

A Probabilistic Solution to High-Dimensional Continuous-Time Macro and Finance Models

Ji Huang*

The Chinese University of Hong Kong

August 5, 2025

Abstract

This paper introduces the probabilistic formulation of continuous-time economic models: forward stochastic differential equations (SDEs) govern the dynamics of backward-looking variables, while backward SDEs capture those of forward-looking variables. Deep learning streamlines the search for probabilistic solutions, which are less sensitive to the “curse of dimensionality.” The paper proposes two types of algorithms and assesses their accuracy and computational speed by applying them to a multi-country model with an explicit solution under symmetric states. Numerical experiments indicate that as the model’s dimensionality increases, the computational speed of the partial differential equation approach deteriorates exponentially, whereas that of the probabilistic approach declines only linearly.

JEL Classification: C63, G21, E44

Keywords: backward stochastic differential equation, deep learning, the curse of dimensionality, economic dynamics

*I am incredibly grateful to Han Jiequn for answering numerous questions on their algorithms and BSDEs. My CUHK colleague Vinci Chow provided tremendous technical support on Tensorflow. Yu Jinghai and Geng Bojun offered superb research assistance. For insightful feedbacks, I would like to thank Hengjie Ai, Ken Judd, Wenlan Luo, Jianjun Miao, Galo Nuno, Shi Qiu, Yuliy Sannikov as well as seminar and conference participants of Zhejiang University, Shanghai JiaoTong, Fudan, SHUFE, Fudan Workshop on Economic Dynamics 2023, CICM 2023, AMES 2023, NASM 2023, SITE 2023 New Frontier in Asset Pricing, CESifo Macro Money, and International Finance 2023, CICF 2024, et al. My research benefits from the financial support of the General Research Fund of Hong Kong SAR (project number: 14501019). Contact Details: 9/F Esther Lee Building, The Chinese University of Hong Kong, Shatin, Hong Kong, China. Email: jihuang@cuhk.edu.hk.

Introduction

With the recent success of artificial intelligence, economists have begun applying deep learning to solve high-dimensional models including those in the continuous-time setting e.g., [Duarte](#), [Duarte and Silva \(2024\)](#), [Fernández-Villaverde, Hurtado and Nuno \(2022\)](#), and papers in the literature session. Thus far, all of these papers employ the analytic formulation and solve for equilibrium using partial differential equations (PDEs). Although deep learning provides a powerful toolbox for approximating high-dimensional functions, solving PDEs still suffers from the curse of dimensionality, particularly due to the Hessian matrices—critical components in PDEs that describe models driven by diffusion processes.

In this paper, I introduce Backward Stochastic Differential Equations (BSDEs)—the probabilistic counterpart to PDEs—to a general audience in economics, and more importantly, I highlight their computational advantages, especially the fact that they bypass the need to compute Hessian matrices. The development of BSDEs began with the seminal paper by [Pardoux and Peng \(1990\)](#). My work is largely inspired by [Han, Jentzen and E \(2018\)](#), the first paper to apply deep learning to high-dimensional BSDEs and the corresponding PDEs.

Due to its technical difficulty, I will first present the key intuition and terminology of BSDEs using a two-period binomial tree model, in which the European call option’s payoff at expiration is given by $Y_j = \max\{0, X_j - K\}$, $j = uu, ud, du, dd$. The no-arbitrage condition implies that the option price equals the value of the replicating portfolio (B_u, S_u) , where B_u is the risk-free saving and S_u is the number of shares of the underlying stock. Replicating the payoff from node u onward requires that (B_u, S_u) solves the following 2-by-2 linear system:

$$Y_{uu} = RB_u + X_{uu}S_u, \quad Y_{ud} = RB_u + X_{ud}S_u. \quad (1)$$

The no-arbitrage condition then yields the option price at node u as $Y_u = B_u + X_uS_u$. The same procedure applies to obtain Y_d and Y_0 . Next, I reinterpret the procedure for solving $\{Y_i, i = 0, u, d\}$ through the lens of a BSDE.

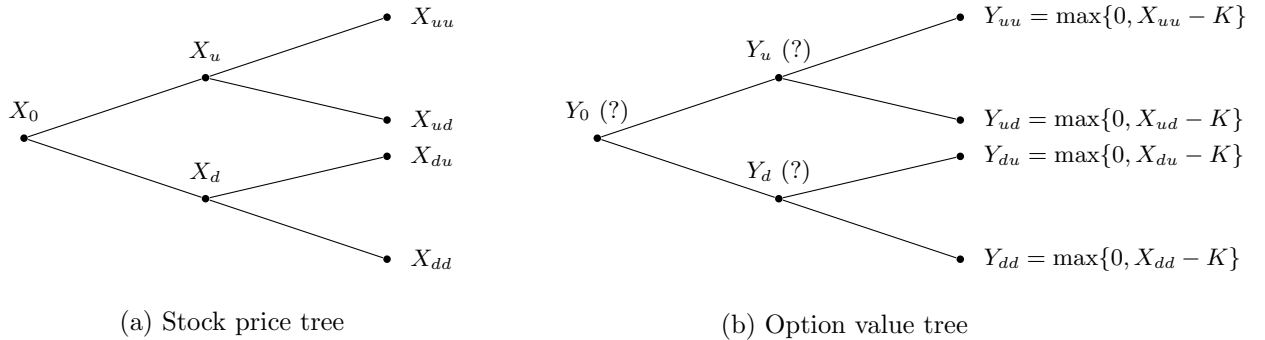


Figure 1: Binomial tree

Unlike Forward SDEs, the initial value Y_0 is determined as part of the solution implied by the terminal condition in a Backward SDE. More precisely, a BSDE’s solution refers to the sequence of pairs (Y_i, S_i) , $i = 0, u, d$. Each pair solves a 2-by-2 linear system, and these three systems, together with the terminal condition, constitute a BSDE.

Consider node u : the pair (Y_u, S_u) is the solution of the linear system

$$Y_{uu} = RY_u + (X_{uu} - RX_u) S_u, \quad Y_{ud} = RY_u + (X_{ud} - RX_u) S_u,$$

which is derived by substituting B_u in equations (1) with $Y_u - X_u S_u$. The pair (Y_u, S_u) can be interpreted as the projection of the vector (Y_{uu}, Y_{ud}) onto the plane spanned by (R, R) and $(X_{uu} - RX_u, X_{ud} - RX_u)$. The latter vector captures the risk of the underlying stock, and its coefficient S_u serves as the control that ensures the portfolio's payoff matches the option's payoff. Similar 2-by-2 linear systems can be constructed at nodes d and 0, yielding the solution pairs (Y_d, S_d) and (Y_0, S_0) , respectively.

Next, I introduce BSDEs using the continuous-time version of the same option pricing model. Suppose the short rate is a constant r and the underlying stock price X_t follows the SDE given in (2). The no-arbitrage condition implies that the option price equals the value of its replicating portfolio Y_t , whose dynamic budget constraint is specified by the SDE in (3):

$$dX_t = X_t (\mu dt + \sigma dW_t), \quad X_0 = x, \quad (2)$$

$$dY_t = (rY_t + (\mu - r) X_t S_t) dt + \sigma X_t S_t dW_t, \quad Y_T = \max\{0, X_T - K\}. \quad (3)$$

SDE (3) cannot be solved forward in time—even if Y_0 is given—because its evolution depends on the replicating strategy $\{S_t\}_{t \geq 0}$. Thus, (3) is a Backward SDE with terminal condition $Y_T = \max\{0, X_T - K\}$. The process $\{Y_t\}_{t \geq 0}$ must be solved backward along with $\{S_t\}_{t \geq 0}$.

To connect with the binomial tree example, consider the time discretization of BSDE (3):

$$\begin{aligned} Y_{t+\Delta} &= (1 + r\Delta) Y_t + ((\mu\Delta + \sigma(W_{t+\Delta} - W_t)) X_t - r\Delta X_t) S_t \\ &= (1 + r\Delta) Y_t + (X_{t+\Delta} - (1 + r\Delta) X_t) S_t, \end{aligned}$$

where each realization of $W_{t+\Delta}$ determines $X_{t+\Delta}$ via SDE (2) and also determines $Y_{t+\Delta}$, either via the terminal condition on Y_T if $t + \Delta = T$, or via backward induction along the path $\{X_s\}_{t+\Delta \leq s \leq T}$.¹ At time t , the pair (Y_t, S_t) is the solution of a system of linear equations, each corresponding to a realization of $W_{t+\Delta}$. In summary, the option pricing model is specified by the Forward-Backward SDE system (2)-(3). Let $y(t, X_t)$ and $s(t, X_t)$ denote its solution pair.

There are two approaches to solving for $y(t, X_t)$ and $s(t, X_t)$: the analytic (PDE) approach and the probabilistic (BSDE or Monte Carlo) approach. Following the PDE approach, apply Itô's formula to $Y_t = y(t, X_t)$:

$$dY_t = \left(y_t(t, X_t) + y_x(t, X_t) \mu X_t + \frac{1}{2} y_{xx}(t, X_t) \sigma^2 X_t^2 \right) dt + y_x(t, X_t) \sigma X_t dW_t.$$

Matching coefficients on dW_t implies $s(t, x) = y_x(t, x)$, and matching coefficients on dt yields:

$$y_t(t, x) + rxy_x(t, x) + \frac{1}{2} \sigma^2 x^2 y_{xx}(t, x) = ry(t, x). \quad (4)$$

¹Recall that Y_u and Y_d in the linear system at period 0 were solved using Y_{uu} and Y_{ud} , which were obtained from the linear systems at nodes u and d in period 1.

Combined with the terminal condition $y(T, x) = \max\{0, x - K\}$, this PDE characterizes the solution pair $y(t, X_t)$ and $s(t, X_t)$. Numerically solving for $y(t, X_t)$ becomes challenging when the state variable X_t is high-dimensional for two reasons: the difficulty of approximating a high-dimensional function and the computational cost of evaluating the Hessian matrix $y_{xx}(t, x)$. The enormous industrial success of deep learning suggests that neural network-based function spaces may address the first challenge. However, the second challenge remains a binding constraint that limits economists from exploring a wider range of models, to which the probabilistic approach can offer significant advantages.

The probabilistic (or Monte Carlo) approach has been widely used in derivative pricing. Next, I introduce how to combine deep learning with the probabilistic approach. A common feature of deep learning-based algorithms is trial and error. Let $y(t, x; \Theta)$ and $s(t, x; \Theta)$ denote parametric approximations of the solution pair, identified by a set of parameters Θ (trial), and let the FBSDE system (2)–(3) provide the assessment of the approximation (error). Machine learning libraries offer a rich set of algorithms that update the “trial” to minimize the “error.”

The assessment procedure is as follows. Suppose the time interval $[t, T]$ is evenly discretized as $t = t_0 \leq t_1 \leq \dots \leq t_N = T$, with step size $\Delta \equiv t_{n+1} - t_n$. Simulate M paths $\{X_n^m, Y_n^m\}_{n=0}^N$ according to the time-discretized FBSDE system:

$$\begin{aligned} X_{n+1}^m &= X_n^m + X_n^m (\mu \Delta + \sigma (W_{n+1}^m - W_n^m)), \\ Y_{n+1}^m &= Y_n^m + (r Y_n^m + (\mu - r) X_n^m S_n^m) \Delta + \sigma X_n^m S_n^m (W_{n+1}^m - W_n^m), \end{aligned}$$

where X_0^m is randomly initialized from the state space, $Y_0^m = y(t_0, X_0^m; \Theta)$, $S_n^m = s(t_n, X_n^m; \Theta)$, and $W_{n+1}^m - W_n^m$ is an independent draw from $N(0, \Delta)$ across both time steps n and paths m . The “error” is quantified by the following loss function:

$$\sum_{m=1}^M (Y_N^m - \max\{0, X_N^m - K\})^2 + \sum_{m=1}^M \sum_{n=1}^{N-1} (Y_n^m - y(t_n, X_n^m; \Theta))^2.$$

A variant of Newton’s gradient descent method can be applied to optimize the parameters Θ so as to minimize the loss function.

The example above illustrates that computing the Hessian matrix is unnecessary when applying the BSDE approach—a distinction that significantly improves numerical tractability. In Section 3, I present a multi-country macro-finance model and compare the computational performance of deep learning-based PDE and BSDE approaches under both CPU and high-performance GPU settings. The numerical experiments demonstrate that the runtime of the PDE approach deteriorates exponentially due to the need to compute Hessian matrices (see Figure 2). Using a CPU, the BSDE-based numerical scheme is more than 400 times faster than the PDE-based scheme when there are 15 countries. Moreover, storing these large Hessian matrices imposes hardware constraints. Although high-performance GPUs can efficiently process large-scale matrices, the PDE approach fails to execute when the model includes 15 countries, as even a GPU with 80 gigabytes of memory cannot store the prohibitively large Hessian matrices.

Mathematically, the PDE formulation defines functions in the state space, while the BSDE formulation does so in the path space. Their connection lies in the fact that the PDE formulation

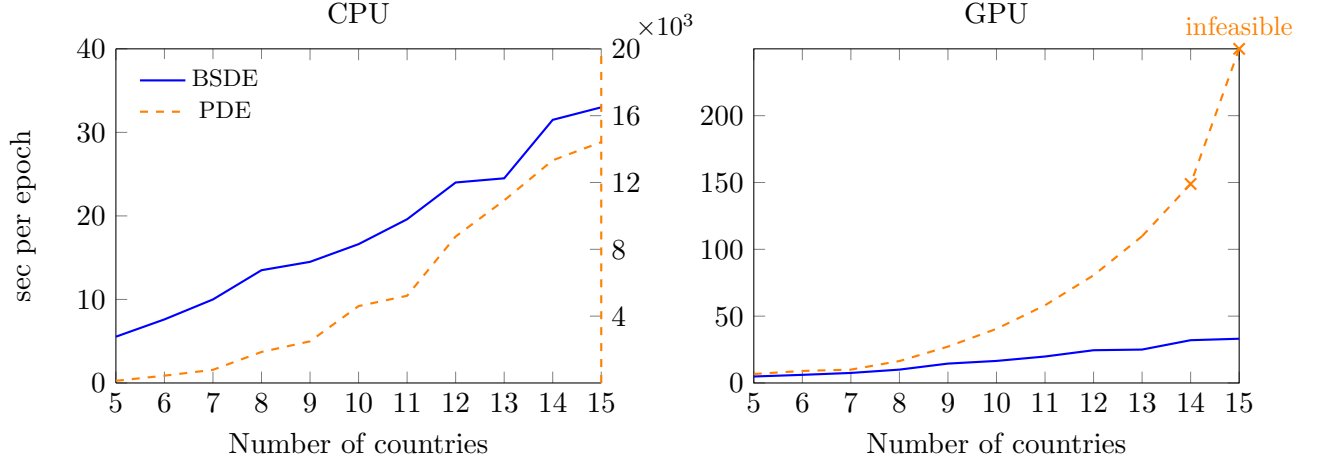


Figure 2: Computation speed (time usage per iteration) of the multi-country example in Section 3

captures the average effect on Y_t across all transition paths of X_t from t to $t + \Delta$. For instance, the left-hand side of PDE (4) is simply the conditional expectation:

$$\frac{1}{\Delta} E^Q [y(t + \Delta, X_{t+\Delta}) - y(t, X_t) | X_t = x],$$

where $E^Q[\cdot]$ denotes the expectation under the risk-neutral probability measure.² Hessian matrices are necessary for the local approximation of this conditional expectation when Δ is small—precisely the type of pre-computation for integrals advocated in discrete-time models by Judd, Maliar, Maliar and Tsener (2017). In contrast, the BSDE formulation defines the solution $y(t, x)$ by prescribing a co-movement between the backward-looking and forward-looking variables through a “control” $s(t, x)$. Although X_t and Y_t evolve according to their respective laws of motion (2) and (3), the “control” $S_t = s(t, X_t)$ ensures that the relationship $Y_t = y(t, X_t)$ holds almost surely along any realized path driven by the Brownian shocks $\{W_t\}_{t=0}^T$.

Most dynamic economic models are cast in an infinite-horizon, time-independent framework, whose solution can be characterized by the following FBSDE system:

$$\begin{aligned} dX_t &= b(X_t, Y_t, Z_t)dt + \sigma(X_t, Y_t, Z_t)dW_t, \\ dY_t &= -h(X_t, Y_t, Z_t)dt + Z_t dW_t, \\ Y_t &= y(X_t), \quad Z_t = z(X_t). \end{aligned}$$

The transition from finite-horizon, time-dependent FBSDEs to infinite-horizon, time-independent systems is analogous to that in dynamic programming. Fixing an arbitrary terminal date T , a finite-horizon FBSDE system maps a given function $y(\cdot)$ —which specifies the terminal condition $Y_T = y(X_T)$ —into a function $Y_0 = \hat{y}(X_0)$ at the initial date. Just as in the recursive formulation of dynamic programming, the solution to the infinite-horizon FBSDE system is the fixed point at which $y(\cdot) = \hat{y}(\cdot)$ for any choice of T .

