Journal of Computational Mathematics
Vol.31, No.3, 2013, 221–248.

http://www.global-sci.org/jcm
doi:10.4208/jcm.1212-m4014

A SPARSE-GRID METHOD FOR MULTI-DIMENSIONAL BACKWARD STOCHASTIC DIFFERENTIAL EQUATIONS*

Guannan Zhang
Department of Scientific Computing, Florida State University, Tallahassee, FL 32306,
Computer Science and Mathematics Division, Oak Ridge National Lab, Oak Ridge, TN 37831
Email: zhangg@ornl.gov
Max Gunzburger
Department of Scientific Computing, Florida State University, Tallahassee, FL 32306
Email: gunzburg@fsu.edu
Weidong Zhao
School of Mathematics, Shandong University, Jinan, Shandong 250100, China
Email: wdzhao@sdu.edu.cn

Abstract

A sparse-grid method for solving multi-dimensional backward stochastic differential equations (BSDEs) based on a multi-step time discretization scheme [31] is presented. In the multi-dimensional spatial domain, i.e. the Brownian space, the conditional mathematical expectations derived from the original equation are approximated using sparse-grid Gauss-Hermite quadrature rule and (adaptive) hierarchical sparse-grid interpolation. Error estimates are proved for the proposed fully-discrete scheme for multi-dimensional BSDEs with certain types of simplified generator functions. Finally, several numerical examples are provided to illustrate the accuracy and efficiency of our scheme.

Mathematics subject classification: 60H10, 60H35, 65C10, 65C20, 65C50.
Key words: Backward stochastic differential equations, Multi-step scheme, Gauss-Hermite quadrature rule, Adaptive hierarchical basis, Sparse grids.

1. Introduction

We consider the following backward stochastic differential equation (BSDE)

$$\begin{cases} -dy_t = f(t, y_t, z_t)dt - z_t dW_t, \quad t \in [0, T), \\ y_T = \xi, \end{cases} \tag{1.1}$$

where T is a fixed positive number, W_t is the standard d-dimensional Brownian motion defined on a complete, filtered probability space $(\Omega, \mathcal{F}, \mathbb{P}, \{\mathcal{F}_t\}_{0\le t\le T})$, $f(t, y_t, z_t)$ is an adapted stochastic process with respect to $\{\mathcal{F}_t\}$ $(0 \le t \le T)$ for each (y_t, z_t), and ξ is an $\{\mathcal{F}_T\}$ measurable random variable. The existence and uniqueness of the solution of the BSDE (1.1) were proved by Pardoux and Peng in [20]. Since then, BSDEs and their solutions have been extensively studied. In [22], Peng obtained a direct relation between forward-backward stochastic differential equations and partial differential equations and then, in [21], he also derived a maximum principle for stochastic control problems. Many important properties of BSDEs and their applications in finance were studied by Karoui et al. in [8].

* Received January 9, 2012 / Revised version received October 30, 2012 / Accepted December 25, 2012 / Published online May 6, 2013 /

Because analytical solutions of BSDEs are often very difficult to obtain, approximate numerical solutions of BSDEs become highly desired in practical applications. There are mainly two types of numerical methods for BSDEs. One is based on the relation between the forward-backward stochastic differential equations (FBSDEs) and corresponding parabolic partial differential equation (PDEs) [13, 14, 22]; the other is directly based on BSDEs or FBSDEs [2, 3, 6, 7, 10, 12, 23, 27, 28, 30, 31]. Zhao et al. proposed a θ-scheme for BSDEs in [28]; in [29], it was extended to a generalized θ-scheme. In [31], a stable multi-step scheme was proposed which is a highly accurate numerical method for BSDEs. Note that for the second type of numerical methods, approximating spatial derivatives at different time-space points for the case of PDEs is converted to approximating conditional mathematical expectations with Gaussian kernels centered at different time-space points.

It should be noted that the BSDEs used in practice usually involve a multi-dimensional Brownian motion, such as the option pricing problem with multiple underlying assets. Existing numerical methods for BSDEs can be theoretically extended to the multi-dimensional cases; however, the computational cost may be unaffordable due to the so-called *curse of dimensionality*. The most popular approach to solving multi-dimensional BSDEs is the Monte Carlo method [3, 28] that is very easy to implement. However, the convergence rate is typically very slow, although having a mild dependence on the dimensionality. Thus, an accurate and efficient numerical method for solving multi-dimensional BSDEs is highly desired in the BSDE community.

In this paper, we extend the multi-step method in [31] using the sparse-grid method for solving multi-dimensional BSDEs. As discussed in [31], the target BSDE (1.1) is discretized by the multi-step scheme in the time direction. In the spatial domain, a quadrature rule is needed to approximate all the conditional mathematical expectations (multi-dimensional integrals) and an interpolation scheme is also needed to evaluate the integrands of the expectations at non-grid quadrature points. The sparse-grid method is highly suitable for the multi-step scheme because it has been demonstrated to be effective and efficient in dealing with multi-dimensional interpolation and quadrature [1, 4, 9, 11, 16–19]. Sparse-grid interpolantion (or quadrature rule) resulting from the Smolyak algorithm depends weakly on dimensionality so the computational expense can be significantly reduced; however, the accuracy can be preserved up to a logarithmic factor compared with tensor-product interpolantion (or quadrature rule). On the other hand, the multi-step method is also highly suitable for the sparse-grid method because no spatial derivatives are involved in the multi-step scheme and the solution can be obtained without solving a linear system. In comparison, the sparse-grid method can be potentially used together with finite difference or finite element method to solve the associated parabolic PDE instead solving the BSDE directly; in this case, spatial derivatives need to be discretized on sparse grids such that the resulting linear system may have stability or conditioning issues and a CFL condition needs to be satisfied for solving the time-dependent problem. In [24, 25], a spectral sparse-grid method was proposed for elliptic problems, which does not have severe stability or conditioning issues. However, in this paper, those issues on linear systems are completely avoided in our method and the CFL condition is not needed either. In addition, the sparse-grid method is also suitable for the θ-scheme in [28] and the generalized θ-scheme in [29]; we focus on the multi-step method because it is more accurate than the other two schemes in the time direction.

The main contributions in this paper are as follows:

- propose a fully-discrete scheme with the sparse-grid method for multi-dimensional BSDEs;

- rigorously analyze the error of the proposed scheme for a particular type of BSDEs.

As what is discussed above, based on the semi-discrete scheme obtained by the multi-step scheme, a quadrature rule and an interpolation scheme are needed to derive a fully-discrete scheme. Several types of sparse-grid quadrature rules and interpolation schemes can be chosen to achieve our objective. For the quadrature rule, the sparse-grid Clenshaw-Curtis (SG-CC) rule can be used within a local truncated domain due to its nested structure; the sparse-grid Gauss-Hermite (SG-GH) rule is preferable because the expectations have symmetric uncorrelated Gaussian kernels. For the interpolation scheme, a straightforward way is to use the Lagrange polynomial interpolation based on Clenshaw-Curtis or Gaussian points as in [17]. However, in this case, the interpolation points are pre-determined so there is no room for adaptivity. Thus, in this paper, we utilize hierarchical sparse-grid (HSG) interpolation with local support, the same idea as in [11, 16], to construct the needed interpolants. Moreover, we also investigate adaptive hierarchical sparse-grid (AHSG) interpolation that can refine the sparse grid locally according to the smoothness of the solution (y_t, z_t). When the generator f is independent of z_t, we rigorously prove the convergence of the proposed scheme based on the SG-GH quadrature and the AHSG interpolation.

The rest of the paper is organized as follows. Some preliminary notions are discussed in Section 2. The semi-discrete scheme based on the multi-step method is given in Section 3 and in Section 4, we present a fully-discrete scheme based on the sparse-grid method for multi-dimensional BSDEs. Error estimates of the proposed scheme are proved in Section 5. Extensive numerical tests and comparisons are given in Section 6; the results are shown to be consistent with the theoretical ones. Finally, some conclusions are given in Section 7.

2. Preliminaries and Problem Definition

Let $\{\Omega, \mathcal{F}, \mathbb{P}, \{\mathcal{F}_t\}_{0\le t\le T}\}$ be a complete, filtered probability space on which a standard d-dimensional Brownian motion W_t is defined, such that $\{\mathcal{F}_t\}_{0\le t\le T}$ is the natural filtration of the Brownian motion W_t and all the $\mathbb{P}$-null sets are augmented to each σ-field $\mathcal{F}_t$. Denote by $|\cdot|$ and $L^2 = L^2_{\mathcal{F}}([0,T];\mathbb{R}^d)$ the standard Euclidean norm in the Euclidean space $\mathbb{R}^m$ or $\mathbb{R}^{m\times d}$ and the set of all $\mathcal{F}_t$-adapted and mean-square-integrable processes valued in $\mathbb{R}^d$, respectively.

A process $(y_t, z_t) : [0,T]\times\Omega \to \mathbb{R}^m \times \mathbb{R}^{m\times d}$ is called an L^2-adapted solution of the BSDE (1.1) if it is $\{\mathcal{F}_t\}$-adapted, square integrable, and satisfies (2.1) in the sense of

$$y_t = \xi + \int_t^T f(s, y_s, z_s)\, ds - \int_t^T z_s\, dW_s, \quad t \in [0,T), \tag{2.1}$$

where $f : [0,T]\times\mathbb{R}^m\times\mathbb{R}^{m\times d} \to \mathbb{R}^m$ is an adapted stochastic process with respect to $\{\mathcal{F}_t\}$ $(0 \le t \le T)$ for each (y_t, z_t) and the third term on the right-hand side is an Itô-type integral. Under certain reasonable regularity conditions for $f(t, y_t, z_t)$, Pardoux and Peng [20] proved the uniquene solvability of the BSDE (2.1). Some properties and applications of BSDEs are given in [8]. We are interested in the numerical solution of the BSDE (2.1). Without loss of generality, assume the BSDE (2.1) admits a unique L^2-adapted solution (y_t, z_t).

We suppose that the terminal value of y_t is of the form $\varphi(W_T)$. Then, the solution (y_t, z_t) of (2.1) (see [8, 15, 22]) can be represented as

$$y_t = u(t, W_t), \quad z_t = \nabla u(t, W_t), \quad \forall\, t \in [0,T), \tag{2.2}$$

where ∇u denotes the gradient of $u(t,x)$ with respect to the spatial variable x and $u(t,x)$ is the solution of the parabolic PDE

$$\frac{\partial u}{\partial t}+\frac{1}{2}\sum_{i=1}^{d}\frac{\partial^2 u}{\partial x_i^2}+f(t,u,\nabla u)=0, \tag{2.3}$$

with terminal condition being $u(T,x)=\varphi(x)$.

It is well known that when the functions f and φ are bounded and smooth with bounded derivatives, the PDE (2.3) has a unique solution $u(t,x)$ which is also bounded and smooth with bounded derivatives. Consequently, the BSDE (2.1) with $\xi=\varphi(W_T)$ has a unique solution (y_t,z_t) which takes the form (2.2). Because of the equivalence of the BSDE (2.1) and PDE (2.3), it may be feasible to solve the BSDE by solving the equivalent PDE with, e.g., a finite difference method or finite element method. However, when dealing with multi-dimensional problems, it becomes very difficult to use such classical numerical methods.

Now we introduce some notations which will be used in the sequel. Let $\mathcal{F}_s^{t,x}(t\le s\le T)$ be a σ-field generated by the Brownian motion $\{x+W_r-W_t,\ t\le r\le s\}$ starting from the time-space point (t,x) and set $\mathcal{F}^{t,x}=\mathcal{F}_T^{t,x}$. Denote by $\mathbb{E}[X]$ the mathematical expectation of the random variable X and by $\mathbb{E}_s^{t,x}[X]$ the conditional mathematical expectation of the random variable X under the σ-field $\mathcal{F}_s^{t,x}(t\le s\le T)$, i.e., $\mathbb{E}_s^{t,x}[X]=\mathbb{E}[X|\mathcal{F}_s^{t,x}]$. When $s=t$, we simply use $\mathbb{E}_t^x[X]$ to denote $\mathbb{E}[X|\mathcal{F}_t^{t,x}]$. Let $x\in\mathbb{R}^d$ mean $x=(x^1,x^2,\cdots,x^d)^\top$ with $x^i\in\mathbb{R}$ $(i=1,\dots,d)$; $(\cdot)^\top$ denotes the transpose operator for a vector or matrix.

3. Multi-Step Semi-Discrete Scheme

In this section, we briefly review the multi-step method proposed by Zhao et al. in [31] which is used in this paper to discretize the target BSDE (2.1) in the time direction.

Let N be a positive integer and consider a uniform partition of the time interval $[0,T]$

$$0=t_0<t_1<\cdots<t_N=T, \tag{3.1}$$

with $t_i=t_0+i\Delta t$ $(i=0,1,\cdots,N)$ and the time step $\Delta t=\frac{T}{N}$. Suppose the BSDE (2.1) involves a standard d-dimensional Brownian motion $W_t=(W_t^1,W_t^2,\cdots,W_t^d)^\top$ with W_t^i $(i=1,2,\cdots,d)$ being independent standard one-dimensional Brownian motions. Let $(y_t,z_t):[0,T]\times\Omega\to\mathbb{R}^m\times\mathbb{R}^{m\times d}$ be the solution of the BSDE (2.1), where $y_t=(y_t^1,y_t^2,\dots,y_t^m)^\top$ and $z_t=(z_t^{i,j})_{m\times d}$. For two given positive integers k and K_y satisfying $1\leqslant k\leqslant K_y\leqslant N$, it is easily shown that

$$y_{t_n}=y_{t_{n+k}}+\int_{t_n}^{t_{n+k}}f(s,y_s,z_s)\,ds-\int_{t_n}^{t_{n+k}}z_s\,dW_s, \tag{3.2}$$

where f is a vector function of dimension m.

