = 4, 5$ in this case. Evidently, the choice of the diffusion process affects the numerical approximation.

Table 8: $|u_{t_0} - u^0|$ of Example 4.2: $c_a = 1.0$, $c_b = 1.0$.

N \ k	1	2	3	4	5	6	7
32	4.79E-03	1.28E-05	1.66E-05	1.61E-06	1.18E-07	8.07E-09	5.31E-10
64	2.35E-03	6.04E-06	2.33E-06	1.17E-07	4.06E-09	1.21E-10	4.37E-12
128	1.16E-03	1.87E-06	3.07E-07	7.93E-09	1.32E-10	2.21E-12	2.81E-14
256	5.78E-04	5.14E-07	3.95E-08	4.97E-10	4.46E-12	3.09E-14	5.62E-17
512	2.88E-04	1.34E-07	5.00E-09	2.96E-11	1.39E-13	5.22E-16	1.45E-17
CR	1.01	1.67	2.93	3.94	4.92	5.97	NaN
T_r	2.82s	8.39s	19s	48.81s	77.01s	114.91s	178.74s
r	11	11	11	20	20	20	20

Table 9: $|u_{t_0} - u^0|$ of Example 4.2: $c_a = 1.0$, $c_b = 1.0$ with large N.

N \ k	2	3	4	5	6
1024	3.432E-08	6.291E-10	1.946E-12	4.406E-15	1.617E-17
2048	8.672E-09	7.890E-11	1.215E-13	1.075E-16	3.131E-19
4096	2.180E-09	9.879E-12	7.605E-15	4.601E-18	5.530E-21
CR	1.99	3.00	4.00	4.95	5.76
T_r	105.86s	309.10s	603.74s	1083.29s	2210.37s

Table 10: $|u_{t_0} - u^0|$ of Example 4.2: $c_a = 1.0$, $c_b = 0.5$.

N \ k	1	2	3	4	5
32	3.22E-03	3.37E-04	2.60E-05	4.54E-05	5.27E-04
64	1.50E-03	8.89E-05	3.75E-06	1.52E-05	2.32E-04
128	7.25E-04	2.29E-05	5.31E-07	2.95E-06	8.17E-05
256	3.56E-04	5.80E-06	6.31E-08	4.56E-07	2.14E-05
512	1.77E-04	1.46E-06	7.90E-09	5.37E-08	4.22E-06
CR	1.04	1.96	2.93	2.45	1.74
T_r	2.85s	7.99s	17.47s	89.49s	124.24s
r	11	11	11	20	24

Example 4.3. Let us now consider the following equation

$$\begin{cases} u_t + \frac{u_{xx}}{2+u_{xx}} - u_x + \frac{\sin(t+x)}{2+\sin(t+x)} = 0, & (t,x) \in [0,T] \times \mathbb{R}, \\ u(T,x) = \sin(T+x), & x \in \mathbb{R}, \end{cases} \quad (4.6)$$

with $T = 1.0$. It is easy to check that the true solution is $u(t,x) = \sin(t+x)$.

The diffusion process is chosen to be Eq. (2.1) with constant drift coefficient $b = c_a$ and

Table 11: $|u_{t_0} - u^0|$ of Example 4.3: $c_a = 1.0$, $c_b = 1.0$.

$N \backslash k$	1	2	3	4	5
32	5.76E-02	1.67E-03	2.75E-04	1.71E-05	1.81E-06
64	2.91E-02	3.77E-04	3.72E-05	1.06E-06	1.03E-08
128	1.46E-02	8.96E-05	4.84E-06	2.03E-08	6.00E-10
256	7.30E-03	2.19E-05	6.16E-07	4.19E-09	1.72E-11
512	3.65E-03	5.40E-06	7.77E-08	2.62E-10	1.65E-12
CR	1.00	2.06	2.95	3.99	4.94
T_r	2.09s	7.48s	21.93s	110.95s	34.48s
r	8	8	8	20	26

diffusion coefficient $\sigma = c_b$. Then the generator of the auxiliary forward backward SDE is

$$\hat{f}(t, x, y, z, \gamma) = \frac{c_b^2 \gamma}{2c_b^4 + \gamma^2} - \frac{1}{2} \gamma - \frac{c_a + 1}{c_b} z + \frac{\sin(t+x)}{2 + \sin(t+x)}.$$

The numerical results for $c_a = 1.0$, $c_b = 1.0$ are shown in Table 11. Again, our multistep schemes admit high order convergence rate.