All forward-looking stochastic processes in economics can be characterized by BSDEs. In

²Under the risk-neutral probability, X_t follows $dX_t = X_t(rdt + \sigma dW_t^Q)$, where W_t^Q is a Brownian motion under the risk-neutral probability.

the option pricing example, the BSDE is implied by the no-arbitrage principle and the dynamic budget constraint of a self-financing strategy. In Section 1, I derive BSDEs for asset prices based on the asset pricing equation using stochastic discount factors and the martingale representation theorem. Section 4 is devoted to the BSDE formulation of dynamic optimization problems, where the forward-looking process Y_t may represent lifetime expected discounted utility or its partial derivatives with respect to controlled state variables. The optimal controls are embedded in the functions $b(\cdot)$, $\sigma(\cdot)$, and $h(\cdot)$ via implicit functions derived from first-order conditions.

Lastly, but not least, I will briefly discuss market clearing conditions—the signature building block that distinguishes economics from related disciplines such as optimal control, mean-field games, multi-agent reinforcement learning, and their variants. Regardless of whether one adopts the PDE or BSDE formulation, market clearing conditions are embodied as static algebraic equations—either linear or nonlinear—which are particularly challenging to solve in high-dimensional models. To the best of my knowledge, no powerful and general method exists for addressing this difficulty other than penalizing “trials” that violate market clearing conditions. With this acknowledgment in place, the remainder of the paper will focus entirely on the BSDE formulation of economic dynamics.

Literature. As a powerful tool for analyzing forward-looking phenomena, BSDEs have been closely connected to problems in economics and finance ever since the seminal work of [Pardoux and Peng \(1990\)](#). In fact, [Duffie and Epstein \(1992\)](#) independently developed a result related to BSDEs in their theoretical work on recursive utility. Prior studies in economics have primarily used the BSDE framework to derive analytical results, e.g., [Schroder and Skiadas \(1999\)](#); [Chen and Epstein \(2002\)](#). This paper is the first to highlight its computational advantages.

Numerically solving BSDEs was difficult prior to the deep learning era, which likely explains why economists have paid little attention to the probabilistic formulation of forward-looking problems. Recently, an influential paper in applied mathematics demonstrated the power of deep learning in solving high-dimensional BSDEs ([Han, Jentzen and E, 2018](#)). In fact, the idea of formulating the solvability of FBSDEs as an optimal control problem dates back to [Ma and Yong \(1995\)](#). The modest contribution of this paper is to tailor the BSDE framework to an infinite-horizon, time-independent setting that is more representative of many economic models.

Broadly speaking, this paper is related to recent work that applies deep learning to solve high-dimensional macroeconomic and finance models. Examples in the discrete-time setting include [Azinovic, Gaegauf and Scheidegger \(2022\)](#), [Maliar, Maliar and Winant \(2021\)](#), and [Han, Yang and E \(2021\)](#); in the continuous-time setting, relevant studies include [Gopalakrishna \(2021\)](#), [Sauzet \(2021\)](#), [Gu, Lauriere, Merkel and Payne \(2024\)](#), [Payne, Rebei and Yang \(2025\)](#), [Gopalakrishna, Payne and Gu \(2024\)](#), and [Fan, Qiao, Jiao, Gu, Li and Lu \(2024\)](#). All the continuous-time studies adopt the analytic approach and solve PDEs using deep learning—specifically, the Physics-Informed Neural Network (PINN) method originally developed in physics.

The organization of the paper is as follows. In Section 1, I use the BSDE formulation of asset pricing as an example to compare with the PDE formulation, and I discuss their connections and differences. Section 2 focuses on two types of numerical schemes for solving Forward-Backward SDE systems. Section 3 is devoted to a multi-country macro-finance model and compares the

computational efficiency of the BSDE and PDE approaches. In Section 4, I present the BSDE formulation of typical dynamic optimization problems. Section 5 demonstrates how to employ the Malliavin derivative to capture the propagation of diffusion shocks in a continuous-time model and illustrates its probabilistic numerical scheme.

1 Asset Pricing under BSDE and PDE Formulations

To fully illustrate the key features of the probabilistic approach, I consider a general asset pricing model in which the laws of motion for the asset price and the aggregate state variable are intertwined. Note that the applications of the BSDE formulation extend well beyond asset pricing. The only reason the detailed discussion of dynamic optimization is deferred to Section 4 is that it contains too much material and could easily overwhelm the audience.

1.1 Backward SDEs

Consider a risky asset, whose price q_t has the following law of motion

$$dq_t = \mu_t^q dt + \sigma_t^q dW_t, \quad (5)$$

where W_t is a standard Brownian motion. The dividend of the asset is given by a function $D(\cdot)$, whose argument is the aggregate state variable X_t . The dynamics of X_t are dictated by

$$dX_t = \mu^x(X_t, q_t, \sigma_t^q) dt + \sigma^x(X_t, q_t, \sigma_t^q) dW_t,$$

where $\mu^x(\cdot)$ and $\sigma^x(\cdot)$ are known functions. Here, I consider a comprehensive setting in which the drift and volatility terms of the state variable depend on the level of the asset price q_t and also its exposure to the aggregate risk σ_t^q . This setting includes many asset pricing examples, such as derivative pricing, where the law of motion for the state variable is independent of the asset pricing result.

The focus of this subsection is to interpret the law of motion for q_t (5) as a BSDE—that is, to derive μ_t^q and σ_t^q based on the asset pricing equation shown below. In addition, assume all variables in this subsection are one-dimensional, which I hope helps the audience quickly digest the probabilistic formulation.

Suppose the stochastic discount factor is given by the function $m(X_t)$, whose law of motion is derived using Itô's formula:

$$dm_t = \underbrace{\left(m'(X_t) \mu^x(X_t, q_t, \sigma_t^q) + 0.5 m''(X_t) (\sigma^x(X_t, q_t, \sigma_t^q))^2 \right)}_{\equiv \mu^m(X_t, q_t, \sigma_t^q)} dt + \underbrace{\left(m'(X_t) \sigma^x(X_t, q_t, \sigma_t^q) \right)}_{\equiv \sigma^m(X_t, q_t, \sigma_t^q)} dW_t$$

The asset pricing equation is

$$m(X_t) q_t = E_t \left[\int_t^{+\infty} m(X_u) D(X_u) du \right] \quad (6)$$

To derive the probabilistic formulation of the asset pricing equation, I first construct a martin-

gale:

$$\int_0^t m(X_u) D(X_u) du + m(X_t) q_t = E_t \left[\int_0^{+\infty} m(X_u) D(X_u) du \right]$$

The martingale representation theorem implies that there exists a progressively measurable stochastic process Z_t such that

$$\int_0^t m(X_u) D(X_u) du + m(X_t) q_t = E_0 \left[\int_0^{+\infty} m(X_u) D(X_u) du \right] + \int_0^t Z_t dW_u. \quad (7)$$

The martingale representation can be interpreted as infinitely many “regressions,” each of which corresponds to a path driven by $\{W_u\}$ realized up to time t . Replacing t with $t + \Delta$, where Δ is a small positive constant,

$$\int_0^{t+\Delta} m(X_u) D(X_u) du + m(X_{t+\Delta}) q_{t+\Delta} = E_0 \left[\int_0^{+\infty} m(X_u) D(X_u) du \right] + \int_0^{t+\Delta} Z_t dW_u.$$

Taking the difference between the two integral equations yields

$$m(X_{t+\Delta}) q_{t+\Delta} + \int_t^{t+\Delta} m(X_u) D(X_u) du = m(X_t) q_t + \int_t^{t+\Delta} Z_t dW_u, \quad (8)$$

$$m(X_{t+\Delta}) q_{t+\Delta} + m(X_t) D(X_t) \Delta = m(X_t) q_t + Z_t (W_{t+\Delta} - W_t). \quad (9)$$

Equation (9) is an approximation of the integral equation (8) for small Δ . Given the information available up to time t , the martingale representation can be considered a linear regression of the left-hand side random variable against 1 and the innovation $W_{t+\Delta} - W_t$, where q_t and Z_t are *unknown* constants to be estimated through the regression. For each path leading up to time t , there is a corresponding constant Z_t , which is what the technical term “progressively measurable” refers to.

Integral equation (8) conveys the dependence of the current asset price q_t on the asset price at the future date $t + \Delta$ and the dividend paid from t to $t + \Delta$. It is common in economics that the current value of a forward-looking variable is determined in a *backward* fashion, as in the linear regression above, which helps explain why Equation (8) is referred to as a Backward SDE in mathematics. Taking the expectation of Equation (8) yields the asset pricing equation more familiar to most economists, connecting the current price of the asset with its future value and the dividend to be paid. The BSDE formulation (8) characterizes the same relationship path-by-path, but additionally requires the solution of Z_t .

The differential form of the integral equation (8) is given by

$$d(m(X_t) q_t) = -m(X_t) D(X_t) dt + Z_t dW_t$$

Apply Itô’s formula to $m_t q_t$ based on their respective laws of motion:

$$d(m_t q_t) = (m_t \mu_t^q + q_t \mu^m(X_t, q_t, \sigma_t^q) + \sigma_t^q \sigma^m(X_t, q_t, \sigma_t^q)) dt + (m_t \sigma_t^q + q_t \sigma^m(X_t, q_t, \sigma_t^q)) dW_t.$$

Matching coefficients in dt and dW_t implies

$$\mu^q(X_t, q_t, \sigma_t^q) = -D(X_t) - \frac{1}{m(X_t)} (q_t \mu^m(X_t, q_t, \sigma_t^q) + \sigma_t^q \sigma^m(X_t, q_t, \sigma_t^q)) \quad (10)$$

$$\sigma_t^q = \frac{1}{m(X_t)} (Z_t - q_t \sigma^m(X_t, q_t, \sigma_t^q)). \quad (11)$$

Equation (10) implies that the drift of q_t can be written as a function of X_t , q_t , and σ_t^q , denoted $\mu^q(X_t, q_t, \sigma_t^q)$. Assuming the regularity of the implicit function defined by Equation (11), the existence of Z_t is equivalent to the existence of σ_t^q . Hereafter, I will work directly with the Forward-Backward SDE system:

$$\text{(backward)} \quad dq_t = \mu^q(X_t, q_t, \sigma_t^q) dt + \sigma_t^q dW_t \quad (12)$$

$$\text{(forward)} \quad dX_t = \mu^x(X_t, q_t, \sigma_t^q) dt + \sigma^x(X_t, q_t, \sigma_t^q) dW_t \quad (13)$$

1.2 State-Space and Path-Space Mappings under Time Discretization

The asset pricing equation (6) suggests a Markovian solution for the asset price $q(\cdot)$, written as

$$m(x) q(x) = E_t \left[\int_t^{+\infty} m(X_u) D(X_u) du \mid X_t = x \right].$$

However, it is still not straightforward to solve for $q(\cdot)$ using direct Monte Carlo simulation because the Forward SDE (13) indicates that the law of motion for X_t depends on the Markovian solution $q(\cdot)$ and its first-order derivative $q'(\cdot)$ evaluated at x . To see the latter, recall that the volatility of the state variable X_t depends on the asset price volatility σ_t^q . Given the Markovian solution $q(\cdot)$, Itô's formula implies that σ_t^q at $X_t = x$ should satisfy

$$\sigma^q = q'(x) \sigma^x(x, q(x), \sigma^q), \quad (14)$$

which depends on both the level $q(x)$ and its local property $q'(x)$.

A viable approach to solving for $q(\cdot)$ is to formulate a functional mapping based on equation (6), under which $q(\cdot)$ is a fixed point. In this subsection, I first review the standard analytic formulation in the *state space*, and then define a functional mapping based on the FBSDE system described above in the *path space*.

Recall that Δ is a small positive constant. Let us rewrite the asset pricing equation (6) as follows:

$$\begin{aligned} m(X_t) q_t &= E \left[\int_t^{t+\Delta} m(X_u) D(X_u) du + \int_{t+\Delta}^{+\infty} m(X_u) D(X_u) du \right] \\ &= m(X_t) D(X_t) \Delta + E_t \left[\int_{t+\Delta}^{+\infty} m(X_u) D(X_u) du \right] \\ &= m(X_t) D(X_t) \Delta + E_t [m(X_{t+\Delta}) q_{t+\Delta}]. \end{aligned}$$

The recursive formulation and the Itô formula above suggest the functional mapping T^a as

$$\begin{aligned} T^a \begin{pmatrix} q(x) \\ \sigma^q(x) \end{pmatrix} &:= \begin{pmatrix} D(x) \Delta + \frac{1}{m(x)} E[m(x_w) q(x_w) | x] \\ q'(x) \sigma^x(x, q(x), \sigma^q(x)) \end{pmatrix}, \\ x_w &= x + \mu^x(x, q(x), \sigma^q(x)) \Delta + \sigma^x(x, q(x), \sigma^q(x)) w, \\ w &\sim N(0, \Delta), \end{aligned}$$

which takes the asset price $q(\cdot)$ at the “end” of the period and the asset volatility $\sigma^q(\cdot)$ as inputs, and returns the asset price and volatility at the “current” state x .

The fixed point of the mapping T^a satisfies

$$T^a \begin{pmatrix} q(x) \\ \sigma^q(x) \end{pmatrix} = \begin{pmatrix} q(x) \\ \sigma^q(x) \end{pmatrix}.$$

The conditional expectation in the mapping T^a averages discounted asset prices over all possible paths in the state space that originate from the initial state x . In practice, this mapping is translated into a PDE, the numerical implementation of which will be discussed later.

The probabilistic formulation also gives rise to a functional mapping, which, however, lies in the path space. The discretization of the FBSDE system leads to

$$\begin{aligned} q_{t+\Delta} &= q_t + \mu^q(X_t, q_t, \sigma_t^q) \Delta + \sigma_t^q (W_{t+\Delta} - W_t), \\ X_{t+\Delta} &= X_t + \mu^x(X_t, q_t, \sigma_t^q) \Delta + \sigma^x(X_t, q_t, \sigma_t^q) (W_{t+\Delta} - W_t). \end{aligned}$$

Let me re-emphasize the interpretation of the Backward SDE with respect to the asset price q_t : the forward-looking nature of the asset price implies that its current value q_t depends on the price in the next period $q_{t+\Delta}$. This relationship holds path-by-path due to the freedom in choosing the “coefficient” σ_t^q —the exposure to the innovation $W_{t+\Delta} - W_t$. The existence of a Markovian solution implies that the time-discretized BSDE above can be rewritten as

$$q(X_{t+\Delta}) = q(X_t) + \mu^q(X_t, q_t, \sigma_t^q) \Delta + \sigma^q(X_t) (W_{t+\Delta} - W_t), \quad (15)$$

which suggests the following functional mapping for $q(\cdot)$ and $\sigma^q(\cdot)$:

$$\begin{aligned} T^p \begin{pmatrix} q(x) \\ \sigma^q(x) \end{pmatrix} &:= \arg \min_{\hat{q}, \hat{\sigma}^q} \sum_{d=1}^D \left(\hat{q} + \mu^q(x, q(x), \sigma^q(x)) \Delta + \hat{\sigma}^q w^d - q(x^d) \right)^2, \\ x^d &= x + \mu^x(x, q(x), \sigma^q(x)) \Delta + \sigma^x(x, q(x), \sigma^q(x)) w^d, \\ w^d &\sim N(0, \Delta), \quad d = 1, \dots, D, \end{aligned}$$

where D is the number of paths. For a given current state x , the functional mapping T^p also takes the asset price $q(\cdot)$ at the “end” of the period and the asset volatility $\sigma^q(\cdot)$ as inputs, and returns the current asset price \hat{q} and its volatility $\hat{\sigma}^q$ such that the BSDE holds along all D sampled paths from the state space. As with the interpretation of the martingale representation above, \hat{q} and $\hat{\sigma}^q$ serve as the intercept and coefficient, respectively, of the least squares problem

specified in mapping T^p . The solution $(q(x), \sigma^q(x))$ is the fixed point:

$$T^p \begin{pmatrix} q(x) \\ \sigma^q(x) \end{pmatrix} = \begin{pmatrix} q(x) \\ \sigma^q(x) \end{pmatrix}.$$

The standard ordinary least squares result in econometrics implies that the optimal estimate \hat{q} in mapping T^p satisfies

$$\hat{q} = -\mu^q(x, q(x), \sigma^q(x))\Delta + \frac{1}{D} \sum_{d=1}^D \left(q(x^d) - \hat{\sigma}^q w^d \right),$$

which approximates the mapping T^a with respect to $q(\cdot)$ when the number of sampled paths D is sufficiently large.³ However, it is critical to note that the minimum number of sampled paths D required to identify \hat{q} is only two in this example—that is, one more than the dimensionality of the exogenous shock.

1.3 Equivalence of BSDE and PDE Representations

As mentioned earlier, the practical way to solve for $q(\cdot)$ under the analytic formulation is to derive a differential equation based on the fixed point defined by the mapping T^a , letting Δ converge to zero. In this subsection, however, I derive the differential equation based on the probabilistic formulation to highlight both the equivalence and the differences between the two approaches.