We now take the conditional mathematical expectation $\mathbb{E}_{t_n}^x[\cdot]$ of both sides of (3.2) and approximate the integral in (3.2) by a multi-step scheme. Choosing Lagrange interpolating polynomials $p_{K_y}^{t_n,x}(s)=(p_{K_y}^{t_n,x,1}(s),\cdots,p_{K_y}^{t_n,x,m}(s))^\top$ based on the support points, $(t_{n+i},\ \mathbb{E}_{t_n}^x[f(t_{n+i},\ y_{t_{n+i}},z_{t_{n+i}})])$ $(i=0,1,\dots,K_y)$, to approximate the integrands $\mathbb{E}_{t_n}^x[f(s,y_s,z_s)]$ on $t_n\le s\le t_{n+k}$, we obtain a reference equation for solving y_{t_n} as

$$y_{t_n}=\mathbb{E}_{t_n}^x[y_{t_{n+k}}]+k\Delta t\sum_{i=0}^{K_y}b_{K_y,i}^k\mathbb{E}_{t_n}^x[f(t_{n+i},y_{t_{n+i}},z_{t_{n+i}})]+R_y^n, \tag{3.3}$$

where $R_y^n = (R_y^{n,1}, R_y^{n,2}, \cdots, R_y^{n,m})^\top$, i.e.,

$$R_y^n = \int_{t_n}^{t_{n+k}} \{\mathbb{E}_{t_n}^x[f(s, y_s, z_s)] - p_{K_y}^{t_n,x}(s)\}\, ds, \tag{3.4}$$

and the coefficient $b_{K_y,i}^k$ for $i = 1, \ldots, K_y$ is defined by

$$b_{K_y,i}^k = \frac{1}{k} \int_0^k \prod_{\substack{j=0 \\ j\neq i}}^{K_y} \left(\frac{s-j}{i-j}\right) ds. \tag{3.5}$$

Next, let us turn to the derivation of a reference equation for solving z_{t_n}. Let $\Delta W_s = (W_s^1 - W_{t_n}^1, \cdots, W_s^d - W_{t_n}^d)^\top$ for $s \geq t_n$; then ΔW_s is an increment of the d-dimensional standard Brownian motion with mean zero and a diagonal covariance matrix $(s - t_n)I_d$. Let l and K_z denote two positive integers satisfying $1 \leq l \leq K_z \leqslant N$. Multipling both sides of (3.2) (with k replaced by l) by $\Delta W_{t_{n+l}}$ and then taking the conditional mathematical expectation $\mathbb{E}_{t_n}^x[\cdot]$ of both sides of the derived equation, we are led to the Itô isometry formula

$$0 = \mathbb{E}_{t_n}^x[y_{t_{n+l}} \Delta W_{t_{n+l}}^\top] + \int_{t_n}^{t_{n+l}} \mathbb{E}_{t_n}^x[f(s, y_s, z_s)\Delta W_s^\top]\, ds - \int_{t_n}^{t_{n+l}} \mathbb{E}_{t_n}^x[z_s]\, ds. \tag{3.6}$$

Using the relation (2.2) between the solution (y_t, z_t) of (2.1) and the solution u of (2.3), it is easy to verify that $\mathbb{E}_{t_n}^x[y_{t_{n+l}} \Delta W_{t_{n+l}}^\top] = l\Delta t \mathbb{E}_{t_n}^x[z_{t_{n+l}}]$ (see [31] for details). Again, similar to the way we have obtained the reference equation (3.3), we obtain the following reference equation for solving z_{t_n}:

$$\begin{aligned} 0 = {} & \mathbb{E}_{t_n}^x[z_{t_{n+l}}] + \sum_{i=0}^{K_z} b_{K_z,i}^l \mathbb{E}_{t_n}^x[f(t_{n+i}, y_{t_{n+i}}, z_{t_{n+i}})\Delta W_{t_{n+i}}^\top] \\ & - \sum_{i=0}^{K_z} b_{K_z,i}^l \mathbb{E}_{t_n}^x[z_{t_{n+i}}] + \frac{1}{l\Delta t} R_z^n, \end{aligned} \tag{3.7}$$

with $R_z^n = R_{z1}^n + R_{z2}^n$ of size $m \times d$, where

$$\begin{aligned} R_{z1}^n &= \int_{t_n}^{t_{n+l}} \left\{ \mathbb{E}_{t_n}^x[f(s, y_s, z_s)\Delta W_s^\top] - \sum_{i=0}^{K_z} b_{K_z,i}^l \mathbb{E}_{t_n}^x[f(t_{n+i}, y_{t_{n+i}}, z_{t_{n+i}})\Delta W_{t_{n+i}}^\top] \right\} ds, \\ R_{z2}^n &= -\int_{t_n}^{t_{n+l}} \left\{ \mathbb{E}_{t_n}^x[z_s] - \sum_{i=0}^{K_z} b_{K_z,i}^l \mathbb{E}_{t_n}^x[z_{t_{n+i}}] \right\} ds. \end{aligned}$$

The equations (3.3) and (3.7) are the two reference equations for the BSDE (2.1).

Let $y^n = (y^{1,n}, y^{2,n}, \cdots, y^{m,n})^\top$, $z^n = (z^{i,j,n})_{m\times d}$, and $K_{yz} = \max\{K_y, K_z\}$. We regard y^n as an approximation to y_t and z^n as an approximation to z_t at the time t_n, respectively. Note that $\Delta W_{t_n}^\top = 0$ in (3.7). Combining the two reference equations (3.3) and (3.7), the semi-discrete scheme for solving the BSDE (2.1) is defined as follows: given random variables y^{N-i} and z^{N-i}, $i = 0, 1, \cdots, K_{yz} - 1$, find random variables y^n and z^n $(n = N - K_{yz}, \ldots, 0)$, such that

$$\begin{aligned} y^n &= \mathbb{E}_{t_n}^x[y^{n+k}] + k\Delta t \sum_{i=0}^{K_y} b_{K_y,i}^k \mathbb{E}_{t_n}^x[f(t_{n+i}, y^{n+i}, z^{n+i})], \\ 0 &= \mathbb{E}_{t_n}^x[z^{n+l}] + \sum_{i=1}^{K_z} b_{K_z,i}^l \mathbb{E}_{t_n}^x[f(t_{n+i}, y^{n+i}, z^{n+i})\Delta W_{t_{n+i}}^\top] - \sum_{i=0}^{K_z} b_{K_z,i}^l \mathbb{E}_{t_n}^x[z^{n+i}]. \end{aligned} \tag{3.8}$$

Remark 3.1. In the reference equations (3.3) and (3.7), we have four positive integers k, K_y, l and K_z. They can be any positive integers satisfying $1 \leqslant k \leqslant K_y \leq N$ and $1 \leqslant l \leqslant K_z \leqslant N$. However, to guarantee the stability of the scheme (3.8), they should be selected properly. According to [31], if we set $k = K_y$, the scheme (3.8) for solving y^n is stable for $K_y = 1, 2, 3, 4, 5, 6, 7, 9$. For the scheme (3.8) for solving z^n, l can only be 1 for $K_z = 1, 2, 3$. So, hereafter, we set $K_y = k$ and $l = 1$. In addition, we can see that several conditional mathematical expectations are involved in (3.8) instead of spatial derivatives as in PDEs. As discussed in [31], the stability of a fully discrete scheme based on (3.8) is not affected by a CFL condition.

In order to solve the BSDE (2.1) using the multi-step scheme (3.8), we need to discretize the spatial domain, i.e., the d-dimensional Euclidean space $\mathbb{R}^d$, and find an approach to approximate all mathematical expectations in (3.8) in the discretized space domain, especially in the multi-dimensional case. This is discussed in the following section.

4. Sparse-Grid Fully-Discrete Scheme

4.1. Properties of the time-space domain

We assume a uniform grid for the interval $[0, T]$ defined by (3.1), which is denoted by $\mathcal{T}$, and focus on the discretization of the spatial domain $\mathbb{R}^d$. Taking y^{n+k} as an example, the conditional mathematical expectation $\mathbb{E}^x_{t_n}[y^{n+k}]$ in (3.8) is defined by

$$\mathbb{E}^x_{t_n}[y^{n+k}] = \frac{1}{(2k\pi\Delta t)^{d/2}} \int_{\mathbb{R}^d} y^{n+k}(v) \exp\left[-\frac{(v-x)^\top (v-x)}{2k\Delta t}\right] dv \tag{4.1}$$

with mean x and covariance matrix $k\Delta t I$ where I is the $d \times d$ identity matrix. The expectation defined over the unbounded domain $\mathbb{R}^d$ is always estimated by some numerical quadrature rules within a truncated bounded domain. Moreover, all the components of the d-dimensional Brownian motion are mutually independent, the expectation $\mathbb{E}^x_{t_n}[\cdot]$ can be estimated within a local bounded symmetric hypercube $[x^1 - r, x^1 + r] \times \cdots \times [x^d - r, x^d + r]$ centered at $x = (x^1, \ldots, x^d)$ with an identical local radius r in all spatial directions. Under the above conditions, according to the semi-discrete scheme (3.8), if we want to obtain the numerical solution of the BSDE (2.1) at the point $(x, t_0) = ((0, \ldots, 0), 0)$ based on the time grid $\mathcal{T}$, we need to solve the equation within the domain $[-r, r]^d$ on time level t_1. Recursively, the needed spatial domain on time level t_n $(n = 0, 1, \ldots, N)$ is $[-nr, nr]^d$. The whole bounded time-space domain is a conically shaped region with vertex at the point (x, t_0) and a base of radius Nr at time level t_N. An illustration of the bounded domain for $d = 1$ is shown in Fig. 4.1.

Note that the magnitude of the local radius r is determined by the numerical quadrature rule used to approximate the integral $\mathbb{E}^x_{t_n}[\cdot]$. For instance, for the Gauss-Hermite quadrature rule used in [30, 31], the local radius is determined by the maximum magnitude of the used Gaussian points. If M Gaussian points $\{\eta_i\}_{i=1}^M$ are involved, the local radius r for computing $\mathbb{E}^x_{t_n}[y^{n+k}]$ in (3.8) is defined by $r = \sqrt{2k\Delta t} \max\limits_{i=1,\ldots,M} |\eta_i|$.

After determining the bounded spatial domain, we need to construct a spatial grid on each time level. A straightforward way is to define the set of all needed quadrature points as a spatial grid on each time level. However, in this case, the number of grid points will increase geometrically rather than arithmetically with the number of time steps N. Thus, in [30,31], a uniform spatial grid is used in each spatial direction and the integrands of $\mathbb{E}^x_{t_n}[\cdot]$ are

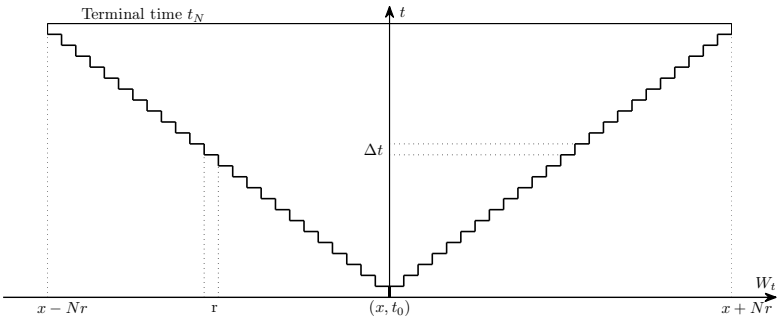

Fig. 4.1. The truncated time-space domain in one-dimensional case($d = 1$).

evaluated at non-grid quadrature points using interpolating polynomials. Combining the Gauss-Hermite quadrature rule and polynomial interpolation, the multi-step scheme (3.8) can be discretized accurately in the bounded spatial domain on each time level. However, in the multi-dimensional case ($d > 1$), the computational cost of the tensor-product Gauss-Hermite rule and the interpolating polynomials based on a tensor-product spatial grid will increase exponentially with the dimension d, i.e., the *curse of dimensionality*. Thus, in the next section we use sparse-grid methods to reduce the complexity for constructing the multivariate interpolating polynomials and for estimating the needed conditional mathematical expectations.

4.2. Smolyak algorithm

The Smolyak algorithm provides an approach to construct multivariate interpolating polynomials based on a small number of points in a multi-dimensional space. In the context of Smolyak method, univariate interpolation formulae are extended to the multivariate case by using tensor products in a special way. This algorithm provides a linear combination of tensor-product interpolants chosen in such a way that the needed number of interpolation points can be reduced significantly but preserve nearly the same accuracy as the full tensor-product interpolation.

