CARF Working Paper

CARF-F-423

Asymptotic Expansion as Prior Knowledge in Deep Learning Method for high dimensional BSDEs

Masaaki Fujii
Quantitative Finance Course, Graduate School of Economics,
The University of Tokyo

Akihiko Takahashi
Quantitative Finance Course, Graduate School of Economics,
The University of Tokyo

Masayuki Takahashi
Quantitative Finance Course, Graduate School of Economics,
The University of Tokyo

First version: October 19, 2017
This version: October 28, 2017

CARF is presently supported by The Dai-ichi Life Insurance Company, Limited, Nomura Holdings, Inc., Sumitomo Mitsui Banking Corporation, Bank of Tokyo-Mitsubishi UFJ, Finatext Ltd., GCI Asset Management, Inc and The University of Tokyo Edge Capital Co.. This financial support enables us to issue CARF Working Papers.

Asymptotic Expansion as Prior Knowledge in Deep Learning Method for high dimensional BSDEs

Masaaki Fujii[*], Akihiko Takahashi[†], Masayuki Takahashi[‡]

First version: October 19, 2017
This version: October 28, 2017

Abstract

We demonstrate that the use of asymptotic expansion as prior knowledge in the *"deep BSDE solver"*, which is a deep learning method for high dimensional BSDEs proposed by Weinan E, Han & Jentzen (2017), drastically reduces the loss function and accelerates the speed of convergence. We illustrate the technique and its implications by Bergman's model with different lending and borrowing rates, and a class of quadratic-growth BSDEs. We also present an extension of the deep BSDE solver for reflected BSDEs using an American basket option as an example.

Keywords : backward stochastic differential equations, reflected BSDEs, non-linear partial differential equations, machine learning, deep neural network

1 Introduction

Since the pioneering works of Bismut (1973) [6] and Pardoux & Peng (1990) [36], backward stochastic differential equations (BSDEs) have attracted many researchers by their deep connections to non-linear partial differential equations and stochastic control problems. Many excellent monographs written on various topics on BSDEs are now available, for example, El Karoui & Mazliak (eds.) (1997) [16], Ma & Yong (2000) [32], Delong (2013) [15], Pardoux & Rascanu (2014), and Zhang (2017) [43]. The relevance of BSDEs for financial problems has also increased recently, particularly since the financial crisis. Early attempts such as Fujii & Takahashi (2012, 2013) [20, 23] and Crepey (2015) [11] have shown that BSDEs are indispensable tools to describe the non-linear effects in various valuation adjustments stemming from collateralization, credit risks, funding and regulatory costs. See Brigo, Morini & Pallavicini (2013) [8] and Crepey & Bielecki (2014) [10] as reviews on the financial problems closely related to BSDEs.

The progress in numerical computation schemes for BSDEs has also been significant. The famous $\mathbb{L}^2$-regularity established by Zhang (2001, 2004) [42, 41] was soon followed by now standard regression based Monte-Carlo simulation scheme developed, among others, by Bouchard & Touzi (2004) [7], Gobet, Lemor & Warin (2005) [28]. There now exist many extensions of these fundamental works to various types of BSDEs. Unfortunately, however, there still remains a big obstacle which has been hindering the successful applications of BSDEs to the realistic financial as well as engineering problems. Although the proposed

[*]Quantitative Finance Course, Graduate School of Economics, The University of Tokyo.
[†]Quantitative Finance Course, Graduate School of Economics, The University of Tokyo.
[‡]Quantitative Finance Course, Graduate School of Economics, The University of Tokyo.

schemes guarantee the convergence in the limit of fine discretization and many sample paths, applications to high dimensional problems required in the practical setups have been almost infeasible due to their numerical burden. This is the notorious problem called the *"curse of dimensionality"* .

A potential breakthrough may come from the recent boom as well as explosive progress of reinforcement machine learning, which makes use of deep neutral networks mimicking the cognitive mechanism of human brains. In fact, Weinan E, Han & Jentzen (2017) [19] motivated by the work of Weinan E & Han (2016) [18] have just demonstrated astonishing power of "deep BSDE solver" for high dimensional problems, which is based on the deep neural networks constructed by the free package *Tensorflow*. The details of the algorithm are explained in Sections 2 and 3, and concrete source codes are available in Appendices in their work [19]. Their main idea is to interpret a Markovian BSDE

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

as a control problem minimizing the square difference $|\Phi(X_T) - \widehat{Y}_T|^2$. Here, $\Phi(X_T)$ is the terminal condition and $\widehat{Y}_T$ the terminal value of forwardly simulated process $(Y_t)_{t\in[0,T]}$ based on the estimated initial value Y_0 as well as the coefficients of Brownian motions $(Z_t)_{t\in[0,T]}$, which are treated as the *control* variables in the minimization problem. Although the mathematical understanding for the deep learning algorithm is still in its infancy, the deep BSDE solver seems to be capable of handling quite high dimensional problems very efficiently in a straightforward manner [1].

Despite its remarkable success for high dimensional problems, it is not free from some important issues. By closely studying the deep BSDE solver given in [19], we find that its direct application to a simple financial problem of Bergman (1995) [5] with different lending and borrowing rates yields persistently high loss function $|\Phi(X_T) - \widehat{Y}_T|^2$ even when the estimated Y_0, which corresponds to the price of the contingent claim, is quite accurate. The slow convergence and large loss function seem to arise partly from non-smooth terminal conditions as well as drivers of BSDEs, which are ubiquitous in financial applications. From the financial viewpoint, the loss function is the square of "replication error" from the dynamic delta-hedging strategy using the estimated $(Z_t)_{t\in[0,T]}$. Therefore, even when Y_0 is known to be accurate, the resultant strategy is not useful when the loss function remains to be high. Worsely, we do not know the accurate value of Y_0 in general. Only available criterion at hand is the famous stability result of BSDEs to guarantee the uniqueness of their solutions, which thus requires the convergence of the loss function to a sufficiently small value. Since many of the financial problems related to the valuation adjustments i.e. XVAs have quite similar form to the Bergman's model, this is not an exceptional problem.

It has been widely known that the prior knowledge to prepare the starting point of the learning process significantly affects the performance of deep learning methods. In this work, we demonstrate that a simple approximation formula based on an asymptotic expansion (AE) of BSDEs serves as very efficient prior knowledge for the deep BSDE solver. Using the method proposed in the works Fujii & Takahashi(2012) [21, 25], one obtains an analytic expression of approximate Z^{AE}. We write $Z = Z^{\mathrm{AE}} + Z^{\mathrm{Res}}$ and apply the reinforcement learning only to the residual term Z^{Res}. We shall show that the use of Z^{AE} drastically reduces the loss function and accelerates the speed of convergence. Moreover, for some examples, the direct application of the deep BSDE solver seems to have some bias in the estimated Y_0. We see that the use of AE as prior knowledge not only reduces the loss function significantly but

[1] See interesting applications of machine learning to various investment strategies, see Nakano et.al. (2017) [33, 34, 35].

also this bias.