Given the Markov solution specified by the BSDE (15), applying a Taylor expansion to the left-hand side, $q(X_{t+\Delta})$, around X_t yields

$$\begin{aligned} q(X_{t+\Delta}) &= q(X_t) + (q'(X_t)) (\mu^x(X_t, q_t, \sigma_t^q) \Delta + \sigma^x(X_t, q_t, \sigma_t^q) (W_{t+\Delta} - W_t)) \\ &\quad + \frac{1}{2} q''(X_t) (\sigma^x(X_t, q_t, \sigma_t^q) (W_{t+\Delta} - W_t))^2, \end{aligned}$$

where terms of order higher than Δ are omitted. Recall that $W_{t+\Delta} - W_t$ is a random variable, and BSDE (15) must hold for any realization of this Brownian increment. Hence, the coefficients of $W_{t+\Delta} - W_t$ on both sides of Equation (15) must be equal, which yields Equation (14) at

³Note

$$\begin{aligned} E_t[m(X_{t+\Delta})q(X_{t+\Delta})] &= E_t[(m(X_{t+\Delta}) - m(X_t) + m(X_t))q(X_{t+\Delta})] \\ &= E_t[(m(X_{t+\Delta}) - m(X_t))q(X_{t+\Delta})] + E_t[m(X_t)q(X_{t+\Delta})] \\ &= E_t[(\mu_t^m \Delta + \sigma_t^m (W_{t+\Delta} - W_t))(q_t + \sigma_t^q \Delta + \sigma_t^q (W_{t+\Delta} - W_t))] + m(X_t) E_t[q(X_{t+\Delta})] \\ &= (\mu_t^m q_t + \sigma_t^m \sigma_t^q) \Delta + m(X_t) E_t[q(X_{t+\Delta})], \end{aligned}$$

where the third equality uses a Taylor expansion or Itô's formula, and the fourth equality omits terms of order higher than Δ . Then, mapping T^a with respect to $q(\cdot)$ can be rewritten as

$$\begin{aligned} &D(X_t) \Delta + \frac{1}{m(X_t)} E[m(X_{t+\Delta})q(X_{t+\Delta}) | X_t] \\ &= \left(D_t + \frac{1}{m_t} (\mu_t^m q_t + \sigma_t^m \sigma_t^q) \right) \Delta + E[q(X_{t+\Delta}) | X_t] \\ &= -\mu_t^q \Delta + E[q(X_{t+\Delta}) | X_t]. \end{aligned}$$

$X_t = x$ —that is, the Itô formula condition for σ_t^q . Under the probabilistic formulation, Itô's condition (14) holds automatically when solving for the fixed point of the mapping T^p , because the volatility term σ_t^q is endogenously identified through the variation in $W_{t+\Delta} - W_t$. This feature is particularly valuable when solving the nonlinear equation (14) numerically would be computationally expensive.

Taking expectations on both sides of Equation (15) (after applying the Taylor expansion) and subtracting $q(X_t)$ yields

$$q'(X_t) \mu^x(X_t, q_t, \sigma_t^q) + \frac{1}{2} q''(X_t) (\sigma^x(X_t, q_t, \sigma_t^q))^2 = \mu^q(X_t, q_t, \sigma_t^q). \quad (16)$$

The left-hand side of the differential equation above is linked to the mapping T^p under the probabilistic formulation via Itô's formula:

$$\begin{aligned} q'(X_t) \mu^x(X_t, q_t, \sigma_t^q) + \frac{1}{2} q''(X_t) (\sigma^x(X_t, q_t, \sigma_t^q))^2 &= \frac{1}{\Delta} (E_t[q(X_{t+\Delta}) - q(X_t)]) \\ &= \frac{1}{\Delta} \left(\frac{1}{D} \sum_{d=1}^D q(x^d) - \hat{q} \right), \end{aligned}$$

where the second equation is an approximation under the mapping T^p when $X_t = x$ and D is sufficiently large.

To summarize, the differential equation that defines the Markov solution $q(\cdot)$ is given by

$$\begin{aligned} 0 &= D(x) + q(x) \frac{\mu^m(x, q(x), \sigma^q)}{m(x)} + \sigma^q \frac{\sigma^m(x, q(x), \sigma^q)}{m(x)} \\ &\quad + q'(x) \mu^x(x, q(x), \sigma^q) + \frac{1}{2} q''(x) (\sigma^x(x, q(x), \sigma^q))^2, \\ \sigma^q &= q'(x) \sigma^x(x, q(x), \sigma^q). \end{aligned}$$

The two equations above correspond to the intercept and coefficient terms of the linear regression specified by the mapping T^p under the probabilistic formulation, respectively.

1.4 Why the Probabilistic Formulation Scales Better

Assuming the audience has understood the intuition behind the probabilistic formulation, I now turn to multi-dimensional cases and discuss the differences between the analytic (PDE) and probabilistic (BSDE) formulations from the perspective of numerical computation.

Suppose the state variable $X_t \in \mathbb{R}^n$ is n -dimensional, and there are m different assets, i.e., $q_t \in \mathbb{R}^m$ is m -dimensional. The Brownian motion W_t is a d -dimensional standard Brownian process. Accordingly, the drift of asset prices μ_t^q takes values in \mathbb{R}^m , and the volatility σ_t^q takes values in the matrix space $\mathbb{R}^{m \times d}$, that is,

$$\sigma_t^q \equiv \begin{bmatrix} \sigma_t^{q,1} \\ \sigma_t^{q,2} \\ \vdots \\ \sigma_t^{q,m} \end{bmatrix} \in \mathbb{R}^{m \times d}, \quad \sigma_t^{q,i} \in \mathbb{R}^{1 \times d}.$$

The drift and volatility terms of the state variable X_t are given by

$$\mu^x(X, q, \sigma^q) : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{m \times d} \rightarrow \mathbb{R}^n, \quad \sigma^x(X, q, \sigma^q) : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{m \times d} \rightarrow \mathbb{R}^{n \times d}.$$

Given the stochastic discount factor function $m(X) : \mathbb{R}^n \rightarrow \mathbb{R}$, the multi-dimensional Itô formula implies

$$\begin{aligned} \mu^m(X_t, q_t, \sigma_t^q) &= \nabla_x m(X_t) \mu^x(X_t, q_t, \sigma_t^q) + \frac{1}{2} \text{Tr} \left\{ \sigma^x(X_t, q_t, \sigma_t^q)^T H_m(X_t) \sigma^x(X_t, q_t, \sigma_t^q) \right\}, \\ \sigma^m(X_t, q_t, \sigma_t^q) &= \nabla_x m(X_t) \sigma^x(X_t, q_t, \sigma_t^q), \end{aligned}$$

where $\text{Tr}\{\cdot\}$ denotes the trace of a matrix, T indicates matrix transpose, ∇_x denotes the gradient (a row vector of dimension n), and H_m is the Hessian matrix of the function $m(\cdot)$. The BSDE with respect to q_t remains the same as in Equation (12).

There is no essential change to the functional mapping T^p based on the probabilistic formulation other than that there are multiple assets or BSDEs to consider

$$\begin{aligned} T^p \left(\begin{array}{c} q(x) \\ \sigma^q(x) \end{array} \right) &: \arg \min_{\hat{q}, \hat{\sigma}^q} \sum_{i=1}^m \sum_{d=1}^D \left(\hat{q}^i + \mu^{q,i}(x, q(x), \sigma^q(x)) \Delta + \hat{\sigma}^{q,i} w^d - q^i(x^d) \right)^2 \\ x^d &= x + \mu^m(x, q(x), \sigma^q(x)) \Delta + \sigma^m(x, q(x), \sigma^q(x)) w^d \\ w^d &\sim N(0, I_\Delta), d = 1, \dots, D \end{aligned}$$

where I_Δ is the product of Δ and a d -dimensional identity matrix. Under the analytic formulation, the PDE system is

$$\begin{aligned} 0 &= D^i(x) + q^i(x) \frac{\mu^m(x, q(x), \sigma^q)}{m(x)} + \sigma^{q,i} \frac{(\sigma^m(x, q(x), \sigma^q))^T}{m(x)} \\ &\quad + \nabla_x q^i(x) \mu^x(x, q(x), \sigma^q) + \frac{1}{2} \text{Tr} \left\{ (\sigma^x(x, q, \sigma^q))^T (H_{q,i}(x)) \sigma^x(x, q, \sigma^q) \right\} \\ \sigma^{q,i} &= \nabla_x q^i(x) \sigma^x(x, q, \sigma^q) \quad i = 1, \dots, m, \end{aligned} \tag{17}$$

where $H_{q,i}(x)$ denotes the Hessian matrix of function $q^i(x)$.

There are two major differences between the probabilistic and analytic formulations that critically impact their computational efficiency. The first is the number of functional evaluations. When solving for the fixed point under either formulation, we approximate $q(\cdot)$ within a certain parametrized functional space, denoted as $\hat{q}(\cdot; \Theta)$. For a given point in the state space, the primary computational cost under the analytic formulation lies in evaluating $\nabla_x q^i(x)$ and the Hessian matrix $H_{q,i}(x)$ in PDE (17), which require approximately n and $0.5n^2$ functional evaluations, respectively. Recall that each direction requires at least one such evaluation. The calculation of the Hessian matrix is especially costly because it is not only time-consuming but also memory-intensive.

In applied mathematics, the challenge of computing high-dimensional derivatives using deep learning is referred to as the “curse of dimensionality” in the context of PINNs (Hu, Shukla, Karniadakis and Kawaguchi, 2024). Although Duarte et al. (2024) propose a technique to

address this challenge, its practical performance remains unclear, as they apply the method only to a special case and do not extend it to other applications in the main text. My understanding of their approach is that it embeds Hessian matrix calculation into automatic differentiation, which nonetheless remains highly demanding in terms of both computation time and resources.

Under the probabilistic formulation, the minimum number of paths required to identify $(\hat{q}^i, \hat{\sigma}^{q,i})$ at a given point in the state space is $1 + d$. Hence, only $1 + d$ functional evaluations of $\hat{q}^i(\cdot; \Theta)$ are involved. In the next section, which focuses on a multi-country macro-finance model, I demonstrate these differences more vividly through a series of numerical exercises.

The second major difference lies in solving for the asset price volatility σ^q in Equation (18). If this equation is nonlinear in σ^q , the fixed point search for $q(\cdot)$ becomes significantly more complex, as each update of $\hat{q}(\cdot; \Theta)$ requires solving a system of nonlinear equations. Even when Equation (18) is linear in σ^q , matrix inversion is still necessary. Moreover, the approximated function $\hat{q}(\cdot; \Theta)$ and its gradient $\nabla_x \hat{q}(x; \Theta)$ appear within the matrices. Therefore, any algorithm that updates $\hat{q}(\cdot; \Theta)$ must internalize its effect on the matrix inversion process.

Smoothness. Applying the PDE approach requires that its classical solution $q(\cdot)$ be twice differentiable everywhere in its domain. For broader applicability, mathematicians have developed a weaker solution concept—viscosity solutions. However, it remains unclear whether this concept is compatible with the auto-differentiation techniques on which deep learning heavily relies.⁴ Non-differentiable points are not uncommon in macro-finance models due to the effects of asset fire-sales, e.g., the logarithmic preference version of Brunnermeier and Sannikov (2014). Economists, however, need not be concerned with such smoothness requirements, as FBSDE systems do not require their solutions to be differentiable.

2 Deep Learning-Based Probabilistic Approach

The focus of this section is to present two types of deep learning-based numerical methods for solving Forward-Backward SDE systems. To keep the presentation as simple as possible, I make two compromises. First, I omit market clearing conditions. Second, I take it for granted that individual dynamic optimality conditions have already been embedded into the FBSDE systems. Section 4 will discuss how to cast standard dynamic optimization problems as BSDEs.

2.1 Forward-Backward SDE

Consider the following infinite-horizon time-independent Forward-Backward SDE system:

$$\begin{aligned} dX_t &= b(X_t, Y_t, Z_t) dt + \sigma(X_t, Y_t, Z_t) dW_t, \\ -dY_t &= h(X_t, Y_t, Z_t) dt - Z_t dW_t, \\ X_0 &= x, \end{aligned}$$

⁴The finite difference method is compatible with viscosity solutions, but it becomes impractical in dimensions higher than four or five due to the curse of dimensionality.

where $x \in \mathbb{R}^n$,

$$\begin{aligned} b &: \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{m \times d} \rightarrow \mathbb{R}^n, \\ \sigma &: \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{m \times d} \rightarrow \mathbb{R}^{n \times d}, \\ h &: \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{m \times d} \rightarrow \mathbb{R}^m. \end{aligned}$$

Here, W_t is a standard d -dimensional Brownian motion defined on a complete probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t \geq 0}, P)$ satisfying the *usual conditions*.⁵ The solution of the FBSDEs, $\{X_t, Y_t, Z_t\}_{t \geq 0}$, is required to be $\{\mathcal{F}_t\}_{t \geq 0}$ -adapted. The well-posedness of the system above is beyond the scope of this paper.⁶ For the rest of the paper, I assume that the system admits a unique solution.

Markov Solution. In view of the infinite horizon setting and the time homogeneity of $b(\cdot)$, $\sigma(\cdot)$, and $h(\cdot)$, it is natural to expect that the FBSDE system admits a Markov solution $\{X_t, Y_t, Z_t\}_{t \geq 0}$, such that there exist functions $y(\cdot)$ and $z(\cdot)$ mapping X_t to Y_t and Z_t , respectively, i.e.,

$$Y_t = y(X_t), \quad Z_t = z(X_t), \quad \forall t \geq 0.$$

Rigorously speaking, the existence of a Markov solution requires the ergodicity of $\{X_t, Y_t, Z_t\}_{t \geq 0}$ and a transversality condition with respect to Y_t . The interpretation of the Markov solution is that, regardless of the initial state $X_0 = x$ and the realized path of $\{W_t\}_{t \geq 0}$, the process $Z_t = z(X_t)$ ensures that the “simulated” paths of X_t and Y_t , starting from X_0 and $Y_0 = y(X_0)$, always satisfy the mapping $Y_t = y(X_t)$ for all $t > 0$. This is the core insight for the design of all algorithms presented below.

2.2 Numerical Methods

The deep learning-based numerical method aims to approximate the functions $y(\cdot)$ and $z(\cdot)$ in a parameterized functional space using feedforward neural networks, such that the Markov property, or the corresponding fixed-point property, holds.⁷ In particular, let $\hat{y}(\cdot; \Theta)$ and $\hat{z}(\cdot; \Theta)$ denote the neural network approximations of $y(\cdot)$ and $z(\cdot)$, respectively, where Θ is the set of parameters specifying the neural network. The parameters Θ are optimized so that, for all sample paths $\{X_t^i, Y_t^i, Z_t^i\}_{t \geq 0}$ simulated according to the FBSDE, the Markov property is satisfied; that is, $Y_t^i = \hat{y}(X_t^i; \Theta)$ for all $t \geq 0$.

The core of the deep learning-based numerical scheme is to construct a loss function that ensures the Markov property, and to minimize this loss over the parameters Θ using deep learning packages in Python, such as TensorFlow or PyTorch. [Goodfellow, Bengio and Courville \(2016\)](#) and [Zhang, Lipton, Li and Smola \(2023\)](#) are standard references for deep neural networks, optimization algorithms, and related topics in deep learning.

The probabilistic formulation allows for various numerical schemes to solve for the Markov solution. In this paper, I consider two types. The first is in the spirit of the functional mapping T^b in Section 1.2, which evaluates the value of a forward-looking variable at the next period

⁵See [Karatzas and Shreve \(1998\)](#) for details of technical terms.

⁶See Chapter 8 of [Zhang \(2017\)](#) for a survey of papers on the well-posedness of FBSDEs.

⁷See Chapter 6 of [Goodfellow et al. \(2016\)](#) for details on feedforward neural networks.

(using $\hat{y}(X_{t+\Delta}; \Theta)$) and solves for its current value Y_t based on the BSDE. The current value is then used to check the fixed-point property $Y_t = \hat{y}(X_t; \Theta)$. This method is referred to as the Backward Euler scheme in mathematics; for details, see Chapter 5 of [Zhang \(2017\)](#).

The second method takes the current value of the forward-looking variable (given by $\hat{y}(X_t; \Theta)$) and simulates its value at the next period $Y_{t+\Delta}$ in a forward fashion based on the BSDE, which is then used to check $Y_{t+\Delta} = \hat{y}(X_{t+\Delta}; \Theta)$. The plausibility of the forward Euler scheme has been greatly enhanced by recent developments in machine learning ([E, Han and Jentzen, 2017](#); [Han, Jentzen and E, 2018](#)).

2.2.1 Backward Euler Scheme

The numerical scheme in this subsection seeks the fixed point of the functional mapping T^p discussed in Section 1.2. First, the scheme samples M initial state points in the state space, denoted by $\{x_i\}_{i=1}^M$, and for each initial state generates D independent Brownian shocks, approximated by a normal distribution with zero mean and standard deviation $\sqrt{\Delta}$, i.e., $\{w_{i,j}\}_{i=1,j=1}^{M,D}$. For brevity, we omit the state index i in the following exposition.

Let K denote

$$K \equiv \begin{bmatrix} 1 & (w_1)^T \\ 1 & (w_2)^T \\ \vdots & \vdots \\ 1 & (w_D)^T \end{bmatrix}$$

which is a $(D) \times (1 + d)$ matrix.⁸

The loss function is constructed as follows:

1. Compute $y = \hat{y}(x; \Theta)$ and $z = \hat{z}(x; \Theta)$;
2. For each $j = 1, \dots, D$, calculate

$$\begin{aligned} x_j &= x + b(x, y, z)\Delta + \sigma(x, y, z)w_j, \\ y_j &= \hat{y}(x_j; \Theta); \end{aligned}$$

3. For $k = 1, \dots, m$, define Y^k as

$$Y^k \equiv \begin{bmatrix} y_1^k + h^k(x, y, z)\Delta \\ y_2^k + h^k(x, y, z)\Delta \\ \vdots \\ y_D^k + h^k(x, y, z)\Delta \end{bmatrix},$$

a D -dimensional vector, where y_j^k denotes the k th element of y_j , and $h^k(\cdot)$ the k th element of $h(\cdot)$. Then compute

$$\begin{bmatrix} \hat{y}^k \\ (\hat{z}^k)^T \end{bmatrix} = (K^T K)^{-1} K^T Y^k$$

for $k = 1, \dots, m$. The k th OLS result is based on the system of equations implied by the

⁸If your K has $1 + D$ rows, ensure that w_0 is included as well, otherwise adjust accordingly.

