

# Solving Elliptic HJB Equations in A Value Space

Jiamin Jian

Jungang Bu

Yiyang Che

## Abstract

In this short report, we introduce the HJB equation and give a benchmark problem. To solve this problem, we will use the reinforcement approach such as value iteration. And we also want to apply the deep neural network to get the numerical solution of the HJB equation. What's more, we will consider what is the highest dimension we can solve.

## 1 Introduction

Stochastic control theory has a long history and it has been widely used in various fields. Generally we introduce the corresponding Hamilton-Jacobi-Bellman equations when we want to solve the stochastic control problem. As the regularity of the system is often not known a prior due to the control process, and since the solutions of the HJB equations are usually highly nonlinear, closed-form solution are rarely obtainable. Thus sometimes it is more convenient and efficient to consider the numerical solution of HJB equation. Numerical methods such as finite element method, finite difference method and spectral method are often used to get the numerical solution of partial differential equations.

Nowadays, with the strengthening of computer computing ability and the continuous development of machine learning and especially deep learning, we have an effective tool to solve some problems. Artificial neural networks can sometimes get unexpected great results in solving high dimensional problems. And there are increasing interests of using the reinforcement learning and deep learning technologies to solve the PDE problems. In this project we want to apply the reinforcement learning and deep learning approach based on the Markov chain approximation methods to solve a type a HJB equations. It involves two steps. Firstly, we approximate the solutions of the underlying stochastic control problems and recast them into discrete time Markov decision processes. Then, we solve the MDPs by using reinforcement learning and deep learning approaches.

## 1.1 Hamilton-Jacobi-Bellman Equation

In optimal control theory, the Hamilton–Jacobi–Bellman (HJB) equation gives a necessary and sufficient condition for optimality of a control with respect to a loss function. It is, in general, a nonlinear partial differential equation in the value function, which means its solution is the value function itself. Once the solution is known, it can be used to obtain the optimal control by taking the maximizer/minimizer of the Hamiltonian involved in the HJB equation.

## 1.2 Markov decision process

A Markov decision process (MDP) is a discrete time stochastic control process. It provides a mathematical framework for modeling decision making in situations where outcomes are partly random and partly under the control of a decision maker. MDPs are useful for studying optimization problems solved via dynamic programming and reinforcement learning.

## 1.3 Reinforcement learning

Reinforcement learning (RL) is an area of machine learning concerned with how software agents ought to take actions in an environment in order to maximize the notion of cumulative reward. Reinforcement learning is one of three basic machine learning paradigms, alongside supervised learning and unsupervised learning. The environment is typically stated in the form of a Markov decision process (MDP), because many reinforcement learning algorithms for this context utilize dynamic programming techniques. The main difference between the classical dynamic programming methods and reinforcement learning algorithms is that the latter do not assume knowledge of an exact mathematical model of the MDP and they target large MDPs where exact methods become infeasible.

# 2 Problem Setup

In this section we discuss the HJB equation associated to the control problem and give a benchmark problem for the demonstration of the computational method.

## 2.1 HJB equation

We consider an elliptic nonlinear HJB equation in the  $d$ -dimensional Euclidean space  $\mathbb{R}^d$  and  $\mathcal{O}$  is the domain,

$$F(x, Du(x), D^2u(x)) = 0, \forall x \in \mathcal{O} \quad (1)$$

with its boundary condition

$$u(x) = g(x), \forall x \in \mathcal{O}^c, \quad (2)$$

where  $F : \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^{d \times d} \mapsto \mathbb{R}$  is a given continuous function of the type

$$F(x, p, q) = \otimes_a \{b(x, a) \cdot p + \frac{1}{2} \text{tr}(q) + l(x, a)\}. \quad (3)$$

In the above, the  $\otimes$  is an operator which satisfies the following conditions:

- $\otimes_a [c\phi_1(x, a) + \phi_2(x)] = c \otimes_a [\phi_1(x, a)] + \phi_2(x)$
- $\otimes_a^x \phi_1(x, a) \leq \otimes_a \phi_2(x, a)$ , whenever  $\phi_1 \leq \phi_2$
- $\otimes_a \phi_1(x, a) - \otimes_a \phi_2(x, a) \leq K \max_a |\phi_1(x, a) - \phi_2(x, a)|$ ,

where  $\phi_1, \phi_2$  are real valued functions and  $c, K$  are constant. Many frequently used operators satisfy the aforementioned conditions. For instance,  $\max_a \phi(x, a)$  and  $\min_a \phi(x, a)$  are related to standard HJB equations and stochastic control problems.

Let's consider a type of  $d$ -dimension HJB as follows:

- Domain:

$$\mathcal{O} = \{x \in \mathbb{R}^d : 0 < x_i < 1, i = 1, 2, \dots, d\}.$$

- Equation on  $\mathcal{O}$ :

$$(\frac{1}{2}\Delta - \lambda)u(x) + \inf_a \left( \sum_{i=1}^d b_i(x, a) \frac{\partial u(x)}{\partial x_i} + \ell(x, a) \right) = 0.$$

- Dirichlet data on the boundary  $\partial\mathcal{O}$ :

$$u(x) = g(x).$$

## 2.2 Benchmark problem

To illustrate the effect of our numerical method, we consider a benchmark problem, on which we can compare the numerical solutions of differential equations with the exact ones. We consider the following  $d$ -dimensional HJB equation:

- Domain:

$$\mathcal{O} = \{x \in \mathbb{R}^d : 0 < x_i < 1, i = 1, 2, \dots, d\}.$$

- Equation on  $\mathcal{O}$ :

$$\frac{1}{2}\Delta u + \inf_{|a| \leq 3} (a \cdot \nabla u + \ell(x, a)) = 0,$$

where  $\ell(x, a) = d + 2|x|^2 + \frac{1}{2}|a|^2$ .