Let us consider a smooth function $f : [-1, 1]^d \to \mathbb{R}$. In the one-dimensional case ($d = 1$), the interpolation formula is

$$\mathcal{U}^i(f) = \sum_{j=1}^{m_i} f(x_j^i) \cdot a_j^i(x), \tag{4.2}$$

where $i \in \mathbb{N}$, $x_j^i (j = 1, \ldots, m_i)$ are the interpolation points, and $a_j^i(x)(j = 1, \ldots, m_i)$ are basis functions. In the multi-dimensional case ($d > 1$), the tensor-product interpolatant is

$$\left(\mathcal{U}^{i_1} \otimes \cdots \otimes \mathcal{U}^{i_d}\right)(f) = \sum_{j_1=1}^{m_{i_1}} \cdots \sum_{j_d=1}^{m_{i_d}} f\left(x_{j_1}^{i_1}, \cdots, x_{j_d}^{i_d}\right)\left(a_{j_1}^{i_1} \otimes \cdots \otimes a_{j_d}^{i_d}\right). \tag{4.3}$$

Clearly, if we put identical high resolution in each direction, i.e., $m_1 = \cdots = m_d$, then the above formula needs $\Pi_{i=1}^d m_i$ function values, which is computationally expensive when d is large. Thus the Smolyak interpolant [1,11] is a linear combination of a series of tensor-product interpolants, each of which is defined on a coarse grid with different resolutions in different

dimensions, i.e.,

$$\mathcal{A}^{q,d}(f) = \sum_{q-d+1 \leqslant |\mathbf{i}| \leqslant q} (-1)^{q-|\mathbf{i}|} \binom{d-1}{q-|\mathbf{i}|} \left(\mathcal{U}^{i_1} \otimes \cdots \otimes \mathcal{U}^{i_d}\right)(f), \tag{4.4}$$

where $q \geq d$, the multi-index $\mathbf{i} = (i_1, \ldots, i_d)$ and $|\mathbf{i}| = i_1 + \cdots + i_d$. Here, $i_k (k = 1, \ldots, d)$ is the level of the tensor-product interpolant $\mathcal{U}^{i_1} \otimes \cdots \otimes \mathcal{U}^{i_d}$ along the kth direction. The Smolyak algorithm builds the interpolant by adding a combination of all tensor-product interpolants satisfying $q - d + 1 \leq |\mathbf{i}| \leq q$. The structure of the algorithm becomes clearer when one considers the incremental interpolant, Δ^i given in [1, 11]

$$\mathcal{U}^0(f) = 0, \quad \Delta^i = \mathcal{U}^i(f) - \mathcal{U}^{i-1}(f). \tag{4.5}$$

The Smolyak interpolant (4.4) is then equivalent to

$$\mathcal{A}^{q,d}(f) = \sum_{|\mathbf{i}| \leqslant q} (\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_d}) = \mathcal{A}^{q-1,d}(f) + \sum_{|\mathbf{i}|=q} (\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_d})(f). \tag{4.6}$$

According to (4.4), to compute $\mathcal{A}^{q,d}(f)$, one only needs function values at the "sparse grid"

$$\mathcal{H}^{q,d} = \bigcup_{q-d+1 \leqslant |\mathbf{i}| \leqslant q} (\chi^{i_1} \times \cdots \times \chi^{i_d}), \tag{4.7}$$

where χ^i denotes the set of interpolation points used by $\mathcal{U}^i$. According to (4.6), to extend the Smolyak interpolant $\mathcal{A}^{q,d}(f)$ from level $q-1$ to q, one only needs to evaluate the function at the incremental grid $\Delta\mathcal{H}^{q,d}$ defined by

$$\Delta\mathcal{H}^{q,d} = \bigcup_{|\mathbf{i}|=q} (\Delta\chi^{i_1} \times \cdots \times \Delta\chi^{i_d}), \tag{4.8}$$

where $\Delta\chi^{i_j} = \chi^{i_j} \setminus \chi^{i_j-1}$, $j = 1, \ldots, d$.

By integrating the interpolant $\mathcal{A}^{q,d}(f)$ over the interval $[-1, 1]^d$, a sparse-grid quadrature rule is obtained as

$$\mathcal{Q}^{q,d}(f) = \sum_{i=1}^{N_s} \omega_i f(x_i^1, \ldots, x_i^d), \tag{4.9}$$

where N_s is the number of points on the sparse grid $\mathcal{H}^{q,d}$ and the weight ω_i is the integration of the basis functions in (4.4) associated with the ith grid point $x_i = (x_i^1, \ldots, x_i^d)$. See [9] for details about the computation of the weights.

Our goal of using the sparse-grid method is to estimate the conditional mathematical expectations in (3.8) at all spatial grid points on each time level. An interpolating polynomial is also needed to evaluate the integrand at non-grid points. Thus, in the multi-dimensional case, we need one sparse-grid quadrature rule used to estimate the multi-dimensional integrals and one sparse-grid interpolation formula to construct approximations of y^n and z^n on time level t_n.

4.3. Choice of sparse-grid quadrature rule

First, we consider the choice of the needed sparse-grid quadrature rule. It is suggested to choose the quadrature points in a nested fashion to obtain many recurring points with increasing

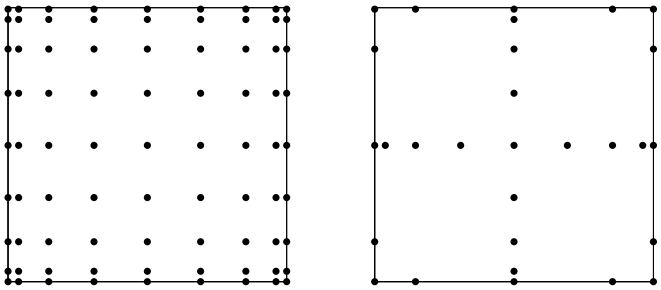

Fig. 4.2. For a two dimensional space ($d = 2$) and maximum level $q = 4$, we plot the full tensor-product grid having 81 points(left) and the sparse grid $\mathcal{H}^{4,2}$ based on SG-CC rule having only 29 points(right).

level q. One such choice is the sparse-grid Clenshaw-Curtis (SG-CC) rule with grid points at the extrema of the Chebyshev polynomials [1,17,18]. The grid points in this case are given by

$$m_i = \begin{cases} 1, & \text{if } i = 1, \\ 2^{i-1} + 1, & \text{if } i > 1, \end{cases} \tag{4.10}$$

$$x_j^i = \begin{cases} 0, & \text{for } j = 1, \text{ if } m_i = 1, \\ -\cos\left(\dfrac{\pi(j-1)}{m_i - 1}\right), & \text{for } j = 1, \ldots, m_i, \text{ if } m_i > 1. \end{cases} \tag{4.11}$$

In this way, the one-dimensional grid is fully nested, i.e. $\chi^i \subset \chi^{i+1}$, and thereby the resulting sparse grid, i.e., $\mathcal{H}^{q,d} \subset \mathcal{H}^{q+1,d}$. Then, if we estimated the expectation $\mathbb{E}_{t_n}^x[\cdot]$ within a truncated domain bounded by a pre-selected local radius r, the Clenshaw-Curtis rule is a good choice due to its nested structure. Fig. 4.2 shows, as an example, the sparse grid $\mathcal{H}^{4,2}$ and the corresponding full tensor-product grid based on Chebshev points. It is easy to see that the sparse grid has many fewer points than the full tensor-product grid. On the other hand, because the expectations in (3.8) have symmetric uncorrelated Gaussian kernels and the Gauss-Hermite quadrature rule was used in one-dimensional cases in [30,31], the sparse-grid Gauss-Hermite(SG-GH) rule based on Hermite polynomials is preferable for solving BSDE (2.1). In this case, the number of grid points m_i is defined to be $m_i = 2^i - 1$ and the grid points on level i is the roots of the Hermite polynomial of degree m_i, i.e.,

$$H_{m_i}(x) = (-1)^{m_i} e^{x^2} \frac{d^{m_i}}{dx^{m_i}}(e^{-x^2}). \tag{4.12}$$

Fig. 4.3 shows, as an example, the sparse grid $\mathcal{H}^{3,2}$ based on the SG-GH rule and the corresponding full tensor-product grid based on Hermite points. The full tensor-product grid has 81 points while the sparse grid has only 22 points. Note that the SG-GH rule is not nested but has higher accuracy. Thus, if we compare the SG-CC rule and the CG-GH rule, the latter will have higher accuracy but also a greater number of points. When measuring efficiency, we really need to balance the cost in quadrature points against the accuracy.

4.4. Choice of sparse-grid interpolation scheme

Here, we consider the choice of the sparse-grid points for constructing the interpolating polynomials for y^n and z^n on time level t_n for $n = 1, \ldots, N$.

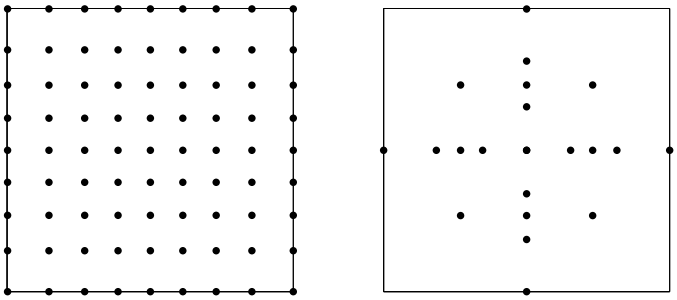

Fig. 4.3. For a two dimensional space ($d = 2$) and maximum level $q = 3$, we plot the full tensor-product grid having 81 points(left) and the sparse grid $\mathcal{H}^{3,2}$ based on the SG-GH rule having only 22 points(right).

4.4.1. Hierarchical sparse-grid interpolant

It is noted that the Lagrange interpolating polynomials based on the Chebyshev points used by the SG-CC rule can be directly employed to construct the needed interpolating polynomials. However, because the grid points are pre-determined as in (4.11), it is not suitable if we want to apply adaptivity. Thus, we propose to use sparse grids based on Newton-Cotes points, i.e., equidistant points. By this, it is easy to refine the grids locally. Moreover, due to the Runge's phenomenon caused by Lagrange interpolating polynomials on uniform grids, we use instead the piecewise linear hat function as the basis function a_j^i in the univariate interpolant (4.2) [4,16]. The piecewise linear function has local support in contrast to the global support of the Lagrange basis function. The grid points on [-1,1] in this case are given by

$$m_i = \begin{cases} 1, & \text{if } i = 1, \\ 2^{i-1} + 1, & \text{if } i > 1, \end{cases} \tag{4.13}$$

$$x_j^i = \begin{cases} 0, & \text{for } j = 1, \text{ if } m_i = 1, \\ \dfrac{2(j-1)}{m_i - 1} - 1, & \text{for } j = 1, \ldots, m_i, \text{ if } m_i > 1. \end{cases} \tag{4.14}$$

For a general interval $[a, b]$, the grid points are simple translations and scalings of (4.14). The nodal basis function a_j^i with local support $[x_j^i - 2^{1-i}, x_j^i + 2^{1-i}]$ is defined as follows. For $i = 1$, $a_1^1 = 1$; for $i > 1$ and $j = 1, \ldots, m_i$,

$$a_j^i = \begin{cases} 1 - \dfrac{m_i - 1}{2} \cdot |x - x_j^i|, & \text{if } |x - x_j^i| < \dfrac{2}{m_i - 1}, \\ 0, & \text{otherwise .} \end{cases} \tag{4.15}$$

Then the multi-linear basis functions used in the tensor-product interpolant (4.3) are defined by

$$a_{\mathbf{j}}^{\mathbf{i}} = a_{j_1}^{i_1} \otimes \cdots \otimes a_{j_d}^{i_d} = \prod_{k=1}^{d} a_{j_k}^{i_k}. \tag{4.16}$$

Substituting into $\mathcal{A}^{q,d}(f)$, the equation (4.4) can be rewritten as

$$\mathcal{A}^{q,d}(f) = \sum_{q-d+1 \leqslant |\mathbf{i}| \leqslant q} \sum_{\mathbf{j}} (-1)^{q-|\mathbf{i}|} \binom{d-1}{q-|\mathbf{i}|} \cdot f(x_{j_1}^{i_1}, \ldots, x_{j_d}^{i_d}) \cdot a_{\mathbf{j}}^{\mathbf{i}}. \tag{4.17}$$

Note that the sparse-grid interpolant (4.17) does not give much information about the smoothness of the function f so that it is not appropriate for an adaptive implementation.