BSDE for Y_t^k :

$$y_j^k = \hat{y}^k - h^k(x, y, z)\Delta + \hat{z}^k w_j, \quad j = 1, \dots, D.$$

Let \hat{z} denote

$$\hat{z} \equiv \begin{bmatrix} \hat{z}^1 \\ \hat{z}^2 \\ \vdots \\ \hat{z}^m \end{bmatrix}.$$

The loss function is then defined as

$$\text{Loss}(\Theta; \{x_i, w_{i,j}\}_{i=1, \dots, M; j=1, \dots, D}) = \frac{1}{M} \sum_{i=1}^M \left(\|y_i - \hat{y}_i\|^2 + \|z_i - \hat{z}_i\|^2 \right),$$

where $\|\cdot\|$ denotes the squared norm.

2.2.2 Forward Euler Scheme

The forward Euler scheme relies on the property of the Markov solution $y(\cdot)$ and $z(\cdot)$ that, for any initial state $X_0 = x$ and any realized path of Brownian shocks $\{W_t\}_{t \geq 0}$, the path $\{X_t, Y_t\}_{t \geq 0}$ generated by the dynamic system

$$\begin{aligned} dX_t &= b(X_t, Y_t, z(X_t)) dt + \sigma(X_t, Y_t, z(X_t)) dW_t, \\ dY_t &= -h(X_t, Y_t, z(X_t)) dt + z(X_t) dW_t, \\ X_0 &= x, \quad Y_0 = y(X_0) \end{aligned}$$

satisfies

$$Y_t = y(X_t), \quad \forall t \geq 0.$$

Without loss of generality, I truncate the infinite horizon at T , discretize the time interval $[0, T]$ as $0 = t_0 < t_1 < t_2 < \dots < t_n = T$, and simulate M sample paths of Brownian motions $\{W_{i,j}\}_{i=0, \dots, n-1; j=1, \dots, M}$ as well as random initial states $\{x_{0,j}\}_{j=1}^M$. For brevity, the path index j is omitted in the following. Set $y_0 = \hat{y}(x_0; \Theta)$, $\Delta_i = t_{i+1} - t_i$, and $w_i = W_{i+1} - W_i$. The following procedure is repeated for $i = 0$ to $n - 1$:

1. Compute $z_i = \hat{z}(x_i; \Theta)$;
2. Calculate x_{i+1} and y_{i+1} via

$$\begin{aligned} x_{i+1} &= x_i + b(x_i, y_i, z_i)\Delta_i + \sigma(x_i, y_i, z_i)w_i, \\ y_{i+1} &= y_i - h(x_i, y_i, z_i)\Delta_i + z_i w_i; \end{aligned}$$

3. Compute $\tilde{y}_{i+1} = \hat{y}(x_{i+1}; \Theta)$.

The simulated sample paths yield the following loss function:

$$\text{Loss}(\Theta; \{x_{0,j}, W_{i,j}\}_{i=0, \dots, n-1; j=1, \dots, M}) = \frac{1}{Mn} \sum_{j=1}^M \sum_{i=1}^n \|y_{i,j} - \tilde{y}_{i,j}\|^2,$$

where $\|\cdot\|$ denotes the squared norm.

2.3 Connections with Applied Mathematics

The earliest papers to apply deep learning to solve high-dimensional BSDEs are [E, Han and Jentzen \(2017\)](#) and [Han, Jentzen and E \(2018\)](#), which simulate backward SDEs in a forward manner and utilize deep learning to search for solutions that satisfy the terminal conditions of BSDEs. Their work has been highly influential in the applied mathematics literature. Related papers that also employ the forward Euler scheme include [Raissi \(2024\)](#), [Zhang and Cai \(2022\)](#), and [Ji, Peng, Peng and Zhang \(2020\)](#).

The backward Euler scheme has a longer history than the forward scheme ([Bouchard and Touzi, 2004](#); [Zhang, 2004](#)). [Huré, Pham and Warin \(2020\)](#) combine the backward scheme with deep learning. Notably, the regression-based approach to numerically solving BSDEs in [Gobet, Lemor and Warin \(2005\)](#) originates from a derivative pricing paper, [Longstaff and Schwartz \(2001\)](#).

In applied mathematics, a major motivation for solving high-dimensional BSDEs is to find the solution to the corresponding PDEs. In economics, however, it is more natural to formulate dynamic models as FBSDE systems. Reformulating an FBSDE system as PDEs is only necessary if solving the PDEs numerically is more advantageous.

In the mathematical literature, the majority of studies and examples focus on finite-horizon and time-dependent FBSDEs. In contrast, most economic models are formulated with infinite horizons and are typically time-independent (time-homogeneous). In this context, terminal conditions for finite-horizon FBSDEs are replaced by the fixed point or Markov property in infinite-horizon, time-homogeneous FBSDEs—a translation which constitutes my primary contribution

3 Implementation: Multiple-Country Macro-Finance

In this section, I consider a multiple-country macro-finance model to demonstrate the implementation of the deep learning-based probabilistic approach and compare its performance with the numerical approach based on the analytic (PDE) formulation. Each country in the model is populated by two groups of agents: experts and savers. All agents have logarithmic preferences with a time discount factor ρ , which leads to two simple optimality conditions. First, the optimal consumption flow to wealth ratio is ρ . Second, an asset’s expected excess return under investors’ optimal portfolio choices equals the covariance between the investor’s portfolio return and the asset’s return.

Only experts in a country can hold the physical capital located in that country, and physical capital cannot move across borders. Experts use physical capital to produce final goods and transform output into new physical capital through capital production. The final (consumption) goods produced in all countries are homogeneous and serve as the numéraire. The critical financial friction is that experts cannot issue outside equity. The international risk-free bond market and the international trade market are both frictionless. All countries are identical in terms of technology and agents’ preferences. This symmetric setting allows for an analytic

solution under symmetric states across different countries, which helps assess the performance of the algorithm.

3.1 Model and Equilibrium

Since the international bond and trade markets are frictionless, it is sufficient to consider a representative **saver** for the world economy. Given logarithmic preferences, the saver's consumption flow c_t equals the time discount factor ρ times her wealth level at time t . Hence, the law of motion for the saver's wealth N_t^s is

$$\frac{dN_t^s}{N_t^s} = (r_t - \rho) dt,$$

where r_t is the risk-free rate.

Without loss of generality, I consider the optimal decisions of a representative expert in each country. Experts across different countries are equipped with the same linear production technology, $y_t = ak_t$, where k_t is the efficiency units of **physical capital** managed by an expert. The capital production technology allows an expert to convert ιk_t units of final goods into $\Phi(\iota)k_t$ units of physical capital, and vice versa, where $\Phi(\iota) = \frac{1}{\psi} \ln(\psi\iota + 1)$. Suppose the market price of physical capital in country i is q_t^i , i.e., the expert can exchange 1 unit of final goods for $1/q_t^i$ units of physical capital in the market. Since the opportunity cost of capital production is $1/q_t^i$ (in units of physical capital), the expert's optimal investment rate, ι_t^i , satisfies

$$\iota_t^i = \arg \max_{\iota} \left\{ \Phi(\iota) - \frac{\iota}{q_t^i} \right\} = \frac{q_t^i - 1}{\psi}.$$

Given the depreciation rate δ , the efficiency units of physical capital held by the expert evolve according to

$$\frac{dk_t^i}{k_t^i} = (\Phi(\iota_t^i) - \delta) dt + \sigma dW_t^i, \quad i = 1, \dots, J,$$

where $\{W_t^1, \dots, W_t^J\}_{t \geq 0}$ is a standard J -dimensional Brownian motion. Note that W_t^j and W_t^i are independent for any $i \neq j$.

Since physical capital is durable, its price q_t^i should be forward-looking; that is, when an expert decides to hold a unit of physical capital, she must take into account its future dividend, capital gain, and related risks. Therefore, the **price of physical capital** in country i follows a backward SDE:

$$\frac{dq_t^i}{q_t^i} = \mu_t^{q,i} dt + \sum_{j=1}^J \sigma_t^{q,i,j} dW_t^j,$$

where foreign shocks influence domestic asset prices via the international bond market and trade. I will derive the drift term $\mu_t^{q,i}$ based on experts' optimality conditions in a subsection dedicated to BSDEs.

Let N_t^i denote the wealth of the representative **expert** in country i . The law of motion for

N_t^i is

$$\frac{dN_t^i}{N_t^i} = (r_t + \varphi_t^i (R_t^i - r_t) - \rho) dt + \sum_{j=1}^J \underbrace{\varphi_t^i (\mathbf{1}\{j=i\} \sigma + \sigma_t^{q,i,j})}_{\equiv \sigma_t^{W,i,j}} dW_t^j,$$

$$\text{where } R_t^i \equiv \frac{a - \iota_t^i}{q_t^i} + \Phi(\iota_t^i) - \delta + \mu_t^{q,i} + \sigma \sigma_t^{q,i,i},$$

where $R_t^i - r_t$ is the expected excess return, and ρ is the consumption flow to wealth ratio. φ_t^i is the expert's asset-to-equity ratio, which satisfies the Euler equation to be specified in the subsection on BSDEs.

Given the aggregate stock of physical assets K_t^i and the aggregate consumption c_t^i , the **market clearing** condition for final goods is

$$a \sum_{j=1}^J K_t^j = \sum_{j=1}^J c_t^j + \iota_t^j K_t^j.$$

The market clearing of physical capital in each country implies

$$\varphi_t^i N_t^i = q_t^i K_t^i.$$

I will focus on the **Markov equilibrium** with $2J - 1$ endogenous state variables: $\eta_t^1, \dots, \eta_t^J$ and $\zeta_t^1, \dots, \zeta_t^{J-1}$, where η_t^i is the ratio of the expert's net worth to the value of physical assets in country i , i.e.,

$$\eta_t^i \equiv \frac{N_t^i}{q_t^i K_t^i},$$

and

$$\zeta_t^j \equiv \frac{q_t^j K_t^j}{\sum_{h=1}^J q_t^h K_t^h}$$

is the proportion of country j 's physical asset value in the world economy. The market clearing condition for physical capital implies

$$\varphi_t^i = \frac{1}{\eta_t^i}.$$

Under **symmetric states**, i.e., $\eta_t^1 = \dots = \eta_t^J$ and $\zeta_t^j = 1/J$, $j = 1, \dots, J - 1$, the final-good market clearing condition yields an analytic solution for q_t^j . Since aggregate consumption is $\rho \sum_{i=1}^J q_t^i K_t^i$, the final-good market clearing condition can be rewritten as

$$\rho = \sum_{j=1}^J \frac{a - \iota_t^j}{q_t^j} \frac{q_t^j K_t^j}{\sum_{i=1}^J q_t^i K_t^i} = \frac{a - \iota_t^J}{q_t^J} \left(1 - \sum_{i=1}^{J-1} \zeta_t^i \right) + \sum_{i=1}^{J-1} \frac{a - \iota_t^i}{q_t^i} \zeta_t^i. \quad (19)$$

Plugging in $\iota_t^i = (q_t^{i-1})/\psi$, under symmetric states,

$$q_t^j = \frac{a\psi + 1}{\rho\psi + 1}, \quad j = 1, \dots, J,$$

since $q_t^i = q_t^j$ for all $i \neq j$. The analytic result helps gauge the performance of the probabilistic solution.

3.2 Backward SDEs

Recall the convenient property of logarithmic preferences: under the expert's optimal portfolio choice, the expected excess return of an asset equals the covariance between its return and the expert's portfolio return. Hence, the Euler equations for experts—i.e., the asset pricing equations—are

$$R_t^i - r_t = \varphi_t^i \sum_{j=1}^J \left(\mathbf{1}\{j=i\} \sigma + \sigma_t^{q,i,j} \right)^2, \quad i = 1, \dots, J,$$

which yield the expression for $\mu_t^{q,i}$ and also the BSDEs with respect to q_t^i , that is,

$$\begin{aligned} dq_t^i &= -h^i \left(\eta_t^i, q_t^i, \{\sigma_t^{q,i,j}\}_{j=1}^J \right) dt + \sum_{j=1}^J q^j \sigma_t^{q,i,j} dW_t^j, \\ h^i(\cdot) &\equiv \frac{a\psi + 1}{\psi} + \frac{q_t^i}{\psi} \ln(q_t^i) - q_t^i \left(\frac{1}{\psi} + \delta \right) + \sigma q_t^i \sigma_t^{q,i,i} - \frac{q_t^i}{\eta_t^i} \sum_{j=1}^J \left(\mathbf{1}\{j=i\} \sigma + \sigma_t^{q,i,j} \right)^2 - q_t^i r_t. \end{aligned} \quad (20)$$

Note that the risk-free rate is also part of the solution. Although there are no BSDEs specifically for $\{r_t\}_{t \geq 0}$, the final-good market clearing condition determines the risk-free rate.

3.3 Forward SDEs

Given the optimal portfolio and capital production decisions, the law of motion for the physical asset value in country i is

$$\frac{d(q_t^i K_t^i)}{q_t^i K_t^i} = \underbrace{\left(-\frac{a\psi + 1}{\psi q_t^i} + \frac{1}{\psi} + \frac{1}{\eta_t^i} \sum_{j=1}^J (\mathbf{1}\{j=i\} \sigma + \sigma_t^{q,i,j})^2 + r_t \right)}_{\equiv \mu_t^{qK,i}} dt + \sum_{j=1}^J \underbrace{(\mathbf{1}\{j=i\} \sigma + \sigma_t^{q,i,j})}_{\equiv \sigma_t^{qK,i,j}} dW_t^j.$$

To apply Itô's formula, I derive the law of motion for $\eta_t^i = \frac{W_t^i}{q_t^i K_t^i}$:

$$\begin{aligned} d\eta_t^i &= b^i \left(\eta_t^i, q_t^i, \{\sigma_t^{q,i,j}\}_{j=1}^J \right) dt + \sum_{j=1}^J (1 - \eta_t^i) (\mathbf{1}\{j=i\} \sigma + \sigma_t^{q,i,j}) dW_t^j, \\ b^i(\cdot) &\equiv \left(\frac{a\psi + 1}{\psi q_t^i} - \frac{1}{\psi} - \rho \right) \eta_t^i + \left(\frac{1}{\eta_t^i} - 1 \right)^2 \eta_t^i \sum_{j=1}^J (\mathbf{1}\{j=i\} \sigma + \sigma_t^{q,i,j})^2. \end{aligned} \quad (21)$$

A key feature of the dynamics of η_t^i is that the solutions to the BSDEs (20) enter both the drift and volatility terms, which is a crucial characteristic of many macro-finance models.

Next, I consider the law of motion for ζ_t^i , $i = 1, \dots, J-1$, to which I also apply Itô's formula:

$$\begin{aligned} d\zeta_t^i &= \zeta_t^i \mu_t^{\zeta_t^i} dt + \sum_{l=1}^J \zeta_t^i \left(\mathbf{1}\{l=i\} \sigma + \sigma_t^{q,i,l} - \sigma_t^{H,l} \right) dW_t^l, \\ \mu_t^{\zeta_t^i} &\equiv \mu_t^{qK,i} - \sum_{j=1}^{J-1} \zeta_t^j \mu_t^{qK,j} - \left(1 - \sum_{j=1}^{J-1} \zeta_t^j \right) \mu_t^{qK,J} - \sum_{l=1}^J \sigma_t^{H,l} (\mathbf{1}\{l=i\} \sigma + \sigma_t^{q,i,l} - \sigma_t^{H,l}), \\ \sigma_t^{H,l} &\equiv \sum_{k=1}^{J-1} \zeta_t^k (\mathbf{1}\{l=k\} \sigma + \sigma_t^{q,k,l}) + \left(1 - \sum_{k=1}^{J-1} \zeta_t^k \right) (\mathbf{1}\{l=J\} \sigma + \sigma_t^{q,J,l}). \end{aligned} \quad (22)$$

Again, the dynamics of ζ_t^i depend on the solutions to the BSDEs (20), specifically $(q_t^i, \sigma_t^{q,i,j})$.

3.4 Markov Equilibrium and Neural Network Approximation

The equilibrium of the model is characterized by the forward SDEs (21) and (22), which govern the laws of motion for the state variables; the backward SDE (20), which determines the dynamics of forward-looking variables; and the market clearing condition for final goods (19). When following the numerical schemes outlined in Section 2, it is essential to ensure that the market clearing condition is always satisfied. Thanks to the convenience of logarithmic preferences, the optimal consumption policy is absorbed into the forward SDE (21), while the optimal portfolio choice is expressed through the backward SDE (20).

The solution to the Markov equilibrium consists of a set of functions of the state variables,

$$\begin{aligned} &q^i(\Omega_t), \quad r(\Omega_t), \quad \sigma^{q,i,j}(\Omega_t), \quad i = 1, \dots, J, \quad j = 1, \dots, J, \\ &\text{where } \Omega_t \equiv \left[\eta_t^1, \dots, \eta_t^J, \zeta_t^1, \dots, \zeta_t^{J-1} \right]. \end{aligned}$$

While each $q^i(\cdot)$ must satisfy its respective BSDE, the collection $\{q^i(\cdot)\}_{i=1}^J$ must also jointly satisfy the market clearing condition at all times. The required degree of freedom is provided by the risk-free rate function $r(\cdot)$, which adjusts the consumption-saving trade-off.