- Dirichlet data on the boundary  $\partial\mathcal{O}$ :

$$u(x) = -|x|^2.$$

And we can get that the exact solution of the above problem is  $u(x) = -|x|^2$  with  $a = 2x$ , and  $F$  can be written as

$$F(x, p, q) = \inf_{|a| \leq 3} \left\{ \frac{1}{2} \text{tr}(q) + a \cdot p + \ell(x, a) \right\}.$$

### 2.2.1 Methods of Approximation

First, we will introduce some notions of finite difference operators. Commonly used first order finite difference operators are Forward Finite Difference (FFD), Backward Finite Difference (BFD), Central Finite Difference (CFD) and Upwind Finite Difference (UFD). Here, to approximate the solution to the HJB equation, we mainly use CFD and UFD operators.

CFD is given below:

$$\frac{\partial}{\partial x_i} u(x) \approx \delta_{\pm h e_i} u(x) := \frac{1}{2}(\delta_{-h e_i} + \delta_{h e_i}) u(x) = \frac{u(x + h e_i) - u(x - h e_i)}{2h}.$$

UFD is given below:

$$\frac{\partial}{\partial x_i} u(x) \approx \begin{cases} \delta_{h e_i} u(x) = \frac{u(x + h e_i) - u(x)}{h}, & b(x) < 0 \\ \delta_{-h e_i} u(x) = \frac{u(x) - u(x - h e_i)}{h}, & b(x) \geq 0 \end{cases}$$

where  $b(\cdot)$  is the coefficient function of  $\frac{\partial}{\partial x_i} u(x)$ .

The Second order finite difference operators are the followings:

$$\frac{\partial^2}{\partial x_i^2} u(x) \approx \delta_{-he_i} \delta_{he_i} u(x) = \frac{u(x + he_i) - 2u(x) + u(x - he_i)}{h^2},$$

and if  $i \neq j$ , we use

$$\begin{aligned} \frac{\partial^2}{\partial x_i \partial x_j} u(x) &\approx \delta_{\pm he_i} \delta_{\pm he_j} u(x) \\ &= \frac{u(x + he_i + he_j) - u(x + he_i - he_j) - u(x - he_i + he_j) + u(x - he_i - he_j)}{4h^2}. \end{aligned}$$

### 2.2.2 CFD on HJB

Now, implement CFD on HJB for approximation, we substitute

$$\frac{\partial}{\partial x_i} u(x) \leftarrow \delta_{\pm he_i} u(x)$$

and

$$\frac{\partial^2}{\partial x_i^2} u(x) \leftarrow \delta_{-he_i} \delta_{he_i} u(x).$$

For simplicity, if we set

$$\gamma = \frac{d}{d + h^2 \lambda}, \quad p^h(x \pm he_i | x, a) = \frac{1}{2d} (1 \pm hb_i(x, a)), \quad \ell^h(x, a) = \frac{h^2 \ell(x, a)}{d},$$

then it yields DPP

$$v(x) = \gamma \inf_a \left\{ \ell^h(x, a) + \sum_{i=1}^d [p^h(x + he_i | x, a) v(x + he_i) + p^h(x - he_i | x, a) v(x - he_i)] \right\}.$$

In our benchmark problem, we have

$$\lambda = 0, \quad \gamma = 1, \quad b(x, a) = a, \quad \ell(x, a) = d + 2|x|^2 + \frac{1}{2}|a|^2.$$

### 2.2.3 UFD on HJB

If choose to implement UFD on HJB, we replace  $\frac{\partial}{\partial x_i} u(x)$  by the equation mentioned before and

$$\frac{\partial^2}{\partial x_i^2} u(x) \leftarrow \delta_{-he_i} \delta_{he_i} u(x).$$

Thus, in a similar way as before, for simplicity we set

$$\gamma = \frac{c}{c + h^2 \lambda}, \quad p^h(x \pm he_i | x, a) = \frac{1}{2c} (1 - 2hb_i^\mp(x, a)), \quad \ell^h(x, a) = \frac{h^2 \ell(x, a)}{c},$$

where  $c = d - h \sum_{i=1}^d |b_i|$ , then it also yields MDP

$$v(x) = \inf_a \gamma \left\{ \ell^h(x, a) + \sum_{i=1}^d [p^h(x + he_i | x, a) v(x + he_i) + p^h(x - he_i | x, a) v(x - he_i)] \right\}.$$

In our benchmark problem, we have

$$\lambda = 0, \quad b(x, a) = a, \quad \ell(x, a) = d + 2|x|^2 + \frac{1}{2}|a|^2.$$

### 3 Methodology

We want to solve the HJB equation by reinforcement learning and deep learning approach. The goals of our project are listed as follows:

- Recover the value iteration on central finite difference method.
- Do the value iteration on upwind finite difference scheme.
- Consider what is the highest dimension we can compute.