Next, let us consider the hierarchical sparse-grid (HSG) interpolant in (4.6) based on the multi-linear basis functions $a_{\mathbf{j}}^{\mathbf{i}}$. Taking advantage of the nested structure of the Newton-Cotes abscissas, i.e. $\chi^{i-1} \subset \chi^i$, the hierarchical basis and the hierarchical surpluses can be derived for the interpolant (4.6). Clearly, $\Delta\chi^i = \chi^i \backslash \chi^{i-1}$ has $m_\Delta^i = m^i - m^{i-1}$ points due to $\chi^{i-1} \subset \chi^i$. By consecutively numbering the points in $\Delta\chi^i$, and denoting the jth point of $\Delta\chi^i$ as x_j^i, the incremental interpolant in (4.5) can be represented by (see [11, 16] for details)

$$\Delta^i(f) = \sum_{j=1}^{m_\Delta^i} a_j^i \cdot \left[f(x_j^i) - \mathcal{U}^{i-1}(f)(x_j^i) \right], \tag{4.18}$$

where $\omega_j^i = f(x_j^i) - \mathcal{U}^{i-1}(f)(x_j^i)$ is defined as the one-dimensional hierarchical surplus on level i, which is just the difference between the values of the interpolating polynomials at level i and $i-1$; the set of the basis functions $\{a_j^i, j = 1, \ldots, m_\Delta^i\}$ is defined as the hierarchical basis functions on level i. By (4.18), the HSG interpolant (4.6) can be rewritten as

$$\begin{aligned} \mathcal{A}^{q,d}(f) &= \mathcal{A}^{q-1,d}(f) + \sum_{|\mathbf{i}|=q} (\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_d})(f) \\ &= \mathcal{A}^{q-1,d}(f) + \sum_{\substack{|\mathbf{i}|=q \\ \mathbf{j} \in B_{\mathbf{i}}}} \omega_{\mathbf{j}}^{\mathbf{i}} \cdot a_{\mathbf{i}}^{\mathbf{j}}(x) = \sum_{|\mathbf{i}| \leqslant q} \sum_{\mathbf{j} \in B_{\mathbf{i}}} \omega_{\mathbf{j}}^{\mathbf{i}} \cdot a_{\mathbf{i}}^{\mathbf{j}}(x), \end{aligned} \tag{4.19}$$

where the multi-index set $B_{\mathbf{i}}$ is

$$B_{\mathbf{i}} = \left\{ \mathbf{j} \in \mathbb{N}^d : x_{j_k}^{i_k} \in \Delta\chi^{i_k} \text{ for } j_k = 1, \ldots, m_\Delta^{i_k}, k = 1, \ldots, d \right\}, \tag{4.20}$$

and the surpluses $\omega_{\mathbf{j}}^{\mathbf{i}}$ are

$$\omega_{\mathbf{j}}^{\mathbf{i}} = f(x_{j_1}^{i_1}, \ldots, x_{j_d}^{i^d}) - \mathcal{A}^{q-1,d}(f)(x_{j_1}^{i_1}, \ldots, x_{j_d}^{i^d}). \tag{4.21}$$

As proved in [1, 11], for smooth functions, the hierarchical surpluses tend to zero as the interpolation level tens to infinity. On the other hand, the magnitude of the surplus is a good indicator to show the smoothness of the interpolated function. The bigger the magnitude is, the stronger the underlying discontinuity is. Thus, the hierarchical surplus can be used for error control and implementation of adaptivity.

4.4.2. Adaptive hierarchical sparse-grid interpolant

As discussed above, if the solution of the BSDE (2.1) is not equally smooth with respect to the Brownian motion W_t over the bounded domain shown in Fig. 4.1, an adaptive sparse grid [16] is preferred, which may place more points in the non-smooth region and fewer points in the smooth region. In this method, the hierarchical surplus $\omega_{\mathbf{j}}^{\mathbf{i}}$ is used as the indicator for adaptation. Analogous to the HSG interpolant (4.19), the adaptive hierarchical sparse-grid (AHSG) interpolant can be defined by modifying the multi-index $B_{\mathbf{i}}$ to

$$B_{\mathbf{i}}^{\varepsilon} = \left\{ \mathbf{j} \in B_{\mathbf{i}} : |w_{\mathbf{j}}^{\mathbf{i}}| > \varepsilon \right\}, \tag{4.22}$$

and the AHSG interpolant is defined by

$$\mathcal{A}_{\varepsilon}^{q,d}(f) = \sum_{|\mathbf{i}|\leqslant q} \sum_{\mathbf{j}\in B_{\mathbf{i}}^{\varepsilon}} \omega_{\mathbf{j}}^{\mathbf{i}} \cdot a_{\mathbf{i}}^{\mathbf{j}}(x), \tag{4.23}$$

where the constant ε is called the threshold of the interpolant $\mathcal{A}_{\varepsilon}^{q,d}(f)$. To increase the level of the interpolant (4.23) from level $q-1$ to q, one needs to evaluate the function f at the points

$$\Delta\mathcal{H}_{\varepsilon}^{q,d} = \left\{x_{\mathbf{j}}^{\mathbf{i}} : |\mathbf{i}| = q \text{ and } \mathbf{j} \in B_{\mathbf{i}}^{\varepsilon}\right\}. \tag{4.24}$$

It is easy to see that a q-level adaptive sparse grid, denoted by $\mathcal{H}_{\varepsilon}^{q,d}$, is a subgrid of the corresponding q-level sparse grid $\mathcal{H}^{q,d}$. If $\varepsilon = 0$, the AHSG interpolant (4.23) is equivalent to the HSG interpolant (4.19); if $\varepsilon > 0$, it will adaptively select which points are added to the sparse grid. Subsequently, the sparse-grid points will become concentrated in the non-smooth region. The refinement algorithm of adaptive sparse grids can be found in [16].

4.5. Approximation errors

First, we discuss the errors of the sparse-grid quadrature rules. The errors are considered in the function space

$$F_d^k(D) = \left\{f : D \to \mathbb{R} | \mathcal{D}^{\boldsymbol{\alpha}} f \text{ continuous if } \alpha_i \leqslant k \text{ for all } i\right\}, \tag{4.25}$$

where $D \subset \mathbb{R}^d$, $\boldsymbol{\alpha} = (\alpha_i, \ldots, \alpha_d) \in \mathbb{N}_0^d$, $|\boldsymbol{\alpha}| = \alpha_1 + \cdots + \alpha_d$ and $\mathcal{D}^{\boldsymbol{\alpha}}$ denotes the d-variate partial derivative of order $|\boldsymbol{\alpha}|$, i.e.,

$$\mathcal{D}^{\boldsymbol{\alpha}} f = \frac{\partial^{|\boldsymbol{\alpha}|} f}{\partial x_1^{\alpha_1} \cdots \partial x_d^{\alpha_d}}. \tag{4.26}$$

As discussed in [1, 19], for any function $f \in F_1^k([-1,1])$, the error of the one-dimensional Clenshaw-Curtis rule is given by

$$\left| \int_D f(x)dx - \mathcal{Q}_{CC}^{q,1}(f) \right| \leqslant C N_s^{-k}, \tag{4.27}$$

where N_s is the number of quadrature points and the constant C depends on the upper bound of the k-th derivative of f. For the SG-CC quadrature rule, we have the following lemma given in [1, 19].

Lemma 4.1. *For any function $f \in F_d^k([-1,1]^d)$, the error of the sparse-grid Clenshaw-Curtis rule is*

$$\left| \int_D f(x)dx - \mathcal{Q}_{CC}^{q,d}(f) \right| \leqslant C_{k,d} N_s^{-k} \left(\log(N_s)\right)^{(k+1)(d-1)}, \tag{4.28}$$

where N_s is the number of sparse-grid quadrature points used by $\mathcal{Q}_{CC}^{q,d}(f)$ and the constant $C_{k,d}$ only depends on d and the upper bound of the k-th derivative of f.

Analogously, as discussed in [26], for any function $f \in F_1^k(\mathbb{R})$ whose growth at infinity satisfies the condition in Theorem 1 in [26], the error of the one-dimensional Gauss-Hermite rule is given by

$$\left| \int_{\mathbb{R}} f(x) e^{-x^2} dx - \mathcal{Q}_{GH}^{q,1}(f) \right| \leqslant C N_s^{-k/2}. \tag{4.29}$$

By conducting the same procedure as that in [1, 19], we can obtain the following lemma:

Lemma 4.2. *For any function $f \in F_d^k(\mathbb{R}^d)$, the error of the sparse-grid Gauss-Hermite rule is*

$$\left| \int_{\mathbb{R}^d} f(x) e^{-x^\top x} dx - \mathcal{Q}_{GH}^{q,d}(f) \right| \leqslant C_{k,d} N_s^{-k/2} \left(\log(N_s)\right)^{(k/2+1)(d-1)}, \tag{4.30}$$

where $x = (x_1, \ldots, x_d)$, N_s is the number of the sparse-grid quadrature points used by $\mathcal{Q}_{GH}^{q,d}(f)$ and the constant $C_{k,d}$ depends only on d and the upper bound of the k-th derivative of f.

Next, we discuss the interpolation errors of the HSG and AHSG interpolation. Analogous to the error analysis of the SG-CC quadrature rule, we can obtain the error for the HSG interpolation scheme as follows [11, 16].

Lemma 4.3. *For any function $f \in F_d^2([a,b]^d)$ where $[a,b]^d$ is bounded, the error of the HSG interpolant* (4.19) *in L^∞ norm is given by*

$$\left\| f - \mathcal{A}^{q,d}(f) \right\|_\infty \leqslant C_d N_s^{-2} \left(\log(N_s)\right)^{3(d-1)}, \tag{4.31}$$

where the constant C_d only depends on d and the upper bound of the second derivative of f.

Relying on Lemma 4.3, we can obtain the error estimate of the AHSG interpolant (4.23) in the following lemma [16].

Lemma 4.4. *For any function $f \in F_d^2([a,b]^d)$ where $[a,b]^d$ is bounded, the error of the AHSG interpolant* (4.23) *with the threshold ε in L^∞ norm is given by*

$$\left\| f - \mathcal{A}_\varepsilon^{q,d}(f) \right\|_\infty \leqslant C_d N_s^{-2} \left(\log(N_s)\right)^{3(d-1)} + \varepsilon N_{m,\varepsilon}, \tag{4.32}$$

where N_s is the number of points of $\mathcal{H}^{q,d}$, $N_{m,\varepsilon}$ is the number of the missing points of the adaptive sparse grid $\mathcal{H}_\varepsilon^{q,d}$ under the threshold ε, the constant C_d depends on d and the upper bound of the second derivative of f.

In fact, for a fixed threshold ε, a q-level HSG interpolant (4.19) can be written as a sum of two terms, i.e.

$$\mathcal{A}^{q,d}(f) = \mathcal{A}_\varepsilon^{q,d}(f) + \sum_{|\mathbf{i}| \leqslant q} \sum_{\mathbf{j} \in B_\mathbf{i} \setminus B_\mathbf{i}^\varepsilon} \omega_\mathbf{j}^\mathbf{i} \cdot a_\mathbf{i}^\mathbf{j}(x), \tag{4.33}$$

where the second term involves all the missing points whose surpluses are below the threshold ε. Since for any piecewise multi-linear basis function $a_\mathbf{j}^\mathbf{i}(x)$, $\|a_\mathbf{j}^\mathbf{i}\|_\infty = 1$, for any function $f \in F_d^2([a,b]^d)$, the interpolation error of the AHSG interpolant is given by [3, 16]

$$\begin{aligned}
\left\| f - \mathcal{A}_\varepsilon^{q,d}(f) \right\|_\infty &= \left\| f - \mathcal{A}^{q,d}(f) + \mathcal{A}^{q,d}(f) - \mathcal{A}_\varepsilon^{q,d}(f) \right\|_\infty \\
&\leqslant \left\| f - \mathcal{A}^{q,d}(f) \right\|_\infty + \left\| \mathcal{A}^{q,d}(f) - \mathcal{A}_\varepsilon^{q,d}(f) \right\|_\infty \\
&= \left\| f - \mathcal{A}^{q,d}(f) \right\|_\infty + \left\| \sum_{|\mathbf{i}| \leqslant q} \sum_{\mathbf{j} \in B_\mathbf{i} \setminus B_\mathbf{i}^\varepsilon} \omega_\mathbf{j}^\mathbf{i} \cdot a_\mathbf{i}^\mathbf{j}(x) \right\|_\infty \\
&\leqslant C_d N_s^{-2} \left(\log(N_s)\right)^{3(d-1)} + \varepsilon N_{m,\varepsilon}.
\end{aligned} \tag{4.34}$$

Note that the number of points, denoted by $N_{s,\varepsilon}$, on an adaptive sparse grid is $N_s - N_{m,\varepsilon}$.

4.6. Fully-discrete scheme

The scheme (3.8) is a stable semi-discrete scheme in the time direction. To develop a fully-discrete scheme for solving the BSDE (2.1), an effective discretization in space is also necessary. According to the discussion of the time-space domain in Section 4.1, for a fixed local radius r, we construct an adaptive sparse grid $\mathcal{H}_{n,\varepsilon}^{q_n,d}$ with threshold ε to discretize the spatial domain $[-rn, rn]^d$ on time level t_n. It is clear that $\mathcal{H}_{n,\varepsilon}^{q_n,d} \subset \mathcal{H}_n^{q_n,d}$. Denote by N_s^n and $N_{s,\varepsilon}^n$ as the total number of points in $\mathcal{H}_n^{q_n,d}$ and $\mathcal{H}_{n,\varepsilon}^{q_n,d}$, respectively. $N_{m,\varepsilon}^n$ is the number of missing points of $\mathcal{H}_{n,\varepsilon}^{q_n,d}$, so that $N_s^n = N_{s,\varepsilon}^n + N_{m,\varepsilon}^n$. By consecutively numbering the points in $\mathcal{H}_{n,\varepsilon}^{q_n,d}$, it can be represented as

$$\mathcal{H}_{n,\varepsilon}^{q_n,d} = \{x_i^n, i = 1, \ldots, N_{s,\varepsilon}^n\}.$$

In addition, because the volume of the spatial domain increases along with time, the level q_n should also increases accordingly. If $\varepsilon = 0$, i.e., no adaptivity, the abscissas of $\mathcal{H}_{n,\varepsilon}^{q_n,d}$ can be determined in advance for a fixed q_n; otherwise ($\varepsilon > 0$), the abscissas cannot be predetermined until the solver gets to the time level t_n.

