THE APPLICATION OF STOCHASTIC MESH METHOD IN BSDES

XIA HAOYANG
(B.S. & B.A., Beijing Normal University)

A THESIS SUBMITTED
FOR THE DEGREE OF MASTER OF SCIENCE
DEPARTMENT OF MATHEMATICS
NATIONAL UNIVERSITY OF SINGAPORE

2015

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

夏浩洋

July 20, 2015

Contents

Abstract

We study the application of stochastic mesh method in BSDEs. We start with the review of stochastic mesh method in American option pricing. Then we introduce BSDEs briefly, and by deducing the drivers and recursion in BSDEs, finally we apply stochastic mesh method to BSDEs. Numerical results are presented, of stochastic mesh method in both American option pricing and BSDEs.

1 Introduction

In practice, Monte-Carlo methods are usually designed to solve problems, which involve conditional expectations of the form:

$$f_u = \mathbb{E}\left[g(S_t)|\mathcal{F}_u\right]$$

where S_t could be some processes, of which g is the function, with $t \geq u$. And stochastic mesh method is one of those, which can solve both the conditional expectations and general optimal stopping problems as well. The most important way to construct the mesh is shown in Figure 1, which is originally introduced by Broadie and Glasserman (2004). In the first step, we simulate m independent paths of the Markovian chain $X_1, X_2, \cdots, X_m$ and each path X_j contains n time steps $X_{0j}, X_{1j}, \cdots, X_{ij}, X_{i+1,j}, \cdots, X_{n-1,j}$, where we denote $X_{0j} = X_0$, fixed for all j. In the second step, we 'ignore' the relationships of the nodes next to each other, i.e. 'forget' which node at time step i generates that at time step $i+1$ in each path. Then in the third step, we interconnect all the nodes at consecutive time steps for the backward induction.[1]

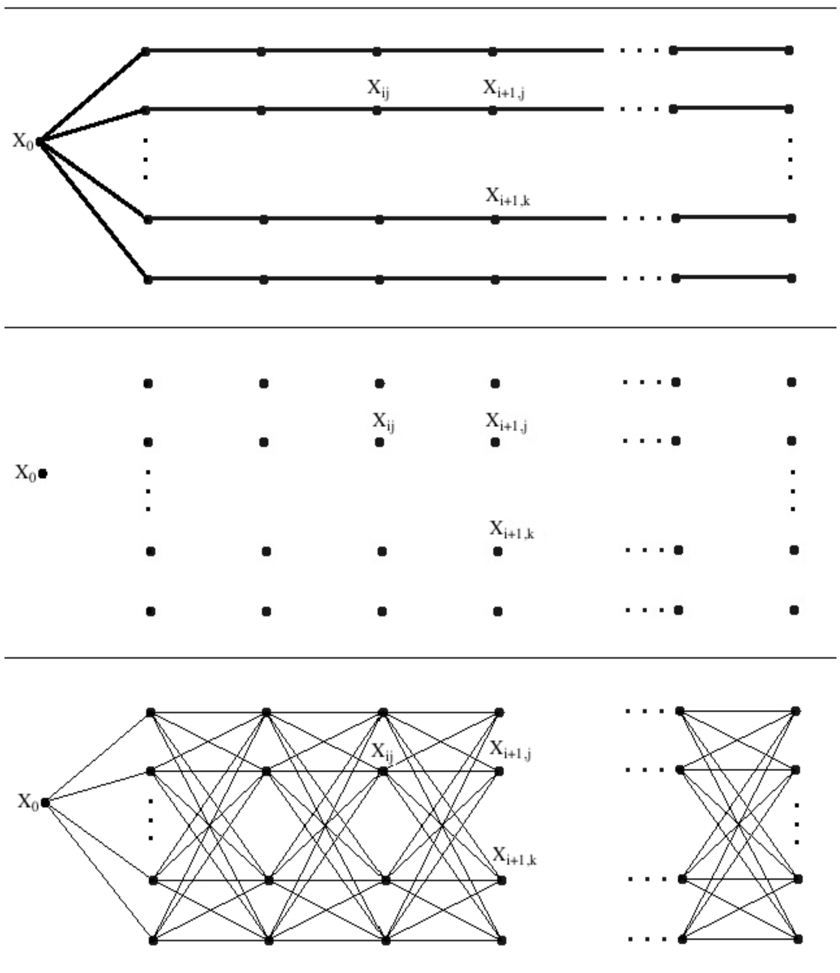

Figure 1: Independent path construction

It's known that no arbitrage implies the price of an American option, V_t, satisfies the following

[1] Glasserman, P. (2004). Monte Carlo Methods in Financial Engineering. *Springer*.

dynamic programming:

$$V_t = \max\left\{\mathbb{E}\left[e^{-r\mathrm{d}t}V_{t+\mathrm{d}t}|\mathcal{F}_t\right], h_t\right\}$$

with terminal condition $V_T = h_T$, where h_t is the payoff function at time t. Besides, American option pricing is also connected to an optimal stopping problem, so it's natural to apply stochastic mesh method to solve American option pricing problem, especially with discrete exercise opportunities[2], as at particular points, immediate exercises are calculated and the continuation values are estimated. Compared with others in American option pricing, like random tree method by Broadie and Glasserman (1997), stochastic mesh method uses values from all nodes at time step $i+1$, rather than just those, which are successors of the current node, when valuing the option at a node at time step i. Thus it avoids the exponential growth characteristic when number of time steps is increasing, however reduces it into linear. This is identical, in particular in high-dimensional cases.

An important issue of the stochastic mesh method is to determine the connection between the nodes at consecutive time steps, the weights. The selection of weights W^i_{jk} is closely related to the sampling of the nodes X_{ij}, as the independent path construction above just provides us one (but never the only one) of the mechanisms, where likelihood ratio weights are applied by Broadie and Glasserman (2004). There are also many others, see, for example, Broadie, Glasserman and Ha (2000) for optimized weights, which avoid the need for the transition density by choosing weights through a constrained optimization problem; Liu and Hong (2009) for binocular weights, which condition on values from nodes at both time steps $i-1$ and $i+1$. To enhance the method, Avramidis and Hyden (1999) propose efficiency improvements, such as bias reduction and importance sampling; Broadie and Yamamoto (2003) use a fast Gauss transform to accelerate backward induction calculations. Moreover, Avramidis and Matzinger (2004) show the convergence of the stochastic mesh estimators.

The notion of backward stochastic differential equations (BSDEs) is firstly introduced by Bismut (1973), for the linear case.[3] Pardoux and Peng (1990, 1992) generalize it and show the connection between BSDEs and parabolic PDEs, as well as prove the existence and uniqueness of the adapted solutions, under proper conditions. The solution of BSDEs typically consists of a pair of adapted process (Y_t, Z_t), satisfying the following equation:

$$-\mathrm{d}Y_t = f(t, Y_t, Z_t)\mathrm{d}t - Z_t\mathrm{d}W_t$$

with terminal condition $Y_T = \xi_T$, where W_t is Brownian motion and f is a function of t, Y_t, Z_t, called the driver. These equations are very useful in financial area, see, for example, El. Karoui, Peng and Quenez (1997) for derivative pricing; Peng (2003) for dynamic risk measures; El. Karoui, Hamadène and Matoussi (2008) for stochastic control, among others. Besides, BSDEs are more robust to fit in uncertainties of different probability models, compared with the theory of derivative pricing studied by Black and Scholes (1973) and Merton (1973, 1991), which could be also expressed in terms of it.

As the cases of PDEs and SDEs, a general problem remained for BSDEs is that they might not usually be solved in analytical formulae. To deal with this, a lot of researches have been proposed to solve BSDEs numerically, particularly in derivative pricing during the decades. Ma, Protter and Yong (1994) show the *four step scheme* and present the explicit relations among the forward and backward components of the adapted solution of FBSDEs; a discrete time approximation is provided

[2] Also known as Bermudan option.

[3] El Karoui, N., Hamadène, S. and Matoussi, A. (2008). Backward stochastic differential equations and applications. *Springer*.

by Bouchard and Touzi (2004) and a regression-based Monte Carlo method is also used by Gobet, Lemor and Warin (2005) to solve the conditional expectations; Bally et al. (2001, 2003) suggest a quantization technique for the solution of reflected BSDEs; besides, those kinds of equations have been applied to solve American option pricing problems by Gobert et al. (2005). Moreover, Peng et al. (2010) develop a parallel algorithm for the BSDEs with application to derivative pricing as well.

However most, if not all, of those methods haven't taken advantages of stochastic mesh method when studying BSDEs. In other words, the exponential growth with respect to the number of time steps might lead to an inefficiency as $n \geq 5$ or in high-dimensional case. So in this paper, we apply stochastic mesh method to BSDEs in some examples to show how they work efficiently together in derivate pricing. We present numerical results of both one dimensional and high-dimensional cases as well.

This paper is organized as follows. In Section 2, we review the stochastic mesh method in American option pricing, which is mainly the original work of Broadie and Glasserman (2004). In Section 3, we introduce some basic knowledge of BSDEs, and detail the application of stochastic mesh method in BSDEs. In Section 4, we present results qualifying the performance of the method in some examples.

2 Stochastic mesh method in American option pricing

In reviewing the method in American option pricing, we mainly follow Broadie and Glasserman (2004). As shown in Figure 2, where we denote $T = (n-1)\Delta t$ and W_{jk}^i the weight between X_{ij} at time step i and $X_{i+1,k}$ at time step $i+1$, the method generates a stochastic mesh of m independent paths within n time steps under several conditions:
(1) $\{\bar{X}_0, \cdots, \bar{X}_{i-1}\}$ and $\{\bar{X}_{i+1}, \cdots, \bar{X}_{n-1}\}$ are independent, where $\bar{X}_i = (X_{i1}, \cdots, X_{im})$, represents all nodes at time step i;
(2) for $1 \leq j, k \leq m$, W_{jk}^i is a deterministic function of $\bar{X}_i$ and $\bar{X}_{i+1}$;
(3)$\forall i = 1, \cdots, n-2$ and $\forall j = 1, \cdots, m$, W_{jk}^i satisfies

$$e^{-r\Delta t} \sum_{k=1}^{m} \mathbb{E}[W_{jk}^i V_{i+1}(X_{i+1,k}) | \bar{X}_i] = C_i(X_{ij}), \tag{2.1}$$

where $C_i(X_{ij})$ denotes the continuation value at node X_{ij} at time step i.

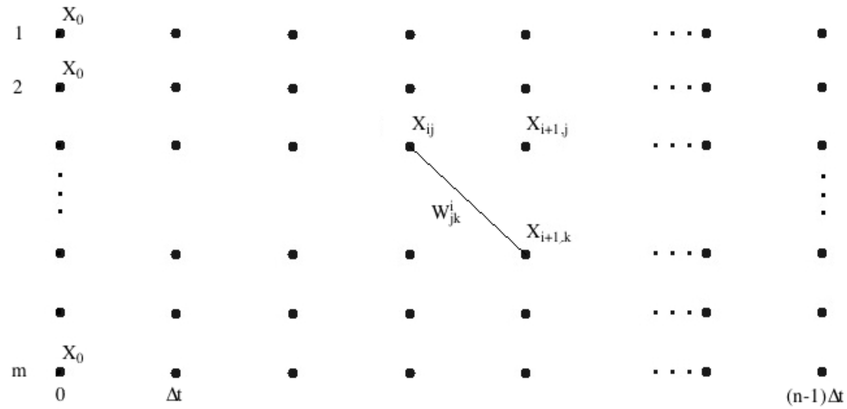

Figure 2: Nodes of the mesh

Example For the detail of the construction of paths and nodes, we take the independent paths of geometric Brownian motion as an example. Suppose under risk-neutral probability, the process of underlying asset satisfies:

$$\mathrm{d}S_t = rS_t\mathrm{d}t + \sigma S_t \mathrm{d}W_t$$

by Itô, we can easily obtain that

$$S_t = S_0 e^{(r-\frac{1}{2}\sigma^2)t+\sigma W_t}$$

and

$$S_{t+\Delta t} = S_t e^{(r-\frac{1}{2}\sigma^2)\Delta t+\sigma(W_{t+\Delta t}-W_t)}. \tag{2.2}$$

Since we know $W_{t+\Delta t} - W_t \sim N(0, \Delta t)$ and the value of $r, \sigma, \Delta t$ and S_0, for each path we can simulate nodes forwardly up to n time steps.Then we repeat the same process to simulate m paths independently[4] , thus to finish the construction.

2.1 Estimators

2.1.1 Mesh estimator

As the backward induction can be used, we now introduce the mesh (high) estimator of the option value. The mesh estimator is defined recursively as follows:

For $(n-1)\Delta t = T$,

$$\hat{V}_{n-1,j} = h_{n-1}(X_{n-1,j})$$

where $h_i(X_{ij})$ is the payoff function of X_{ij}.