In constructing neural network approximations for $q^i(\cdot)$, I leverage the expressive power of neural networks to embed the market clearing condition (19) directly into the architecture, ensuring by construction that the approximated functions $q^i(\cdot; \Theta)$ satisfy the condition exactly. This is accomplished as follows. Substituting the optimal consumption $\iota_t^i = (q_t^i - 1)/\psi$ into the market clearing condition yields

$$\begin{aligned} \rho &= \sum_{j=1}^{J-1} \xi^j \zeta_t^j + \xi^J \left(1 - \sum_{j=1}^{J-1} \zeta_t^j \right), \\ \text{where } \xi^i &\equiv \frac{a\psi - q_t^i + 1}{\psi q_t^i} = \frac{a\psi + 1}{\psi q^i} - \frac{1}{\psi}. \end{aligned}$$

Based on this transformation, let $\tilde{\xi}^j(\Omega_t; \Theta)$ denote the output of a generic neural network

for each $j = 1, \dots, J$. Define

$$\Xi \equiv \sum_{j=1}^{J-1} \tilde{\xi}^j \zeta_t^j + \tilde{\xi}^J \left(1 - \sum_{j=1}^{J-1} \zeta_t^j \right).$$

Then, re-scale the intermediate outputs and generate q^j via

$$\xi^j = \frac{\rho}{\Xi} \tilde{\xi}^j, \quad q^j = \frac{a\psi + 1}{\psi \xi^j + 1}, \quad j = 1, \dots, J.$$

It is straightforward to see that, by this construction, $q^j(\Omega_t; \Theta)$ for $j = 1, \dots, J$ automatically satisfy the market clearing condition at any point in the state space, for any realization of the neural network. [Azinovic and Zemlicka \(2024\)](#) also take advantage of flexibility of the deep neural network architecture to embed market clearing conditions into their solutions.

3.5 Numerical Example Solved with Forward Euler Scheme

As a numerical experiment, I implement the forward Euler scheme illustrated in Section 2.2 to solve for the equilibrium of a 5-country version with parameter values: $a = 0.1, \delta = 0.05, \sigma = 0.023, \psi = 5$, and $\rho = 0.03$. In the 5-country model, there are 9 endogenous state variables: $\eta_t^i, i = 1, \dots, 5, \zeta_t^i, i = 1, \dots, 4$, and 31 functions:

$$q^i(\cdot), i = 1, \dots, 5; r(\cdot); \\ \sigma^{q,i,j}(\cdot), i = 1, \dots, 5, j = 1, \dots, 5.$$

Therefore, the neural network I construct involves nine inputs and 31 outputs. The network has three hidden layers, each with 256 inputs and 256 outputs. The activation function is $f(x) = \sin(x)$. I utilize the TensorFlow library in Python to solve the deep learning problem.

The initial values of $\eta_0^i, i = 1, \dots, 5$ are randomly selected from $[0.2 \ 0.8]$ according to a uniform distribution; those of $\zeta_0^i, i = 1, \dots, 4$ are chosen from $[0.03 \ 0.26]$ in the same manner. $\Delta_i = 0.001, T = 0.2$, and the number of sample paths $M = 20,000$. Notice that if I use the finite difference method and pick only four grid points along each dimension, the total number of grid points would be $4^9 = 262,144$.

Test Sample. I generate a separate 500 test sample paths to assess the performance of the learning outcome. These 500 sample paths are NOT used to train the network in the search for optimal Θ . To plug in the test sample, the value of the loss function indicates the average discrepancy between q_i and \tilde{q}_i is around 5×10^{-5} in absolute value.⁹

Symmetric Cases. Note that all countries are identical regarding their exogenous technologies and agents' preferences. Therefore, when the state variables across different countries are the

⁹ q_i is the value updated according to the FBSDE, and \tilde{q}_i is the corresponding network output given the same state variables.

Table 1: Symmetric State

This table presents the equilibrium solution under five symmetric states.

$\eta_t^i = 0.3, \zeta_t^j = 0.2$					
	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$
q^i	1.3035	1.3036	1.3039	1.3038	1.3070
$\sigma^{q,i,1}$	0.00093	-0.00025	-0.00027	-0.00027	-0.00013
$\sigma^{q,i,2}$	-0.00017	0.00101	-0.00017	-0.00018	-0.00048
$\sigma^{q,i,3}$	-0.00020	-0.00019	0.00097	-0.00018	-0.00039
$\sigma^{q,i,4}$	-0.00026	-0.00026	-0.00027	0.00086	-0.00007
$\sigma^{q,i,5}$	-0.00014	-0.00015	-0.00015	-0.00013	0.00053
$\eta_t^i = 0.4, \zeta_t^j = 0.2$					
	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$
q^i	1.3045	1.3048	1.3046	1.3045	1.3034
$\sigma^{q,i,1}$	0.00079	-0.00021	-0.00023	-0.00023	-0.00013
$\sigma^{q,i,2}$	-0.00014	0.00086	-0.00014	-0.00015	-0.00045
$\sigma^{q,i,3}$	-0.00017	-0.00016	0.00083	-0.00015	-0.00035
$\sigma^{q,i,4}$	-0.00022	-0.00022	-0.00023	0.00073	-0.00006
$\sigma^{q,i,5}$	-0.00013	-0.00014	-0.00013	-0.00012	0.00053
$\eta_t^i = 0.5, \zeta_t^j = 0.2$					
	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$
q^i	1.3056	1.3059	1.3053	1.3052	1.2998
$\sigma^{q,i,1}$	0.00066	-0.00018	-0.00019	-0.00018	-0.00014
$\sigma^{q,i,2}$	-0.00011	0.00072	-0.00011	-0.00012	-0.00042
$\sigma^{q,i,3}$	-0.00014	-0.00013	0.00069	-0.00012	-0.00031
$\sigma^{q,i,4}$	-0.00018	-0.00019	-0.00019	0.00060	-0.00005
$\sigma^{q,i,5}$	-0.00011	-0.00012	-0.00012	-0.00010	0.00052
$\eta_t^i = 0.6, \zeta_t^j = 0.2$					
	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$
q^i	1.3067	1.3070	1.3060	1.3058	1.2962
$\sigma^{q,i,1}$	0.00053	-0.00014	-0.00015	-0.00014	-0.00014
$\sigma^{q,i,2}$	-0.00008	0.00058	-0.00008	-0.00008	-0.00039
$\sigma^{q,i,3}$	-0.00011	-0.00011	0.00055	-0.00009	-0.00027
$\sigma^{q,i,4}$	-0.00015	-0.00015	-0.00015	0.00047	-0.00005
$\sigma^{q,i,5}$	-0.00009	-0.00010	-0.00010	-0.00008	0.00051
$\eta_t^i = 0.7, \zeta_t^j = 0.2$					
	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$
q^i	1.3078	1.3081	1.3067	1.3065	1.2927
$\sigma^{q,i,1}$	0.00039	-0.00010	-0.00010	-0.00010	-0.00015
$\sigma^{q,i,2}$	-0.00005	0.00044	-0.00005	-0.00005	-0.00035
$\sigma^{q,i,3}$	-0.00008	-0.00008	0.00041	-0.00006	-0.00023
$\sigma^{q,i,4}$	-0.00011	-0.00012	-0.00012	0.00035	-0.00004
$\sigma^{q,i,5}$	-0.00007	-0.00008	-0.00008	-0.00006	0.00051

same, i.e.,

$$\begin{aligned}\eta_t^1 &= \eta_t^2 = \eta_t^3 = \eta_t^4 = \eta_t^5, \\ \zeta_t^1 &= \zeta_t^2 = \zeta_t^3 = \zeta_t^4 = 0.2,\end{aligned}$$

the endogenous variables, such as asset prices and their volatility terms, are supposed to be identical. In particular, the asset price under any symmetric cases is 1.3043 across all countries, given the chosen parameter values. At symmetric states, positive domestic shocks increase the net worth of domestic experts and also the price of domestic assets. The market clearing condition implies that the price of foreign assets must decline. Hence, at symmetric states

$$\begin{aligned}\sigma_t^{q,i,j} &> 0 \text{ if } i = j; \\ \sigma_t^{q,i,j} &< 0 \text{ if } i \neq j.\end{aligned}$$

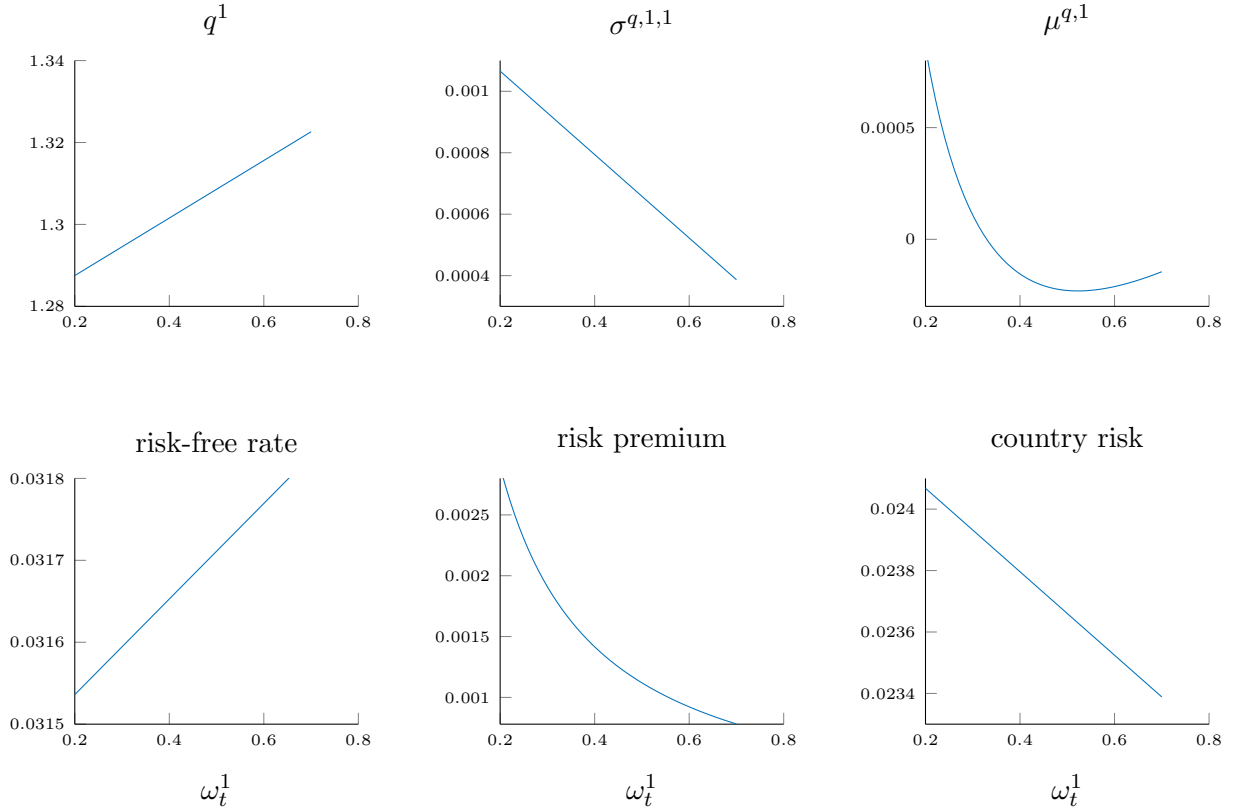


Figure 3: Dynamics

This figure shows asset price, asset price volatility driven by domestic shocks, the drift of asset price, risk-free rate, risk premium, and country-wide risk in country 1 as functions of its expert wealth share when other state variables stay constant ($\eta_t^i = 0.45, i = 2, \dots, 5, \zeta_t^i = 0.2, i = 1, \dots, 4$).

Table 1 displays the solutions of the FBSDE under five symmetric states: $\eta_t^i = 0.3, \eta_t^i = 0.4, \eta_t^i = 0.5, \eta_t^i = 0.6$, and $\eta_t^i = 0.7$. Note that the training process is not guided by any information about the symmetric property of the model. The solution appears fairly reasonable. The accuracy declines when there are fewer data samples to learn during the training process.

The approximation of the volatility terms is challenging, as the symmetric solution is a

knife-edge case. Table 1 show that the signs of the domestic risk exposure, $\sigma_t^{q,i,i}, i = 1, \dots, 5$ are positive. The signs of all foreign shock exposure terms $\sigma_t^{q,i,j}, i \neq j$, are negative, and their magnitudes are close.

Dynamics. Figure 3 displays how endogenous variables of country 1 respond to the variation in η_t^1 its expert's wealth share while other state variables stay unchanged ($\eta_t^i = 0.45, i = 2, \dots, 5, \zeta_t^i = 0.2, i = 1, \dots, 4$). When the expert's net worth increases, the domestic asset price rises, and the endogenous risk due to financial amplification declines. In the economy of five countries, the change in the financial condition of one country has limited influence on the world's risk-free rate. When the country-1 expert's net worth is low, she demands a high risk premium, which comes with a low relatively low risk-free rate. Due to the financial friction, the aggregate risk in country 1

$$(\sigma + \sigma^{q,1,1})^2 + (\sigma^{q,1,2})^2 + (\sigma^{q,1,3})^2 + (\sigma^{q,1,4})^2 + (\sigma^{q,1,5})^2$$

is decreasing in the wealth share of its productive agent.

3.6 Comparison with the PDE approach

Section 1.4 highlights that solving PDEs becomes exponentially more costly as the model's dimensionality increases. I will illustrate this point more vividly with a series of numerical exercises based on the multi-country macro-finance model.

To establish the PDEs, it is necessary to derive the volatility terms of asset prices, $\sigma_t^{q,i,j}$ for $i = 1, \dots, J; j = 1, \dots, J$, according to equation (18), which is

$$q_t^i \sigma_t^{q,i,j} = \sum_{h=1}^J \frac{\partial q_t^i}{\partial \eta_t^h} (1 - \eta_t^h) \left(\mathbf{1}\{h = j\} + \sigma_t^{q,h,j} \right) + \sum_{h=1}^{J-1} \frac{\partial q_t^i}{\partial \zeta_t^h} \zeta_t^h \left(\mathbf{1}\{h = j\} + \sigma_t^{q,h,j} - \sigma_t^{H,j} \right), \quad (23)$$

for each i and j in this example. Note that $\sigma_t^{H,j}$ is a linear function of $\sigma_t^{q,i,j}$ for $i = 1, \dots, J$ (see equation 22). Thus, for each Brownian shock to the j th country, there is a system of J linear equations with J unknowns, all of the same form as (23):

$$A^j(q(\cdot), \nabla q(\cdot), \sigma^q(\cdot)) \begin{bmatrix} \sigma_t^{q,1,j} \\ \sigma_t^{q,2,j} \\ \vdots \\ \sigma_t^{q,J,j} \end{bmatrix} = \sigma b^j(q(\cdot), \nabla q(\cdot), \sigma^q(\cdot)), \quad j = 1, \dots, J. \quad (24)$$

Here, $A^j(\cdot)$ is a $J \times J$ matrix and b^j is a J -dimensional vector. Both the matrix and the vector depend on $q(\cdot)$, $\nabla q(\cdot)$, and $\sigma^q(\cdot)$; their exact expressions can be found in the Appendix.

Given $\sigma^{q,i,j}$, I can determine $\mu_t^{q,i}$ from the asset pricing equation, i.e., the Euler equation in Section 3.2. Equation (16) in Section 1.3 suggests that I only need to apply the Itô formula to

establish the PDE with respect to $q^i(\cdot)$:

$$\begin{aligned}
q_t^i \mu_t^{q,i} = & \sum_{h=1}^J \frac{\partial q^i}{\partial \eta^h} b^i(\cdot) + \sum_{h=1}^{J-1} \frac{\partial q^i}{\partial \zeta^h} \zeta_t^h \mu_t^{\zeta,h} + \frac{1}{2} \sum_{h=1}^J \frac{\partial^2 q^i}{\partial (\eta^h)^2} (1 - \eta_t^h)^2 \sum_{j=1}^J \left(\sigma_t^{qK,h,j} \right)^2 \\
& + \frac{1}{2} \sum_{h=1}^{J-1} \frac{\partial^2 q^i}{\partial (\zeta^h)^2} (\zeta_t^h)^2 \sum_{j=1}^J \left(\sigma_t^{qK,h,j} - \sigma_t^{H,j} \right)^2 \\
& + \sum_{h=1}^J \sum_{\substack{l=1 \\ l \neq h}}^J \frac{\partial^2 q^i}{\partial \eta^h \partial \eta^l} (1 - \eta_t^h) (1 - \eta_t^l) \sum_{j=1}^J \sigma_t^{qK,h,j} \sigma_t^{qK,l,j} \\
& + \sum_{h=1}^{J-1} \sum_{\substack{l=1 \\ l \neq h}}^{J-1} \frac{\partial^2 q^i}{\partial \zeta^h \partial \zeta^l} \zeta_t^h \zeta_t^l \sum_{j=1}^J \left(\sigma_t^{qK,h,j} - \sigma_t^{H,j} \right) \left(\sigma_t^{qK,l,j} - \sigma_t^{H,j} \right) \\
& + \sum_{h=1}^J \sum_{l=1}^{J-1} \frac{\partial^2 q^i}{\partial \eta^h \partial \zeta^l} (1 - \eta_t^h) \zeta_t^l \sum_{j=1}^J \sigma_t^{qK,h,j} \left(\sigma_t^{qK,l,j} - \sigma_t^{H,j} \right).
\end{aligned}$$

It is straightforward to apply deep learning to solve for $q^i(\cdot)$ using this PDE formulation. Similar to the approach taken under the BSDE formulation, I employ a neural network to approximate $q^i(\cdot, \cdot)$, denoted by $\hat{q}^i(\cdot, \cdot; \Theta)$. The loss function is defined as the mean squared error of the PDE residual, measuring the difference between the left- and right-hand sides of the equation. An optimization algorithm from a machine learning library is used to choose parameters Θ that minimize this loss.