For further examples, we have presented an extension of the deep BSDE solver for reflected BSDEs. Using an American basket option as an example, we have shown the flexibility of the method [19] and the usefulness of the asymptotic expansion for variety of financial applications. Finally, we have tested the deep BSDE solver for a special class of quadratic growth BSDEs (qg-BSDEs) which allows a closed form solution by a Cole-Hopf exponential transformation. Despite the notorious difficulty to obtain stable numerical results for qg-BSDEs [9], the deep BSDE solver with asymptotic expansion can handle the problem efficiently.

2 An application to Bergman's model

2.1 Model

Let us consider the filtered probability space $(\Omega, \mathbb{F}=(\mathcal{F}_t)_{t\in[0,T]}, \mathcal{F}_T, \mathbb{P})$ generated by a d-dimensional Brownian motion $(W^\alpha)_{\alpha=1}^d$, which is assumed to satisfy the usual conditions. We suppose that the d risky assets follow the dynamics

$$X_t^i = x_0^i + \int_0^t \mu^i X_s^i ds + \int_0^t X_s^i \sigma^i \sum_{\alpha=1}^d \rho_{i,\alpha} dW_s^\alpha, \ t\in[0,T], i=1,\cdots,d \tag{2.1}$$

where $x_0^i > 0$ is the initial value, $\mu^i, \sigma^i > 0$ are constants and $(\rho_{i,\alpha})_{i,\alpha=1}^d$ is the square root of the (instantaneous) correlation matrix among X^i, normalized as $(\rho\rho^\top)_{i,i}=1,\ i=1,\cdots,d$. ρ is assumed to be invertible. There are two interest rates, one is for lending $r>0$ and the other $R>r$ for borrowing. The dynamics of portfolio value $(Y_t)_{t\in[0,T]}$ under the least-borrowing self-financing strategy for replicating the terminal payoff $\Phi(X_T)$, where $\Phi:\mathbb{R}^d\to\mathbb{R}$ is a Lipschitz continuous function, is given by the following BSDE:

$$\begin{aligned} Y_t &= \Phi(X_T) - \int_t^T \Big\{ rY_s + \sum_{i,\alpha=1}^d Z_s^\alpha (\rho^{-1})_{\alpha,i} \frac{\mu^i - r}{\sigma^i} - \Big(\sum_{i,\alpha=1}^d Z_s^\alpha (\rho^{-1})_{\alpha,i} \frac{1}{\sigma^i} - Y_s \Big)^+ (R-r) \Big\} ds \\ &\quad - \int_t^T \sum_{\alpha=1}^d Z_s^\alpha dW_s^\alpha, \quad t\in[0,T] \ . \end{aligned} \tag{2.2}$$

The existence of unique solution is guaranteed by the standard results for the Lipschitz BSDEs. Note that the cash amount invested to the ith risky asset at time t is given by $\pi_t^i = \sum_{\alpha=1}^d Z_t^\alpha (\rho^{-1})_{\alpha,i}/\sigma^i$.

2.2 Asymptotic expansion based on driver's linearization

We adopt an asymptotic expansion method proposed in Fujii & Takahashi [21] which is based on a perturbative expansion of the non-linear driver of the BSDE around a linear term. Mathematical justification of the expansion is available in Takahashi & Yamada (2015) [40]. Its numerical implementation using the particle methods proposed in Fujii & Takahashi (2015) [24] has been successfully applied to large scale simulations in many works such as [3, 12, 13, 14] using the second order approximations. See for example [22] as a simple analytic example.

In this work, we only use the leading term of the asymptotic expansion. For higher order corrections, see discussions and examples available in [21, 40, 22]. From Appendix A, one obtains the leading order approximation $(Y_t^{(0)})_{t\in[0,T]}$ of $(Y_t)_{t\in[0,T]}$ as

$$Y_t^{(0)} = e^{-r(T-t)}\mathbb{E}^{\mathbb{Q}}\Big[\Phi(X_T)|\mathcal{F}_t\Big], \tag{2.3}$$

with the probability measure $\mathbb{Q}$ defined by

$$\frac{d\mathbb{Q}}{d\mathbb{P}} = \mathcal{E}\Big(-\int_0^T \sum_{\alpha,j=1}^{d}(\rho^{-1})_{\alpha,j}\frac{\mu^j - r}{\sigma^j}dW_s^\alpha\Big)$$

where $\mathcal{E}$ is Doléans-Dade exponential. Since $Y^{(0)}$ is equal to the price process in Black-Scholes model with the risk-free rate r, $Z^{(0)}(=: Z^{\mathrm{AE}})$ is obtained as deltas multiplied by $\sigma^i X^i$. For example, if $d=1$ and $\Phi(X_T) = \max(X_T - K, 0)$, one has $Z_t^{(0)} = N(d_+)\sigma X_t$ where $N(\cdot)$ is the distribution function of the standard normal and $d_+ = \frac{1}{\sigma\sqrt{T-t}}\Big(\log\Big(\frac{F_t}{K}\Big) + \frac{1}{2}\sigma^2(T-t)\Big)$, $\quad F_t = e^{r(T-t)}X_t$.

We should emphasize that an analytical expression can be obtained even when Φ and the process X have more general forms. This is a well-known application of asymptotic expansion technique to European contingent claims. See Takahashi (1999, 2015) [38, 39] and Kunitomo & Takahashi (2003) [31] for details on this topic. In the following, in order to focus on the implications of AE as prior knowledge in the deep BSDE solver instead of deriving AE formulas for general setups, we only deal with the terminal conditions consisting of call/put options and the log-normal process for X.

2.3 Numerical examples

2.3.1 Purely call terminals

Suppose that the terminal condition is given by

$$\Phi(X_T) = \sum_{i=1}^{d} q^i \max(X_T^i - K^i, 0), \quad q^i > 0,\ i = 1,\cdots,d.$$

In this case, the one who tries to replicate the terminal payoff must always hold a long position for every risky asset. Since this implies she must always borrow cash to finance her hedging position, the BSDE (2.2) becomes

$$Y_t \;=\; \Phi(X_T) - \int_t^T \Big\{RY_s + \sum_{i,\alpha=1}^{d} Z_s^\alpha(\rho^{-1})_{\alpha,i}\frac{\mu^i - R}{\sigma^i}\Big\}ds - \int_t^T \sum_{\alpha=1}^{d} Z_s^\alpha dW_s^\alpha$$

Notice that this holds true irrespective of the correlation among X's. After a simple measure change, one sees that the exact solution of Y_0 is given by the corresponding Black-Scholes formula with r replaced by R.