For $0 < i < n-1$,

$$\hat{V}_{ij} = \max\left\{ e^{-r\Delta t} \sum_{k=1}^{m} W^i_{jk} \hat{V}_{i+1,k}, h_i(X_{ij}) \right\} \tag{2.3}$$

for some set of weights W^i_{jk}.

For $i = 0$,

$$\hat{V}_0 = e^{-r\Delta t} \frac{1}{m} \sum_{k=1}^{m} \hat{V}_{1k}.$$

[4]Note that each path constructed as this way is a Markov chain.

Lemma 1. *(High biased) The mesh estimator $\hat{V}_0$ defined as above is biased high.*

Proof We've already known, at the terminal time step,

$$\hat{V}_{n-1,j} = h_{n-1}(X_{n-1,j}) = V_{n-1}(X_{n-1,j}),$$

for $j = 1, \cdots, m$. Then suppose for some $0 \leq i < n-1$,

$$\mathbb{E}[\hat{V}_{i+1,j} | \bar{X}_{i+1}] \geq V_{i+1}(X_{i+1,j})$$

holds for $j = 1, \cdots, m$. We now consider $\hat{V}_{ij}$ at time step i.

First of all, by using Jensen's inequality, from (2.3) we could obtain:

$$\begin{aligned}\mathbb{E}[\hat{V}_{ij} | \bar{X}_i] &= \mathbb{E}\left[\max\left\{e^{-r\Delta t}\sum_{k=1}^{m} W_{jk}^i \hat{V}_{i+1,k}, h_i(X_{ij})\right\}\middle| \bar{X}_i\right] \\ &\geq \max\left\{\mathbb{E}\left[e^{-r\Delta t}\sum_{k=1}^{m} W_{jk}^i \hat{V}_{i+1,k}\middle| \bar{X}_i\right], h_i(X_{ij})\right\}\end{aligned} \tag{2.4}$$

Now, we're investigating the conditional expectation of the right hand side above. By further conditioning on $\bar{X}_{i+1}$, it can be deduced as:

$$\begin{aligned}\mathbb{E}\left[e^{-r\Delta t}\sum_{k=1}^{m} W_{jk}^i \hat{V}_{i+1,k}\middle| \bar{X}_i, \bar{X}_{i+1}\right] &= e^{-r\Delta t}\sum_{k=1}^{m} W_{jk}^i \mathbb{E}\left[\hat{V}_{i+1,k}\middle| \bar{X}_i, \bar{X}_{i+1}\right] \\ &= e^{-r\Delta t}\sum_{k=1}^{m} W_{jk}^i \mathbb{E}\left[\hat{V}_{i+1,k}\middle| \bar{X}_{i+1}\right] \\ &\geq e^{-r\Delta t}\sum_{k=1}^{m} W_{jk}^i V_{i+1}(X_{i+1,k})\end{aligned}$$

Then by taking conditional expectation (r.w.t. $\bar{X}_i$) on both hand sides of the inequality above, from (2.1) we get:

$$\begin{aligned}\mathbb{E}\left[e^{-r\Delta t}\sum_{k=1}^{m} W_{jk}^i \hat{V}_{i+1,k}\middle| \bar{X}_i\right] &\geq \mathbb{E}\left[e^{-r\Delta t}\sum_{k=1}^{m} W_{jk}^i V_{i+1}(X_{i+1,k})\middle| \bar{X}_i\right] \\ &= C_i(X_{ij})\end{aligned}$$

Plug this inequality back into (2.4) and we can easily obtain:

$$\begin{aligned}\mathbb{E}[\hat{V}_{ij} | \bar{X}_i] &\geq \max\left\{\mathbb{E}\left[e^{-r\Delta t}\sum_{k=1}^{m} W_{jk}^i \hat{V}_{i+1,k}\middle| \bar{X}_i\right], h_i(X_{ij})\right\} \\ &\geq \max\left\{C_i(X_{ij}), h_i(X_{ij})\right\} \\ &= V_i(X_{ij})\end{aligned}$$

Thus we complete the induction and finish the proof.

QED

From the induction, we find that the conditions on mesh are requisite when we use mesh estimators $\hat{V}_{ij}$ to simulate the true values $V_i(X_{ij})$. Broadie and Glasserman (2004) prove the convergence of the mesh estimator when $\{\bar{X}_0, \cdots, \bar{X}_{n-1}\}$ are independent of each other, $X_{i1}, \cdots, X_{im}$ are i.i.d. for each i, and each weight W^i_{jk} is a function only of X_{ij} at time step i and $X_{i+1,k}$ at time step $i+1$. Avramidis and Matzinger (2004) derive an upper bound on the error of the mesh estimator for a dependence case under the same conditions on mesh as well.

2.1.2 Path estimator

We now introduce a low biased estimator, to define which Broadie and Glasserman (2004) use a stopping rule as follows:

First of all, the same nodes of mesh are constructed as shown in Figure 2 and the weight W^i_{jk} is extended from $X_{i1}, \cdots, X_{im}$ to all points in the state space at time step i, denoted by $W^i_k(x)$ as that between the state x at time step i and node $X_{i+1,k}$ at time step $i+1$. To make it compatible to the extant result, we assume $W^i_k(X_{ij}) = W^i_{jk}$. By this extension of weight function, we define a continuation value at state x at time step i as:

$$\hat{C}_i(x) = e^{-r\Delta t} \sum_{k=1}^{m} W^i_k(x) \hat{V}_{i+1,k} \tag{2.5}$$

where $\hat{V}_{ij}$ is calculated as (2.3). So for the extant paths, $\hat{C}_i(X_{ij})$ coincides with the continuation value estimated by mesh at node X_{ij}, for $j = 1, \cdots, m$.

With the mesh fixed, we simulate a new path of Markovian chain, independent from the original m ones as shown in Figure 3. We define the stopping time $\hat{\tau}$ as

$$\hat{\tau} = \min\left\{ i : \hat{C}_i(X_{i,m+1}) \le h_i(X_{i,m+1}) \right\}$$

which is the first time step that the immediate exercise value is greater than continuation value. So with the definition of the stopping rule, the path estimator is

$$\check{V} = e^{-r\hat{\tau}} h_{\hat{\tau}}(X_{\hat{\tau},m+1}).$$

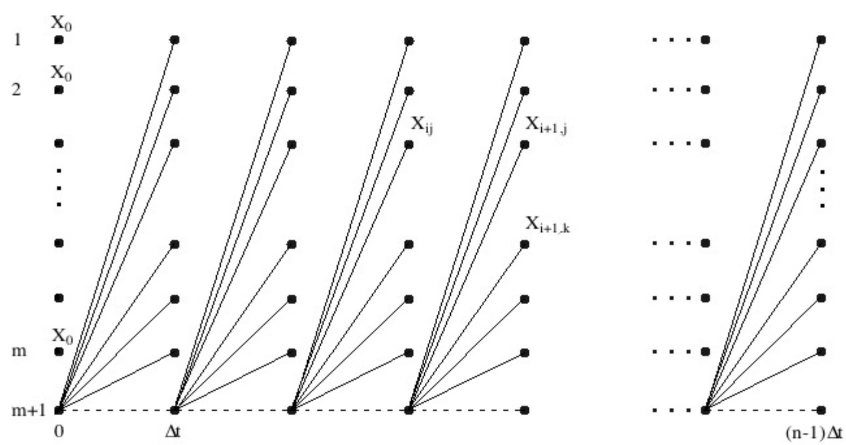

Figure 3: A new path besides the original m paths

Lemma 2. *(Low biased) The path estimator $\check{V}$ defined as above is biased low.*

Proof As introduced, the price of American option V_t is the solution to an optimal stopping problem, i.e.

$$V_t = \sup_{\tau \in \mathcal{S}[t,T]} \mathbb{E}\left[e^{-r(\tau - t)} h_\tau \middle| \mathcal{F}_t \right]$$

where $\mathcal{S}[t,T]$ is the set of stopping time between t and T. Thus the lemma is proved, indeed, since no policy can be better than the optimal one.

QED

We then repeat the same process to generate m' paths independently, with each following the same stopping rule defined above, so we can calculate an average low estimator conditional on the mesh. With the independence assumption of the construction of nodes in the mesh, Broadie and Glasserman (2004) give conditions, under which the low estimator is asymptotically unbiased, i.e. $\mathbb{E}[\check{V}] \to V_0(X_0)$, as $m \to \infty$.

2.1.3 Average estimator

With the independent replications of high and low estimators, we can calculate the sample mean and standard deviation, then form a confidence interval of each estimator. By combining the lower bound of low estimator and upper bound of high estimator, we can thus get a so-called interval estimator. However, there is another way to produce a more accurate value by blending those two techniques rather than simply keep the two sources of bias, called the average estimator, which is defined by Avramidis and Hyden (1999) as follows:

Again, we first construct the same nodes of mesh of m independent paths within n time steps, as shown in Figure 2 and calculate the immediate exercise function h_i, estimators $\hat{V}_{ij}$ as in the cases of mesh and path estimators, and estimated continuation function $\bar{C}_i$[5]. Then we obviate the influence of one of nodes at time step $i+1$, say, $X_{i+1,k}$ on the node X_{ij} at time step i, and calculate the new estimated continuation value $\bar{C}_i^{-k}$. For the next step, we make two criteria: (1) if $\bar{C}_i^{-k}(X_{ij}) \leq h_i(X_{ij})$, we take the value $h_i(X_{ij})$ as our estimator $\check{V}_{ij}^k$ at node X_{ij}; (2) otherwise, we take the continuation value $\bar{C}_i^k(X_{ij})$ as $\check{V}_{ij}^k$, which is obtained by only considering the influence of $X_{i+1,k}$ on X_{ij}. By making k over all $1, \cdots, m$, we then define:

$$\check{V}_{ij} = \frac{1}{m} \sum_{k=1}^{m} \check{V}_{ij}^k.$$

Lemma 3. *(Low biased) The estimators $\check{V}_{ij}$ defined as above is biased low for all i, j.*

Proof See Avramidis and Hyden (1999).

QED

By combining the low estimator $\check{V}_{ij}$ and the preciously calculated high estimator $\hat{V}_{ij}$, we define the average estimator at node X_{ij} as:

$$\bar{V}_{ij} = \frac{1}{2}(\hat{V}_{ij} + \check{V}_{ij}) \tag{2.6}$$

Then by applying backward induction, we obtain the average estimator $\bar{V}_0$ of V_0. This procedure takes advantages of both high and low estimators, alternating between two and taking average.

[5] $\bar{C}_i(x) = e^{-r\Delta t} \sum_{k=1}^{m} W_k^i(x) \bar{V}_{i+1,k}$, where the definition of $\bar{V}_{i+1,k}$ comes from the induction, which we'll illustrate soon.

2.2 Weights

In this part, we discuss how to define weights W^i_{jk} that satisfy conditions on mesh. This is a very important issue in stochastic mesh method, since by calculating the weights, we can simulate the conditional expectations in the continuation functions of American option pricing, which indeed, is one of the main reasons we propose this method. Here we basically talk about likelihood ratio weights by Broadie and Glasserman (2004) and optimized weights by Broadie, Glasserman and Ha (2000).

2.2.1 Likelihood ratio weights

To begin with, the transition densities $f_1, \cdots, f_{n-1}$ between the states of Markov chain $\bar{X}_0, \cdots, \bar{X}_{n-1}$ (if exist) are defined as follows:

$$\mathbb{P}[\bar{X}_i \in A | \bar{X}_{i-1} = x] = \int_A f_i(x, y)\mathrm{d}y$$

for $\bar{X}_i \in \mathbb{R}^m$, where $i = 1, \cdots, n-1$, and $A \subseteq \mathbb{R}^m$. And the marginal densities of $\bar{X}_1, \cdots, \bar{X}_{n-1}$ are defined by induction:

For the marginal density of $\bar{X}_1$,

$$l_1(\cdot) = f_1(\bar{X}_0, \cdot).$$

For the marginal density of $\bar{X}_i$, $i = 2, \cdots, n-1$,

$$l_i(y) = \int l_{i-1}(x) f_i(x, y)\mathrm{d}x.$$

We denote $g_1, \cdots, g_{n-1}$ as mesh density functions, from which $\bar{X}_1, \cdots, \bar{X}_{n-1}$ are generated as i.i.d. samples. Then the likelihood ratio weights W^i_{jk} between X_{ij} at time step i and $X_{i+1,k}$ at time step $i+1$ are defined by:

$$W^i_{jk} = \frac{f_{i+1}(X_{ij}, X_{i+1,k})}{m \cdot g_{i+1}(X_{i+1,k})}. \tag{2.7}$$

The choice of the densities g_i is crucial to the practical success of the mesh method, for example, if we choose the marginal density functions as our mesh density functions, i.e. to set $g_i = l_i$, Broadie and Glasserman (2004) show that this choice can lead to a trouble that the variance of estimators grows exponentially w.r.t. the number of time steps. However, they provide a better choice by using average density function:

$$g_{i+1}(\cdot) = \frac{1}{m}\sum_{j=1}^{m} f_{i+1}(X_{ij}, \cdot)$$

i.e. the average of the transition densities out of the nodes at time step i. Then the corresponding likelihood ratio weight is:

$$W^i_{jk} = \frac{f_{i+1}(X_{ij}, X_{i+1,k})}{\sum_{j=1}^{m} f_{i+1}(X_{ij}, X_{i+1,k})}, \tag{2.8}$$

which also implies:

$$\sum_{j=1}^{m} W^i_{jk} = 1.$$

Using the average density function corresponds to generating m independent paths, and then 'forgetting' the path to which each sampled node belongs at time step $i = 0, \cdots, n-1$. Moreover, the average density converges to marginal density:

$$\frac{1}{m}\sum_{j=1}^{m} f_{i+1}(X_{ij}, \cdot) \to l_{i+1}(\cdot),$$

as $m \to \infty$.