As discussed in Section 1.4, the primary computational cost of the PDE approach arises from calculating Hessian matrices. To illustrate this, I apply deep learning to solve for the Markov equilibrium of the multi-country macro-finance model using the PDE formulation and compare its computational speed to the backward Euler scheme described in Section 2.2. To isolate the effect of Hessian matrix calculations, I use equation (24) to recover $\sigma_t^{q,i,j}$ even when implementing the backward Euler scheme, although this step is not strictly necessary.

For a model with J countries, there are J functions to solve, i.e., $q^j(\cdot)$, $j = 1, \dots, J$, and the number of state variables is $2J - 1$. Therefore, the number of second-order partial derivatives (components of the Hessian matrices) is approximately $0.5J(2J - 1)^2$. When using the backward Euler scheme, I set $D = J + 1$, the minimum number of paths required to identify \hat{q}^i and $\hat{\sigma}^{q,i,j}$. Accordingly, the number of function evaluations is $J(J + 1)$. The computational complexity ratio between the PDE and BSDE formulations is thus on the order of $2J$, growing with the problem's dimensionality.

To quantify these differences, I solved the model using both the PDE and BSDE formulations for $J = 5, \dots, 15$. Table 2 presents the details of these numerical experiments. On a CPU, the computation time per neural network update step differs by orders of magnitude between the two formulations. For $J = 15$, the algorithm based on the BSDE formulation is more than 400 times faster than its PDE-based counterpart. Using Nvidia's high-performance A800 GPU, the difference in computation time between the two approaches is more pronounced as J increases, although the gap is somewhat narrower than on a CPU. However, a significant hardware limitation arises with the PDE formulation: when the model size reaches $J = 15$, the

A800 GPU can no longer process the numerical scheme because the memory required to store the Hessian matrices exceeds the GPU’s 80 GB capacity. For example, calculating the PDE approximation error requires the GPU to store 15 Hessian matrices of size 29×29 for each sampled point in the state space, drastically increasing memory demand.

Table 2: PDE vs. BSDE

This table presents the time (in seconds) required per round of neural network parameter updating (i.e., per epoch) under four different specifications: (i) whether the PDE or BSDE formulation is used, and (ii) whether an A800 GPU is employed. The CPU is an AMD EPYC 7763, with 16 cores and 160 gigabytes of memory allocated. The number of sample points for network training is $J \times 10,000$, and the batch size is 512.

	J	5	6	7	8	9	10	11	12	13	14	15
CPU	PDE	129	438	786	1856	2494	4601	5220	8767	10928	13329	14419
	BSDE	5.54	7.61	10.0	13.5	14.5	16.6	19.6	24.0	24.5	31.5	33.0
GPU	PDE	6.7	9.00	10.0	16.4	27.2	40.5	58.1	80.6	110	149	infeasible
	BSDE	4.84	6.08	7.51	10.0	14.5	16.5	19.8	24.5	25.0	32.0	33.1

4 Probabilistic Formulation of Dynamic Optimization

Dynamic optimization for a single agent is a fundamental building block of any dynamic economic model. This section focuses exclusively on the probabilistic formulation of standard dynamic optimization problems, as there is too much content to cover within a single paper. I begin with the BSDE formulation of recursive utility, given a consumption policy, and then proceed to the special case of homothetic preferences with a linear budget constraint, which frequently arises in macroeconomic and asset pricing models. The core of the section is devoted to the probabilistic formulation of general dynamic optimization problems, in which the state variables are backward-looking and evolve according to diffusion processes. Finally, I discuss potential applications of the probabilistic approach to models with impulse or singular controls, as well as to problems with forward-looking constraints.

4.1 Recursive Utility

Given a consumption process $\{c_t\}_{t \geq 0}$, [Duffie and Epstein \(1992\)](#) outline the conditions under which an agent’s utility at time t can be defined recursively as

$$V_t = E_t \left[\int_t^\infty f(c_s, V_s) ds \right],$$

where f is an aggregator, and $E_t[\cdot]$ denotes the expectation conditional on information available up to time t .¹⁰ If $f(c, v) = u(c) - \beta v$, then V_t is the standard expected discounted utility, i.e.,

$$V_t = E_t \left[\int_t^\infty e^{-\beta(s-t)} u(c_s) ds \right].$$

¹⁰The results for the infinite horizon case are provided in Appendix C of [Duffie and Epstein \(1992\)](#) in collaboration with Costis Skiadas.

To derive the BSDE formulation of recursive utility, first construct a martingale as

$$\int_0^t f(c_s, V_s) ds + V_t = E_t \left[\int_0^\infty f(c_s, V_s) ds \right].$$

The martingale representation theorem implies that there exists a progressively measurable stochastic process $(Z_t)_{t \geq 0}$ such that

$$\int_0^t f(c_s, V_s) ds + V_t = V_0 + \int_0^t Z_s dW_s \quad \forall t > 0.$$

The equation above implies that, for any $T > t$,

$$V_t = V_T + \int_t^T f(c_s, V_s) ds - \int_t^T Z_s dW_s,$$

which is the integral form of the BSDE that defines the recursive utility. This form can be interpreted as stating that the current recursive utility V_t depends on the future utility V_T path by path. Taking V_T as a given random variable whose value is realized at T , the above equation yields a finite-horizon BSDE with solution $(V_t, Z_t)_{0 \leq t < T}$.¹¹

Suppose, in a Markov equilibrium, the consumption policy $c(\cdot)$ is a function of a state variable X_t , which evolves as

$$\begin{aligned} dX_t &= \mu(X_t) dt + \sigma(X_t) dW_t \\ X_0 &= x. \end{aligned} \tag{25}$$

Along with the SDE (25), the following BSDE,

$$dV_t = -f(c(X_t), V_t) dt + Z_t dW_t,$$

gives rise to the fixed point mapping $V_t = V(X_t) \quad \forall t \geq 0$.

4.2 Homothetic Preference with a Linear Budget Constraint

In models with portfolio choices, it is often assumed that investors have homothetic preferences with linear budget constraints. Here, I consider the general case in which an investor chooses portfolio weights α_t over time to maximize her recursive utility:

$$\begin{aligned} V_t &= E_t \left[\int_t^\infty f(c_u, V_u) du \right], \\ f(c, V) &= \frac{1}{1-\psi} \left\{ \frac{\rho c^{1-\psi}}{[(1-\gamma)V]^{(\gamma-\psi)/(1-\gamma)}} - \rho(1-\gamma)V \right\}, \end{aligned}$$

¹¹More generally, the aggregator f could include Z_t as an argument to accommodate ambiguity aversion (Chen and Epstein, 2002).

while facing a linear budget constraint

$$dN_t = (\mu^N(X_t, \alpha_t) N_t - c_t) dt + \sigma^N(X_t, \alpha_t) N_t dW_t.$$

The function $\mu^N(X_t, \alpha_t)$ denotes the instantaneous return of the investor's overall portfolio, and $\sigma^N(X_t, \alpha_t)$ captures the portfolio's risk exposure. The state variable X_t follows the law of motion (25). Both the state and control variables, X_t and α_t respectively, as well as the Brownian shocks W_t , are allowed to be multi-dimensional. If the investor is the marginal investor, then the stochastic discount factor used in Section 1, m_t , can be derived as

$$m_t \equiv \exp\left(\int_0^t f_v(c_s, V_s) ds\right) f_c(c_t, V_t),$$

given the optimal consumption and investment policy and the associated recursive utility.

The conventional approach to solving the investor's dynamic optimization problem is to conjecture that V_t takes the following functional form:

$$V_t = \frac{(\xi_t N_t)^{1-\gamma}}{1-\gamma},$$

and to solve for ξ_t as a function of the state variables, denoted as $\xi(\cdot)$. Here, I show that ξ_t follows a BSDE:

$$d\xi_t = \xi_t \left(\mu_t^\xi dt + \sigma_t^\xi dW_t \right),$$

and derive the drift term of ξ_t . Recall that V_t follows the BSDE

$$dV_t = -f(c_t, V_t) dt + Z_t dW_t,$$

and by applying Itô's formula to V_t as a function of ξ_t and N_t , we have

$$\begin{aligned} \frac{dV_t}{\xi_t^{1-\gamma} N_t^{1-\gamma}} &= (\mu^N(X_t, \alpha_t) - \hat{c}_t) dt + \sigma^N(X_t, \alpha_t) dW_t \\ &\quad + \mu_t^\xi dt + \sigma_t^\xi dW_t \\ &\quad + \left(-\frac{1}{2} \gamma \sigma_t^\xi (\sigma_t^\xi)^T + (1-\gamma) \sigma^N(X_t, \alpha_t) (\sigma_t^\xi)^T \right. \\ &\quad \left. - \frac{1}{2} \gamma \sigma^N(X_t, \alpha_t) \sigma^N(X_t, \alpha_t)^T \right) dt, \end{aligned}$$

where $\hat{c}_t \equiv \frac{c_t}{N_t}$. The drift terms of V_t must agree whether derived via the BSDE or via Itô's formula. Thus, plugging in the functional form of $f(c, V)$ yields the drift term of ξ_t :

$$\begin{aligned} -\mu_t^\xi &= \frac{1}{1-\psi} \left\{ \rho \left(\frac{\hat{c}_t}{\xi_t} \right)^{1-\psi} - \rho \right\} + \mu^N(X_t, \alpha_t) - \hat{c}_t \\ &\quad - \frac{1}{2} \gamma \sigma_t^\xi (\sigma_t^\xi)^T + (1-\gamma) \sigma^N(X_t, \alpha_t) (\sigma_t^\xi)^T \\ &\quad - \frac{1}{2} \gamma \sigma^N(X_t, \alpha_t) \sigma^N(X_t, \alpha_t)^T, \end{aligned} \tag{26}$$

which matches the HJB equation when \hat{c}_t and α_t are chosen optimally. To see this, apply Itô's formula to $\xi(X_t)$:

$$\begin{aligned}\sigma_t^\xi &= (\nabla_x \xi(X_t))^T \sigma(X_t), \\ \mu_t^\xi &= (\nabla_x \xi(X_t))^T \mu(X_t) + \frac{1}{2} \text{Tr} \left\{ \sigma(X_t)^T H_\xi(X_t) \sigma(X_t) \right\},\end{aligned}$$

where $H_\xi(\cdot)$ denotes the Hessian matrix of $\xi(\cdot)$. The HJB equation is obtained by substituting σ_t^ξ and μ_t^ξ in equation (26) with their expressions just given above. Equation (26) also yields the optimality conditions with respect to \hat{c}_t and α_t , that is,

$$\begin{aligned}\hat{c}_t &= \rho^{\frac{1}{\psi}} \xi_t^{\frac{\psi-1}{\psi}}, \\ \nabla_\alpha \mu^N(X_t, \alpha_t) &= \gamma \nabla_\alpha \sigma^N(X_t, \alpha_t) \sigma^N(X_t, \alpha_t)^T - (1 - \gamma) \nabla_\alpha \sigma^N(X_t, \alpha_t) \left(\sigma_t^\xi \right)^T,\end{aligned}$$

where ∇_α denotes the partial derivative with respect to the control variable α .

The critical step in the **numerical implementation** is to calculate the drift term μ_t^ξ specified by equation (26), which depends on ξ_t , σ_t^ξ , \hat{c}_t , and α_t . The values of ξ_t and σ_t^ξ are given by approximations of the functions $\xi(X_t)$ and $\sigma^\xi(X_t)$, respectively. The quantities \hat{c}_t and α_t are obtained from their respective optimality conditions above. The fixed point $\xi(\cdot)$ is defined by the time-discretized version of the BSDE with respect to ξ_t :

$$\xi(X_{t+\Delta}) = \xi(X_t) + \xi(X_t) \mu_t^\xi \Delta + \xi(X_t) \sigma^\xi(X_t) (W_{t+\Delta} - W_t),$$

which holds path by path for each realized value of $W_{t+\Delta}$. The main difficulty of solving for $\xi(\cdot)$ using the HJB equation with high-dimensional X_t lies in computing the Hessian matrix of $\xi(\cdot)$, which is unnecessary in the probabilistic approach.

4.3 Bellman's Principle of Optimality and Pontryagin's Maximum Principle

In this subsection, I present the probabilistic representation of dynamic programming and relate it to the stochastic maximum principle, which has traditionally been developed via a probabilistic approach. Consider an agent who maximizes the objective function

$$\max_{\{\alpha_s\}_0^\infty} E_0 \left[\int_0^\infty e^{-\rho s} f(X_s, X_s^0, \alpha_s) ds \right]$$

where ρ is the discount factor, the control variable α_t takes values in a convex subset of a k -dimensional space, and X_t and X_t^0 are the controlled and uncontrolled state variables, respectively. The process X_t^0 , which takes values in an n^0 -dimensional space, is interpreted as an exogenous state variable (e.g., an aggregate state variable). The process $\{X_t\}$ follows the controlled law of motion

$$dX_t = \mu(X_t, X_t^0, \alpha_t) dt + \sigma(X_t, X_t^0, \alpha_t) dW_t + \sigma^0(X_t, X_t^0, \alpha_t) dW_t^0, \quad (27)$$

where W_t and W_t^0 are independent d - and d^0 -dimensional Brownian motions, respectively. The dynamics of X_t^0 are given by

$$dX_t^0 = b(X_t^0) dt + \Sigma(X_t^0) dW_t^0,$$

which is an exogenous process from the agent's perspective. In the case where X_t^0 is an aggregate state variable in the model, W_t can be interpreted as idiosyncratic shocks, and W_t^0 as aggregate shocks.

Let $V(X_t, X_t^0)$ denote

$$V(X_t, X_t^0) = \max_{\{\alpha_s\}_t^\infty} E_t \left[\int_t^\infty e^{-\rho(s-t)} f(X_s, X_s^0, \alpha_s) ds \right]$$

for arbitrary X_t and X_t^0 . **Bellman's principle of optimality** indicates that the function $V(\cdot, \cdot)$ satisfies

$$V(X_t, X_t^0) = \max_{\{\alpha_s\}_t^T} E_t \left[\int_t^T e^{-\rho(s-t)} f(X_s, X_s^0, \hat{\alpha}_s) ds + e^{-\rho(T-t)} V(X_T, X_T^0) \right],$$

for arbitrary $T > t$. Supposing $V(\cdot, \cdot)$ is sufficiently smooth, the recursive formulation of $V(\cdot, \cdot)$ implies the HJB equation:

$$\begin{aligned} \rho V(x, x^0) = & \max_{\alpha} H(x, x^0, \alpha, V_x, 0.5V_{xx}\sigma(x, x^0, \alpha), 0.5V_{xx}\sigma^0(x, x^0, \alpha) + V_{xx^0}\Sigma(x^0)) \\ & + V_{x^0}(x, x^0)b(x^0) + 0.5 \text{Tr} \left\{ \Sigma(x^0)^T V_{x^0 x^0}(x, x^0) \Sigma(x^0) \right\}, \end{aligned}$$

where $H(\cdot)$ is defined by

$$\begin{aligned} H(x, x^0, \alpha, y, z, z^0) \equiv & f(x, x^0, \alpha) + \mu(x, x^0, \alpha)^T y + \text{Tr} \left\{ \sigma(x, x^0, \alpha)^T z \right\} \\ & + \text{Tr} \left\{ \sigma^0(x, x^0, \alpha)^T z^0 \right\}. \end{aligned}$$

The first-order condition with respect to α is

$$0 = \nabla_{\alpha} H(x, x^0, \alpha, V_x, V_{xx}\sigma(x, x^0, \alpha), V_{xx}\sigma^0(x, x^0, \alpha) + V_{xx^0}\Sigma(x^0)), \quad (28)$$

where $\nabla_{\alpha} H$ denotes the partial derivative of $H(\cdot)$ with respect to its third argument. The policy function is $\alpha(x, x^0) = \hat{\alpha}$, where $\hat{\alpha}$ solves the first-order condition above, given the values of $V_x(x, x^0)$, $V_{xx}(x, x^0)$, and $V_{xx^0}(x, x^0)$. The critical step in applying dynamic programming is to solve the HJB equation numerically, which becomes substantially more difficult as the dimensionality of X_t , X_t^0 , or α_t increases.