Let us start from the simplest one-dimensional example with setA := $\{d=1, q=1, \mu=0.05, r=0.01, R=0.06, \sigma=0.3, T=0.5, K=103\}$. In this case, the above discussion gives $Y_0 = 8.4672$ as the exact solution. We have used n_time=50 (time discretization), batch_size=64, n_layer =4, learning_rate $= 10^{-3}$ in the deep BSDE solver [19]. The loss function is estimated with 1024 paths. As explained before, we have used only the leading order approximation as Z^{AE} and put $Z = Z^{\mathrm{AE}} + Z^{\mathrm{Res}}$ in the deep BSDE solver, where only the residual term Z^{Res} and Y_0 are used as the targets of the training process. In Figure 1, we have compared the performance of the deep BSDE solver with and without AE as prior

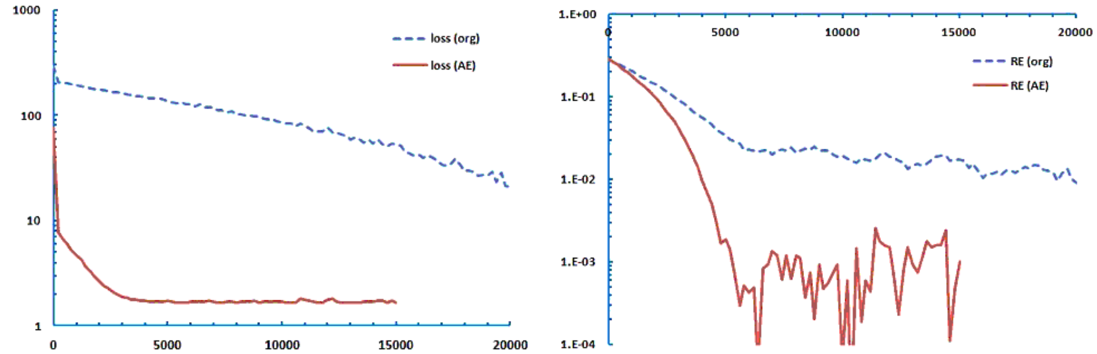

Figure 1: Comparison of the loss function and the relative error w.r.t. the exact solution between the direct use of the deep BSDE solver (indicated by a dashed line) and the one using AE as prior knowledge (indicated by a solid line) for the 1-dimensional case. The horizontal axis is the number of iteration steps of the learning process.

knowledge. It is observed that one achieves much quicker convergence and roughly by one order of magnitude smaller relative error when one uses AE as prior knowledge. After roughly 5,000 iterations, its loss function reaches 1.7. Since the option is around at-the-money, the gamma at the last stage is huge in many paths. If the delta-hedging at the last period $\Delta t = 1/100$ completely fails, its contribution to the loss function is estimated roughly by $(100 \times 0.3 \times \sqrt{\Delta t} \times 0.5)^2 = 2.25$. This estimate implies that the deep BSDE solver with AE reaches its limit performance already at 5,000 iterations. When AE is not used, one sees that the loss function (and hence the replication error) remains rather big and slow to converge.

Notice that the deep BSDE solver uses **tf.train.AdamOptimizer** available in the TensorFlow package for optimizing the coefficient matrices usually denoted by $\mathbf{w}$. This is the algorithm proposed in [1], in which the learning_rate 10^{-3} is recommended as a default value. Although one can speed up the learning process by increasing the learning rate, this is not always recommendable. Let us study the effects of the learning rate using the next 30-dimensional example.

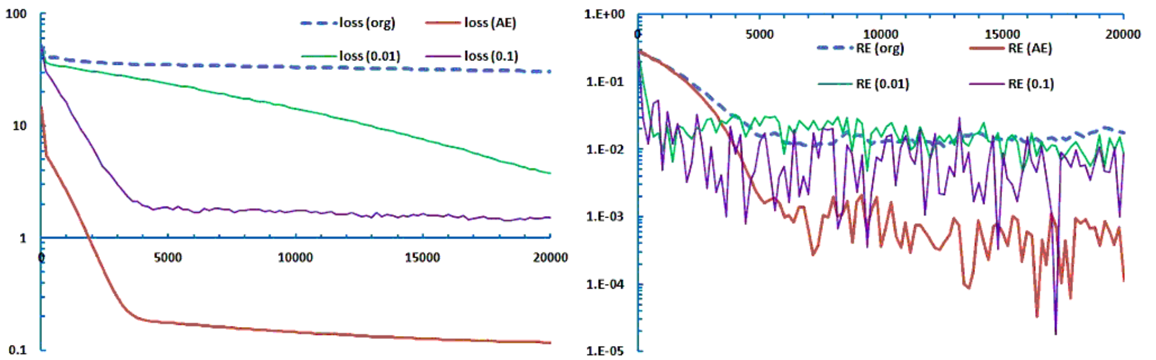

Figure 2: Comparison of the loss function and the relative error w.r.t. the exact solution with and without AE as prior knowledge for the 30-dimensional case with correlation implied by $\gamma = 0.06$ in (2.4). The two *thin* solid lines (green and purple) denote the cases without using AE but the learning_rate replaced by 0.01 and 0.1, respectively. The horizontal axis is the number of iteration steps of the learning process.

We study the case with $d = 30$ where the setA is replaced by $q^i = 1/d$ and $\mu^i, \sigma^i, K^i,\ i = 1, \cdots d$ are common and the same with those in setA i.e., Φ is given by the average of the call options. The matrix ρ is assumed to have the form

$$\rho = \frac{1}{\sqrt{1+(d-1)\gamma^2}} \begin{pmatrix} 1 & \cdots & \gamma \\ \vdots & \ddots & \vdots \\ \gamma & \cdots & 1 \end{pmatrix} \tag{2.4}$$

with $\gamma = 0.06$. This implies that the correlation for every pair (X^i, X^j) is about 20%. The exact value of Y_0 must be the same $Y_0 = 8.4672$. In Figure 2, we have provided the numerical results for this case. In addition to those with the default learning_rate $= 10^{-3}$, we have added two cases with learning_rate $= 10^{-2}$ and 10^{-1} without AE as prior knowledge. One sees, for example, learning_rate $= 10^{-1}$ yields a fast decline of the loss function in the first 5,000 steps comparable to the case with AE, but it stops at the level 10 times higher than the case with AE. Moreover, the estimated Y_0 (and hence the relative error) exhibits strong instability. The use of asymptotic expansion with the default learning rate yields a more stable and accurate estimate. Notice that the instability associated with a higher learning rate is more prominent for lower dimensional problems. The 30-dimensional example we have just considered, the instability is somewhat mitigated by the diversification effects from the imperfectly correlated 30 assets.

Dynamically choosing the optimal learning rate is an important issue and, in fact, is a popular topic for researchers on computation algorithms. At the moment, however, there exists no established rule and it looks to depend on a specific problem under consideration. As we have seen above, the optimal choice depends on the dimension of the forward process X as well as their correlation even if the form of the BSDE is the same. One may possibly use the results obtained with AE as convenient benchmarks to optimize the dynamical choice of the learning rate. In the reminder of the paper, we shall fix the learning rate to the default value 10^{-3} for the **tf.train.AdamOptimizer** unless explicitly stated otherwise.

2.3.2 Call spread

We next study the terminal function $\Phi(X_T) = (X_T - K_1)^+ - 2(X_T - K_2)^+$ with $\{d = 1, \mu = 0.05, r = 0.01, R = 0.06, \sigma = 0.2, T = 0.25, K_1 = 95, K_2 = 105\}$. For this 1-dimensional example of call spread, there is no closed-form solution anymore. However, it is estimated as $Y_0 = 2.96 \pm 0.01$ in Bender & Steiner (2012) [4] using the regression based Monte Carlo scheme [28] improved by the martingale basis functions. The numerical results are given in Figure 3. A much quicker convergence and smaller loss function are observed as before.