Example As an illustration, we're going to consider the case of geometric Brownian motion. First of all, let's deduce the transition density. Consider the process:

$$\mathrm{d}S_t = rS_t\mathrm{d}t + \sigma S_t \mathrm{d}W_t,$$

so that

$$S_t = S_0 e^{(r-\frac{1}{2}\sigma^2)t + \sigma W_t}$$

which has lognormal distribution. By letting $X\Delta t = (r - \frac{1}{2}\sigma^2)\Delta t + \sigma \Delta W_t$, where $\Delta W_t = W_{t+\Delta t} - W_t$, we then get

$$S_{t+\Delta t} = S_t e^{X\Delta t}$$

thus,

$$\begin{aligned}
\mathbb{P}[S_{t+\Delta t} \le y | S_t] &= \mathbb{P}[X\Delta t \le \ln(\frac{y}{S_t})] \\
&= \mathbb{P}[\frac{\Delta W_t}{\sqrt{\Delta t}} \le \frac{\ln(\frac{y}{S_t}) - (r - \frac{1}{2}\sigma^2)\Delta t}{\sigma\sqrt{\Delta t}}] \\
&= \int_{-\infty}^{\frac{\ln(\frac{y}{S_t}) - (r-\frac{1}{2}\sigma^2)\Delta t}{\sigma\sqrt{\Delta t}}} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \mathrm{d}x \\
&= \int_{-\infty}^{\ln(\frac{y}{S_t})} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{z-(r-\frac{1}{2}\sigma^2)\Delta t}{\sigma\sqrt{\Delta t}})^2} \frac{1}{\sigma\sqrt{\Delta t}} \mathrm{d}z \qquad (z = (r - \frac{1}{2}\sigma^2)\Delta t + \sigma\sqrt{\Delta t}x) \\
&= \int_{-\infty}^{y} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{\ln(\frac{a}{S_t})-(r-\frac{1}{2}\sigma^2)\Delta t}{\sigma\sqrt{\Delta t}})^2} \frac{1}{a\sigma\sqrt{\Delta t}} \mathrm{d}a \qquad (a = S_t e^z)
\end{aligned}$$

Since $a = e^z = S_t e^{(r-\frac{1}{2}\sigma^2)\Delta t + \sigma\sqrt{\Delta t}x}$, has the same distribution as $S_{t+\Delta t}$, then the transition density from S_t to $S_{t+\Delta t}$ is

$$f(S_t, S_{t+\Delta t}) = \frac{1}{S_{t+\Delta t}\sigma\sqrt{\Delta t}} \phi(\frac{\ln(\frac{S_{t+\Delta t}}{S_t}) - (r - \frac{1}{2}\sigma^2)\Delta t}{\sigma\sqrt{\Delta t}}) \tag{2.9}$$

where ϕ is the normal probability density function.

So if applied in the mesh of m independent paths within n time steps constructed as before, we can write the transition density function as:

$$f_{i+1}(X_{ij}, X_{i+1,k}) = \frac{1}{X_{i+1,k}\sigma\sqrt{\Delta t}} \phi(\frac{\ln(\frac{X_{i+1,k}}{X_{ij}}) - (r - \frac{1}{2}\sigma^2)\Delta t}{\sigma\sqrt{\Delta t}}), \tag{2.10}$$

then the corresponding likelihood ratio weight is:

$$W_{jk}^{i} = \frac{\frac{1}{X_{i+1,k}\sigma\sqrt{\Delta t}}\phi(\frac{\ln(\frac{X_{i+1,k}}{X_{ij}})-(r-\frac{1}{2}\sigma^2)\Delta t}{\sigma\sqrt{\Delta t}})}{\sum_{j=1}^{m}\frac{1}{X_{i+1,k}\sigma\sqrt{\Delta t}}\phi(\frac{\ln(\frac{X_{i+1,k}}{X_{ij}})-(r-\frac{1}{2}\sigma^2)\Delta t}{\sigma\sqrt{\Delta t}})}. \tag{2.11}$$

2.2.2 Optimized weights

Now, we're going to extend the method from its original work to a more general setting where we don't need to know the transition density but choose the mesh weights through a constrained optimization problem, which is also proposed by Broadie and Glasserman (2000). Innovatively, they introduced two criteria for use in the optimization, which are maximum entropy and least square. And the constraints they impose make sure that the mesh value of some basic quantities could match its theoretical values. Generally speaking, the number of constrains is much smaller than that of weights, so that the problem is underdetermined.

The maximum entropy weights w_{ij} aim to maximize

$$L = L_0 + \lambda_0(\sum_j w_{ij} - 1) + \sum_{k,j}\lambda_k(B_{kj} - b_k)w_{ij}$$

where $L_0 = -\sum_{j=1}^{M} w_{ij}\log(w_{ij})$ is the entropy criterion, λ_k's are Lagrange multipliers and B is a $K \times M$ matrix, b is a K-dimensional vector, which give us K linear constraints[6] at each node i by

$$\sum_{j=1}^{M} B_{kj}w_{ij} = b_k$$

where $k = 1, \cdots, K$.[7]

Under first order condition, we could obtain

$$\frac{\partial L}{\partial w_{ij}} = -(\log(w_{ij}) + 1) + \lambda_0 + \sum_k \lambda_k(B_{kj} - b_k) = 0$$

$$\frac{\partial L}{\partial \lambda_0} = \sum_j w_{ij} = 1$$

then

$$w_{ij} = \exp(\lambda_0 + \sum_k \lambda_k(B_{kj} - b_k) - 1)$$

and

$$e^{1-\lambda_0} = \sum_j \exp(\sum_k \lambda_k(B_{kj} - b_k))$$

thus

$$w_{ij} = \frac{\exp(\sum_k \lambda_k(B_{kj} - b_k))}{\sum_j \exp(\sum_k \lambda_k(B_{kj} - b_k))}.$$

[6] Such as 'natural' constraints: commensurate with first and higher order moments for the process of underlying asset, which we'll describe in detail later.

[7] There is also an intuitive constrain: $\sum_j w_{ij} = 1$.

So finally, if the matrix B and vector b are known, by solving the Lagrange multiplies, we could work out the maximum entropy weights.

Although w_{ij} could be guaranteed to be nonnegative(obviously), we yet choose to use the second criteria—least square in our case, which is far easier in computation. By Taylor expansion

$$-\log(w) = 1 - w + o(w),$$

we approximate the L_0 in maximum entropy criterion via simply replacing $-\log(w)$ by $(1-w)$, which leave us to deal with an equivalent problem by choosing weights through a least square criterion of minimizing $\sum_j w_{ij}^2$. By solving first order equation and others, we find that the weights vector $w_i = (w_{i1}, \cdots, w_{iM})$ at each node i, could be expressed w.r.t the matrix B and vector b as

$$w_i = B^\top (BB^\top)^{-1} b \tag{2.12}$$

this gives the direct answer of weights w_{ij} by knowing the parameter B and b, which also provides us the advantage of computing speed compared with the original maximum entropy, the nonnegativity is no longer ensured though.

Example Now we introduce the 'natural' constraints, which make sure that the weights w_{ij} will match first and higher order moments for the process of underlying asset. For example, in the 1^{st} order moment case, where the process of underlying asset satisfies geometric Brownian motion:

$$\mathrm{d}S_t = rS_t\mathrm{d}t + \sigma S_t \mathrm{d}W_t,$$

we know

$$\begin{aligned}\mathbb{E}[S_{t+\Delta t}|S_t] &= \mathbb{E}[S_t e^{r\Delta t + \sigma \Delta W_t}|S_t] \\ &= S_t e^{r\Delta t + \frac{1}{2}\sigma^2 \Delta t}.\end{aligned}$$

As an approximation of the conditional expectation, we assume:

$$\mathbb{E}[S_{t+\Delta t}|S_t] = \sum_{j=1}^{M} w_{ij} S_{t+\Delta t}(j)$$

so to let w_{ij} match moment conditions, we obtain

$$\sum_{j=1}^{M} w_{ij} S_{t+\Delta t}(j) = S_t e^{r\Delta t + \frac{1}{2}\sigma^2 \Delta t}.$$

In the same way, we write out the cases of 2^{nd} to 4^{th} order moments as follows:

$$\mathbb{E}[S_{t+\Delta t}^2|S_t] = S_t^2 e^{2r\Delta t + \frac{1}{2}(2\sigma)^2 \Delta t}$$

$$\mathbb{E}[S_{t+\Delta t}^3|S_t] = S_t^3 e^{3r\Delta t + \frac{1}{2}(3\sigma)^2 \Delta t}$$

$$\mathbb{E}[S_{t+\Delta t}^4|S_t] = S_t^2 e^{4r\Delta t + \frac{1}{2}(4\sigma)^2 \Delta t}$$

then we can also write out all four linear constraints for node i at time t:

$$\sum_{j=1}^{M} w_{ij} S_{t+\Delta t}(j) = S_t(i) e^{r\Delta t + \frac{1}{2}\sigma^2 \Delta t}.$$

$$\sum_{j=1}^{M} w_{ij} S_{t+\Delta t}^2(j) = S_t^2(i) e^{2r\Delta t + \frac{1}{2}(2\sigma)^2 \Delta t}.$$

$$\sum_{j=1}^{M} w_{ij} S_{t+\Delta t}^3(j) = S_t^3(i) e^{3r\Delta t + \frac{1}{2}(3\sigma)^2 \Delta t}.$$

$$\sum_{j=1}^{M} w_{ij} S_{t+\Delta t}^4(j) = S_t^4(i) e^{4r\Delta t + \frac{1}{2}(4\sigma)^2 \Delta t}.$$

Thus we obtain the parameters B and b for the node i at time t as

$$(B)_{k,j} = S_{t+\triangle t}^k(j)$$

$$b_k = S_t^k(i) e^{kr\Delta t + \frac{1}{2}(k\sigma)^2 \Delta t}$$

where $k = 1, \cdots, 4$, or equivalently,

$$B = \begin{pmatrix} S_{t+\Delta t}(1) & \cdots & S_{t+\Delta t}(j) & \cdots & S_{t+\Delta t}(M) \\ S_{t+\Delta t}^2(1) & \cdots & S_{t+\Delta t}^2(j) & \cdots & S_{t+\Delta t}^2(M) \\ S_{t+\Delta t}^3(1) & \cdots & S_{t+\Delta t}^3(j) & \cdots & S_{t+\Delta t}^3(M) \\ S_{t+\Delta t}^4(1) & \cdots & S_{t+\Delta t}^4(j) & \cdots & S_{t+\Delta t}^4(M) \end{pmatrix}_{4\times M}$$

$$b = \begin{pmatrix} S_t(i) e^{r\Delta t + \frac{1}{2}\sigma^2 \Delta t} \\ S_t^2(i) e^{2r\Delta t + \frac{1}{2}(2\sigma)^2 \Delta t} \\ S_t^3(i) e^{3r\Delta t + \frac{1}{2}(3\sigma)^2 \Delta t} \\ S_t^4(i) e^{4r\Delta t + \frac{1}{2}(4\sigma)^2 \Delta t} \end{pmatrix}_{4\times 1}$$

By using the formula (2.12), we can easily get the solution of the weights.