To formulate the BSDE that characterizes the value function $V(\cdot, \cdot)$, i.e., the **probabilistic formulation of dynamic programming**, I first construct the martingale

$$\int_0^t e^{-\rho s} f(X_s, X_s^0, \hat{\alpha}_s) ds + e^{-\rho t} V_t = E_t \left[\int_0^\infty e^{-\rho s} f(X_s, X_s^0, \hat{\alpha}_s) ds \right],$$

where V_t is defined under the optimal policy $\hat{\alpha}_t = \alpha(X_t, X_t^0)$ by

$$V_t = E_t \left[\int_t^\infty e^{-\rho(s-t)} f(X_s, X_s^0, \hat{\alpha}_s) ds \right].$$

The martingale representation theorem implies the existence of progressively measurable processes Z_t and Z_t^0 such that

$$\int_0^t e^{-\rho s} f(X_s, X_s^0, \hat{\alpha}_s) ds + e^{-\rho t} V_t = V_0 + \int_0^t e^{-\rho s} Z_s dW_s + \int_0^t e^{-\rho s} Z_s^0 dW_s^0,$$

whose differential form is

$$dV_t = - \left(f(X_t, X_t^0, \hat{\alpha}_t) - \rho V_t \right) dt + Z_t dW_t + Z_t^0 dW_t^0.$$

The BSDE above, along with the first-order condition (28) and the laws of motion for X_t and X_t^0 , defines the fixed point $V(\cdot, \cdot)$. That is, starting from any initial state X_0 and X_0^0 with $V_0 = V(X_0, X_0^0)$, any realized path $\{V_t, X_t, X_t^0\}$ driven by $\{W_t, W_t^0\}$ via the FBSDE system satisfies $V_t = V(X_t, X_t^0)$ at any time $t > 0$.

The alternative approach to derive the optimal control $\hat{\alpha}_t$ is the stochastic version of **Pontryagin's maximum principle**. The generalized current-value Hamiltonian is precisely the function $H(\cdot)$. The optimal control $\hat{\alpha}_t$ satisfies

$$\hat{\alpha}_t = \arg \max_{\alpha} H \left(X_t, X_t^0, \alpha, Y_t, Z_t^y, Z_t^{y,0} \right),$$

where $\{Y_t, Z_t^y, Z_t^{y,0}\}$ follows the BSDE

$$dY_t = - \left(\nabla_x H \left(X_t, X_t^0, \hat{\alpha}_t, Y_t, Z_t^y, Z_t^{y,0} \right) - \rho Y_t \right) dt + Z_t^y dW_t + Z_t^{y,0} dW_t^0.$$

Along with the forward SDEs of X_t and X_t^0 , this BSDE defines the fixed point $Y(\cdot, \cdot)$. A critical connection between dynamic programming and the maximum principle is

$$Y(X_t, X_t^0) = \nabla_x V(X_t, X_t^0) \quad \forall t > 0,$$

which means the costate or adjoint variable Y_t captures the impact on the continuation value of a marginal change in the controlled state X_t .¹² The marginal impact of the uncontrolled state variable is not a concern from the perspective of optimal control.

Not surprisingly, dynamic programming and the maximum principle lead to the same optimality condition. To see this, note that Itô's formula implies

$$\begin{aligned} Z_t^y &= Y_x \sigma(X_t, X_t^0, \hat{\alpha}_t) = V_{xx} \sigma(X_t, X_t^0, \hat{\alpha}_t), \\ Z_t^{y,0} &= Y_x \sigma^0(X_t, X_t^0, \hat{\alpha}_t) + Y_{x^0} \Sigma(X_t) = V_{xx} \sigma^0(X_t, X_t^0, \hat{\alpha}_t) + V_{xx^0} \Sigma(X_t). \end{aligned}$$

Plugging in the expressions for Y_t, Z_t^y , and $Z_t^{y,0}$ into the generalized Hamiltonian above, it

¹²See Chapter 5 of [Yong and Zhou \(1999\)](#) for a rigorous analysis of the connection between the dynamic programming approach and the stochastic maximum principle.

is straightforward to observe that the maximum principle gives rise to the same first-order condition as does dynamic programming.

The **computational** advantage of the probabilistic formulation over the analytic (PDE) formulation is that it avoids calculating the first- and second-order derivatives of the value function with respect to the uncontrolled state variable, i.e., V_{x^0} and $V_{x^0x^0}$. In [Huang \(2024\)](#), I exploit this feature in heterogeneous-agent models, in which the dimensionality of the aggregate state variable is high or infinite, while the individual state variable and aggregate shocks are low-dimensional. If the controlled state variable is high-dimensional, it becomes more challenging to numerically solve the corresponding dynamic optimization problem because computing the high-dimensional objects V_x , V_{xx} , and V_{xx^0} is unavoidable. In [Chen and Huang \(2025\)](#), we consider a simpler case in which the control α_t does not affect the volatility terms of the controlled state X_t 's dynamics. Therefore, V_{xx} and V_{xx^0} are omitted from the first-order condition.

Remarks. If the control variable takes values in a nonconvex set, one needs to establish a more general Hamiltonian and trace adjoint variables of the second order. For detailed treatment on this, see Chapter 3 of [Yong and Zhou \(1999\)](#). In economics, we often encounter constraints on states such as the wealth of an agent cannot be negative. Nevertheless, there is still lack of general treatment in the stochastic control literature on state constraints. The solution would be on a case-by-case basis.

4.4 Two Classes of Problems Left Untouched

The law of motion for the controlled state (27) is too standard to cover all dynamic optimization problems encountered in economics. In this subsection, I discuss potential applications of the probabilistic formulation to two types of deviations: impulse and singular controls, and forward-looking constraints.

The law of motion (27) implies that the controlled state X_t has a continuous realized path. However, in some economic models, state variables may be discrete, such as a binary variable. For instance, whether a consumer, firm, or sovereign has bankruptcy records affects the terms of their new borrowings. Besides **impulse controls**, the state X_t may also be driven by a **singular control** process. The term “singularity” means the realized path of the control process has a zero time derivative except at countably many points where the first-order time derivatives do not exist. Examples with this feature include portfolio choices with proportional transaction costs, inventory management problems with proportional adjustment costs, and investment problems with linear and convex adjustment costs. The latter two examples are discussed extensively in [Stokey \(2008\)](#).

The main difficulty of solving dynamic optimization problems with impulse or singular controls lies in characterizing the inaction region of the controlled state X_t , where impulse or singular controls are not exercised. The variational inequality method based on HJB equations has been widely used for low-dimensional problems. For relevant high-dimensional problems, the situation becomes extremely difficult to solve. Such problems, however, often arise in macroeconomics—for example, the price-setting model in Chapter 12 of [Stokey \(2008\)](#) and the consumption of durable goods model in [McKay and Wieland \(2021\)](#). In the discrete-

time setting, models with similar features include [Khan and Thomas \(2008, 2013\)](#). Although much work remains, it is foreseeable that the probabilistic formulation could contribute to the numerical solution of the value function within the inaction region, due to its advantage over the HJB/PDE formulation.

A **forward-looking constraint** is an important modeling ingredient in both macroeconomics and finance. The widely used Ph.D.-level macroeconomic textbook [Ljungqvist and Sargent \(2018\)](#) devotes eight chapters to this topic. In the study of security design, dynamic incentives are critical due to information asymmetry ([Sannikov, 2013](#)). The key feature of this class of model is that an agent must respect other parties' incentive compatibility constraints when solving their own dynamic optimization problems. Consequently, the continuation value of other parties serves as a state variable for the agent in question. This deviates from the law of motion (27), which only characterizes the dynamics of backward-looking state variables.

The main difficulty in dynamic optimization problems with forward-looking constraints is to characterize the boundary of forward-looking state variables. Currently, there are two related approaches to circumvent this obstacle. The first is to transform primal optimization problems into their dual problems, under which the dynamics of forward-looking state variables are replaced by the dynamics of backward-looking co-state variables with clear boundaries ([Marcet and Marimon, 2019](#); [Pavoni, Sleet and Messner, 2018](#)). The second, known as the Negishi approach (following [Lucas and Stokey \(1984\)](#)), characterizes the set of all agents' continuation values sustained by all incentive-compatible equilibria. Given a precise characterization of this set, any single agent's dynamic optimization problem becomes identifying an element in the set that maximizes the agent's lifetime expected utility.

The Negishi approach utilizes a public randomization device to convexify the set of all sustainable equilibrium payoffs, and identifies the convex set with a support function—a standard method in convex analysis ([Bloise and Siconolfi, 2022](#); [Faingold and Sannikov, 2020](#)). The multipliers of the support function, interpreted as Pareto weights assigned by a social planner, become backward-looking state variables, much like the co-state variables in the dual approach. The entire problem is reduced to characterizing the support function—loosely speaking, the weighted sum of all agents' continuation values.

For both the dual and Negishi approaches, one ultimately has to solve an HJB equation numerically for high-dimensional models in the continuous-time setting. Examples include [Miao and Zhang \(2015\)](#) and [Pegoraro \(2023\)](#) (dual approach), and [Faingold and Sannikov \(2020\)](#) (Negishi approach). I speculate that the corresponding BSDE formulation of the HJB equation could potentially elevate computational efficiency.

5 Propagation of Diffusion Shocks

In the continuous-time setting with diffusion shocks, the Malliavin derivative is an essential mathematical tool for capturing the impact of diffusion shocks on economic time series. The term structure of expected impacts can be obtained by solving PDEs. For a comprehensive survey of this literature, readers are referred to [Borovička and Hansen \(2016\)](#).

In this section, I demonstrate that applying the chain rule to the FBSDE system of an eco-

nomic model directly yields a linear FBSDE system that characterizes the Malliavin derivatives, which can be numerically solved using deep learning. The computed Malliavin derivatives can subsequently be used to calculate the average impact of a realized shock over time, thereby circumventing the need to solve PDEs for high-dimensional models.

5.1 A Brief Introduction to Malliavin Derivative

Recall the d -dimensional Brownian motion $\{W_t\}_{t \geq 0}$ and the probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t \geq 0}, P)$ introduced in Section 2. Let \mathbf{S} denote the set of random variables of the form

$$F = f \left(\int_0^\infty \langle h_t^1, dW_t \rangle, \dots, \int_0^\infty \langle h_t^n, dW_t \rangle \right),$$

where f is an infinitely differentiable function with bounded partial derivatives of any order, and h_t^1, \dots, h_t^n are square-integrable functions mapping from $\Omega \times [0, \infty)$ to \mathbb{R}^d . The Malliavin derivative of F is defined as the d -dimensional process

$$\mathcal{D}_t F = \sum_{i=1}^n \frac{\partial f}{\partial x_i} \left(\int_0^\infty \langle h_s^1, dW_s \rangle, \dots, \int_0^\infty \langle h_s^n, dW_s \rangle \right) h_t^i.$$

The space $\mathbb{D}^{1,2}$ denotes the closure of \mathbf{S} with respect to the norm

$$\|F\|_{1,2} = \left[E|F|^2 + E \left(\int_0^\infty \|\mathcal{D}_t F\|^2 dt \right) \right]^{1/2}.$$

It can be shown that the Malliavin derivative has a closed extension to the space $\mathbb{D}^{1,2}$. A relatively accessible reference for Malliavin derivatives is [Nualart and Nualart \(2018\)](#). Intuitively, the Malliavin derivative $\mathcal{D}_t F$ captures the impact of the Brownian shock $W_{t+\Delta} - W_t$ on the random variable F as the length of the time interval Δ becomes arbitrarily small.

Similar to the ordinary derivative, the chain rule also applies to the Malliavin derivative. Suppose $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuously differentiable function with bounded partial derivatives, and $F_1, \dots, F_n \in \mathbb{D}^{1,2}$. Then $\phi(F_1, \dots, F_n) \in \mathbb{D}^{1,2}$ and

$$\mathcal{D}_t \phi(F_1, \dots, F_n) = \sum_{i=1}^n \frac{\partial \phi}{\partial x_i}(F_1, \dots, F_n) \mathcal{D}_t F_i.$$

5.2 Malliavin Derivatives of FBSDE solutions

Following the notation in Section 2, the Malliavin derivative of the equilibrium dynamics $(\mathcal{D}_u^i X_t, \mathcal{D}_u^i Y_t, \mathcal{D}_u^i Z_t, u \leq t \leq T)$ with respect to the i 'th dimension of W_t satisfies a linear

FBSDE system:

$$\begin{aligned}
\mathcal{D}_u^i X_t &= \mathcal{D}_u^i X_u + \int_u^t \frac{\partial b}{\partial x}(X_s, Y_s, Z_s) \mathcal{D}_u^i X_s + \frac{\partial b}{\partial y}(X_s, Y_s, Z_s) \mathcal{D}_u^i Y_s + \frac{\partial b}{\partial z}(X_s, Y_s, Z_s) \mathcal{D}_u^i Z_s ds \\
&\quad + \int_u^t \frac{\partial \sigma}{\partial x}(X_s, Y_s, Z_s) \mathcal{D}_u^i X_s + \frac{\partial \sigma}{\partial y}(X_s, Y_s, Z_s) \mathcal{D}_u^i Y_s + \frac{\partial \sigma}{\partial z}(X_s, Y_s, Z_s) \mathcal{D}_u^i Z_s dW_s, \\
\mathcal{D}_u^i Y_t &= \frac{\partial y}{\partial x}(X_T) \mathcal{D}_u^i X_T + \int_t^T \frac{\partial h}{\partial x}(X_s, Y_s, Z_s) \mathcal{D}_u^i X_s + \frac{\partial h}{\partial y}(X_s, Y_s, Z_s) \mathcal{D}_u^i Y_s + \frac{\partial h}{\partial z}(X_s, Y_s, Z_s) \mathcal{D}_u^i Z_s ds \\
&\quad - \int_t^T \mathcal{D}_u^i Z_s dW_s,
\end{aligned}$$

assuming that $b(\cdot)$, $\sigma(\cdot)$, $h(\cdot)$, and $y(\cdot)$ are continuously differentiable with uniformly bounded derivatives. Moreover,

$$[\mathcal{D}_u^1 X_u, \dots, \mathcal{D}_u^d X_u] = \sigma(X_u, Y_u, Z_u), \quad (29)$$

$$[\mathcal{D}_u^1 Y_u, \dots, \mathcal{D}_u^d Y_u] = Z_u. \quad (30)$$

Equation (29) is a general result for forward SDEs (see Chapter 7.5 of [Nualart and Nualart \(2018\)](#)), and Equation (30) is a result from Proposition 5.9 in [El Karoui, Peng and Quenez \(1997\)](#). These two equations indicate that Malliavin derivatives coincide with the volatility terms with respect to the immediate impact of diffusion shocks. The solution of the linear FBSDE system above, $(\mathcal{D}_u X_t, \mathcal{D}_u Y_t, \mathcal{D}_u Z_t, u \leq t \leq T)$, characterizes the propagation of the Brownian shock $(W_{u+\Delta}^i - W_u^i)$ in the dynamic system from u to T .

Malliavin derivatives are no longer time-homogeneous, and the time t now serves as a state variable. The Malliavin derivatives of equilibrium state variables, $\mathcal{D}_u^i X_t$, also serve as state variables. Since the FBSDEs satisfied by $(\mathcal{D}_u^i X_t, \mathcal{D}_u^i Y_t, \mathcal{D}_u^i Z_t)$ are independent of those satisfied by $(\mathcal{D}_u^j X_t, \mathcal{D}_u^j Y_t, \mathcal{D}_u^j Z_t)$ when $i \neq j$, the state variables for the forward-looking processes $(\mathcal{D}_u^i Y_t, \mathcal{D}_u^i Z_t)$ are the time t , the equilibrium state variable X_t , and its Malliavin derivative $\mathcal{D}_u^i X_t$.

5.3 Computation

While numerically solving for $(\mathcal{D}_u^j X_t, \mathcal{D}_u^j Y_t, \mathcal{D}_u^j Z_t, u \leq t \leq T)$, I focus on the mapping denoted by $\Sigma : (t, X_t, \mathcal{D}_u^j X_t) \rightarrow \mathcal{D}_u^j Z_t$, which is approximated by a feedforward neural network $\hat{\Sigma}(\cdot; \Theta)$. Note that when $t = u$, the initial state X_u is fixed, and the parametric approximation $\hat{\Sigma}$ reduces to $m \times d$ variables $\mathcal{D}_u^j Z_u$.

As in Section 2.2.2, I discretize the time interval $[u, T]$ as $u = t_0 < t_1 < t_2 < \dots < t_n = T$ and simulate M sample paths of Brownian motions $\{W_{i,k}\}_{i=0}^{n-1}^M$. According to the equilibrium FBSDE, I first generate M sample paths of $\{x_{i,k}, y_{i,k}, z_{i,k}\}$. The sample path index k will be omitted in subsequent notation. Equations (29) and (30) are used to obtain $\mathcal{D}_u^j x_0$ and $\mathcal{D}_u^j y_0$, respectively. Next, the following procedure is repeated for $i = 0$ to $i = n - 1$:

1. Compute $\mathcal{D}_u^j z_i$ given the network $\hat{\Sigma}(\cdot; \Theta)$, x_i , and $\mathcal{D}_u^j x_i$;

2. Calculate $\mathcal{D}_u^j x_{i+1}$ and $\mathcal{D}_u^j y_{i+1}$ according to

$$\begin{aligned}\mathcal{D}_u^j x_{i+1} &= \mathcal{D}_u^j x_i + \frac{\partial b}{\partial x} \Big|_i \mathcal{D}_u^j x_i \Delta_i + \frac{\partial b}{\partial y} \Big|_i \mathcal{D}_u^j y_i \Delta_i + \frac{\partial b}{\partial z} \Big|_i \mathcal{D}_u^j z_i \Delta_i \\ &\quad + \frac{\partial \sigma}{\partial x} \Big|_i \mathcal{D}_u^j x_i w_i + \frac{\partial \sigma}{\partial y} \Big|_i \mathcal{D}_u^j y_i w_i + \frac{\partial \sigma}{\partial z} \Big|_i \mathcal{D}_u^j z_i w_i, \\ \mathcal{D}_u^j y_{i+1} &= \mathcal{D}_u^j y_i - \frac{\partial h}{\partial x} \Big|_i \mathcal{D}_u^j x_i \Delta_i - \frac{\partial h}{\partial y} \Big|_i \mathcal{D}_u^j y_i \Delta_i - \frac{\partial h}{\partial z} \Big|_i \mathcal{D}_u^j z_i \Delta_i + \mathcal{D}_u^j z_i w_i.\end{aligned}$$

The loss function is given by

$$\text{Loss} \left(\Theta; \{W_{i,k}\}_{i=0}^{n-1} \right) = \frac{1}{Mn} \sum_{m=1}^M \sum_{i=1}^n \left\| \mathcal{D}_u^j y_i - \frac{\partial y}{\partial x}(x_i) \mathcal{D}_u^j x_i \right\|^2,$$

which is motivated by the chain rule as well as the terminal condition.