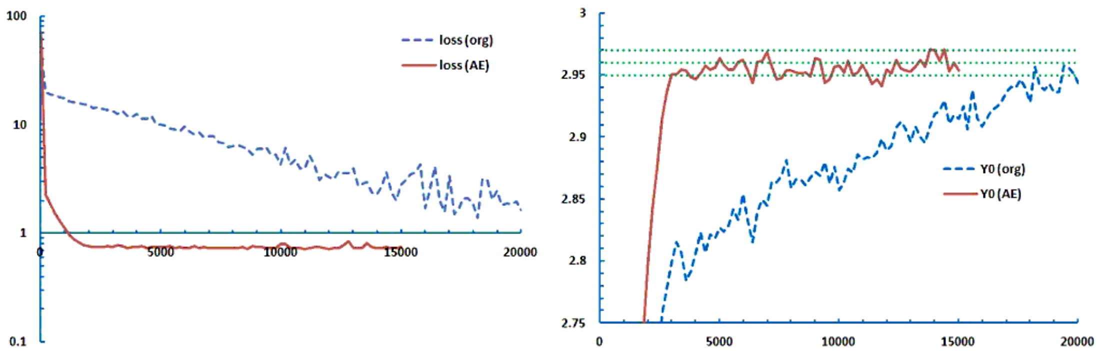

Figure 3: Comparison of the loss function and the estimated Y_0 between the direct use of the deep BSDE solver (indicated by a dashed line) and the one using AE as prior knowledge (indicated by a solid line) for the 1-dimensional call spread. The three dotted lines denote 2.96 ± 0.01 the value obtained in [4] using the regression-based Monte Carlo simulation.

Finally, we provide the numerical results for a high dimensional setup with

$$\Phi(X_T) = \frac{1}{d}\sum_{i=1}^{d}\Big((X_T^i - K_1)^+ - 2(X_T^i - K_2)^+\Big),$$

$\{d = 30, (\mu^i)_{i=1}^d = 0.05, r = 0.01, R = 0.06, (\sigma^i)_{i=1}^d = 0.2, T = 0.25, K_1 = 95, K_2 = 105\}$ and the same matrix ρ with $\gamma = 0.06$ given in the last subsection. The numerical comparison is given in Figure 4. Probably due to the diversification effects, one observes the quicker convergence of the estimated Y_0 for both cases. However, the loss function is more than one magnitude smaller when AE is used as prior knowledge. Moreover, there remains a gap in estimated Y_0 between the two cases.

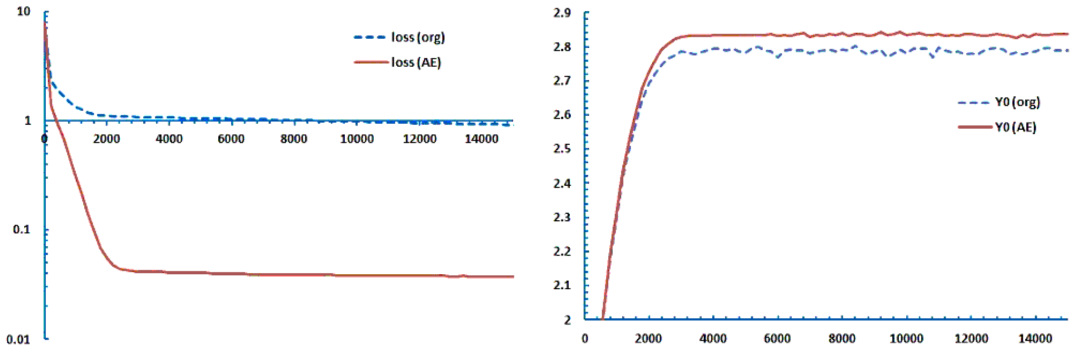

Figure 4: Comparison of the loss function and the estimated Y_0 between the direct use of the deep BSDE solver (indicated by a dashed line) and the one using AE as prior knowledge (indicated by a solid line) for the 30-dimensional call spreads with correlation implied by $\gamma = 0.06$ in (2.4).

Remark 2.1. *In the deep BSDE solver, one needs to set* Yini *which is the interval (say,* Yini $= [0.3, 0.6]$ *in the sample code given in [19]) from which a set of Y_0's are randomly sampled under the uniform distribution to start the very first stage of the learning process. When AE is not used as prior knowledge, we have observed that the choice of* Yini *does not affect the convergence speed meaningfully. However, when AE is used, the interval* Yini *set close to the true value produces significantly quicker convergence.*

3 An example of a reflected BSDE

3.1 Model

We now study a reflected BSDE corresponding to the replication problem of an American option:

$$Y_t = \Phi(X_T) - \int_t^T \Big\{ rY_s + \sum_{i,\alpha=1}^{d} Z_s^\alpha (\rho^{-1})_{\alpha,i} \frac{\mu^i + y^i - r}{\sigma^i} \Big\} ds - \int_t^T \sum_{\alpha=1}^{d} Z_s^\alpha dW_s^\alpha + L_T - L_t,$$

$$Y_t \geq \Phi(X_t),\ t \geq 0, \qquad \int_0^T [Y_t - \Phi(X_t)] dL_t = 0, \tag{3.1}$$

where $y^i > 0$ is a dividend yield of the ith security $\{i = 1, \cdots, d\}$, $(L_t, t \in [0, T])$ is the reflecting process that keeps the solution Y_t from going below the barrier $\Phi(X_t)$ for every $t \in [0, T]$. The other assumptions made in Section 2.1 are still in force.

Instead of using the penalization method [17], we extend the deep BSDE solver so that it learns the process $(L_t, t \in [0,T])$ directly in addition to Y_0 and $(Z_t, t \in [0,T])$. We adopt the loss function

$$|\Phi(X_T) - \widehat{Y}_T|^2 + w \int_0^T \max(\Phi(X_t) - \widehat{Y}_t, 0)^2 dt$$

where the weight $w := 2.0/T$ is used to take a balance between the terminal and the lateral conditions. Remember that $\widehat{Y}$ is the forwardly simulated process based on the estimated Y_0 and $(Z_t, L_t)_{t\in[0,T]}$. We apply dL_t to update the process $\widehat{Y}$ only when $\widehat{Y}_t \le \Phi(X_t)$ so that we can avoid the explicit inclusion of the second condition of (3.1) into the loss function. Since it is impossible to make the loss function exactly zero, the weight w slightly affects the estimated Y_0 (as well as the size of the loss function).

Remark 3.1. *Changing the code for the penalization method [17] is very simple. The solution of the penalized BSDE, which is obtained by replacing $L_T - L_t$ with*

$$\frac{1}{\epsilon}\int_t^T \max(\Phi(X_s) - Y_s, 0) ds \ , \tag{3.2}$$

is known to converge to that of (3.1) in the limit of $\epsilon \downarrow 0$. Although there is no need to estimate the process L, we have found that the numerical results depend quite sensitively on the size of ϵ, and hence not recommendable in general. It is useful, however, for double checking the correct implementation by comparing the numerical results.