If applied in the mesh of m independent paths within n time steps as we constructed, the optimized weights between X_{ij} at time step i and every node $X_{i+1,1}, \cdots, X_{i+1,m}$ at time step $i+1$, can be written out as:

$$(W_{j1}^i, \cdots, W_{jm}^i) = B^\top (BB^\top)^{-1} b, \tag{2.13}$$

where

$$B = \begin{pmatrix} X_{i+1,1} & \cdots & X_{i+1,k} & \cdots & X_{i+1,m} \\ X_{i+1,1}^2 & \cdots & X_{i+1,k}^2 & \cdots & X_{i+1,m}^2 \\ X_{i+1,1}^3 & \cdots & X_{i+1,k}^3 & \cdots & X_{i+1,m}^3 \\ X_{i+1,1}^4 & \cdots & X_{i+1,k}^4 & \cdots & X_{i+1,m}^4 \end{pmatrix},$$

and

$$b = \begin{pmatrix} X_{ij} e^{r\Delta t + \frac{1}{2}\sigma^2 \Delta t} \\ X_{ij}^2 e^{2r\Delta t + \frac{1}{2}(2\sigma)^2 \Delta t} \\ X_{ij}^3 e^{3r\Delta t + \frac{1}{2}(3\sigma)^2 \Delta t} \\ X_{ij}^4 e^{4r\Delta t + \frac{1}{2}(4\sigma)^2 \Delta t} \end{pmatrix}.$$

Thus by following the same process at each time step i, we can obtain all the optimized weights.

3 Stochastic mesh method in BSDEs

3.1 Introduction

The introduction of BSDEs starts from the motivation: Characterize the solution of an PDE by solving an SDE.

The semi-linear parabolic PDEs we're interested in is as follows:

$$\partial_t u(t,x) + \mathcal{L}u(t,x) + f(t,x,u(t,x),D_x u(t,x)) = 0 \tag{3.1}$$

where $\mathcal{L}$ is the action of the infinitesimal generator of stochastic process S_t on C^2-functions:

$$\mathrm{d}S_t = b(S_t)\mathrm{d}t + \sigma(S_t)\mathrm{d}W_t$$

and

$$\mathcal{L}\varphi(x) = \sum_{i=1}^{d} b_i(x)\partial_i\varphi(x) + \sum_{i,j=1}^{d} a_{ij}(t,x)\partial_{i,j}\varphi(x)$$

where $a_{i,j} = \frac{1}{2}(\sigma\sigma^T)_{i,j}$

Under certain proper conditions, equation (3.1) has a solution u. Now, let's derive the SDE by describing the dynamics of $Y_t := u(t,S_t)$.

By Itô,

$$\begin{aligned}
\mathrm{d}Y_t &= \mathrm{d}u(t,S_t)\\
&= \partial_t u(t,S_t)\mathrm{d}t + D_x u(t,S_t)\mathrm{d}S_t + \frac{1}{2}D^2_{x,x}u(t,S_t)\mathrm{d}[S,S]^2_t\\
&= \partial_t u(t,S_t)\mathrm{d}t + D_x u(t,S_t)(b(S_t)\mathrm{d}t + \sigma(S_t)\mathrm{d}W_t) + \frac{1}{2}D^2_{x,x}u(t,S_t)\sigma\sigma^T\mathrm{d}t\\
&= (\partial_t u(t,S_t) + D_x u(t,S_t)b(S_t) + \frac{1}{2}D^2_{x,x}u(t,S_t)\sigma\sigma^T)\mathrm{d}t + D_x u(t,S_t)\sigma(S_t)\mathrm{d}W_t\\
&= (\partial_t u(t,S_t) + \mathcal{L}u(t,S_t))\mathrm{d}t + D_x u(t,S_t)\sigma(S_t)\mathrm{d}W_t\\
&= -f(t,S_t,u(t,S_t),D_x u(t,S_t))\mathrm{d}t + D_x u(t,S_t)\sigma(S_t)\mathrm{d}W_t
\end{aligned}$$

Thus we obtain that

$$\mathrm{d}Y_t = -f(t,S_t,u(t,S_t),D_x u(t,S_t))\mathrm{d}t + D_x u(t,S_t)\sigma(S_t)\mathrm{d}W_t$$

If u solves this equation (3.1), for $Y_t = u(t,S_t)$, $Z_t = D_x u(t,S_t)\sigma(S_t)$, we get:

$$\mathrm{d}Y_t = -f(t,S_t,Y_t,Z_t)\mathrm{d}t + Z_t\mathrm{d}W_t. \tag{3.2}$$

A solution of this equation (3.2) consists of a pair of process (Y,Z).

Consider (3.2) as a backward equation, where we know: $Y_T = \xi_T$, Then for the case $f \equiv 0$,

$$\begin{aligned}
Y_t &= \mathbb{E}[Y_t|\mathcal{F}_t]\\
&= \mathbb{E}[\xi_T - \int_t^T Z_s\mathrm{d}W_s|\mathcal{F}_t]\\
&= \mathbb{E}[\xi_T|\mathcal{F}_t]
\end{aligned}$$

So Y_t is a martingale. If the filtration is generated by Brownian motion W_t, then by martingale representation property, there exists a unique predictable process Z, s.t. $Y_t = Y_0 + \int_0^t Z_s \mathrm{d}W_s$, which yields

$$Y_t = Y_T - \int_0^T Z_s \mathrm{d}W_s = \xi_T - \int_0^T Z_s \mathrm{d}W_s.$$

Through this investigation, we obtain an expression of BSDEs, which leads to our next section to talk about the existence and uniqueness of the solution of BSDEs, where we mainly follow the notation from El Karoui et al. (1997).

3.2 Existence and Uniqueness

Let $(\Omega, \mathcal{F}, \mathrm{P})$ be a probability space on which a d-dimensional Brownian motion $W = (W_t)_{t \leq T}$ is defined. Let $(\mathcal{F}_t)_{t \leq T}$ be the completion of $\sigma\{W_s : 0 \leq s \leq t\}$. Let's define the following spaces:
$\mathcal{P}_n$ the set of R^n-valued, $\mathcal{F}_t$-adapted processes on $\Omega \times [0, T]$.
$L_n^2(\mathcal{F}_t) = \{\eta : \mathcal{F}_t - \text{measurable random } \mathrm{R}^n - \text{valued variable s.t. } \mathbb{E}[|\eta|^2] < \infty\}$.
$\mathcal{S}_n^2(0, T) = \{\varphi \in \mathcal{P}_n \text{with continuous paths, s.t. } \mathbb{E}[\sup_{t \leq T} |\varphi_t|^2] < \infty\}$.
$\mathcal{H}_n^2(0, T) = \{Z \in \mathcal{P}_n \text{ s.t. } \mathbb{E}[\int_0^T |Z_s|^2 \mathrm{d}s] < \infty\}$.
$\mathcal{H}_n^1(0, T) = \{Z \in \mathcal{P}_n \text{ s.t. } \mathbb{E}[(\int_0^T |Z_s|^2 \mathrm{d}s)^{\frac{1}{2}}] < \infty\}$.

Definition 1. Let $\xi_T \in L_m^2(\mathcal{F}_t)$ be a $\mathbb{R}^m$-valued terminal condition and let f be $\mathbb{R}^m$-valued, $\mathcal{P}_m \otimes \mathcal{B}(\mathbb{R}^m \times \mathbb{R}^{m \times d})$-measurable. A solution for the m - dimensional BSDE associated with parameters (f, ξ_T) is a pair of adapted processes $(Y, Z) := (Y_t, Z_t)_{t \leq T}$ with values in $\mathbb{R}^m \otimes \mathbb{R}^{m \times d}$ s.t.

$$Y \in \mathcal{S}_m^2, \quad Z \in \mathcal{H}_{m \times d}^2$$

$$-\mathrm{d}Y_t = f(t, Y_t, Z_t)\mathrm{d}t - Z_t \mathrm{d}W_t, \quad Y_T = \xi_T.$$

f is called the driver and ξ the terminal value of the BSDE.

Theorem 1. *(Pardoux and Peng) Under the standard assumption as follows:*
(i) $(f(t, 0, 0))_{t \leq T} \in \mathcal{H}_m^2$
(ii) f *is uniformly Lipschitz with respect to* (y, z)*: there exists a constant* $C \geq 0$ *s.t.*

$$\forall (y, y^{'}, z, z^{'}) \quad |f(t, x, y, z) - f(t, x, y^{'}, z^{'})| \leq C(|y - y^{'}| + |z - z^{'}|), \quad \mathrm{d}t \otimes \mathrm{d}\mathbb{P} \quad a.s.$$

then there exists a unique solution (Y, Z) *of the BSDE with parameters* (f, ξ_T)*.*

Proof See Pardoux and Peng (1990).

QED

3.3 Deduction of the driver

An explicit form of the driver is very critical when solving BSDEs either analytically or numerically (as can be seen in later content). With the basic preliminaries above, we can deduce the driver of BSDEs now. Here we present two examples.

3.3.1 In Black-Scholes model

First, we deduce the driver when the process S_t satisfies Black-Scholes model. Suppose the dynamics of an underlying asset are given by geometric Brownian motion:

$$\mathrm{d}S_t = \mu S_t \mathrm{d}t + \sigma S_t \mathrm{d}W_t$$

where μ and σ is constant, W_t is standard Brownian motion. An agent has the money market account:

$$\mathrm{d}\beta_t = r\beta_t \mathrm{d}t$$

where r is risk-free rate.

By self-financing condition, the wealth process Y_t of the agent satisfies:

$$\mathrm{d}Y_t = a_t \mathrm{d}S_t + b_t \mathrm{d}\beta_t$$

where a_t, b_t are both processes.

We then denote $Y_t = w(t, S_t)$, where $Y_T = w(T, S_T)$ is the terminal condition. By Itô,

$$\begin{aligned}
\mathrm{d}Y_t &= \frac{\partial w}{\partial t}(t,x)\mathrm{d}t + \frac{\partial w}{\partial x}(t,x)\mathrm{d}S_t + \frac{1}{2}\frac{\partial^2 w}{\partial x^2}(t,x)\mathrm{d}[S,S]_t \\
&= \frac{\partial w}{\partial t}\mathrm{d}t + \frac{\partial w}{\partial x}\mu S_t \mathrm{d}t + \frac{\partial w}{\partial x}\sigma S_t \mathrm{d}W_t + \frac{1}{2}\frac{\partial^2 w}{\partial x^2}\sigma^2 S_t^2 \mathrm{d}t \\
&= (\frac{\partial w}{\partial t} + \frac{\partial w}{\partial x}\mu S_t + \frac{1}{2}\frac{\partial^2 w}{\partial x^2}\sigma^2 S_t^2)\mathrm{d}t + \frac{\partial w}{\partial x}\sigma S_t \mathrm{d}W_t
\end{aligned}$$

As we've already known, the self-financing condition implies:

$$\mathrm{d}Y_t = (a_t \mu S_t + b_t r \beta_t)\mathrm{d}t + a_t \sigma S_t \mathrm{d}W_t,$$

then Doob-Meyer decomposition theorem yields:

$$a_t \mu S_t + b_t r \beta_t = \frac{\partial w}{\partial t} + \frac{\partial w}{\partial x}\mu S_t + \frac{1}{2}\frac{\partial^2 w}{\partial x^2}\sigma^2 S_t^2$$

$$a_t \sigma S_t = \frac{\partial w}{\partial x}\sigma S_t$$

thus we obtain:

$$a_t = \frac{\partial w}{\partial x}$$

$$b_t = \frac{Y_t - \frac{\partial w}{\partial x} S_t}{\beta_t}$$

Now, we can rewrite Y_t as in BSDE:

$$\begin{aligned}
Y_T &= Y_t + \int_t^T \mathrm{d}Y_u \\
&= Y_t + \int_t^T \frac{\partial w}{\partial x}\mathrm{d}S_u + \int_t^T \frac{Y_u - \frac{\partial w}{\partial x} S_u}{\beta_u}\mathrm{d}\beta_u \\
&= Y_t + \int_t^T (\frac{\partial w}{\partial x}\mu S_u + rY_u - \frac{\partial w}{\partial x} r S_u)\mathrm{d}u + \int_t^T \frac{\partial w}{\partial x}\sigma S_u \mathrm{d}W_u \\
&= Y_t + \int_t^T (\frac{\partial w}{\partial x}(\mu - r)S_u + rY_u)\mathrm{d}u + \int_t^T \frac{\partial w}{\partial x}\sigma S_u \mathrm{d}W_u
\end{aligned}$$

where $Y_T = w(T, S_T)$.

Thus we obtain:

$$Y_t = Y_T + \int_t^T f(u, Y_u, Z_u)\mathrm{d}u - \int_t^T Z_u \mathrm{d}W_u$$

which can be also expressed in the differential form:

$$-\mathrm{d}Y_t = f(t, Y_t, Z_t)\mathrm{d}t - Z_t \mathrm{d}W_t$$

where

$$f(t, Y_t, Z_t) = \frac{r-\mu}{\sigma} Z_t - rY_t \tag{3.3}$$

with $Z_t = \frac{\partial w}{\partial x}\sigma S_t$.