Based on the semi-discrete scheme and sparse-grid methods, we propose the fully-discrete scheme for solving the BSDE (2.1) as follow: given the random variables y_i^{N-l}, $i = 1, \ldots, N_{s,\varepsilon}^{N-l}$ and $l = 0, 1, \cdots, K_{yz} - 1$, find the random variables (y_i^n, z_i^n), $i = 1, \ldots, N_{s,\varepsilon}^n$ and $n = N - K_{yz}, \cdots, 0$, such that

$$y_i^n = \widehat{\mathbb{E}}_{t_n}^{x_i^n}[\widehat{y}^{n+k}] + K_y \Delta t \sum_{j=1}^{K_y} b_{K_y,j}^{K_y} \widehat{\mathbb{E}}_{t_n}^{x_i^n}[f(t_{n+j}, \widehat{y}^{n+j}, \widehat{z}^{n+j})] + K_y \Delta t f(t_n, y_i^n, z_i^n), \tag{4.35a}$$

$$0 = \widehat{\mathbb{E}}_{t_n}^{x_i^n}[\widehat{z}^{n+1}] + \sum_{j=1}^{K_z} b_{K_z,j}^1 \widehat{\mathbb{E}}_{t_n}^{x_i^n}[f(t_{n+j}, \widehat{y}^{n+j}, \widehat{z}^{n+j}) \Delta W_{t_{n+j}}^\top] - \sum_{j=1}^{K_z} b_{K_z,j}^1 \widehat{\mathbb{E}}_{t_n}^{x_i^n}[\widehat{z}^{n+j}] - b_{K_z,0}^1 z_i^n. \tag{4.35b}$$

Here, $\widehat{y}^{n+1}$ and $\widehat{z}^{n+1}$ are the AHSG interpolants on $\mathcal{H}_{n+1,\varepsilon}^{q_{n+1},d}$ defined by

$$\widehat{y}^{n+1} = \mathcal{A}_{n+1,\varepsilon}^{q_{n+1},d}(y^{n+1}) \quad \text{and} \quad \widehat{z}^{n+1} = \mathcal{A}_{n+1,\varepsilon}^{q_{n+1},d}(z^{n+1}), \tag{4.36}$$

and $\widehat{y}^{n+j}$, $\widehat{z}^{n+j}$ are defined for $j = 2, \ldots, K_y$ or K_z in a similar way. $\mathbb{E}_{t_n}^{x_i^n}[y^{n+1}]$ in (3.8) is approximated by $\widehat{\mathbb{E}}_{t_n}^{x_i^n}[\widehat{y}^{n+1}]$ within a local hypercube $[x_i^n - r, x_i^n + r]$ by Clenshaw-Curtis or Gauss-Hermite sparse-grid quadrature rule (4.9), i.e.,

$$\widehat{\mathbb{E}}_{t_n}^{x_i^n}[\widehat{y}^{n+1}] = \mathcal{Q}^{q,d}(\widehat{y}^{n+1}) = \sum_{i=1}^{N_s^Q} \omega_i \widehat{y}^{n+1}(\eta_i), \tag{4.37}$$

where N_s^Q is the number of quadrature points, ω_i, η_i are the weights and quadrature points for $i = 1, \ldots, N_s^Q$. The same number of quadrature points are used for computing the expectation at any time-space point (t_n, x_i^n). Because some quadrature points may not belong to the sparse grid $\mathcal{H}_{n+1,\varepsilon}^{q_{n+1},d}$, we use the interpolating polynomial $\widehat{y}^{n+1}$ to evaluate the integrand y^{n+1} at non-grid points. This discussion is also applied for approximating other expectations in (4.35).

5. Error Estimates

In this section, we carry out error analysis for the fully-discrete scheme (4.35) for solving the BSDE (2.1) with the generator function f being independent of the random variable z_t,

i.e.,

$$y_t = \varphi(W_T) + \int_t^T f(s, y_s)ds - \int_t^T z_s dW_s. \tag{5.1}$$

For simplicity of the analysis, we consider only the case of $m = 1$ and $d > 1$, i.e., there is only one BSDE but $W_t = (W_t^1, \ldots, W_t^d)^\top$ is a vector of d mutually independent Brownian motions. The error estimates obtained in the sequel also hold for a system of BSDEs. In such a simplified case, the reference equations (3.3) and (3.7) with $k = K_y$ and $l = 1$ have the following reduced form:

$$\begin{aligned} y_{t_n}^{x_i^n} &= \mathbb{E}_{t_n}^{x_i^n}[y_{t_{n+K_y}}] + K_y\Delta t \sum_{j=0}^{K_y} b_{K_y,j}^{K_y} \mathbb{E}_{t_n}^{x_i^n}[f(t_{n+j}, y_{t_{n+j}})] + R_y^n, \\ 0 &= \mathbb{E}_{t_n}^{x_i^n}[z_{t_{n+1}}] + \sum_{j=0}^{K_z} b_{K_z,j}^1 \mathbb{E}_{t_n}^{x_i^n}[f(t_{n+j}, y_{t_{n+j}})\Delta W_{t_{n+j}}] - \sum_{j=0}^{K_z} b_{K_z,j}^1 \mathbb{E}_{t_n}^{x_i^n}[z_{t_{n+j}}] + \frac{1}{\Delta t} R_z^n. \end{aligned} \tag{5.2}$$

The corresponding fully-discrete scheme (4.35) becomes:

$$\begin{aligned} y_i^n &= \widehat{\mathbb{E}}_{t_n}^{x_i^n}[\widehat{y}^{n+k}] + K_y\Delta t \sum_{j=1}^{K_y} b_{K_y,j}^{K_y} \widehat{\mathbb{E}}_{t_n}^{x_i^n}[f(t_{n+j}, \widehat{y}^{n+j})] + K_y\Delta t f(t_n, y_i^n), \\ 0 &= \widehat{\mathbb{E}}_{t_n}^{x_i^n}[\widehat{z}^{n+1}] + \sum_{j=1}^{K_z} b_{K_z,j}^1 \widehat{\mathbb{E}}_{t_n}^{x_i^n}[f(t_{n+j}, \widehat{y}^{n+j})\Delta W_{t_{n+j}}^\top] - \sum_{j=0}^{K_z} b_{K_z,j}^1 \widehat{\mathbb{E}}_{t_n}^{x_i^n}[\widehat{z}^{n+j}], \end{aligned} \tag{5.3}$$

for $n = N - K, \ldots, 0$ with $K = \max(K_y, K_z)$. In the following analysis, $\widehat{\mathbb{E}}_{t_n}^{x_i}[\cdot]$ is defined using the sparse-grid Gauss-Hermite rule $\widehat{y}^{n+j}$ is constructed using the AHSG interpolation.

To simplify the presentation, we assume that the following two assumptions hold.

Assumption 5.1. *The positive integers K_y and K_z are chosen such that the roots of the characteristic polynomial*

$$\rho_{K_z}^1(\lambda) = \lambda^{K_z - 1} - \sum_{i=0}^{K_z} b_{K_z,i}^1 \lambda^{K_z - i} \tag{5.4}$$

satisfy the root conditions.

Assumption 5.2. *The functions $f(t, y_t)$ and $\varphi(W_T)$ are bounded and have bounded derivatives of order up to k.*

Next, we present two lemmas as follows.

Lemma 5.1. *Let R_y^n and R_z^n be the local truncation errors defined in the reference equations* (3.3) *and* (3.7)*. Then under Assumption 5.2, we have the local estimates*

$$|R_y^n| \le C(\Delta t)^{K_y+2}, \quad |R_z^n| \le C(\Delta t)^{K_z+2}, \tag{5.5}$$

where $C > 0$ is a generic constant only depending on T, the upper bounds of φ and f and their derivatives.

Lemma 5.2. *Suppose N and K are two non-negative integers with $N \ge K$ and Δt is any positive number. Let $\{\eta_n\}$ be a series satisfying*

$$|\eta_n| \le \beta + \alpha\Delta t \sum_{j=n+1}^{N} |\eta_j|, \qquad n = N - K, N - K - 1, \cdots, 0, \tag{5.6}$$

where α and β are two positive constants. Let $M_0 = \max_{N-K<j\leq N} |\eta_j|$ and $T = N\Delta t$, then

$$|\eta_n| \leq e^{\alpha T}(\beta + \alpha K \Delta t M_0), \qquad n = N-K, N-K-1, \cdots, 0. \tag{5.7}$$

Proofs of the above two lemmas are similar to that of Lemma 3.1 in [30] and Lemma 3 in [31] and thus we omit them here.

Now we give an error estimate of $y_{t_n}^{x_i^n} - y_i^n$ in the following theorem.

Theorem 5.1. *Let $y_{t_n}^{x_i^n}$ and y_i^n be the solution of the BSDE* (5.1) *and the fully-discrete scheme* (5.3)*, respectively. Suppose Assumption 5.2 holds and the initial values satisfy*

$$\max_{\substack{N-K_y\leqslant j\leqslant N \\ i=1,\cdots,N_{s,\varepsilon}^j}} |y_{t_j}^{x_i^j} - y_i^j| = O((\Delta t)^{K_y+1}).$$

Then for sufficiently small time step Δt, we have

$$\begin{aligned}\max_{\substack{0\leqslant j\leqslant N \\ i=1,\cdots,N_s^j}} |y_{t_j}^{x_i^j} - y_i^j| \leqslant \frac{C}{\Delta t}\Bigg\{ & \left(N_s^Q\right)^{-k/2}\left(\log(N_s^Q)\right)^{(k/2+1)(d-1)} \\ & + \max_{i=2,\ldots,N}\left[\left(N_{s,\varepsilon}^i\right)^{-2}\left(\log(N_{s,\varepsilon}^i)\right)^{3(d-1)} + \varepsilon N_{m,\varepsilon}^i\right]\Bigg\} + C(\Delta_t)^{K_y+1},\end{aligned} \tag{5.8}$$

where $C > 0$ is a generic constant only depending on T, upper bounds for the functions φ and f and their derivatives, and the levels of used sparse grids q_n $(n = 2, \ldots, N)$.

Proof. Let $e_i^n = y_{t_n}^{x_i^j} - y_i^n$ for $n = N, N-1, \ldots, 0$. From (5.2) and (5.3), we obtain

$$\begin{aligned} e_i^n &= \mathbb{E}_{t_n}^{x_i^n}[y_{t_{n+K_y}}] - \widehat{\mathbb{E}}_{t_n}^{x_i^n}[\widehat{y}^{n+K_y}] \\ &\quad + K_y\Delta t\sum_{j=0}^{K_y} b_{K_y,j}^{K_y}\left\{\mathbb{E}_{t_n}^{x_i^n}[f(t_{n+j}, y_{t_{n+j}})] - \widehat{\mathbb{E}}_{t_n}^{x_i^n}[f(t_{n+j}, \widehat{y}^{n+j})]\right\} + R_y^n \\ &= I_1 + I_2 + R_y^n, \end{aligned} \tag{5.9}$$

where

$$\begin{aligned} I_1 &= \mathbb{E}_{t_n}^{x_i^n}[y_{t_{n+K_y}}] - \widehat{\mathbb{E}}_{t_n}^{x_i^n}[\widehat{y}^{n+K_y}], \\ I_2 &= K_y\Delta t\sum_{j=0}^{K_y} b_{K_y,j}^{K_y}\left\{\mathbb{E}_{t_n}^{x_i^n}[f(t_{n+j}, y_{t_{n+j}})] - \widehat{\mathbb{E}}_{t_n}^{x_i^n}[f(t_{n+j}, \widehat{y}^{n+j})]\right\}. \end{aligned} \tag{5.10}$$

We rewrite I_1 as

$$I_1 = \mathbb{E}_{t_n}^{x_i^n}[y_{t_{n+K_y}}] - \widehat{\mathbb{E}}_{t_n}^{x_i^n}[y_{t_{n+K_y}}] + \widehat{\mathbb{E}}_{t_n}^{x_i^n}[y_{t_{n+K_y}} - \widehat{y}_{t_{n+K_y}}] + \widehat{\mathbb{E}}_{t_n}^{x_i^n}[\widehat{y}_{t_{n+K_y}} - \widehat{y}^{n+K_y}]. \tag{5.11}$$

Based on Assumption 5.2 and the error of the SG-GH quadrature rule given in Lemma 4.2, we have

$$\left|\mathbb{E}_{t_n}^{x_i^n}[y_{t_{n+K_y}}] - \widehat{\mathbb{E}}_{t_n}^{x_i^n}[y_{t_{n+K_y}}]\right| \leqslant C_{k,d}\left(N_s^Q\right)^{-k/2}\left(\log(N_s^Q)\right)^{(k/2+1)(d-1)}. \tag{5.12}$$

If the AHSG interpolant is used to approximate $y_{t_{n+K_y}}$ with threshold being ε, then based on Lemma 4.4, we have

$$\left|\widehat{\mathbb{E}}_{t_n}^{x_i^n}[y_{t_{n+K_y}} - \widehat{y}_{t_{n+K_y}}]\right| \leqslant C_d\left(N_{s,\varepsilon}^{n+K_y}\right)^{-2}\left(\log(N_{s,\varepsilon}^{n+K_y})\right)^{3(d-1)} + \varepsilon N_{m,\varepsilon}^{n+K_y}, \tag{5.13}$$

and

$$\left|\widehat{\mathbb{E}}_{t_n}^{x_i^n}[\widehat{y}_{t_{n+K_y}} - \widehat{y}^{n+K_y}]\right| \leqslant \max_{i=1,\ldots,N_{s,\varepsilon}^{n+K_y}} |e_i^{n+K_y}|. \tag{5.14}$$

Combining (5.12), (5.13) and (5.14), we have

$$\begin{aligned} |I_1| \leqslant & C_{k,d}\left(N_s^Q\right)^{-k/2}\left(\log(N_s^Q)\right)^{(k/2+1)(d-1)} \\ & + C_d\left(N_{s,\varepsilon}^{n+K_y}\right)^{-2}\left(\log(N_{s,\varepsilon}^{n+K_y})\right)^{3(d-1)} + \varepsilon N_{m,\varepsilon}^{n+K_y} + \max_{i=1,\ldots,N_{s,\varepsilon}^{n+K_y}} |e_i^{n+K_y}|. \end{aligned} \tag{5.15}$$