3.2 Numerical examples

In the following, we adopt $\Phi(X_T) = \max\Bigl(\frac{1}{d}\sum_{i=1}^d X_T^i - K, 0\Bigr)$ corresponding to an American basket call option. The leading order asymptotic expansion is still given by $Y_t^{(0)} = e^{-r(T-t)}\mathbb{E}^{\mathbb{Q}}\Bigl[\Phi(X_T)|\mathcal{F}_t\Bigr]$ and $Z^{(0)}(=: Z^{\rm AE})$ as its deltas multiplied by $\sigma^i X^i$. Although one cannot obtain the exact solution, it is not difficult to expand the solution in terms of σ to obtain

$$Z_t^{\mathrm{AE},\alpha} = N(d_c)\frac{1}{d}\sum_{i=1}^d e^{-y^i(T-t)}\sigma^i X_t^i \rho_{i,\alpha} + \mathcal{O}(\sigma^2) \ , \alpha = 1, \cdots, d$$

where

$$\begin{aligned} d_c &= \frac{1}{\widetilde{\sigma}(t)\sqrt{T-t}}\Bigl(\frac{1}{d}\sum_{i=1}^d X_t^i e^{(r-y^i)(T-t)} - K\Bigr) \\ \widetilde{\sigma}(t) &= \frac{1}{d}\Bigl(\sum_{i,j=1} \sigma^i X_t^i e^{(r-y^i)(T-t)}(\rho\rho^\top)_{i,j}\sigma^j X_t^j e^{(r-y^j)(T-t)}\Bigr)^{1/2} . \end{aligned}$$

See Appendix B for some details.

3.2.1 American call option

Let us first check the general performance by studying one-dimensional example. We use $\{\mu = 0.02, y = 0.07, r = 0.03, T = 0.5, \sigma = 0.2, K = 100, x_0 = 110\}$. Note that the choice of μ should not affect Y_0. The benchmark price taken from [30] is 11.098, while the corresponding European option price is 10.421. The comparison of the loss function and the relative error is

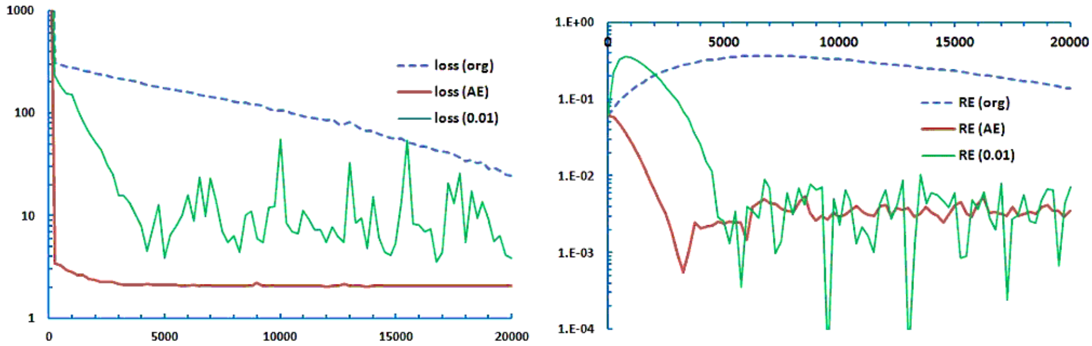

Figure 5: Comparison of the loss function and the relative error between the direct use of the deep BSDE solver and the one using AE as prior knowledge for the 1-dimensional American call option. For comparison, the case with learning rate 10^{-2} (without using AE) is given by a thin solid line.

given in Figure 5. In order to achieve an accurate estimate, a fine discretization (n_time= 100) is used. Since the direct use of the deep BSDE solver with the default learning rate yields very slow convergence, we have also provided the case with the learning rate 10^{-2}. The associated instability in the loss function as well as Y_0 suggests that one needs to tune the learning rate dynamically in the deep BSDE solver for achieving the comparable performance to the case with AE. [2]

3.2.2 American 50-dimensional basket call option

We now study a 50-dimensional American basket call option. Let us use $\{\mu^i = 0.02, y^i = 0.07, x_0^i = 110, \sigma^i = 0.2\}_{i=1}^{50}$, $K = 100, T = 0.5, r = 0.03$ and $\gamma = 0.07$ in (2.4), which implies around 30% correlation for every pair of X's. We have used (n_time= 100) time partition as before. The price of the corresponding European option is estimated as 8.46 by a simulation with 500,000 paths.

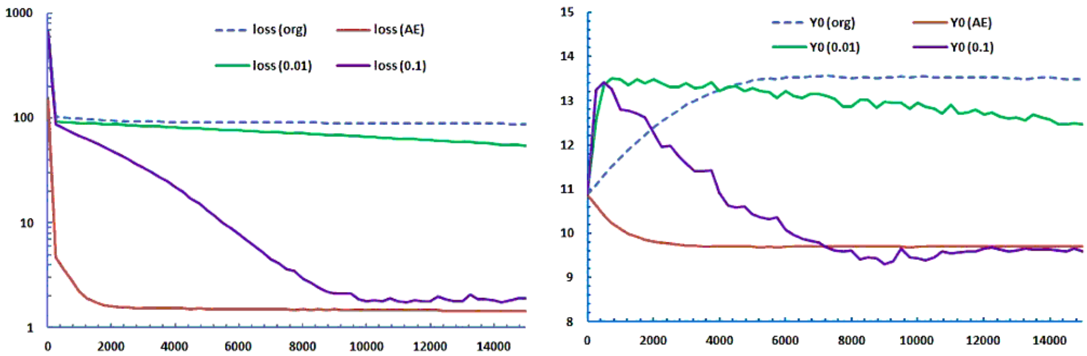

Figure 6: Comparison of the loss function and the estimated option price Y_0 between the direct use of the deep BSDE solver and the one using AE as prior knowledge for the 50-dimensional American basket call option. For comparison, the cases with learning rate 10^{-2} and 10^{-1} (without using AE) are also given.

As one can see from Figure 6, the convergence is very quick when AE is used as prior knowledge. After 2,000 iterations, Y_0 is settled around 9.7. When AE is not used, the convergence is very slow. Even if we use an extremely large learning_rate= 0.1, it takes

[2] In this example, a different choice of weight $w = 1.0/T$ gives a slightly smaller (by roughly 1% relative difference) value of Y_0.

more than 10,000 iterations to give a comparable size of loss function. The stability of estimated Y_0 with AE clearly stands out from the others. In the deep BSDE solver, we estimate (Z_t, L_t) *at each time step* using neural networks with multiple layers. Therefore quick convergence brought by a simple AE approximation is a great advantage, in particular, for the problems that require many time steps for accurate estimates. Comparing to the previous one-dimensional example, one clearly sees that the optimal choice of the learning rate depends on the details of the settings (such as, the number of assets and the correlation among them) even for the same BSDE.

4 A solvable example of Quadratic BSDE

4.1 Model

We now consider the following qg-BSDE

$$Y_t = \Phi(X_T) + \int_t^T \frac{a}{2}|Z_s|^2 - \int_t^T \sum_{\alpha=1}^{d} Z_s^\alpha dW_s^\alpha,\ t \in [0,T] \tag{4.1}$$

where $a \in \mathbb{R}$ is a constant, W a d-dimensional Brownian motion, $\Phi : \mathbb{R}^d \to \mathbb{R}$ is a bounded Lipschitz continuous function. For simplicity, we assume that the associated forward process is given by

$$X_t^i = x_0 + \int_0^t \sigma X_s^i \sum_{\alpha=1}^{d} \rho_{i,\alpha} dW_s^\alpha,\ t \in [0,T],\ i = 1, \cdots, d$$

with a common initial value $x_0 > 0$, and a volatility $\sigma > 0$. $\rho = (\rho_{i,j})_{i,j=1}^d$ is a square root of correlation matrix among X^i and assumed to be invertible.