Example In the case of European option pricing, we suppose that the agent wish to buy an European call option at time t, with the payoff function $h_T = (S_T - K)^+$. So no arbitrage implies that the price of European call option at time t should satisfy:

$$-\mathrm{d}Y_t = (\frac{r-\mu}{\sigma} Z_t - rY_t)\mathrm{d}t - Z_t \mathrm{d}W_t$$

with terminal condition:

$$Y_T = (S_T - K)^+,$$

which can be solved explicitly as:[8]

$$Y_t = S_t \Phi(d_+) - Ke^{-r(T-t)}\Phi(d_-),$$

where

$$d_\pm = \frac{\log(\frac{S_t}{K}) + (r \pm \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{(T-t)}}$$

and Φ is the cumulative distribution function of normal distribution.

3.3.2 In the model with different interest rates

Then, we deduce the driver when there are different interest rates. Similarly, suppose the dynamics of an underlying asset satisfy geometric Brownian motion:

$$\mathrm{d}S_t = \mu S_t \mathrm{d}t + \sigma S_t \mathrm{d}W_t.$$

Now, the agent has two money market accounts:

$$\mathrm{d}\alpha_t = R\alpha_t \mathrm{d}t$$

$$\mathrm{d}\beta_t = r\beta_t \mathrm{d}t$$

where R, r represent borrowing and lending rates respectively, with the assumption $R > r$. Then by self-financing condition, the wealth process Y_t satisfies:

$$\mathrm{d}Y_t = a_t \mathrm{d}S_t + (\frac{Y_t - a_t S_t}{\beta_t})^+ \mathrm{d}\beta_t - (\frac{Y_t - a_t S_t}{\alpha_t})^- \mathrm{d}\alpha_t$$

[8] Black-Scholes formula for the call option.

where a_t is a process.

We then denote $Y_t = w(t, S_t)$ with its terminal case $Y_T = w(T, S_T)$. By Itô, together with self-financing condition, we obtain:

$$\mathrm{d}Y_t = (\frac{\partial w}{\partial t} + \frac{\partial w}{\partial x}\mu S_t + \frac{1}{2}\frac{\partial^2 w}{\partial x^2}\sigma^2 S_t^2)\mathrm{d}t + \frac{\partial w}{\partial x}\sigma S_t \mathrm{d}W_t,$$

then

$$\mathrm{d}Y_t = (a_t\mu S_t + (Y_t - a_t S_t)^+ r - (Y_t - a_t S_t)^- R)\mathrm{d}t + a_t\sigma S_t\mathrm{d}W_t.$$

Thus by Doob-Meyer decomposition theorem, we get $a_t = \frac{\partial w}{\partial x}$, and we can rewrite Y_t as in BSDE:

$$\begin{aligned}
Y_T &= Y_t + \int_t^T \mathrm{d}Y_u \\
&= Y_t + \int_t^T (\frac{\partial w}{\partial x}\mu S_u + (Y_u - \frac{\partial w}{\partial x}S_u)^+ r - (Y_u - \frac{\partial w}{\partial x}S_u)^- R)\mathrm{d}u + \int_t^T \frac{\partial w}{\partial x}\sigma S_u \mathrm{d}W_u \\
&= Y_t + \int_t^T (\frac{\partial w}{\partial x}\mu S_u + (Y_u - \frac{\partial w}{\partial x}S_u) r + (Y_u - \frac{\partial w}{\partial x}S_u)^- r - (Y_u - \frac{\partial w}{\partial x}S_u)^- R)\mathrm{d}u + \int_t^T \frac{\partial w}{\partial x}\sigma S_u \mathrm{d}W_u \\
&= Y_t + \int_t^T (\frac{\partial w}{\partial x}(\mu - r)S_u + rY_u - (Y_u - \frac{\partial w}{\partial x}S_u)^-(R - r))\mathrm{d}u + \int_t^T \frac{\partial w}{\partial x}\sigma S_u \mathrm{d}W_u \\
(\quad \text{or} \quad &= Y_t + \int_t^T (\frac{\partial w}{\partial x}(\mu - R)S_u + RY_u - (Y_u - \frac{\partial w}{\partial x}S_u)^+(R - r))\mathrm{d}u + \int_t^T \frac{\partial w}{\partial x}\sigma S_u \mathrm{d}W_u \quad)
\end{aligned}$$

where $Y_T = w(T, S_T)$.

So we obtain:

$$Y_t = Y_T + \int_t^T f(u, Y_u, Z_u)\mathrm{d}u - \int_t^T Z_u \mathrm{d}W_u$$

which can be also expressed in the differential form:

$$-\mathrm{d}Y_t = f(t, Y_t, Z_t)\mathrm{d}t - Z_t\mathrm{d}W_t$$

where

$$\begin{aligned}
f(t, Y_t, Z_t) &= \frac{r - \mu}{\sigma}Z_t - rY_t + (Y_t - \frac{Z_t}{\sigma})^-(R - r) \\
&= \frac{R - \mu}{\sigma}Z_t - RY_t + (Y_t - \frac{Z_t}{\sigma})^+(R - r)
\end{aligned} \tag{3.4}$$

with $Z_t = \frac{\partial w}{\partial x}\sigma S_t$.

Example Again, in this case of European option pricing with final payoff $h_T = (S_T - K)^+$ at time T , the price of European call option at time t should satisfy:

$$\begin{aligned}
-\mathrm{d}Y_t &= (\frac{r - \mu}{\sigma}Z_t - rY_t + (Y_t - \frac{Z_t}{\sigma})^-(R - r))\mathrm{d}t - Z_t\mathrm{d}W_t \\
&= (\frac{R - \mu}{\sigma}Z_t - RY_t + (Y_t - \frac{Z_t}{\sigma})^+(R - r))\mathrm{d}t - Z_t\mathrm{d}W_t
\end{aligned}$$

with terminal condition:

$$Y_T = (S_T - K)^+.$$

3.4 Recursion in BSDEs

With the drivers deducted, now we apply stochastic mesh method to BSDEs. First of all, we change the BSDEs in a numerical setting to obtain the recursion in BSDEs as follows:

As summarised above[9],

$$-\mathrm{d}Y_t = f(t, Y_t, Z_t)\mathrm{d}t - Z_t \mathrm{d}W_t \tag{3.5}$$

with $Y_T = \xi_T$, where $\mathrm{d}Y_t = Y_{t+\mathrm{d}t} - Y_t$, then we can obtain:

$$Y_t = Y_{t+\mathrm{d}t} + \int_t^{t+\mathrm{d}t} f(t, Y_t, Z_t)\mathrm{d}s - \int_t^{t+\mathrm{d}t} Z_s \mathrm{d}W_s.$$

Now, let's take conditional expectation of $\mathcal{F}_t$ on both sides of the equation and then change "d" to "Δ", numerically,

$$\mathbb{E}[Y_t|\mathcal{F}_t] = \mathbb{E}[Y_{t+\Delta t}|\mathcal{F}_t] + \mathbb{E}[f(t, Y_t, Z_t)\Delta t|\mathcal{F}_t]$$

$\Longrightarrow$

$$Y_t = \mathbb{E}[Y_{t+\Delta t}|\mathcal{F}_t] + f(t, Y_t, Z_t)\Delta t \tag{3.6}$$

Then, multiplying $\mathrm{d}W_t$ to both sides of equation (3.5) yields:

$$-\mathrm{d}Y_t \mathrm{d}W_t = f(t, Y_t, Z_t)\mathrm{d}t\mathrm{d}W_t - Z_t \mathrm{d}[W, W]_t$$

Again, by taking conditional expectation on $\mathcal{F}_t$ and changing "d" to "Δ", we get:

$$\mathbb{E}[(Y_t - Y_{t+\Delta t})\Delta W_t|\mathcal{F}_t] = \mathbb{E}[-Z_t \Delta t|\mathcal{F}_t]$$

i.e.

$$\mathbb{E}[-Y_{t+\Delta t}\Delta W_t|\mathcal{F}_t] = -Z_t \Delta t$$

$\Longrightarrow$

$$Z_t = \frac{1}{\Delta t}\mathbb{E}[Y_{t+\Delta t}\Delta W_t|\mathcal{F}_t] \tag{3.7}$$

where $\Delta W_t = W_{t+\Delta t} - W_t$.

Combine (3.6) and (3.7), then we obtain the recursion. With the terminal condition $Y_T = \xi_T$ known, by choosing the weights to simulate the conditional expectations of both equations and doing backward induction as in the dynamic programming of American option pricing through stochastic mesh method, we can solve the BSDEs. We take examples as illustration.

3.4.1 Linear BSDEs

For general Linear BSDE, $f(t, Y_t, Z_t) = \phi_t + \alpha_t Y_t + \gamma_t Z_t$. Then by (3.6),

$$Y_t = \mathbb{E}[Y_{t+\Delta t}|\mathcal{F}_t] + (\phi_t + \alpha_t Y_t + \gamma_t Z_t)\Delta t$$

$\Longrightarrow$

$$Z_t = \frac{1}{\Delta t}\mathbb{E}[Y_{t+\Delta t}\Delta W_t|\mathcal{F}_t] \tag{3.8}$$

$$Y_t = \frac{1}{1 - \alpha_t \Delta t}(\mathbb{E}[Y_{t+\Delta t}|\mathcal{F}_t] + \phi_t \Delta t + \gamma_t Z_t \Delta t) \tag{3.9}$$

[9]See (3.2).

So if we can know the terminal condition of Y_T and solve the condition expectations of both (3.8) and (3.9), by doing backward induction, we can solve the pair (Y, Z).

Example In a special case of European option pricing, we suppose under risk-neutral probability, the process of underlying asset is given by:

$$\mathrm{d}S_t = rS_t\mathrm{d}t + \sigma S_t\mathrm{d}W_t$$

where r is risk-free rate and σ is constant, and agent's money market account is:

$$\mathrm{d}\beta_t = r\beta_t\mathrm{d}t.$$

We denote $Y_t = w(t, S_t)$ as our wealth process, where $Y_T = w(T, S_T) = (S_T - K)^+$ is the final payoff as in call option case. By the deduction of the driver in Black-Scholes model, i.e. (3.3), we can obtain: $f(t, Y_t, Z_t) = -rY_t$ and then rewrite it as in BSDE:

$$Y_0 = Y_T + \int_0^T (-rY_s)\mathrm{d}s - \int_0^T Z_s\mathrm{d}W_s$$

$$Y_T = (S_T - K)^+$$

where $Z_t = \frac{\partial w}{\partial x}(t, S_t)\sigma S_t$.

Then we get its recursion in this special case.

$\Longrightarrow$

$$Z_t = \frac{1}{\Delta t}\mathbb{E}[Y_{t+\Delta t}\Delta W_t|\mathcal{F}_t]$$

$$Y_t = \frac{1}{1 + r\Delta t}\mathbb{E}[Y_{t+\Delta t}|\mathcal{F}_t]$$

If applied in the mesh of m independent paths within n time steps[10] as we constructed, by choosing weights W^i_{jk} from the mesh , we can rewrite the recursion at node j at time step i as:

$$Z_{ij} = \frac{1}{\Delta t}\sum_{k=1}^{m} W^i_{jk}Y_{i+1,k}(B_{i+1,k} - B_{ij}) \tag{3.10}$$

$$Y_{ij} = \frac{1}{1 + r\Delta t}\sum_{k=1}^{m} W^i_{jk}Y_{i+1,k} \tag{3.11}$$

where B_{ij} is the value of standard Brownian motion at node j at time step i.[11]

3.4.2 Non-linear BSDEs

For other BSDEs, we can by no means follow the same steps as in the LBSDEs case to put a single Y_t on the left hand side of the equation and the expression $Y_{t+\Delta t}$ (without Y_t) on the right hand side of it. However, there are still a lot of applications of other BSDEs in finance, like American option pricing via RBSDEs by Gobert et al. (2005). Here we provide another simple example of

[10] Here the number of time steps n no longer represents the number of exercise dates but the simply the number of times we operate calculation of recursion.

[11] Note: Z_{ij} is actually not involved in the computation of Y_t in this special case.

non-linear BSDEs in finance.

Example Suppose we have an underlying asset, satisfying the geometric Brownian motion:

$$\mathrm{d}S_t = \mu S_t \mathrm{d}t + \sigma S_t \mathrm{d}W_t$$

And there are two constants R and r representing the borrowing and lending interest rates respectively, where $R > r$. By our previous results, i.e. (3.4), we obtain the driver of BSDE in this setting as $f(t, y, z) = -(yr + z\theta - (y - \frac{z}{\sigma})^-(R - r))$, where $\theta = \frac{\mu - r}{\sigma}$. And we can also rewrite it as (non-linear) BSDE:

$$Y_0 = Y_T + \int_0^T -(Y_s r + Z_s\theta - (Y_s - \frac{Z_s}{\sigma})^-(R - r))\mathrm{d}s - \int_0^T Z_s \mathrm{d}W_s$$

$$Y_T = (S_T - K)^+$$

where $\theta = \frac{\mu - r}{\sigma}$.