Example 4.4. Consider the following fully nonlinear HJB equation, which has applications in economics and finance.

$$\begin{cases} u_t + rxu_x - \frac{(\mu-r)^2}{2\sigma^2} \frac{u_x^2}{u_{xx}} = 0, & (t, x) \in [0, T] \times (0, +\infty), \\ u(T, x) = -\log(x), & x \in (0, +\infty), \end{cases} \quad (4.7)$$

where μ , r and σ are all constants. The exact solution $u(t, x) = -\log(x) - (r + \frac{(\mu-r)^2}{2\sigma^2})(T-t)$.

Let $c = \frac{\mu-r}{\sigma}$ and the forward SDE be

$$dX_t = (r + c^2)X_t dt + cX_t dW_t,$$

with initial $X_0 = 1.0$. Then the generator \hat{f} of the auxiliary forward backward SDE is

$$\hat{f}(t, x, y, z, \gamma) = -\frac{1}{2} \left(\frac{c^2 z^2}{\gamma - cz} + \gamma + cz \right).$$

Numerical results by using our multistep schemes are reported in Table 12. Similar as other examples, high order convergence rates are obtained.

Table 12: $|u_{t_0} - u^0|$ of Example 4.4: $\mu = 0.1$, $r = 0.05$, $\sigma = 1$.

N\k	1	2	3	4	5
32	9.92E-04	4.15E-04	7.20E-05	2.69E-06	9.61E-07
64	4.90E-04	5.24E-05	2.22E-07	6.11E-08	3.78E-08
128	2.47E-04	1.17E-05	2.04E-06	6.84E-09	1.29E-09
256	1.23E-04	3.62E-06	2.54E-08	4.62E-10	4.76E-11
512	6.13E-05	9.24E-07	6.57E-09	3.13E-11	1.75E-12
CR	1.00	2.15	3.00	3.98	4.78
T_r	1.26s	3.79s	14.76s	45.2s	58.92
r	3	5	7	7	7

5 Conclusion

We have proposed high-order probabilistic numerical schemes for fully nonlinear parabolic partial differential equations. The idea is to solve their associated second order forward backward SDE. Several numerical experiments are presented to show that the proposed k -step scheme admits a k th-order convergence rate for $1 \leq k \leq 6$.

There are, however, some other related topics that need to be investigated:

- High dimensional problems. Note the methods here can be easily extended to high dimensional problems. However, we have proposed the local Lagrange interpolation methods here in our schemes. For high dimensional problems, this would results in the tensor Lagrange interpolation methods, which may be time consuming. Thus, we would suggest more feasible techniques such as the sparse grid interpolation, RBF interpolation etc. This would be part of our future studies.
- Rigorous stability and convergence analysis for our multistep schemes are still missing. This is also our ongoing project.
- An elegant way of making the best choice of the diffusion process is also left for future studies.

Acknowledgments

This work is partially supported by the National Natural Science Foundations of China under grant numbers 91130003, 11201461 and 11171189. The authors would like to thank the referees for their valuable comments, which improve much of the quality of the paper.

References

- [1] C. Bender and J. Zhang, Time discretization and Markovian iteration for coupled FBSDEs, Ann. Appl. Probab., 18(2008), pp. 143-177.