Thanks to this special form, it is easy to derive a closed form solution

$$Y_t = \frac{1}{a}\log\Big(\mathbb{E}\Big[\exp\big(a\Phi(X_T)\big)\big|\mathcal{F}_t\Big]\Big),\ t \in [0,T] \tag{4.2}$$

by applying Itô formula to e^{aY_t}. Note however that the numerical evaluation of (4.1) is known to be very hard despite its simple appearance. See discussions in Imkeller & Reis (2010) [29], Chassagneux & Richou (2015) [9] and Fujii & Takahashi (2016) [26].

A formal application of the method [21] to the current case gives $Y_t^{(0)} = \mathbb{E}\Big[\Phi(X_T)|\mathcal{F}_t\Big]$ as the leading order asymptotic expansion, and hence $Z_t^{(0)} (=: Z_t^{\mathrm{AE}})$ can be derived as deltas in exactly the same manner as in the last section. Although the asymptotic expansion methods in [21, 40] are only proved for the Lipschitz BSDEs, using Malliavin's differentiability and the associated representation theorem for Z given in Ankirchner, Imkeller & Dos Reis (2007) [2], one can justify the method also for the quadratic case in a similar way. The asymptotic expansion for the Lipschitz BSDEs with jumps in [25] can also be extendable to a quadratic-exponential growth BSDEs by using the results of Fujii & Takahashi (2017) [27]. The details may be given in different opportunities.

4.2 Numerical examples

We suppose a bounded terminal condition defined by

$$\Phi(X_T) = \frac{1}{d}\sum_{i=1}^{d}\Big(\max(X_T^i - K_1, 0) - \max(X_T^i - K_2, 0)\Big) \tag{4.3}$$

with two constants $0 < K_1 < K_2$. As a first example, we have tested a 50-dimensional model with zero correlation: $\mathrm{set}_0 := \{d = 50, a = 1.0, T = 0.25, K_1 = 95, K_2 = 105, \sigma = 0.2, x_0 = 100, \rho = \mathbf{I}_{d\times d}\}$. The solution (4.2) is estimated as $Y_0 = 5.01$ by a simulation with one million paths. We use n_time = 25 as discretization. In Figure 7, we have compared the performance of the deep BSDE solver with and without AE as prior knowledge. When the asymptotic expansion is used, the convergence is achieved just after a few thousands iterations and the relative error becomes ≤0.1%. On the other hand, the learning process proceeds very slowly when the deep BSDE solver is directly applied without using AE. Even after 30,000 iterations, both of the loss function and the relative error are still larger than the former by more than an order of magnitude. It seems that clever dynamic tuning of the learning rate is necessary for achieving comparable speed of convergence and stability to those for the case with asymptotic expansion.

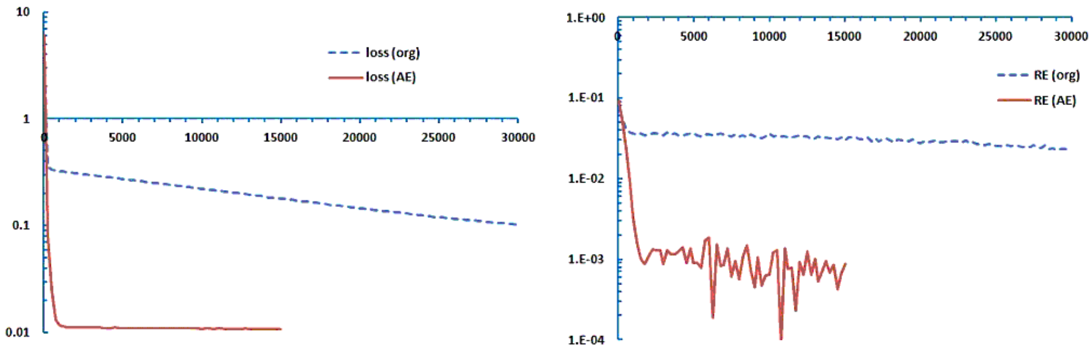

Figure 7: Comparison of the loss function and the relative error between the direct use of the deep BSDE solver (indicated by a dashed line) and the one using AE as prior knowledge (indicated by a solid line) for the 50-dimensional model with zero correlation.

Next, we have studied the impact of correlation among X's. Since the regressors have non-zero correlation, one can expect that the learning process becomes harder to proceed. We have used the same parameters in set_0 except that the matrix ρ is now defined by (2.4) with $\gamma = 0.07$, which implies about 30% correlation among X's. In this case, the solution (4.2) estimated by one million paths is $Y_0 = 6.78$. The comparison of the performance is given in Figure 8. Although the accuracy is deteriorated in the both cases, the deep BSDE solver with the asymptotic expansion still achieves the relative error $3 \sim 4\%$ after 5,000 iterations. When AE is not used, the relative error remains more than 25% even after 30,000 iteration steps. As it is the case for the Bergman's model, the use of AE as prior knowledge effectively ameliorates the problem of correlated inputs without doing any other adjustments in the algorithm.

Finally, let us study a bit extreme situation with a large quadratic coefficient as well as volatility. We set $\{d = 50, a = 5.0, T = 0.25, K_1 = 95, K_2 = 105, \sigma = 1.0, x_0 = 100, \rho = \mathbf{I}_{d\times d}\}$ and increase the number of time partition to n_time= 50. The solution (4.2) estimated by a million paths of Monte Carlo simulation with the same step size is given by $Y_0 = 5.17 \pm 0.01$. We have provided the numerical results in Figure 9. Although the loss function becomes

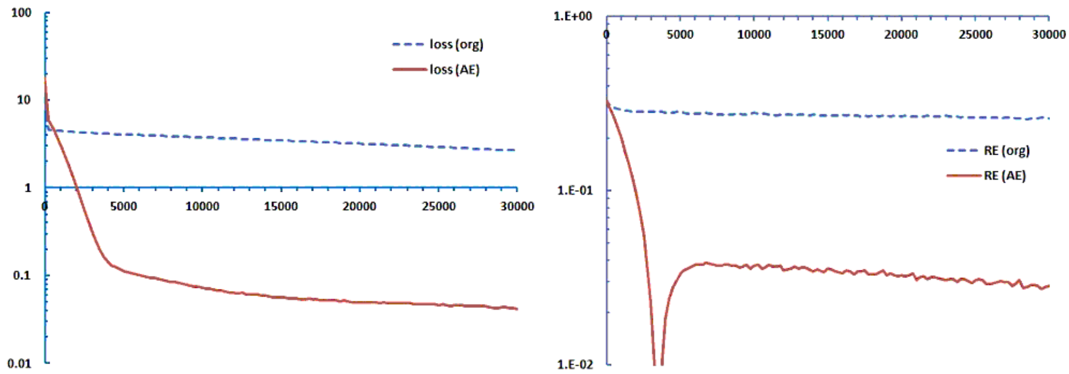

Figure 8: Comparison of the loss function and the relative error between the direct use of the deep BSDE solver (indicated by a dashed line) and the one using AE as prior knowledge (indicated by a solid line) for the 50-dimensional model with correlation implied by $\gamma = 0.07$ in (2.4).

larger by a factor of few, the deep BSDE solver with AE quickly reaches its equilibrium after a few thousands steps and the relative error is around 2%. As is clearly seen from the graph, the estimated Y_0 obtained without using AE is quite far from the target even after 30,000 iterations steps.