By a similar procedure, we obtain

$$\begin{aligned} |I_2| \leqslant & LK_yC_{k,d}\Delta t\left(N_s^Q\right)^{-k/2}\left(\log(N_s^Q)\right)^{(k/2+1)(d-1)} \\ & + K_y\Delta t\sum_{j=0}^{K_y} b_{K_y,j}^{K_y}\left[C_d\left(N_{s,\varepsilon}^{n+j}\right)^{-2}\left(\log(N_{s,\varepsilon}^{n+j})\right)^{3(d-1)} + \varepsilon N_{m,\varepsilon}^{n+j} + \max_{i=1,\ldots,N_{s,\varepsilon}^{n+j}} |e_i^{n+j}|\right], \end{aligned} \tag{5.16}$$

where L is the Lipchitz constant. To simplify the presentation, we define some notations in the following derivation:

$$\begin{aligned} & \|e^n\| = \max_{i=1,\ldots,N_{s,\varepsilon}^n} |e_i^n|, \\ & e_{sg1} = C_{k,d}\left(N_s^Q\right)^{-k/2}\left(\log(N_s^Q)\right)^{(k/2+1)(d-1)}, \\ & e_{sg2} = \max_{i=2,\ldots,N}\left[C_d\left(N_{s,\varepsilon}^i\right)^{-2}\left(\log(N_{s,\varepsilon}^i)\right)^{3(d-1)} + \varepsilon N_{m,\varepsilon}^i\right], \\ & \widetilde{R}_y^n = (1 + LK_y\Delta t)(e_{sg1} + e_{sg2}) + R_y^n, \end{aligned} \tag{5.17}$$

where $n = 1, \ldots, N - K_y$. By these notations, substituting (5.15) and (5.16) into (5.9), we get

$$\|e^n\| \leqslant \|e^{n+K_y}\| + LK_y\Delta t\sum_{j=0}^{K_y} b_{K_y,j}^{K_y}\|e^{n+j}\| + |\widetilde{R}_y^n|. \tag{5.18}$$

Let $N_{K_y} = \left[\dfrac{N-n}{K_y}\right]$. For an integer s satisfying $1 \leqslant s \leqslant N_{K_y}$, we similarly have the estimate

$$\|e^{n+(s-1)K_y}\| \leqslant \|e^{n+sK_y}\| + LK_y\Delta t\sum_{j=0}^{K_y} b_{K_y,j}^{K_y}\|e^{n+(s-1)K_y+j}\| + |\widetilde{R}_y^{n+(s-1)K_y}|. \tag{5.19}$$

Now we add up the above inequalities (5.19) over $s = 1, 2, \ldots, N_{K_y}$ and obtain

$$\|e^n\| \leqslant \|e^{n+N_{K_y}K_y}\| + 2LK_y\Delta t\sum_{j=0}^{N_{K_y}K_y} b_{K_y,j}^{K_y}\|e^{n+j}\| + \sum_{j=0}^{N_{K_y}-1} |\widetilde{R}_y^{n+jK_y}|, \tag{5.20}$$

which is equivalent to

$$(1 - 2LK_y\Delta t)\|e^n\| \leqslant \|e^{n+N_{K_y}K_y}\| + 2LK_y\Delta t\sum_{j=1}^{N_{K_y}K_y} b_{K_y,j}^{K_y}\|e^{n+j}\| + \sum_{j=0}^{N_{K_y}-1} |\widetilde{R}_y^{n+jK_y}|. \tag{5.21}$$

Let $D = \frac{1}{1-2LK_y\Delta t}$, $N_1 = \frac{2LK_y}{1-2LK_y\Delta t}$, and

$$M_0 = \max_{\substack{N-K_y\leqslant j\leqslant N \\ i=1,\cdots,N^j_{s,\varepsilon}}} |y^{x_i^j}_{t_j} - y^j_i| + \sum_{j=0}^{N_{K_y}-1} |\widetilde{R}^{n+jK_y}_y|.$$

For sufficiently small time step Δt, D and N_1 are clearly positive and bounded by a positive constant. Then by Lemma 5.2 and (5.21), we obtain the following inequality

$$\|e^n\| \leqslant e^{N_1T}(DM_0 + \Delta tK_yN_1M_0) = e^{N_1T}(D + \Delta tK_yN_1)M_0. \tag{5.22}$$

On the other hand, by the fact in Lemma 5.1 that

$$R^n_y \leqslant C(\Delta t)^{K_y+2}, \tag{5.23}$$

and the given condition $|y^{x_i^j}_{t_j} - y^j_i| \sim O((\Delta t)^{K_y+1})$ in the theorem, we obtain

$$\begin{aligned} M_0 =& \max_{\substack{N-K_y\leqslant j\leqslant N \\ i=1,\cdots,N^j_{s,\varepsilon}}} |y^{x_i^j}_{t_j} - y^j_i| + \sum_{j=0}^{N_{K_y}-1} |\widetilde{R}^{n+jK_y}_y| \\ \leqslant& C(\Delta t)^{K_y+1} + N(1+LK_y\Delta t)(e_{sg1}+e_{sg2}) + \sum_{j=0}^{N_{K_y}-1} R^{n+jK_y}_y \\ \leqslant& \frac{C}{\Delta t}\Bigg\{ \left(N^Q_s\right)^{-k/2}\left(\log(N^Q_s)\right)^{(k/2+1)(d-1)} \\ &+ \max_{i=2,\ldots,N}\left[\left(N^i_{s,\varepsilon}\right)^{-2}\left(\log(N^i_{s,\varepsilon})\right)^{3(d-1)} + \varepsilon N^i_{m,\varepsilon}\right]\Bigg\} + C(\Delta_t)^{K_y+1}. \end{aligned} \tag{5.24}$$

Combining the inequalities (5.22) and (5.24), we immediately get (5.8) and the proof is completed. □

In the following we present an estimation of $z^{x_i^n}_{t_n} - z^n_i$. Because both $z^{x_i^n}_{t_n}$ and z^n_i are vectors of d elements, we measure the error by the L^∞ norm, i.e., $\|z^{x_i^n}_{t_n} - z^n_i\|_\infty$. Note that he error $y^{x_i^n}_n - y^n_i$ of the fully-discrete scheme (5.3) in Theorem 5.1 consists of two parts. One is the time-discretization error of order $(\Delta t)^{K_y+1}$ provided in Theorem 1 in [31]; the other is the space-discretization error caused by sparse-grid approximation. Similarly the error $z^{x_i^n}_{t_n} - z^n_i$ can be bounded combining the proof of Theorem 2 in [31] for the semi-discrete scheme and interpolation error bounds of the spares-grid method given in Section 4.5. Therefore, we only provide a conclusion but omit the proof.

Theorem 5.2. *Let $z^{x_i^n}_{t_n}$ and z^n_i be the solution of the BSDE* (5.1) *and the fully-discrete scheme* (5.3)*, respectively. Suppose Assumption 5.1 and 5.2,the hypotheses of Theorem 5.1 hold, and the initial values satisfy*

$$\max_{\substack{N-K_y\leqslant j\leqslant N \\ i=1,\cdots,N^j_{s,\varepsilon}}} \|z^{x_i^j}_{t_j} - z^j_i\|_\infty = O((\Delta t)^{K_z}).$$

Then for sufficiently small time step Δt, we have

$$\max_{\substack{0\leqslant j\leqslant N\\ i=1,\cdots,N_s^j}} \|z_{t_j}^{x_i^j} - z_i^j\|_\infty \leqslant \frac{C}{\Delta t}\Bigg\{ \left(N_s^Q\right)^{-k/2} \left(\log(N_s^Q)\right)^{(k/2+1)(d-1)} + \max_{i=2,\ldots,N}\left[\left(N_{s,\varepsilon}^i\right)^{-2}\left(\log(N_{s,\varepsilon}^i)\right)^{3(d-1)} + \varepsilon N_{m,\varepsilon}^i\right]\Bigg\} + C(\Delta_t)^{K_z}, \tag{5.25}$$

where $C>0$ is a generic constant only depending on T, the upper bounds for the functions φ and f and their derivatives, and the levels of used sparse grids q_n $(n=2,\ldots,N)$.

Remark 5.1. From the conclusions of Theorem 5.1 and 5.2, we can see that, for the same number of grid points, our method with (adaptive) sparse grids is much more accurate than existing numerical methods with full tensor-product grids. For example, if we use the existing multi-step scheme with full tensor-product grids where the number of quadrature points N_f^Q is set to N_s^Q and the number of interpolation points N_f^i is set to $N_{x,\varepsilon}^i$ for $i=1,\ldots,N$, then the error estimate of z_i^n in (5.25) becomes

$$\max_{\substack{0\leqslant j\leqslant N\\ i=1,\cdots,N_f^j}} \|z_{t_j}^{x_i^j} - z_i^j\|_\infty \leqslant \frac{C}{\Delta t}\left\{ \left(N_f^Q\right)^{-\frac{k}{2d}} + \max_{i=2,\ldots,N}\left(N_f^i\right)^{-\frac{2}{d}}\right\} + C(\Delta_t)^{K_z}. \tag{5.26}$$

Apparently, when the dimension d is large, the accuracy of the scheme with full tensor-product grid is much worse than our scheme with (adaptive) sparse grids. As discussed in Section 3, the total computational cost mainly depends on the number of grid points because at each point x_i^j, we need to approximate 5 conditional mathematical expectations and solve a nonlinear equation to obtain the values y_i^j and z_i^j. Thus, under the same computational cost, our method is more accurate than existing methods in solving multi-dimensional BSDEs. On the other hand, our method with sparse grids can attain a prescribed accuracy with much fewer grid points than the existing methods with full tensor-product grids, which shows much improved efficiency of our scheme. Some comparisons between schemes with sparse grids and full tensor-product grids are provided in Example 6.1 for the dimension $d=2,3,4$.

6. Numerical Examples

In this section, we report on the results of two numerical tests that illustrate the accuracy and efficiency of the proposed scheme (4.35) based on the sparse-grid method. Denote by W_t the standard d-dimensional Brownian motion. In the experiments, we take uniform partitions in time with the time step denoted by Δt. The time partition number N is then given by $N=\frac{T}{\Delta t}$, where T is the finite terminal time.

The errors $y_{t_n}^{x_i^n} - y_i^n$ and $z_{t_n}^{x_i^n} - z_i^n$ arise from three causes:

1. the time discretization for obtaining the semi-discrete scheme;
2. the approximation of the conditional mathematical expectation $\mathbb{E}_{t_n}^{x_i^n}[\cdot]$ by $\widehat{\mathbb{E}}_{t_n}^{x_i^n}[\cdot]$;
3. the interpolation for computing $\widehat{y}^{n+j}$ and $\widehat{z}^{n+j}$.

In order to obtain optimal numerical solutions, the errors from the three parts should often be balanced. Because the time-disretization error has been studied in [31], in this paper, the time

step Δt is set small enough such that the error contributed by time disretization is very small. On the other hand, from the error bounds given in Theorem 5.1 and 5.2, the interpolation error for computing $\widehat{y}^{n+j}$ and $\widehat{z}^{n+j}$ dominate the error from space-discretization. Therefore, in the following examples, our task is to investigate the errors caused by the HSG interpolant and the AHSG interpolant. Hereafter, we define $e_y = y_{t_0}^{x_0^0} - y_0^0$ and $e_z = z_{t_0}^{x_0^0} - z_0^0$ to be the errors of the numerical solution at the time-space point $(t_0, x_0^0) = (0, (0, \ldots, 0))$.

Example 6.1. In this example, we consider a d-dimensional BSDE with d from 2 to 4. Let $W_t = \left(W_t^1, \cdots, W_t^d\right)^\top$ be a d-dimensional Brownian motion. W_t^i $(i = 1, \cdots, d)$ are d independent standard one-dimensional Brownian motions. The BSDE of interest is

$$\begin{cases} -dy_t = \left[(d-1)y_t + 2\sum_{i=1}^{d} W_t^i z_t^i\right] dt - z_t dW_t, \\ y_T = \exp\left[T - \sum_{i=1}^{d} \left(W_T^i\right)^2\right], \end{cases} \tag{6.1}$$

where $z_t = (z_t^1, \cdots, z_t^d)$. The analytical solution of (6.1) is given by

$$\begin{cases} y_t = \exp\left[t - \sum_{i=1}^{d} (W_t^i)^2\right], \\ z_t^i = -2W_t^i \exp\left[t - \sum_{i=1}^{d} (W_t^i)^2\right], \quad i = 1, \cdots, d. \end{cases} \tag{6.2}$$

Note that the kernel f has a more general form containing W_t. Although we do not analyze properties of this type of kernel, it still can be used to test the performance of our scheme. The exact solution (y_t, z_t) at the time $t = 0$ is $y_0 = 1$ and $z_0^i = 0$ $(i = 1, \cdots, d)$. In the time direction, set the terminal time $T = 0.1$, $N = 17$, $K_y = 2$ and $K_z = 3$, so that the fully-discrete scheme (4.35) is of third-order [31] in the time direction with $\Delta t = 0.00625$. In space, a 3-level sparse-grid Gauss-Hermite quadrature rule is used to compute all mathematical expectations. The number of quadrature points is 22, 37, 57 in the cases of $d = 2, 3, 4$ [5]. Thus, in the following we focus on the interpolation error caused by the approximations $\widehat{y}^n$ and $\widehat{z}^n$.