- [2] B. Bouchard and N. Touzi, Discrete-time approximation and Monte-Carlo simulation of backward stochastic differential equations, *Stochastic Process. Appl.*, 111(2004), pp. 175-206.
- [3] J.F. Chassagneux and D. Crisen, Runge-Kutta schemes for BSDEs, to appear in *Ann. Appl. Probab.*, 2014.
- [4] P. Cheridito, H. M. Soner, N. Touzi, and Nicolas Victoir, Second-order backward stochastic differential equations and fully nonlinear parabolic PDEs, *Communications on Pure and Applied Mathematics*, Vol. LX (2007), pp. 1081-1110.
- [5] D. Crisan and K. Manolarakis, Solving backward stochastic differential equations using the cubature method, *SIAM J. Math. Finance*, (3)2012, pp. 534-571.
- [6] F. Delarue and S. Menozzi, A forward-backward stochastic algorithm for quasi-linear PDEs, *Ann. Appl. Probab.*, 16(2006), pp. 140-184.
- [7] F. Delarue, and S. Menozzi, An interpolated stochastic algorithm for quasi-linear pdes. *Mathematics of Computation* 77, 261 (2008), 125-158.
- [8] J. Douglas, J. Ma and P. Protter, Numerical methods for forward-backward stochastic differential equations, *Ann. Appl. Probab.*, 6(1996), pp. 940-968.
- [9] Arash Fahim, Nizar Touzi, and Xavier Warin, A probabilistic numerical method for fully nonlinear parabolic PDEs, *Ann. Appl. Probab.*, 4(2011), pp. 1322-1364.
- [10] X. Feng, R. Glowinski, and M. Neilan, Recent developments in numerical methods for fully nonlinear second order partial differential equations. *SIAM Review* 55, 2(2013), 205-267.
- [11] Y. Fu, W. Zhao, and T. Zhou, Efficient sparse grid approximations for multi-dimensional coupled forward backward stochastic differential equations, submitted, 2015.
- [12] W. Guo, J. Zhang, and J. Zhuo, A Monotone Scheme for High Dimensional Fully Nonlinear PDEs, arXiv:1212.0466, to appear in *Ann. Appl. Probab.*, 2015.
- [13] Tao Kong, Weidong Zhao, and Tao Zhou, High order numerical schemes for second order FBSDEs with applications to stochastic optimal control, arXiv:1502.03206, 2015.
- [14] Bernt øksendal, *Stochastic Differential Equations: An Introduction with Applications*, 6th edition (2014) Springer.
- [15] J. P. Lemor, E. Gobet and X. Warin, A regression-based Monte Carlo method for backward stochastic differential equations, *Ann. Appl. Probab.*, 15(2005), pp. 2172-2202.
- [16] G. N. Milstein and M. V. Tretyakov, Discretization of Forward-Backward Stochastic Differential Equations And Related Quasi-linear Parabolic Equations, *SIAM J. Numer. Anal.*, 27(2007), 24-34.
- [17] E. Pardoux and S. Peng, Backward stochastic differential equations and quasilinear parabolic partial differential equations, *Lecture Notes in CIS*, Springer, 176 (1992), 200-217.
- [18] E. Pardoux and S. Tang, Forward-backward stochastic differential equations and quasilinear parabolic PDEs, *Probab. Theory Relat. Fields*, 114(1999), pp. 123-150.
- [19] S. Peng, Nonlinear Expectations and Stochastic Calculus under Uncertainty, preprint (2010), arXiv:1002.4546v1.
- [20] S. G. Peng, Probabilistic interpretation for systems of quasilinear parabolic partial differential equations, *Stoch. Stoch. Repts.*, 37 (1991), pp. 61-74.
- [21] H. M. Soner, N. Touzi, and J. Zhang, Wellposedness of second order backward SDEs, *Probab. Theory Relat. Fields*, Vol. 153(2012), pp:149-190.
- [22] X. Tan, Probabilistic Numerical Approximation for Stochastic Control Problems, preprint, 2011.
- [23] X. Tan, A splitting method for fully nonlinear degenerate parabolic PDEs, preprint, 2011.
- [24] T. Tang, W. Zhao, and T. Zhou, Deferred correction methods for forward backward stochastic differential equations, submitted, 2015.

- [25] W. Zhao, L. Chen and S. Peng, A new kind of accurate numerical method for backward stochastic differential equations, *SIAM J. Sci. Comput.*, 28(2006), pp. 1563-1581.
- [26] W. Zhao, Y. Fu, and T. Zhou, New kinds of high-order multistep schemes for coupled forward backward stochastic differential equations, *SIAM J. Sci. Comput.*, 36(4), pp. A1731-1751, 2014.
- [27] W. Zhao, G. Zhang and L. Ju, A stable multistep scheme for solving backward stochastic differential equations, *SIAM J. Numer. Anal.*, 48(2010), pp. 1369-1394.
- [28] W. Zhao, W. Zhang and L. Ju, A numerical method and its error estimates for the decoupled forward-backward stochastic differential equations, *Commun. Comput. Phys.*, 15(2014), pp. 618-646.