Then we get its recursion.

$\Longrightarrow$

$$Z_t = \frac{1}{\Delta t}\mathbb{E}[Y_{t+\Delta t}\Delta W_t|\mathcal{F}_t]$$

$$Y_t = \mathbb{E}[Y_{t+\Delta t}|\mathcal{F}_t] - Y_t r\Delta t - Z_t\theta\Delta t + (Y_t - \frac{Z_t}{\sigma})^-(R - r)\Delta t$$

Note: The assumption that $R > r$ is remarkable, otherwise the BSDE (or recursion) of this non-linear case will collapse immediately to the linear case if $R = r$ as we showed in previous content. Since it's a non-linear BSDE (or recursion), we can no longer use the same method to solve it numerically as linear BSDE, however, we provide two ways to deal with it as follows:

- (*Implicit*) In our case mainly to deal with the function $(Y_t - \frac{Z_t}{\sigma})^-(R - r)\Delta t$, we can create a criterion as follows: First, we suppose $Y_t - \frac{Z_t}{\sigma} < 0$, then we rewrite its recursion for Y_t as

  $$Y_t = \frac{1}{1 + r\Delta t - (R - r)\Delta t}\left\{\mathbb{E}[Y_{t+\Delta t}|\mathcal{F}_t] - Z_t\theta\Delta t - \frac{Z_t}{\sigma}(R - r)\Delta t\right\},$$

  thus in which way we can calculate the value of Y_t. Second, we use the value of Y_t to make a comparison with $\frac{Z_t}{\sigma}$: if $Y_t - \frac{Z_t}{\sigma} < 0$ still holds, we just valuate Y_t with the calculated value; otherwise, we should reject the calculated value and then by making the assumption $(Y_t - \frac{Z_t}{\sigma})^- = 0$, valuate Y_t with

  $$Y_t = \frac{1}{1 + r\Delta t}\left\{\mathbb{E}[Y_{t+\Delta t}|\mathcal{F}_t] - Z_t\theta\Delta t\right\}.$$

- (*Explicit*) Instead of creating a criterion, we simply replace Y_t of the non-linear (or even whole) part of the driver with $Y_{t+\Delta t}$ as an estimation, which is also quite intuitive since $Y_{t+\Delta t} \to Y_t$ as $\Delta t \to 0$. This gives us a new recursion which can be solved immediately:

  $$Y_t = \frac{1}{1 + r\Delta t}\left\{\mathbb{E}[Y_{t+\Delta t}|\mathcal{F}_t] - Z_t\theta\Delta t + (Y_{t+\Delta t} - \frac{Z_t}{\sigma})^-(R - r)\Delta t\right\}$$

  $$(\quad \text{or} \quad Y_t = \mathbb{E}[Y_{t+\Delta t}|\mathcal{F}_t] - Y_{t+\Delta t} r\Delta t - Z_t\theta\Delta t + (Y_{t+\Delta t} - \frac{Z_t}{\sigma})^-(R - r)\Delta t \quad).$$

Remark: In more general cases of non-linear BSDEs, we can also use the *explicit* way rather than create criteria (*implicitly*) to deal with the recursion, by simply replacing Y_t with $Y_{t+\Delta t}$ of the non-linear part.

If applied in the mesh of m independent paths within n time steps constructed as before, by choosing weights W_{jk}^i from the mesh , we can rewrite the recursion at node j at time step i as:

$$Z_{ij} = \frac{1}{\Delta t}\sum_{k=1}^{m} W_{jk}^i Y_{i+1,k}(B_{i+1,k} - B_{ij}) \tag{3.12}$$

which is the same as (3.10), where B_{ij} is the value of standard Brownian motion at node j at time step i. However,

$$Y_{ij} = \frac{1}{1+r\Delta t}\left[\sum_{k=1}^{m} W_{jk}^i Y_{i+1,k} - Z_{ij}\theta\Delta t + (Y_{i+1,j} - \frac{Z_{ij}}{\sigma})^-(R-r)\Delta t\right] \tag{3.13}$$

$$(\quad \text{or} \quad Y_{ij} = \sum_{k=1}^{m} W_{jk}^i Y_{i+1,k} - Y_{i+1,j}r\Delta t - Z_{ij}\theta\Delta t + (Y_{i+1,j} - \frac{Z_{ij}}{\sigma})^-(R-r)\Delta t \quad). \tag{3.14}$$

4 Results

Now we present several numerical results to illustrate stochastic mesh method in both American option pricing and BSDEs.

4.1 In American option pricing

4.1.1 One-dimensional example

In order to show how stochastic mesh method works, first we repeat the similar example by Broadie, Glasserman and Ha (2000) to price a one-dimensional American call option as a test, since it could be compared with a known solution by binomial tree method. For each estimator below, we repeat the calculation for *nor* times, denoted by a vector $\vec{a}$, then we compute the empirical mean and standard deviation to form a 95% confident interval as:

$$[\text{mean}(\vec{a}) - 1.96 * \text{std}(\vec{a})/\sqrt{\text{nor}}, \text{mean}(\vec{a}) + 1.96 * \text{std}(\vec{a})/\sqrt{\text{nor}}].$$

Suppose a one-dimensional Black-Scholes model with parameters[12]

r	σ	δ	**T**	S_0	**K**
0.1	0.2	0.05	1	100	100

and final payoff function of an American option: $V_T = (S_T - K)^+$. The results of mesh (high) estimator using weights both from transition densities and from optimization are shown in Table 1[13]. From the table, we note that results with weights from optimization are more time consuming

[12]The parameter δ denotes the dividend factor, which does not violate our deductions in examples we presented in previous content, if we simply replace r with $r - \delta$.

[13]The $-$ and $+$ in the table, represent the lower and upper bound of the 95% confident interval respectively, for estimators with each number of paths.

than those with weights from transition densities under same conditions. The approaching of the mesh estimator with its 95% confidence interval is also shown in Figure 4, in both two methods as the number of mesh increases, from which we conclude that stochastic mesh method works very well in American option pricing with the weights chosen from both ways.

# of repeat: 25	# of time steps: 6		
	via transition density (42.577 s)		
# of paths	mean	-	+
50	11.7205	11.1930	12.2480
100	11.0053	10.4912	11.5195
200	10.4187	10.0870	10.7503
400	10.0450	9.8196	10.2704
800	10.2125	10.0311	10.3940
1600	10.0731	9.9915	10.1548
	via optimization (162.478 s)		
# of paths	mean	-	+
50	9.8709	9.5973	10.1445
100	10.0872	9.8282	10.3462
200	9.8918	9.7572	10.0265
400	10.0234	9.9108	10.1360
800	10.0617	9.9757	10.1478
1600	10.0513	9.9677	10.1349
BIN. TREE	AME CALL PRICE:	10.053692508920713	

Table 1: Results of mesh estimator in one-dimensional American option pricing

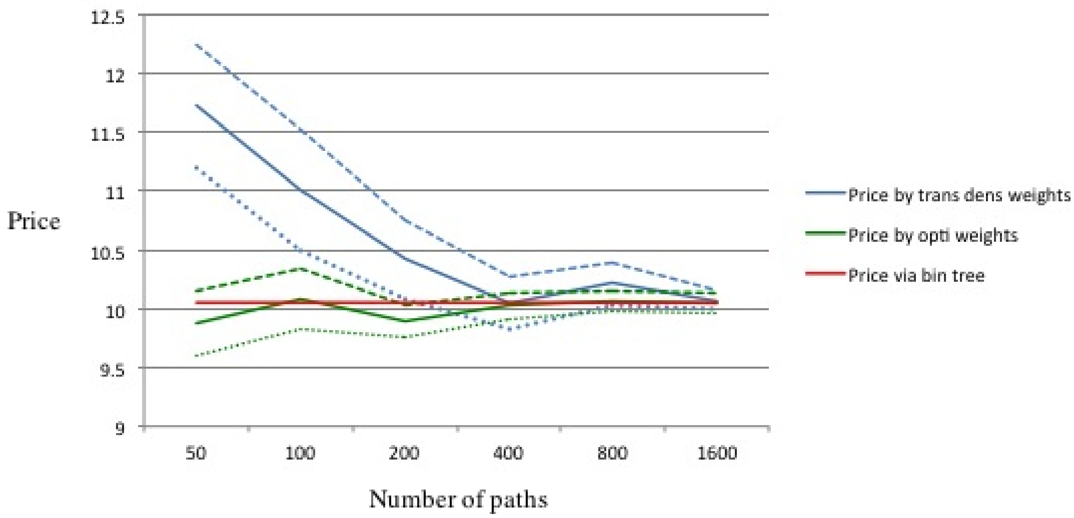

Figure 4: Approaching of mesh estimator by different weights and binomial tree

The results of path (low) estimator and average estimator using weights from transition densities are also shown in Table 2 and Table 3 respectively. From Table 2, compared with the results of Table 1, we find that the path estimator is less biased than mesh estimator, but it takes much more time than the latter one. However, the average estimator is more accurate than any of the other two estimators, but its time consumption is between mesh and path estimators since it blends the characteristics of both estimators as can be seen in Table 3.

# of repeat: 25	# of time steps: 6		
	via transition density (209400.090 s)		
# of paths (m_1/m_2)	mean	-	+
50/500	9.4758	9.2787	9.6729
100/1000	9.7544	9.5919	9.9169
200/2000	9.7649	9.6375	9.8923
400/4000	9.8589	9.7585	9.9592
800/8000	9.8630	9.8088	9.9172
1600/16000	9.8702	9.8222	9.9182
BIN. TREE	AME CALL PRICE: 10.053692508920713		

Table 2: Results of path estimator in one-dimensional American option pricing

# of repeat: 25	# of time steps: 6		
	via transition density (4459.615 s)		
# of paths	mean	-	+
50	9.6891	9.0032	10.3750
100	10.0480	9.6170	10.4791
200	9.6226	9.1454	10.0998
400	9.8839	9.5104	10.2574
800	9.8076	9.6170	9.9982
1600	9.9526	9.8345	10.0707
BIN. TREE	AME CALL PRICE:	10.053692508920713	

Table 3: Results of average estimator in one-dimensional American option pricing

Combining the results from all above, the approaching of three estimators using weights from transition densities is shown in Figure 5.

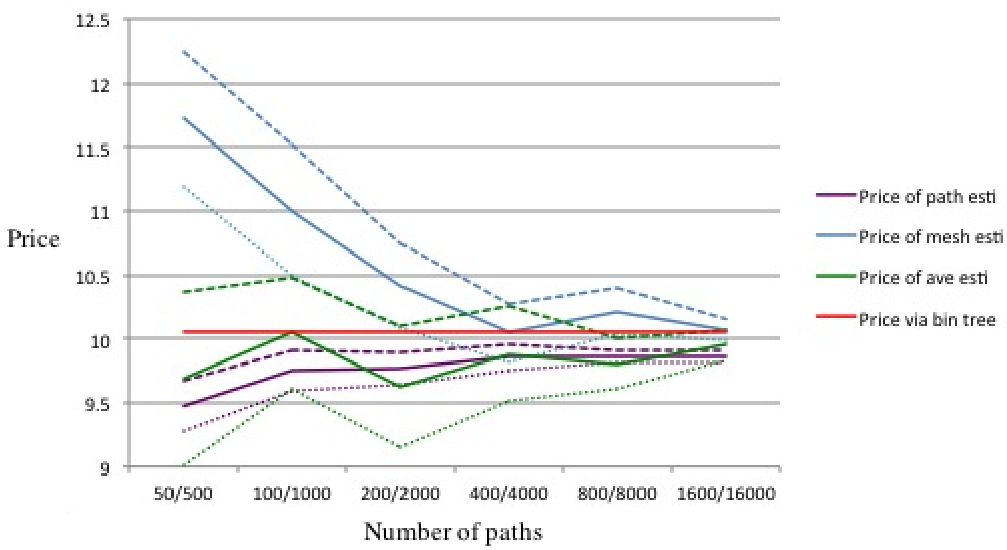

Figure 5: Approaching of three estimators by weights from transition density and binomial tree

4.1.2 Geometric average of multiple underlying assets

Next, we give numerical results of pricing an American geometric average option on 7 independent underlying assets, which are also presented by Broadie and Glasserman (2004), where the problem could be reduced to a single dimension to obtain an accurate answer by binomial tree method, thus could also provide us a test of algorithms in multi-dimensional cases. Here we consider a call option and each the 7 independent assets is identically modeled as geometric Brownian motion with parameters[14]

[14]The parameter D denotes the dimension.

r	σ	δ	T	D	K
0.03	0.4	0.05	1	7	100

and the final payoff function: $V_T = (\sqrt[7]{\prod_{d=1}^{7} S_T^{(d)}} - K)^+$. Accordingly, by the independence of the 7 underlying assets, the transition density from one (7-dimensional) node to another is as the product of one dimensional densities, then from (2.10), we obtain:

$$f_{i+1}(X_{ij}, X_{i+1,k}) = \prod_{d=1}^{7} \frac{1}{X_{i+1,k}^{(d)} \sigma\sqrt{\Delta t}} \phi(\frac{\ln(\frac{X_{i+1,k}^{(d)}}{X_{ij}^{(d)}}) - (r - \frac{1}{2}\sigma^2)\Delta t}{\sigma\sqrt{\Delta t}})$$

where $X_{ij} = (X_{ij}^{(1)}, X_{ij}^{(2)}, \cdots, X_{ij}^{(7)})$ at time step i and $X_{i+1,k} = (X_{i+1,k}^{(1)}, X_{i+1,k}^{(2)}, \cdots, X_{i+1,k}^{(7)})$ at time step $i+1$. The results of mesh estimator using weights from transition densities with different initial values and number of time steps are shown in Table 4.