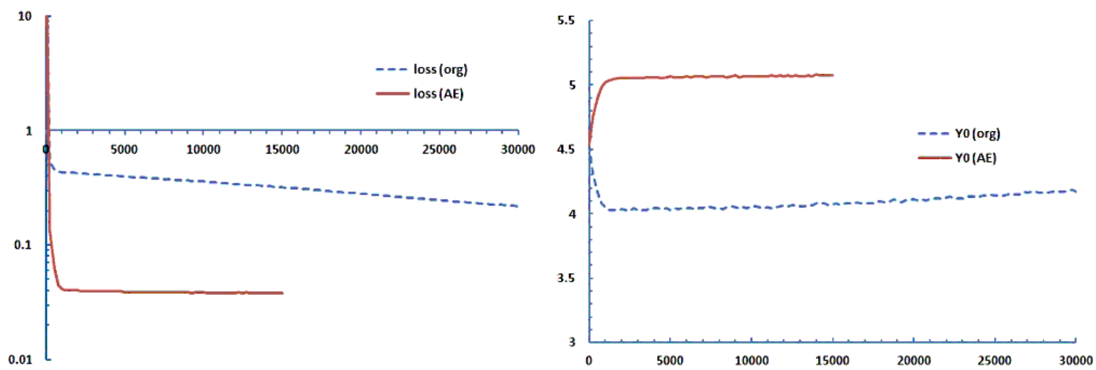

Figure 9: Comparison of the loss function and the estimated Y_0 between the direct use of the deep BSDE solver (indicated by a dashed line) and the one using AE as prior knowledge (indicated by a solid line) for the 50-dimensional model with $a = 5.0, \sigma = 1.0$.

5 Conclusions and further research directions

In this work, we have demonstrated that the use of a simple formula based on an asymptotic expansion (AE) as prior knowledge significantly improve the performance of the deep BSDE solver proposed in [19]. We have illustrated the technique and its performance by studying Bergman's model that has a typical form of BSDEs often encountered in financial applications. Furthermore, we provided an extension of the deep BSDE solver for reflected BSDEs by directly learning the reflecting process L_t in addition to Z_t. We have also observed a similar advantage of the use of AE for a class of quadratic growth BSDEs which are known to be very hard to obtain stable numerical estimates.

Applying asymptotic expansion techniques for other type of BSDEs is a topic for the future research. Projecting the terminals and drivers of BSDEs to smooth functions (such as Chebyshev polynomials) may allow a systematic application of asymptotic expansions, whose result can then be applied to the *original* BSDEs to accelerate the speed of convergence. Applying the proposed technique to the so-called *large-investor* problems is quite promising. Although the relevant BSDEs have non-linear terms arising from the price impacts of the investor's actions, their leading order asymptotic expansions corresponding to the market without feed-back effects should give reasonable approximations. More fundamentally, application of the deep learning methods to the BSDEs with jumps remains as an important challenge. Even if we restrict the jump space to a finite set, small to medium size jump intensities (such as in credit models) is expected to make the learning process very hard to proceed. An analytic approximation of the jump coefficients available by the asymptotic expansion in [25] may mitigate the difficulty.

A Leading order asymptotic expansion for the Bergman's model

According to [21], we consider (2.2) as the perturbed model around the linear driver:

$$\begin{aligned} Y_t^\epsilon &= \Phi(X_T) - \int_t^T \Big\{ rY_s^\epsilon + \sum_{i,\alpha=1}^d Z_s^{\alpha,\epsilon}(\rho^{-1})_{\alpha,i}\frac{\mu^i - r}{\sigma^i} - \epsilon\Big(\sum_{i,\alpha=1}^d Z_s^{\alpha,\epsilon}(\rho^{-1})_{\alpha,i}\frac{1}{\sigma^i} - Y_s^\epsilon\Big)^+ (R-r)\Big\} ds \\ &\quad - \int_t^T \sum_{\alpha=1}^d Z_s^{\alpha,\epsilon} dW_s^\alpha, \quad t \in [0,T] \ . \end{aligned}$$

The idea of the approximation [21] is to expand $(Y^\epsilon, Z^{\alpha,\epsilon})$ around $\epsilon = 0$. The leading order $(Y^{(0)}, Z^{\alpha,(0)}) := (Y^\epsilon, Z^{\alpha,\epsilon})|_{\epsilon=0}$ follows

$$\begin{aligned} Y_t^{(0)} &= \Phi(X_T) - \int_t^T \Big\{ rY_s^{(0)} + \sum_{i,\alpha=1}^d Z_s^{\alpha,(0)}(\rho^{-1})_{\alpha,i}\frac{\mu^i - r}{\sigma^i}\Big\} ds \\ &\quad - \int_t^T \sum_{\alpha=1}^d Z_s^{\alpha,(0)} dW_s^\alpha, \quad t \in [0,T] \ , \end{aligned}$$

which immediately gives (2.3).

B Small diffusion expansion

The approximation used in Section 3.2 is based on the well-known *small-diffusion* expansion technique [38, 31]. We perturb the forward process X as

$$\begin{aligned} dX_s^{i,\epsilon} &= (r - y^i)X_s^{i,\epsilon} ds + \epsilon X_s^{i,\epsilon}\sigma^i \sum_\alpha \rho_{i,\alpha} dW_s^{\mathbb{Q},\alpha}, s \geq t \\ X_t^{i,\epsilon} &= X_t^i \ . \end{aligned}$$

Notice the fact that the expansion of X^ϵ as a power series of ϵ is equivalent to that of σ after setting $\epsilon = 1$. One sees that the 0th order expansion corresponds to the deterministic forward process $X_s^{i,(0)} = X_t^i e^{(r-y)(s-t)}, s \geq t$ and that the next order expansion is a Gaussian process defined by $dX_s^{i,1} = (r - y^i)X_s^{i,1}ds + X_s^{i,0}\sigma^i \sum_\alpha \rho_{i,\alpha} dW_s^{\mathbb{Q},\alpha}, s \geq t$ and $X_t^{i,1} = 0$. Using the expansion up to the first oder, it is easy to derive the results in Section 3.2 since the process is now Gaussian. Although one can continue the expansion to an arbitrary higher order, it is expected to give only marginal effects when used in the deep BSDE solver.

Acknowledgement

The research is partially supported by Center for Advanced Research in Finance (CARF).

References

[1] Kingma, D. P. and Ba, J.L., 2015, *ADAM: A method for stochastic optimization, arXiv:1412.6980.*

[2] Ankirchner, S., Imkeller, P. and Dos Reis, G., Classical and Variational Differentiability of BSDEs with Quadratic Growth, Electronic Journal of Probability, Vol. 12, 1418-1453.

[3] Armenti, Y. and Crepey, S., 2017, Central Clearing Valuation Adjustment, SIAM J. Financial Math., Vol. 8, 274-313.