First, we investigate the convergence of our scheme (4.35) along with the increasing number of grid points. For $d = 2, 3, 4$, the AHSG interpolation is used with threshold being $\varepsilon = 10^{-4}, 10^{-5}, 10^{-6}$. In comparison, HSG interpolation without adaptivity and full-grid interpolation are also conducted. The results are shown in Fig. 6.1. Note that because the number of grid points varies on different time levels, the averaged number of points over all time levels are used in Fig. 6.1. It is clear that both the HSG and AHSG methods are more accurate than the full-grid interpolation for the same number of points. The convergence rate of the HSG interpolation is consistent with the theoretical analysis in Theorem 5.1 and 5.2, i.e. $O(N_s^{-2} \cdot \log(N_s)^{3(d-1)})$. In order to compare the convergence rate between the HSG and AHSG methods, we choose the same maximum interpolation level for both methods. However, note that the AHSG method has almost the same convergence rate as the HSG method. The reason is the analytical solutions y_t and z_t in (6.2) are almost equally smooth over the spatial domain so that the surplus decreases almost equally fast in the interpolant (4.17). This is illustrated for $d = 2$ in Fig. 6.2 in which the shapes of z_t^1, z_t^2 and the corresponding adaptive sparse grids on a particular time level are plotted. It is clear that equally decreasing of the surplus leads to an almost equal density of grid points over the entire domain.

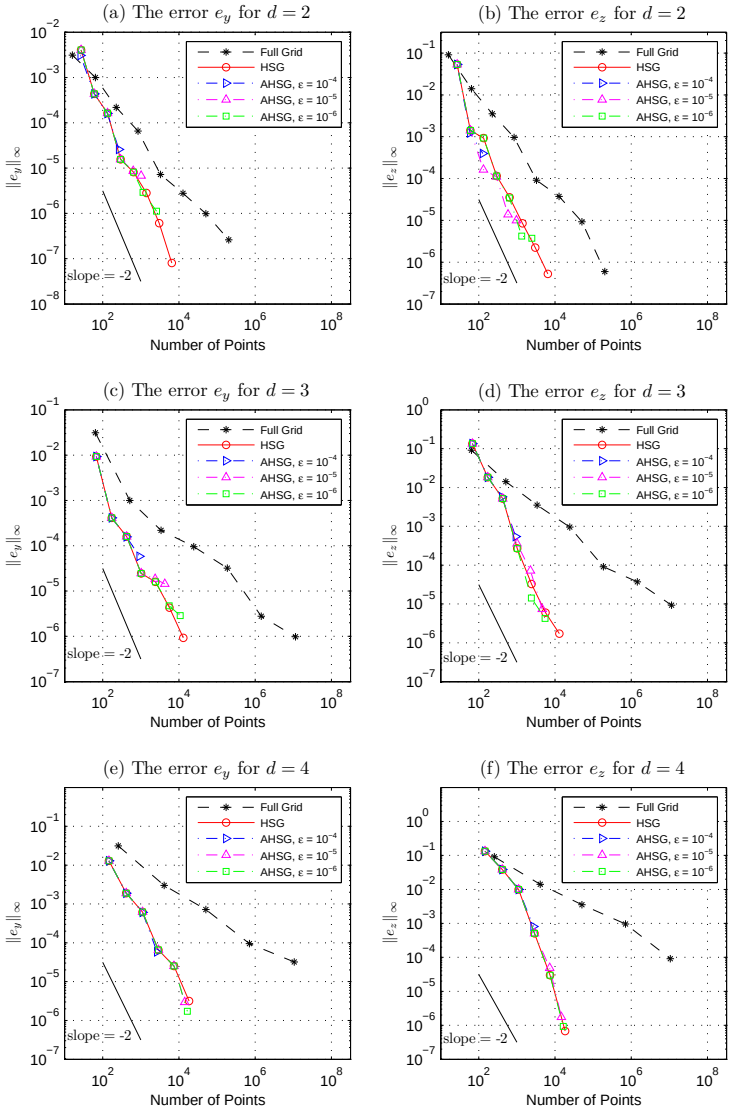

Fig. 6.1. The errors e_y and e_z with respect to the number of grid points for $d = 2, 3, 4$ in Example 6.1.

Next, we set the maximum interpolation level to be large enough to investigate the convergence of the interpolation error of AHSG method with respect to the threshold $\varepsilon = 10^{-4}$. Fig. 6.3 shows that the convergence rate is consistent with the theoretical result in Theorem 5.1 and 5.2, i.e., first order convergence $O(\varepsilon)$.

Example 6.2. In this example, we present an application of our scheme to financial problems. As discussed in [8], BSDEs appear in numerous financial problems, such as pricing and hedging of European and American options. Here, we consider the pricing of a basket call option in the Black-Scholes model. Denote p_t and $S_t = (S_t^1, \ldots, S_t^d)$ as the bond price and the prices of d

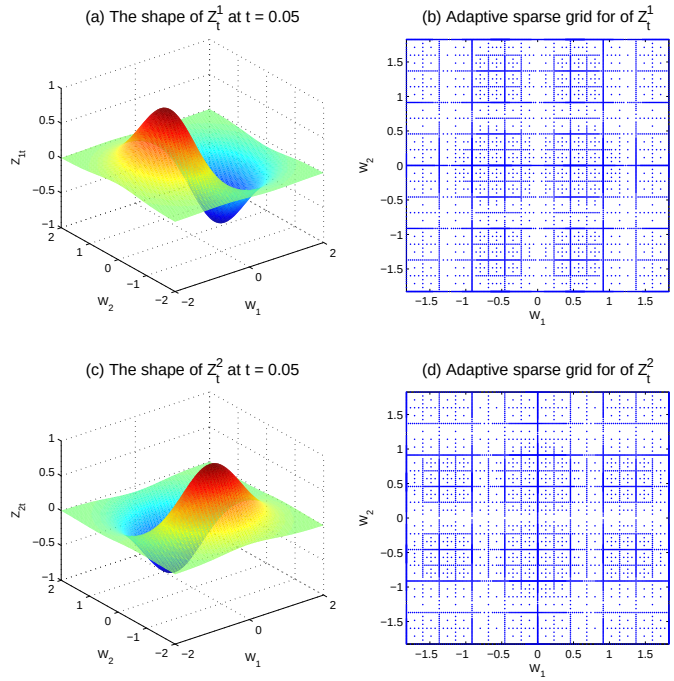

Fig. 6.2. For $d = 2$, the shapes of the solution z_t^1, z_t^2 in (a),(c); and the corresponding adaptive sparse grids in (b), (d) with the threshold $\varepsilon = 10^{-4}$ in Example 6.1.

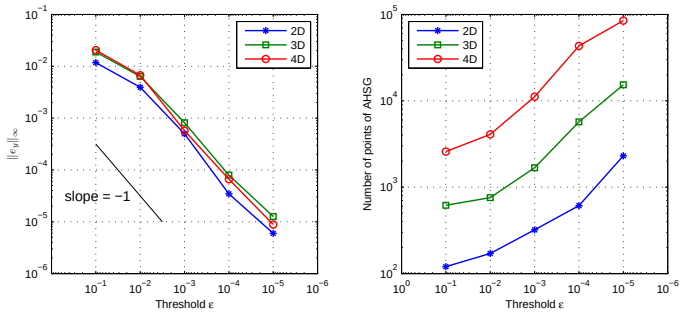

Fig. 6.3. The error e_y with respect to the threshold ε for the AHSG method(left); The growth of the number of grid points with respect to the threshold(right) in Example 6.1.

independent stocks, respectively. Assume that p_t and S_t satisfy

$$\begin{aligned} dp_t &= r_t p_t dt, && t \geqslant 0, \\ dS_t^i &= b_t^i S_t^i dt + \sigma_t^i S_t^i dW_t^i, && i = 1, \ldots, d \text{ and } t \geqslant 0 \end{aligned} \tag{6.3}$$

with initial conditions $p_0 = p$ and $S_0 = x$, where r_t is the return rate of the bond, b_t^i is the expected return rate of the i-th stock and σ_t^i is the volatility of the i-th stock. $W_t = (W_t^1, \ldots, W_t^d)$ is a vector of d mutually independent standard Brownian motions. Note that r_t, b_t^i, σ_t^i, $1/\sigma_t^i$ for $i = 1, \ldots, d$ are all bounded.

An investor with wealth y_t at time t puts π_t^i money to buy the i-th stock and uses $y_t - \sum_{i=1}^d \pi_t^i$

to buy the bond. Suppose that the stocks pay dividends continuously with a bounded dividend rate q_t^i at the time instant t. Then the processes y_t and π_t^i $(i = 1, \ldots, d)$ satisfy the following stochastic differential equation [8]:

$$-dy_t = -\left[r_t y_t + \sum_{i=1}^{d}(b_t^i - r_t + q_t^i)\pi_t^i\right]dt - \sum_{i=1}^{d}\sigma_t^i \pi_t^i dW_t^i, \tag{6.4}$$

Let $z_t = (z_t^1, \ldots, z_t^d) = (\sigma_t^1\pi_t^1, \ldots, \sigma_t^d\pi_t^d)$. Then (y_t, z_t) satisfies

$$-dy_t = -\left[r_t y_t + \sum_{i=1}^{d}\frac{b_t^i - r_t + q_t^i}{\sigma_t^i}z_t^i\right]dt - \sum_{i=1}^{d} z_t^i dW_t^i. \tag{6.5}$$

For the European call option, the terminal condition for the equation (6.5) is given at the mature time T by

$$y_T = \max\left\{\prod_{i=1}^{d}(S_t^i)^{\alpha_i} - K, 0\right\}, \tag{6.6}$$

where $\alpha_i > 0$, $\sum_{i=1}^{d}\alpha_i = 1$, S_T is the solution of S_t at the mature time T and K is the strike price. When $r_t = r$, $b_t = b$, $\sigma_t^i = \sigma^i$ and $q_t^i = q^i$, then the analytical solution can be obtained based on the classic Black-Scholes formula, i.e.,

$$\begin{cases} y_t = V(t, S_t) = e^{-\hat{q}(T-t)}\left(\prod_{j=1}^{d}(S_t^j)^{\alpha_j}\right)N(\hat{d}_1) - e^{-r(T-t)}KN(\hat{d}_2), \\ z_t^i = \dfrac{\partial V}{\partial S^i}\sigma^i = \alpha_i e^{-\hat{q}(T-t)}\left(\prod_{j=1}^{d}(S_t^j)^{\alpha_j}\right)N(\hat{d}_1)\sigma^i, \quad i = 1, \ldots, d, \\ \hat{d}_1 = \dfrac{\log\frac{\prod_{j=1}^{d}(S_t^j)^{\alpha_j}}{K} + \left[r - \hat{q} + \frac{\hat{\sigma}^2}{2}\right](T-t)}{\hat{\sigma}\sqrt{T-t}}, \\ \hat{d}_2 = \hat{d}_1 - \hat{\sigma}\sqrt{T-t}, \quad \hat{\sigma}^2 = \sum_{j=1}^{d}(\sigma_j\alpha_j)^2, \\ \hat{q} = \sum_{j=1}^{d}\alpha_j\left(q_j + \dfrac{\sigma_j^2}{2}\right) - \dfrac{\hat{\sigma}^2}{2}. \end{cases} \tag{6.7}$$

For our test, we set $T = 0.1$, $K = 100$, $r_t = 0.03$; and for $i = 1, \ldots, d$, set $S_0^i = 100$, $b_t^i = 0.05$, $d_t^i = 0.04$ and $\sigma_t^i = 0.2$. In the time direction, let $K_y = 2$, $K_z = 3$ and $N = 17$, so that the fully-discrete scheme (4.35) is of third order in time with $\Delta t = 0.00625$. As in Example 6.1, we also choose a 4-level sparse-grid Gauss-Hermite quadrature rule to approximate all conditional mathematical expectations and focus on the interpolation error caused by $\widehat{y}^n$ and $\widehat{z}^n$.

First, we solve this problem for $d = 2, 3, 4$ in order to compare with Example 6.1. It has already been demonstrated in Example 6.1 that the sparse-grid interpolation (both HSG and AHSG method) can attain higher accuracy than the full-grid interpolation for the same number of grid points, so in this example, we just solve the BSDE using HSG and AHSG method. The threshold ε is set to $10^{-3}, 10^{-4}, 10^{-5}$ for the AHSG method. The computational results are shown in Figs. 6.4, 6.5 and 6.6. It is noted in Fig. 6.4 that our scheme with HSG method achieves the theoretical convergence rate $\mathcal{O}(N_s^{-2}\cdot(\log(N_s))^{3(d-1)})$ proved in Theorem 5.1 and

5.2. Also note that many fewer points are needed for the AHSG method than the HSG method to achieve the same accuracy for solving y_i^n. In the two-dimensional case, the shape of y_t at a certain time level and the evolution of the adaptive sparse grid for the threshold 10^{-3} is shown in Fig. 6.5. Because of the shape of the terminal condition in (6.6), the solution y_t is not equally smooth over the entire region. The region around the diagonal line, where the derivative of y_t has relatively large variation, can be detected by the AHSG method, so that more grid points are placed in this region. In comparison, many fewer points are placed in the off-diagonal regions because y_t is much smoother in these regions.