# of repeat: 25									
$S_0 = 90$									
	# of time steps: 4 (293.747 s)			# of time steps: 6 (452.646 s)			# of time steps: 8 (566.174 s)		
# of paths	mean	-	+	mean	-	+	mean	-	+
50	0.9658	0.8150	1.1166	1.0895	0.8833	1.2957	1.2961	1.0697	1.5225
100	0.9807	0.8665	1.0948	0.9137	0.7993	1.0281	1.1796	1.0603	1.2988
200	0.8861	0.8088	0.9634	1.0664	0.9870	1.1459	1.1068	0.9957	1.2180
400	0.8354	0.7762	0.8946	1.0488	0.9840	1.1137	1.1157	1.0607	1.1707
800	0.8158	0.7801	0.8514	0.9997	0.9621	1.0373	1.1107	1.0694	1.1520
1600	0.8247	0.8023	0.8471	0.9498	0.9279	0.9717	1.0864	1.0545	1.1184
BIN. TREE	AME CALL PRICE:	0.761							
$S_0 = 100$									
	# of time steps: 4 (285.675 s)			# of time steps: 6 (454.981 s)			# of time steps: 8 (556.159 s)		
# of paths	mean	-	+	mean	-	+	mean	-	+
50	4.1503	3.7816	4.5190	4.6592	4.2419	5.0765	5.3759	5.0389	5.7130
100	3.7468	3.4938	3.9999	4.6118	4.3511	4.8725	4.9529	4.6670	5.2389
200	3.6331	3.4212	3.8450	4.5487	4.3791	4.7182	4.8993	4.6449	5.1536
400	3.4647	3.3614	3.5680	4.2671	4.1445	4.3897	4.8822	4.7282	5.0362
800	3.4253	3.3445	3.5060	4.1533	4.0613	4.2454	4.6781	4.5729	4.7832
1600	3.3338	3.2855	3.3821	3.9499	3.8881	4.0118	4.4627	4.3892	4.5363
BIN. TREE	AME CALL PRICE:	3.270							
$S_0 = 110$									
	# of time steps: 4 (292.912 s)			# of time steps: 6 (445.394 s)			# of time steps: 8 (557.037 s)		
# of paths	mean	-	+	mean	-	+	mean	-	+
50	10.7675	10.2293	11.3057	12.2434	11.6557	12.8310	13.7896	13.1735	14.4058
100	9.7591	9.3973	10.1208	12.1251	11.7418	12.5083	13.9677	13.5557	14.3796
200	9.8885	9.6410	10.1359	11.7953	11.5329	12.0577	13.6119	13.3116	13.9121
400	9.3339	9.1293	9.5384	11.2289	11.0398	11.4180	12.7890	12.5953	12.9826
800	9.1460	9.0583	9.2337	10.7223	10.6145	10.8302	12.1793	12.0390	12.3196
1600	8.9063	8.8079	9.0047	10.2400	10.1528	10.3272	11.4677	11.3671	11.5683
BIN. TREE	AME CALL PRICE:	10.000							

Table 4: Results of mesh estimator in 7-dimensional American geometric average option pricing

From the table, we note that the results are basically matched with those of binomial tree, and the time consumption is nearly linear with respect to the number of time steps. However, the accuracy of our results is far from satisfactory, since to which the number of time steps disturbs remarkably in our algorithms, which also means there needs further research for adjustments or enhancements as presented by Broadie and Glasserman (2004).

4.1.3 Maximum of multiple underlying assets

Similarly, we consider an American maximum option on 5 independent underlying assets, of which each is identically modeled as geometric Brownian motion with parameters

r	σ	δ	**T**	**D**	S_0	**K**
0.05	0.2	0.1	3	5	100	100

and the final payoff function: $V_T = (\max_d S_T^{(d)} - K)^+$, where $d = 1, \cdots, 5$. The results of mesh estimator using weights from transition densities are shown in Table 5. Since there is no exact result of this problem, we take those by Broadie and Glasserman (2004) as a comparison. We note that our results become more closed to theirs as number of paths increases, and most closed to it when the number of paths is 3200, i.e. 25.274.

# of repeat: 25	# of time steps: 4 (743.542 s)		
# of paths	mean	-	+
50	30.8093	29.6121	32.0065
100	30.6224	29.6152	31.6295
200	29.2292	28.5765	29.8820
400	28.3005	27.9720	28.6290
800	27.7021	27.3948	28.0095
1600	26.6629	26.4663	26.8595
3200	26.1855	26.0762	26.2949

Table 5: Results of mesh estimator in 5-dimensional American maximum option pricing

4.2 In BSDEs

Now, we present some examples of option pricing by stochastic mesh method in BSDEs, which are mainly in European style as their drivers we've deducted in previous content. Since optimal stopping problems are not involved in our discussion here, we mainly obtain the results of mesh estimators rather than those of the other two estimators.

4.2.1 One-dimensional expamples

European option pricing

First, We show a one-dimensional European option pricing by stochastic mesh method in BSDE as a test of algorithms. Suppose a one-dimensional Black-Scholes model with parameters

r	σ	T	S_0	K
0.01	0.3	1	100	100

and the final payoff function of a European option: $Y_T = (K - S_T)^+$ (or $Y_T = (S_T - K)^+$). Since under risk-neutral probability, the driver in this model is: $f(t, Y_t, Z_t) = -rY_t$, then we simulate the conditional expectations choosing weights both from transition densities as well as optimization, and operate the backward induction of the BSDE(LBSDE) to obtain the mesh estimator. The results are reported in Table 6 and Table 7, which show put and call respectively, both compared with Black-Scholes price and bin tree price as well. And from the tables, we note that results with weights from optimization are more time consuming that those with weights from transition densities, the same as we summarized in American option pricing. We finally conclude that the stochastic mesh method in BSDE works well in European option pricing.

# of repeat: 250	# of time steps: 11		
	via transition density (966.120 s)		
# of mesh	mean	-	+
50	7.3377	7.1297	7.5456
100	7.1914	7.0514	7.3313
200	7.3183	7.225	7.4115
400	7.3239	7.2528	7.3949
800	7.3191	7.2682	7.37
1600	7.2967	7.2621	7.3313
	via optimization (3269.783 s)		
# of mesh	mean	-	+
50	5.2054	4.9473	5.4635
100	6.1868	6.0345	6.3391
200	6.6665	6.5812	6.7518
400	6.909	6.854	6.9641
800	7.0696	7.0353	7.104
1600	7.0907	7.0686	7.1128
BIN. TREE	EUR PUT PRICE:	7.412492924175051	
BLACK-SCHOLES	EUR PUT PRICE:	7.217875919378269	

Table 6: Results of European put option pricing

# of repeat: 250	# of time steps: 11		
	via transition density (1008.034 s)		
# of mesh	mean	-	+
50	16.9271	16.5328	17.3214
100	16.9405	16.6713	17.2096
200	16.8508	16.6336	17.0679
400	16.7598	16.6087	16.911
800	16.8473	16.7414	16.9532
1600	16.9198	16.8453	16.9944
	via optimization (3349.198 s)		
# of mesh	mean	-	+
50	15.5993	15.4091	15.7896
100	16.3935	16.2812	16.5059
200	16.4425	16.3642	16.5208
400	16.7464	16.6989	16.7939
800	16.7729	16.7387	16.807
1600	16.8104	16.7886	16.8322
BIN. TREE	EUR CALL PRICE:	16.928751120579168	
BLACK-SCHOLES	EUR CALL PRICE:	16.734134115782318	

Table 7: Results of European call option pricing

European option with different interest rates

We now consider a one dimensional Black-Scholes model with parameters

μ	σ	**r**	**R**	**T**	S_0	**K**
0.06	0.2	0.04	0.06	0.5	100	100

and payoff function of a European option: $Y_T = (S_T - K)^+$, where R and r represent the borrowing and lending interest rates respectively. According to (3.4), the drivers of this model are:

$$
\begin{aligned}
f(t, Y_t, Z_t) &= \frac{r-\mu}{\sigma} Z_t - rY_t + (Y_t - \frac{Z_t}{\sigma})^-(R-r) \\
&= \frac{R-\mu}{\sigma} Z_t - RY_t + (Y_t - \frac{Z_t}{\sigma})^+(R-r).
\end{aligned}
$$

Since it's non-linear, for the simplicity of calculation, we use the *explicit* way to solve it numerically.[15] Specifically, Y_0 can be given by the Black-Scholes formula evaluated with interest rate R.[16] Then the results of mesh estimators under both drivers in (non-linear) BSDE are presented in Table 8 and Table 9, compared with the Black-Scholes price. From the tables, we note that results of both drivers match together.

[15] The same way as we use in the following content.

[16] Gobet, E., Lemor, J.P. and Warin, X (2005). A regression-based Monte Carlo method to solve backward stochastic differential equations. *The Annals of Applied Probability* .

# of repeat: 250	# of time steps: 6		
	via transition density (495.592 s)		
# of mesh	mean	-	+
50	7.1310	6.9596	7.3023
100	7.1796	7.0611	7.2980
200	7.2116	7.1253	7.2979
400	7.1507	7.0885	7.2129
800	7.1503	7.1067	7.1939
1600	7.1501	7.1189	7.1812
	via optimization(1713.825 s)		
# of mesh	mean	-	+
50	7.1755	7.1542	7.1969
100	7.1508	7.1376	7.1640
200	7.1791	7.1695	7.1887
400	7.1759	7.1696	7.1822
800	7.1806	7.1755	7.1856
1600	7.1824	7.1788	7.1860
BLACK-SCHOLES	EUR CALL PRICE(with R):	7.15	

Table 8: Results of European call option pricing with different interest rates under 1st driver

# of repeat: 250	# of time steps: 6		
	via transition density (503.350 s)		
# of mesh	mean	-	+
50	7.1334	6.9562	7.3106
100	7.0978	6.9638	7.2318
200	7.1921	7.1087	7.2756
400	7.1212	7.0607	7.1816
800	7.1858	7.1446	7.2271
1600	7.1360	7.1053	7.1667
	via optimization(1739.820 s)		
# of mesh	mean	-	+
50	7.1813	7.1605	7.2021
100	7.1577	7.1441	7.1713
200	7.1617	7.1518	7.1716
400	7.1811	7.1751	7.1871
800	7.1730	7.1680	7.1779
1600	7.1807	7.1773	7.1840
BLACK-SCHOLES	EUR CALL PRICE(with R):	7.15	

Table 9: Results of European call option pricing with different interest rates under 2nd driver

European combination with different interest rates

We again consider a one dimensional Black-Scholes model with parameters

μ	σ	**r**	**R**	**T**	S_0	K_1	K_2
0.05	0.2	0.01	0.06	0.25	100	95	105

and payoff function of a European option: $Y_T = (S_T - K_1)^+ - 2(S_T - K_2)^+$, where K_1 and K_2 are the strike prices, R and r represent the borrowing and lending interest rates respectively. The drivers of the model are the same as those in prior example. Then the results of mesh estimators under the 1st driver in (Non-Linear) BSDE are shown in Table 10, compared with the estimated Black-Scholes price by Gobet et al. (2005).