[4] Bender, C. and Steiner, J., 2012, Least-squares Monte Carlo for Backward SDEs, Numerical Methods in Finance (edited by Carmona et al.), 257-289, Springer, Berlin.

[5] Bergman, Y. Z., 1995, Option pricing with different interest rates, Review of Financial Studies, Vol. 8, 2, 475-500.

[6] Bismut, J.M., 1973, Conjugate convex functions in optimal stochastic control, J. Math. Anal. Apl. 44, 384-404.

[7] 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.

[8] Brigo, D, M Morini and A Pallavicini (2013), Counterparty Credit Risk, Collateral and Funding, Wiley, West Sussex.

[9] Chassagneux, J.F. and Richou, A., 2016, Numerical Simulation of Quadratic BSDEs, Annals of Applied Probabilities, Vol. 26, No. 1, 262-304.

[10] Crépey, S, T Bielecki with an introductory dialogue by D Brigo (2014), Counterparty Risk and Funding, CRC press, NY.

[11] Crepey, S., 2015, Bilateral Counterparty Risk under Funding Constraints Part I: Pricing, Part II: CVA Mathematical Finance, Vol. 25, No. 1, 1-22, 23-50.

[12] Crepey, S. and Song, S., 2016, Counterparty risk and funding: immersion and beyond, Finance and Stochastics, Vol. 20, Issue 4, 901-930.

[13] Crepey S., Nguyen T.M., 2016, Nonlinear Monte Carlo Schemes for Counterparty Risk on Credit Derivatives. In: Glau K., Grbac Z., Scherer M., Zagst R. (eds) Innovations in Derivatives Markets. Springer Proceedings in Mathematics & Statistics, vol 165. Springer, Cham.

[14] Crepey, S. and Song, S., 2017, Invariance properties in the dynamic Gaussian copula model, ESAIM: Proceedings and surveys, Vol. 60, 21-37.

[15] Delong, L., 2013, Backward Stochastic Differential Equations with Jumps and Their Actuarial and Financial Applications, Springer-Verlag, LN.

[16] *El Karoui, N. and Mazliak, L. (eds.), 1997, Backward stochastic differential equations, Addison Wesley Longman Limited, U.S..*

[17] *El Karoui, N., Kapoudjian, C., Pardoux, E., Peng, S. and Quenez, M.C., 1997, Reflected solutions of backward SDE's and related obstacle problems for PDE's, Annals of Probability, Vol. 25, 2, 702-737.*

[18] *E, W. and Han, J., 2016, Deep learning approximation for stochastic control problems, arXiv:1611.07422.*

[19] *E, W., Han, J. and Jentzen, A., 2017, Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations, arXiv:1706.04702.*

[20] *Fujii, M. and Takahashi, A., 2012, Collateralized credit default swaps and default dependence, Journal of Credit Risk, Vol. 8, No. 3, 97-113.*

[21] *Fujii, M. and Takahashi, A., 2012, Analytical approximation for non-linear FBSDEs with perturbation scheme, International Journal of Theoretical and Applied Finance, 15, 5, 1250034 (24).*

[22] *Fujii, M. and Takahashi, A., 2012, Perturbative expansion of FBSDE in an incomplete market with stochastic volatility, Quarterly Journal of Finance, Vol. 2, No. 3, 1250015 (24).*

[23] *Fujii, M. and Takahashi, A., 2013, Derivative pricing under asymmetric and imperfect collateralization and CVA Quantitative Finance, Vol. 13, Issue 5, 749-768.*

[24] *Fujii, M. and Takahashi, A., 2015, Perturbative expansion technique for non-linear FBSDEs with interacting particle method, Asia-Pacific Financial Markets, Vol. 22, 3, 283-304.*

[25] *Fujii, M. and Takahashi, A., 2015, Asymptotic Expansion for Forward-Backward SDEs with Jumps, Working paper, CARF-F-372. Available in arXiv.*

[26] *Fujii, M. and Takahashi, A., 2016, Solving backward stochastic differential equations by connecting the short-term expansions, Working paper, arXiv:1606.04285.*

[27] *Fujii, M. and Takahashi, A., 2017, Quadratic-exponential growth BSDEs with jumps and their Malliavin's differentiability, Stochastic Processes and their Applications, in press. DOI: 10.1016/j.spa.2017.09.002.*

[28] *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, Vol. 15, No. 3, 2172-2202.*

[29] *Imkeller, P. and Dos Reis, G., 2010, Path regularity and explicit convergence rate for BSDEs with truncated quadratic growth, Stochastic Processes and their Applications, 120, 348-379. Corrigendum for Theorem 5.5, 2010, 120, 2286-2288.*

[30] *Ju, N. and Zhong, R., 1999, An approximate formula for pricing American options, Journal of Derivatives, Vol. 7, No. 2, 31-40.*

[31] *Kunitomo, N. and Takaashi, A., 2003, On validity of the asymptotic expansion approach in contingent claim analysis, The Annals of Applied Probability, Vol. 13, No. 3, 914-952.*

[32] *Ma, J. and Yong, J., 2000, Forward-backward stochastic differential equations and their applications,Springer, Berlin.*

[33] *Nakano, M., Takahashi, A. and Takahashi, S., 2017, Fuzzy logic-based portfolio selection with particle filtering and anomaly detection, Knowledge-Based Systems, Vol 131, 113-124.*

[34] *Nakano, M., Takahashi, A. and Takahashi, S., 2017, Robust technical trading with fuzzy knowledge-based systems, forthcoming in Frontiers in Artificial Intelligence and Applications.*

[35] *Nakano, M., Takahashi, A. and Takahashi, S., 2017, Creating investment scheme with state space modeling, Expert Systems with Applications, Vol. 81, 53-66.*

[36] *Pardoux, E. and Peng, S., 1990, Adapted solution of a backward stochastic differential equations, Systems Control Lett., 14, 55-61.*

[37] *Pardoux, E. and Rascanu, A., 2014, Stochastic Differential Equations, Backward SDEs, Partial Differential Equations, Springer International Publishing, Switzerland.*

[38] *Takahashi, A., 1999, An asymptotic expansion approach to pricing financial contingent claims, Asia-Pacific Financial Markets, Vol. 6, No. 2, 115-151.*

[39] *Takahashi, A., 2015, Asymptotic Expansion Approach in Finance in Large Deviations and Asymptotic Methods in Finance, edited by Friz, P., Gatheral, J., Gulisashvili, A., Jacquier, A. and Teichman, J. Springer Proceedings in Mathematics and Statistics, Springer, N.Y.*

[40] *Takahashi, A. and Yamada, T., 2015, An asymptotic expansion of forward-backward SDEs with a perturbed driver, International Journal of Financial Engineering, Vol. 02, Issue 02, 1550020 (29).*

[41] *Zhang, J., 2004, A numerical scheme for BSDEs, The Annals of Applied Probability, Vol 14, No. 1, 459-488.*

[42] *Zhang, J., 2001, Some fine properties of backward stochastic differential equations, Ph.D Thesis, Purdue University.*

[43] *Zhang, J., 2017, Backward Stochastic Differential Equations, Probability Theory and Stochastic Modelling 86, Springer, N.Y.*