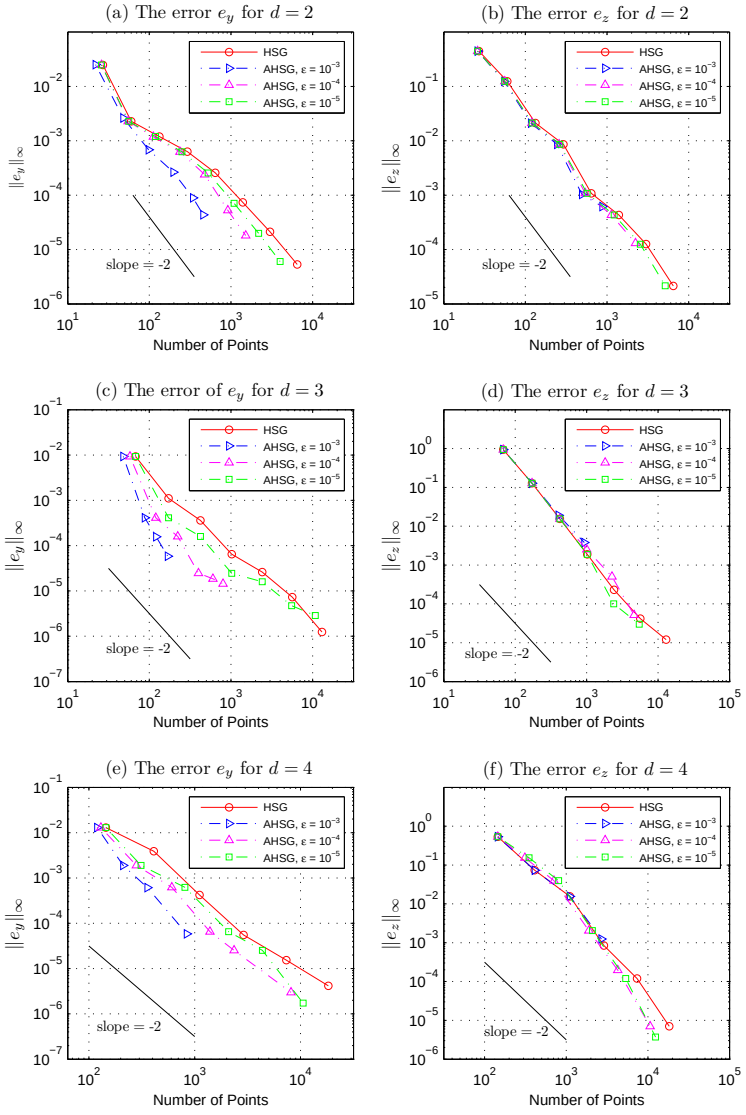

Fig. 6.4. The errors e_y and e_z with respect to the number of grid points for $d = 2, 3, 4$ in Example 2.

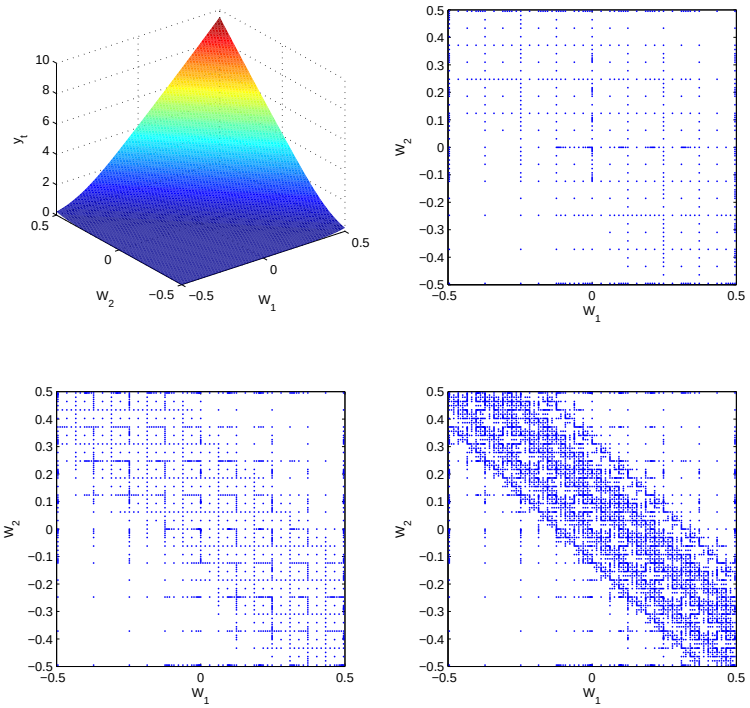

Fig. 6.5. The evolution of the adaptive sparse grid at the time level t_{N-1} with the threshold $\varepsilon = 10^{-3}$ in Example 2.

Next, to test the performance of our scheme in solving high-dimensional BSDEs, we consider the BSDE (6.5) for a European call option in dimension 8. In addition, because long-term computation is of great interest in real financial applications, we set the maturity time is $T = 10$. In the temporal domain, we still use the third-order multi-step scheme with $K_y = 2$, $K_z = 3$ and set $\Delta t = 0.01$; in the spatial domain, a 4-level sparse-grid Gauss-Hermite quadrature rule is used to approximate the conditional mathematical expectations. Thus, the interpolation error of interest caused by $\widehat{y}^n$ and $\widehat{z}^n$ will dominate the total error. The results are shown in Fig. 6.6. In Fig. 6.6(a) and 6.6(b), the convergence of our scheme with the HSG method and AHSG method ($\varepsilon = 10^{-1}, 10^{-2}$) are plotted. In Fig. 6.6(c) and 6.6(d), the error and the needed number of points with respect to threshold ε are provided. All the results are consistent with our theoretical analysis. As discussed in Section 2 Remark 3.1, the chosen values of K_y and K_z guarantee the stability of our scheme in the time direction. Furthermore, because there is no approximation of spatial derivatives involved in our scheme, the stability of our scheme is not affect by any CFL condition.

7. Conclusions and Future Work

In this paper, we propose a sparse-grid method for solving multi-dimensional backward stochastic differential equations. The BSDE is discretized by the multi-step scheme [31] in time and the sparse-grid method in space. It has been shown that the combination of the multi-step

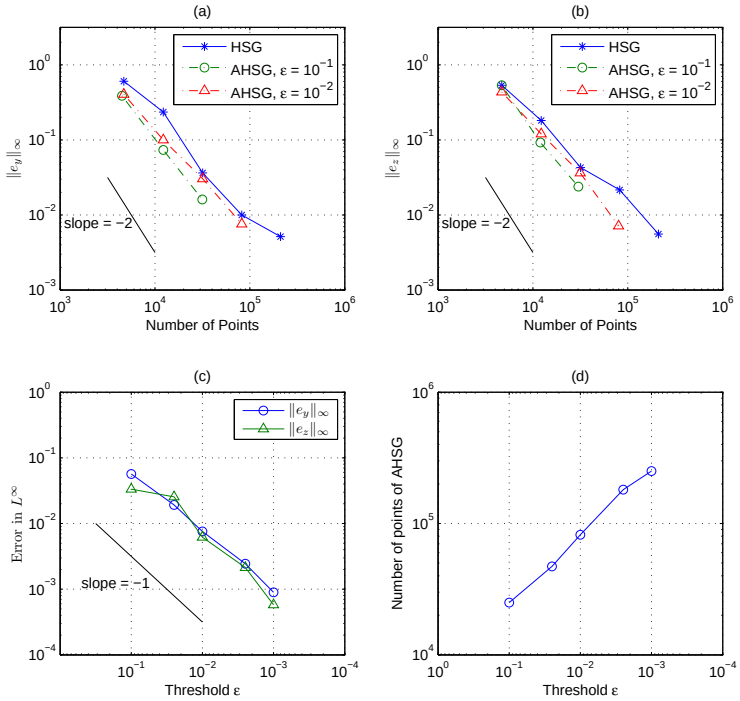

Fig. 6.6. (a,b) The errors e_y and e_z with respect to the number of grid points for $d = 8$, $T = 10$; (c)the error e_y, e_z with respect to the threshold ε for the AHSG method; (d) the growth of the number of grid points with respect to the threshold in Example 2.

method and the sparse-grid method is a highly suitable choice for solving multi-dimensional BSDEs. Moreover, the sparse-grid method can be use together with several one-step methods [28, 29] in a similar way. The numerical experiments demonstrate the effectiveness of our scheme and verify the consistency between theoretical analysis and computational results. For future study, we are going to extend our method to solve coupled forward-backward stochastic differential equations (FBSDEs) which are more general and applicable in real-world applications. When solving FBSDEs, sparse-grid approximations need to be constructed for solutions of both forward and backward equations, so that the sparse-grid mesh refinement will also depend on the smoothness of the driving process, i.e. the forward solution that may be highly non-smooth such as Lèvy processes with jumps. In addition, if a FBSDE system is not fully coupled, i.e. the forward solution does not depends on the backward solution, then the forward and backward equation can still be solved separately, in which case sparse-grid methods can be used with relative ease. However, if a FBSDE system is fully coupled, then the forward and backward equations must be solved simultaneously, in which case how to do sparse-grid mesh refinement will become more challenging.

Acknowledgments. The first author was supported by the US Air Force Office of Scientific Research under grant FA9550-11-1-0149. The first author was also supported by the Advanced Simulation Computing Research (ASCR), Department of Energy, through the Householder Fellowship at ORNL. The ORNL is operated by UT-Battelle, LLC, for the United States Depart-

ment of Energy under Contract DE-AC05-00OR22725. The second author was supported by the US Air Force Office of Scientific Research under grant FA9550-11-1-0149. The third author was supported by the Natural Science Foundation of China under grant 11171189. The third author was also supported by the Natural Science Foundation of China under grant 91130003. The thrid author was also supported by Shandong Province Natural Science Foundation under grant ZR2001AZ002.

References

[1] V. Barthelmann, E. Novak and K. Ritter, High dimensional polynomial interpolation on sparse grids, *Advances in Computational Mathematics*, **12**:4 (2000), 273–288.

[2] C. Bender and R. Denk, A forward scheme for backward SDEs, *Stochastic Processes and their Applications*, **117**:12 (2007), 1793–1812.

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

[4] H.-J. Bungartz and M. Griebel, Sparse grids, *Acta Numerica* **13** (2004), 147–269.

[5] J. Burkardt, Sparse Grids Based on Gauss-Hermite Rules, *http://people.sc.fsu.edu/ ∼jburkardt /cpp_src/sparse_grid_hermite/sparse_grid_hermite.html*.

[6] J. Cvitanić and J. Zhang, The steepest descent method for forward-backward SDEs. *Electronic Journal of Probability*, **10**:45 (2005), 1468–1495.

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

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

[9] T. Gerstner and M. Griebel, Numerical integration using sparse grids, *Numerical Algorithms* **18** (1998), 209–232.

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

[11] A. Klimke and B. Wohlmuth, Algorithm 847: Spinterp: piecewise multilinear hierarchical sparse grid interpolation in MATLAB, *ACM Transactions on Mathematical Software (TOMS)* , **31**:4 (2005), 561–579.

[12] Y. Li and W. Zhao, L^p-error estimates for numerical schemes for solving certain kinds of backward stochastic differential equations, *Statistics & Probability Letters*, **80**:21-22 (2010), 1612–1617.

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

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

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

[16] X. Ma and N. Zabaras, An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations, *Journal of Computational Physics*, **228**:8 (2009), 3084–3113.

[17] F. Nobile, R. Tempone and C. Webster, A sparse grid stochastic collocation method for partial differential equations with random input data, *SIAM J. Numer. Anal*, **46**:5 (2008), 2309–2345.

[18] F. Nobile, R. Tempone and C. Webster, An anisotropic sparse grid stochastic collocation method for elliptic partial differential equations with random input data. *SIAM J. Numer. Anal*, **46**:5 (2008), 2411–2442.

[19] E. Novak and K. Ritter, High dimensional integration of smooth functions over cubes. *Numerische Mathematik*, **75**:1 (1996), 79–97.

[20] E. Pardoux and S. Peng, Adapted solution of a backward stochastic differential equation. *Systems and Control Letters*, **14**:1 (1990), 55–61.

[21] S. Peng, A general stochastic maximum principle for optimal control problems, *SIAM Journal on control and optimization*, **28**:4 (1990), 966–979.

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

[23] S. Peng, A linear approximation algorithm using BSDE, *Pacific Economic Review*, **4**:3 (1999), 285–292.

[24] J. Shen and H. Yu, Efficient spectral sparse grid methods and applications to high-dimensional elliptic problems, *SIAM J. Sci. Comput.*, **32**:6 (2010), 3228–3250.

[25] J. Shen and H. Yu, Efficient spectral sparse grid methods and applications to high-dimensional elliptic equations II. Unbounded Domains, *SIAM J. Sci. Comput.*, **34**:2 (2012), 1141–1164.

[26] W. E. Smith, I. H. Sloan and A. H. Opie, Product Integration I . Rules Based on the Zeros of Hermite Polynomials, *Mathematics of Computation*, **40**:162 (1983), 519–535.

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

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

[29] W. Zhao, Y. Li and G. Zhang, A generalized θ-scheme for solving backward stochastic differential equations, *Discrete and Continuous Dynamical Systems - Series B*, **17**:5 (2012), 1585-1603.

[30] W. Zhao, J. Wang and S. Peng, Error estimates of the θ-scheme for backward stochastic differential equations, *Discrete and Continuous Dynamical Systems - Series B*, **12**:4 (2009), 905–924.

[31] W. Zhao, G. Zhang and L. Ju A stable multistep scheme for solving backward stochastic differential equations, *SIAM Journal on Numerical Analysis*, **48**:4 (2010), 1369–1394.