# of repeat: 250	# of time steps: 6		
	via transition density (461.684 s)		
# of mesh	mean	-	+
50	2.9568	2.8809	3.0327
100	2.9158	2.8618	2.9699
200	2.9035	2.8671	2.9399
400	2.9185	2.8940	2.9431
800	2.9192	2.9007	2.9377
1600	2.9245	2.9101	2.9388
	via optimization(1666.420 s)		
# of mesh	mean	-	+
50	2.6433	2.5733	2.7134
100	2.8161	2.7767	2.8554
200	2.9051	2.8819	2.9283
400	2.9308	2.9137	2.9478
800	2.9524	2.9399	2.9649
1600	2.9644	2.9563	2.9726
ESTIMATED BS PRICE:	2.95		

Table 10: Results of European call combination option pricing with different interest rates

4.2.2 Geometric average of multiple underlying assets

Next, we give an example of multi-dimensional case. Here we consider a European geometric average call option on 7 independent assets, of which each is identically modeled as the same geometric Brownian motion with parameters

r	σ	δ	**T**	**D**	S_0	**K**
0.03	0.4	0.05	1	7	100	100

and payoff function: $Y_T = (\sqrt[7]{\prod_{d=1}^{7} S_T^{(d)}} - K)^+$. The driver of this model is also: $f(t, Y_t, Z_t) = -rY_t$, and the weights are chosen only from transition densities. Since the Brownian motion generated in this example is 7-dimensional, which is actually formed by 7 independent one-dimensional Brownian motions, then we take the average of them in our calculation of mesh estimator. The results are presented in Table 11. As known in previous American geometric average option pricing, this problem can be reduced to a single dimension one, where binomial tree method provide us an accurate value[17], i.e. 2.419.

[17] Broadie, M. and Glasserman P. (2004). A stochastic mesh method for pricing high-dimensional American options. *Journal of Computational Finance.*

# of repeat: 25	$S_0 = 100$								
	# of time steps: 4 (284.054 s)			# of time steps: 6 (418.238 s)			# of time steps: 8 (546.969 s)		
# of paths	mean	-	+	mean	-	+	mean	-	+
50	2.4659	2.0373	2.8944	2.4405	2.1534	2.7277	2.3497	2.0675	2.6319
100	2.3657	2.1851	2.5463	2.5171	2.2687	2.7655	2.3124	2.0424	2.5823
200	2.3718	2.2065	2.5370	2.3612	2.2302	2.4923	2.2760	2.1263	2.4258
400	2.4391	2.3117	2.5665	2.3713	2.2361	2.5065	2.4418	2.2958	2.5877
800	2.3671	2.2634	2.4709	2.3973	2.3081	2.4866	2.4571	2.3738	2.5405
1600	2.4460	2.3884	2.5035	2.3986	2.3369	2.4603	2.4191	2.3649	2.4734
BIN. TREE	EUR CALL PRICE:	2.419							

Table 11: Results of mesh estimator in 7-dimensional European geometric average option pricing

From the table, we note that in this example of European geometric average option pricing, we succeed to avoid the disturbance of the number of time steps to our results, which happens before in American case. Besides, the time consumption is nearly linear with respect to the number of time steps as well. Moreover, the accuracy here is also acceptable.

4.2.3 Maximum of multiple underlying assets

Similarly, we consider a European maximum call option on 5 independent underling assets, of which each is identically modeled as geometric Brownian motion with parameters

r	σ	δ	**T**	**D**	S_0	**K**
0.05	0.2	0.1	3	5	100	100

and the final payoff function: $Y_T = (\max_d S_T^{(d)} - K)^+$, where $d = 1, \cdots, 5$. The driver of the model is the same as prior one. This kind of option is priced by Johnson (1987), and we take the results by Broadie and Glasserman (2004) as a comparison. The results of mesh estimator using weights from transition densities are shown in Table 12.

# of repeat: 25	# of time steps: 4 (742.986 s)		
# of paths	mean	-	+
50	23.8665	22.6327	25.1003
100	25.0915	24.0018	26.1811
200	24.5982	23.8875	25.3088
400	24.2702	23.8931	24.6472
800	24.2186	23.8358	24.6014
1600	24.3495	24.1261	24.5729
3200	24.4619	24.3043	24.6194
B&G	EUR CALL PRICE:	23.052	

Table 12: Results of mesh estimator in 5-dimensional European maximum option pricing I

As mentioned in our previous discussion, the number of time steps n doesn't represent the exercise dates but the number of times we operate the calculation of recursion of BSDE, so $n = 4$

might be too small to provide us accurate results, then the results of $n = 8$ with the same other parameters are shown in Table 13, which are indeed more accurate.

# of repeat: 25	# of time steps: 8 (2299.470 s)		
# of paths	mean	-	+
50	23.0482	21.7225	24.3739
100	23.6059	22.4657	24.7460
200	23.2390	22.4541	24.0240
400	23.9167	23.2676	24.5657
800	23.6582	23.3076	24.0088
1600	23.6666	23.4124	23.9208
3200	23.5832	23.4323	23.7340
B&G	EUR CALL PRICE:	23.052	

Table 13: Results of mesh estimator in 5-dimensional European maximum option pricing II

4.2.4 Geometric average of multiple underlying assets with different interest rates

Now we present an example, which combines European call option with different interest rates and geometric average of 7 independent underlying assets. Suppose each of the assets is identically modeled by geometric Brownian motion with parameters

μ	σ	**r**	**R**	**T**	**D**	S_0	**K**
0.06	0.2	0.04	0.06	0.5	7	100	100

and the final payoff function: $Y_T = (\sqrt[7]{\prod_{d=1}^{7} S_T^{(d)}} - K)^+$, where R and r represent the borrowing and lending interest rates respectively. From above, here we take the first driver of this model:

$$f(t, Y_t, Z_t) = \frac{r - \mu}{\sigma} Z_t - rY_t + (Y_t - \frac{Z_t}{\sigma})^-(R - r).$$

The results of mesh estimators using weights from transition densities in (non-linear) BSDE are presented in Table 14.

# of repeat: 25	# of time steps: 6 (406.489 s)		
# of paths	mean	-	+
50	2.6725	2.4672	2.8779
100	2.5765	2.4581	2.6950
200	2.6629	2.5928	2.7330
400	2.7300	2.6693	2.7908
800	2.7053	2.6609	2.7497
1600	2.7184	2.6787	2.7580

Table 14: Results of 7-dimensional European geo-average option pricing with different interest rates

4.2.5 Maximum of multiple underlying assets with different interest rates

Similarly, we consider a European maximum call option with different interest rates on 5 independent underlying assets, of which each is identically modeled as geometric Brownian motion with parameters

μ	σ	**r**	**R**	**T**	**D**	S_0	**K**
0.06	0.2	0.04	0.06	0.5	5	100	100

and the final payoff function: $Y_T = (\max_d S_T^{(d)} - K)^+$, where $d = 1, \cdots, 5$, R and r represent the borrowing and lending interest rates respectively. The driver of the model is the same as that in prior example. Then the results of mesh estimators using weights from transition densities in (non-linear) BSDE are presented in Table 15.

# of repeat: 25	# of time steps: 6 (281.909 s)		
# of paths	mean	-	+
50	19.3563	18.8002	19.9123
100	19.1840	18.6900	19.6780
200	19.6158	19.3000	19.9316
400	19.4089	19.2242	19.5937
800	19.2094	19.0375	19.3813
1600	19.2389	19.1526	19.3252

Table 15: Results of 5-dimensional European maximum option pricing with different interest rates

4.2.6 Geometric average combination of multiple underlying assets with different interest rates

Finally, we consider a European geometric average call combination option on 7 independent underlying assets, of which each is identically modeled as geometric Brownian motion with parameters

μ	σ	**r**	**R**	**T**	**D**	S_0	K_1	K_2
0.05	0.2	0.01	0.06	0.25	7	100	95	105

and payoff function of a European option: $Y_T = (\sqrt[7]{\prod_{d=1}^{7} S_T^{(d)}} - K_1)^+ - 2(\sqrt[7]{\prod_{d=1}^{7} S_T^{(d)}} - K_2)^+$, where K_1 and K_2 are the strike prices, R and r represent the borrowing and lending interest rates respectively. The driver of the model is also the same as prior one. Then the results of mesh estimators in (non-linear) BSDE are shown in Table 16.

# of repeat: 25	# of time steps: 6 (419.701 s)		
# of paths	mean	-	+
50	4.8434	4.6878	4.9990
100	4.7504	4.6432	4.8576
200	4.6789	4.5952	4.7627
400	4.7198	4.6541	4.7855
800	4.6977	4.6648	4.7307
1600	4.6819	4.6569	4.7069

Table 16: Results of 7-d European geo-ave combination option pricing with different interest rates

Conclusion

In this paper we show the application of stochastic mesh method in BSDEs. We review the origin of this method, which is developed for pricing American option. We introduce the BSDEs and detail the process of deduction of drivers and recursion in BSDEs, where we finally apply stochastic mesh method. Numerical results are also included to illustrate the performance of the method in BSDEs of some examples, particularly, in derivative pricing. Although we're mainly concerned with the European option problem, we can still go a bit further in future research for the other types of option pricing to check the availability of this method in BSDEs, for example, the American option pricing in reflected BSDEs as mentioned, of which we can obtain the results directly by stochastic mesh method as well. Besides, there are a lot more fields we can apply our method as long as we can solve the BSDEs, for instance, in complete markets, the Föllmer-Schweizer strategy is just given by the solution of a BSDE[18]. Even on the efforts of previous research, there are still quite a few parts that can be improved in the method. One possible way suggested by Broadie is that we might try to reduce the connection between the node j at time step i and all the nodes at time step $i+1$ to that between the node j at time step i and some fixed (not all) nodes at time step $i+1$. This is remarkable, since it reduce the calculation effort quadratic to the number of nodes at each time step into linear, but the definition of the weights need to be concerned indeed.

[18] Gobet, E., Lemor, J.P. and Warin, X (2005). A regression-based Monte Carlo method to solve backward stochastic differential equations. *The Annals of Applied Probability.*

References

Avramidis, A. N. and Hyden, P. (1999). Efficiency improvements for pricing American options with a stochastic mesh. *Proceedings of the 31st conference on Winter simulation: Simulation — a bridge to the future* **1**, 344-350.

Avramidis, A. N. and Hyden, P. (1999). Efficient Simulation Techniques for Pricing American Option. *Unpublished manuscript.*

Avramidis, A. N. and Matzinger H. (2004). Convergence of the stochastic mesh estimator for pricing Bermudan options. *Journal of Computational Finance* **7**(2), 73-91.

Bally, V. (1997). Approximation scheme for solutions of BSDE. *Pitman Research Notes in Mathematics Series* **364**, 177-191.

Bally, V. and Pagès, G. (2003). A quantization algorithm for solving multidimensional discrete-time optimal stopping problems. *Bermoulli* **9**, 1003-1049.

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

Bismut, J.M. (1973). Conjugate Convex Functions in Optimal Stochastic Control. *Journal of Mathematical Analysis and Applications* **44**, 384-404.

Black, F. and Scholes, M. (1973). The pricing of options and corporate liabilities. *Journal of Political Economy* **81**, 637-654.

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

Broadie, M. and Glasserman P. (1997). Pricing Amrican-style securities by simulation. *Journal of Economic Dynamics and Control* **21**, 1323-1352.

Broadie, M. and Glasserman P. (2004). A stochastic mesh method for pricing high-dimensional American options. *Journal of Computational Finance* **7**(4), 35-72.

Broadie, M., Glasserman P. and Ha, Z. (2000). Pricing American options by simulation using a stochastic mesh with optimized weights. *Probabilistic Constrained Optimization*, 32-50.

Broadie, M. and Yamamoto, Y (2003). Application of the fast Gauss transform to option pricing. *Management Science* **49**(8), 1071-1088.

El Karoui, N., Peng, S. and Quenez, M.C. (1997). Backward Stochastic Differential Equations in Finance. *Mathematical Finance* **7**, 1-71.

El Karoui, N., Hamadène, S. and Matoussi, A. (2008). Backward stochastic differential equations and applications. *Springer* **27**, 267-320.

Föllmer, H. and Schweizer, M. (1990). Hedging of contingent claims under incomplete information. *Applied Stochastic Analysis* **5**, 389-414.

Glasserman, P. (2004). Monte Carlo Methods in Financial Engineering. *Springer.*

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

Johnson, H. (1987). Options on the maximum or the minimum of several assets. *Journal of Financial and Quantitative Analysis* **22**(3), 277-283.

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

Merton, R. (1973). Theory of Rational Option Pricing. *Bell Journal of Economics and Management Science* **4**, 141-183.

Merton, R. (1991). Continuous Time Finance. *Basil Blackwell.*

Pardoux, E. and Peng, S. (1990). Adapted Solution of a Backward Stochastic Differential Equation. *Systems & Control Letters* **14**, 55-61.

Pardoux, E. and Peng, S. (1992). Backward Stochastic Differential Equations and Quasilinear Parabolic Partial Differential Equations. *Lecture Notes in CIS* **176**, 200-217.

Peng, S. (2003). Dynamically consistent evaluations and expectations. *Technical report, Institute Mathematics, Shandong University.*

Peng, Y., Gong, B., Liu, H. and Zhang, Y. (2010). Parallel computing for option pricing based on the backward stochastic differential equation. *High Performance Computing and Applications*, 325-330.