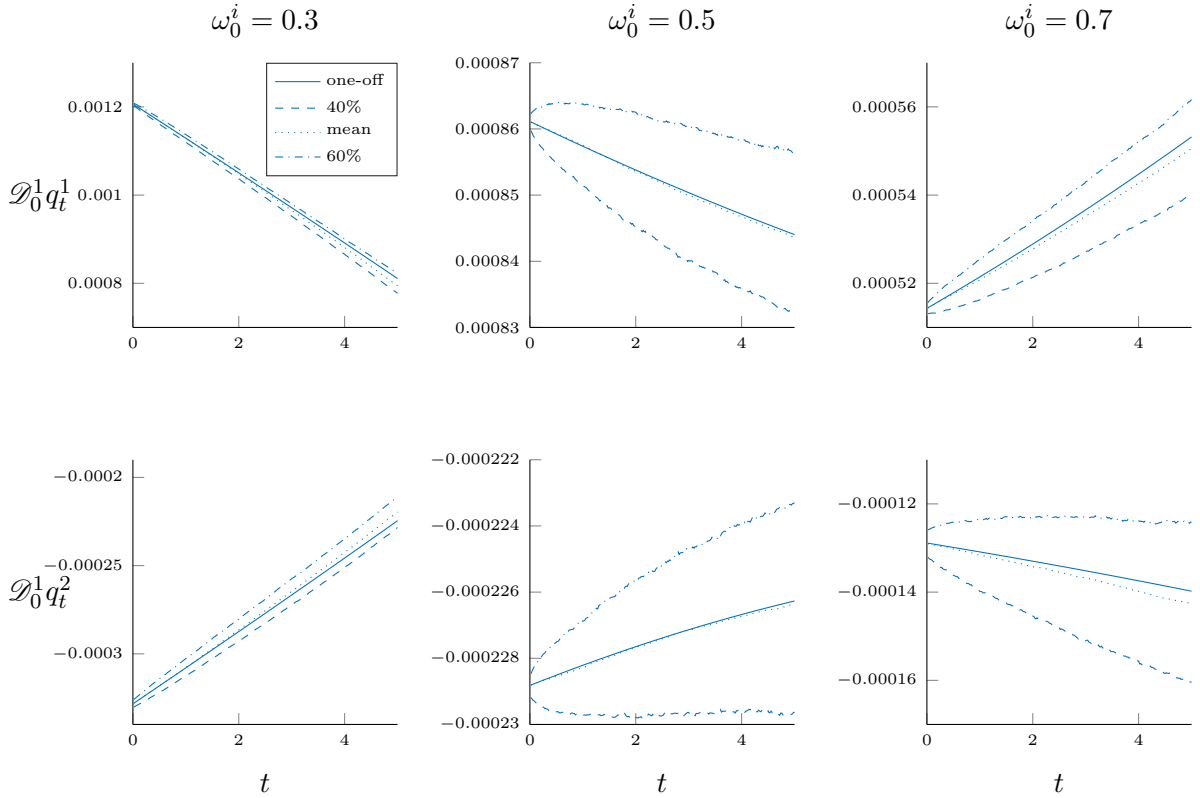


Figure 4: Propagation of Diffusion Shock

This figure displays the impacts of the Brownian shock to country 1 at $t = 0$ on the asset prices in countries 1 and 2 from $t = 0$ to $t = 5$ at three initial states: $(\eta_0^i = 0.3, \zeta_0^j = 0.2)$, $(\eta_0^i = 0.5, \zeta_0^j = 0.2)$, and $(\eta_0^i = 0.7, \zeta_0^j = 0.2)$

Example. I use Malliavin derivatives to characterize the propagation of Brownian shocks in the economy introduced in Section 3. Figure 4 depicts the impacts of country 1's Brownian shock at $t = 0$ on the domestic asset price q_t^1 and country 2's asset price q_t^2 from $t = 0$ to $t = 5$. In the figure, solid curves correspond to the zero-probability scenario where the economy experiences only the one-off shock, i.e., no further shocks occur between $t = 0$ and $t = 5$; dotted, dashed,

and dash-dotted lines represent the mean as well as the 40th and 60th percentiles of Malliavin derivatives in a sample of 20,000 simulations. Figure 4 shows that the short-term impacts of the Brownian shock are relatively large due to significant financial amplification effects when experts' wealth levels are low. Nevertheless, these impacts gradually decay over the longer term. If experts have strong financial conditions ($\eta_0^i = 0.7$), the effects of the Brownian shock are small but remain persistent.

Impulse Response. The impulse response function (IRF) characterizes the propagation of an exogenous shock along the sample path of the one-off shock shown in Figure 4. By contrast, Malliavin derivatives capture the propagation along arbitrary sample paths. It is easier to compute the impulse response function since the one-off shock sample path eliminates the need to compute $\mathcal{D}_0^i Z_t$. Nevertheless, Figure 4 shows that the IRF does not capture the expected impacts of the Brownian shock at $t = 0$ on endogenous variables from $t = 0$ to $t = 5$. The numerical example indicates that the deviation of the IRF from the expected impact increases over time, reaching up to 2 percent at $t = 5$ given the initial state ($\eta_0^i = 0.3$, $\zeta_0^j = 0.2$). This result is natural for a model with nonlinear dynamics.

6 Final Remarks

This paper shows that the equilibrium of a continuous-time macroeconomic or financial model can be characterized by a system of coupled infinite-horizon FBSDEs. Unlike the conventional analytic approach that transforms FBSDEs into PDEs, I emphasize that high-dimensional FBSDEs can be solved directly using deep learning. This idea originates from the applied mathematics literature, where the problem of solving high-dimensional PDEs is reformulated as a system of FBSDEs, which are then solved via deep learning. The algorithms I propose are straightforward to implement, with a level of difficulty comparable to simulating economic dynamics given the equilibrium solution.

When the number of state variables becomes a secondary concern in model construction, economists can treat parameters as time-invariant or pseudo-state variables, which substantially extends the scope of structural estimation. Recent works along this line include [Kase, Melosi and Rottner \(2022\)](#) on macroeconomic models and [Chen, Didisheim and Scheidegger \(2023\)](#) on option pricing.

References

- Azinovic, Marlon, Luca Gaegauf, and Simon Scheidegger (2022) "Deep equilibrium nets," *International Economic Review*.
- Azinovic, Marlon and Jan Zemlicka (2024) "Intergenerational consequences of rare disasters."
- Bloise, Gaetano and Paolo Siconolfi (2022) "A Negishi Approach to Recursive Contracts," *Econometrica*, Vol. 90, pp. 2821–2855.

- Borovička, Jaroslav and Lars Peter Hansen (2016) “Term structure of uncertainty in the macroeconomy,” in *Handbook of Macroeconomics*, Vol. 2: Elsevier, pp. 1641–1696.
- Bouchard, Bruno and Nizar Touzi (2004) “Discrete-time approximation and Monte-Carlo simulation of backward stochastic differential equations,” *Stochastic Processes and their applications*, Vol. 111, pp. 175–206.
- Brunnermeier, Markus K and Yuliy Sannikov (2014) “A Macroeconomic Model with a Financial Sector,” *The American Economic Review*, Vol. 104, pp. 379–421.
- Chen, Hongxiao and Ji Huang (2025) “Applications of Deep Learning-Based Probabilistic Approach to Economic Models with High-Dimensional Controls,” *Available at SSRN 5199943*.
- Chen, Hui, Antoine Didisheim, and Simon Scheidegger (2023) “Deep surrogates for finance: With an application to option pricing,” *Available at SSRN 3782722*.
- Chen, Zengjing and Larry Epstein (2002) “Ambiguity, risk, and asset returns in continuous time,” *Econometrica*, Vol. 70, pp. 1403–1443.
- Duarte, Victor, Diogo Duarte, and Dejanir H Silva (2024) “Machine learning for continuous-time finance,” *The Review of Financial Studies*, Vol. 37, pp. 3217–3271.
- Duffie, Darrell and Larry G Epstein (1992) “Stochastic differential utility,” *Econometrica: Journal of the Econometric Society*, pp. 353–394.
- E, Weinan, Jiequn Han, and Arnulf Jentzen (2017) “Deep Learning-Based Numerical Methods for High-Dimensional Parabolic Partial Differential Equations and Backward Stochastic Differential Equations,” *Communications in Mathematics and Statistics*, Vol. 5, pp. 349–380.
- El Karoui, Nicole, Shige Peng, and Marie Claire Quenez (1997) “Backward stochastic differential equations in finance,” *Mathematical finance*, Vol. 7, pp. 1–71.
- Faingold, Eduardo and Yuliy Sannikov (2020) “Moral hazard in stochastic differential games: Beyond markov equilibrium,” *Unpublished working paper*.
- Fan, Benjamin, Edward Qiao, Anran Jiao, Zhouzhou Gu, Wenhao Li, and Lu Lu (2024) “Deep learning for solving and estimating dynamic macro-finance models,” *Computational Economics*, pp. 1–37.
- Fernández-Villaverde, Jesús, Samuel Hurtado, and Galo Nuno (2022) “Financial frictions and the wealth distribution,” Technical report, National Bureau of Economic Research.
- Gobet, Emmanuel, Jean-Philippe Lemor, and Xavier Warin (2005) “A Regression-Based Monte Carlo Method to Solve Backward Stochastic Differential Equations,” *The Annals of Applied Probability*, Vol. 15, pp. 2172–2202.
- Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016) *Deep learning*: MIT press.
- Gopalakrishna, Goutham (2021) “Aliens and continuous time economies,” *Swiss Finance Institute Research Paper*.

- Gopalakrishna, Goutham, Jonathan Payne, and Zhouzhou Gu (2024) “Asset pricing, participation constraints, and inequality,” *Participation Constraints, and Inequality* (November 8, 2024).
- Gu, Zhouzhou, Mathieu Lauriere, Sebastian Merkel, and Jonathan Payne (2024) “Global solutions to master equations for continuous time heterogeneous agent macroeconomic models,” *arXiv preprint arXiv:2406.13726*.
- Han, Jiequn, Arnulf Jentzen, and Weinan E (2018) “Solving high-dimensional partial differential equations using deep learning,” *Proceedings of the National Academy of Sciences*, Vol. 115, pp. 8505–8510.
- Han, Jiequn, Yucheng Yang, and Weinan E (2021) “Deepham: A global solution method for heterogeneous agent models with aggregate shocks,” *arXiv preprint arXiv:2112.14377*.
- Hu, Zheyuan, Khemraj Shukla, George Em Karniadakis, and Kenji Kawaguchi (2024) “Tackling the curse of dimensionality with physics-informed neural networks,” *Neural Networks*, Vol. 176, p. 106369.
- Huang, Ji (2024) “Breaking the Curse of Dimensionality in Heterogeneous-Agent Models: A Deep Learning-Based Probabilistic Approach,” *Available at SSRN 4649043*.
- Huré, Côme, Huyên Pham, and Xavier Warin (2020) “Deep backward schemes for high-dimensional nonlinear PDEs,” *Mathematics of Computation*, Vol. 89, pp. 1547–1579.
- Ji, Shaolin, Shige Peng, Ying Peng, and Xichuan Zhang (2020) “Three algorithms for solving high-dimensional fully coupled FBSDEs through deep learning,” *IEEE Intelligent Systems*, Vol. 35, pp. 71–84.
- Judd, Kenneth L, Lilia Maliar, Serguei Maliar, and Inna Tsener (2017) “How to solve dynamic stochastic models computing expectations just once,” *Quantitative Economics*, Vol. 8, pp. 851–893.
- Karatzas, Ioannis and Steven Shreve (1998) *Brownian motion and stochastic calculus*, Vol. 113: Springer Science & Business Media.
- Kase, Hanno, Leonardo Melosi, and Matthias Rottner (2022) *Estimating nonlinear heterogeneous agents models with neural networks*: Centre for Economic Policy Research.
- Khan, Aubhik and Julia K Thomas (2008) “Idiosyncratic shocks and the role of nonconvexities in plant and aggregate investment dynamics,” *Econometrica*, Vol. 76, pp. 395–436.
- (2013) “Credit shocks and aggregate fluctuations in an economy with production heterogeneity,” *Journal of Political Economy*, Vol. 121, pp. 1055–1107.
- Ljungqvist, Lars and Thomas J Sargent (2018) *Recursive macroeconomic theory*: MIT press.
- Longstaff, Francis A and Eduardo S Schwartz (2001) “Valuing American options by simulation: a simple least-squares approach,” *The review of financial studies*, Vol. 14, pp. 113–147.

- Lucas, Robert E and Nancy L Stokey (1984) “Optimal growth with many consumers,” *Journal of Economic Theory*, Vol. 32, pp. 139–171.
- Ma, Jin and Jiongmin Yong (1995) “Solvability of Forward-Backward SDES and the Nodal Set of Hamilton-Jacobi-Bellman Equations.,” in *Chinese Annals of Mathematics, Series B*, Vol. 16, pp. 279–298.
- Maliar, Lilia, Serguei Maliar, and Pablo Winant (2021) “Deep learning for solving dynamic economic models.,” *Journal of Monetary Economics*, Vol. 122, pp. 76–101.
- Marcet, Albert and Ramon Marimon (2019) “Recursive contracts,” *Econometrica*, Vol. 87, pp. 1589–1631.
- McKay, Alisdair and Johannes F Wieland (2021) “Lumpy durable consumption demand and the limited ammunition of monetary policy,” *Econometrica*, Vol. 89, pp. 2717–2749.
- Miao, Jianjun and Yuzhe Zhang (2015) “A duality approach to continuous-time contracting problems with limited commitment,” *Journal of Economic Theory*, Vol. 159, pp. 929–988.
- Nualart, David and Eulalia Nualart (2018) *Introduction to Malliavin calculus*, Vol. 9: Cambridge University Press.
- Pardoux, Etienne and Shige Peng (1990) “Adapted solution of a backward stochastic differential equation,” *Systems & Control Letters*, Vol. 14, pp. 55–61.
- Pavoni, Nicola, Christopher Sleet, and Matthias Messner (2018) “The dual approach to recursive optimization: theory and examples,” *Econometrica*, Vol. 86, pp. 133–172.
- Payne, Jonathan, Adam Rebei, and Yucheng Yang (2025) “Deep learning for search and matching models,” *Available at SSRN 5123878*.
- Pegoraro, Stefano (2023) “Incentives and performance with optimal money management contracts,” *Journal of Political Economy*, Vol. 131, pp. 2920–2968.
- Raissi, Maziar (2024) “Forward–backward stochastic neural networks: deep learning of high-dimensional partial differential equations,” in *Peter Carr Gedenkschrift: Research Advances in Mathematical Finance*: World Scientific, pp. 637–655.
- Sannikov, Yuliy (2013) “Dynamic security design and corporate financing,” in *Handbook of the Economics of Finance*, Vol. 2: Elsevier, pp. 71–122.
- Sauzet, Maxime (2021) “Projection methods via neural networks for continuous-time models,” *Available at SSRN*.
- Schroder, Mark and Costis Skiadas (1999) “Optimal consumption and portfolio selection with stochastic differential utility,” *Journal of Economic Theory*, Vol. 89, pp. 68–126.
- Stokey, Nancy L (2008) *The Economics of Inaction: Stochastic Control models with fixed costs*: Princeton University Press.

Yong, Jiongmin and Xun Yu Zhou (1999) *Stochastic controls: Hamiltonian systems and HJB equations*, Vol. 43: Springer Science & Business Media.

Zhang, Aston, Zachary C Lipton, Mu Li, and Alexander J Smola (2023) *Dive into deep learning*: Cambridge University Press.

Zhang, Jianfeng (2004) “A numerical scheme for BSDEs,” *The annals of applied probability*, Vol. 14, pp. 459–488.

——— (2017) *Backward stochastic differential equations: from linear to fully nonlinear theory*: Springer.

Zhang, Wenzhong and Wei Cai (2022) “FBSDE based neural network algorithms for high-dimensional quasilinear parabolic PDEs,” *Journal of Computational Physics*, Vol. 470, p. 111557.

Appendix

A PDE formulation

The exact expressions of $A^j(\cdot)$ and $b^j(\cdot)$ in Section 3.6 are

$$\begin{bmatrix} \frac{\partial q^1}{\partial \eta^1} (1 - \eta_t^1) + \frac{\partial q^1}{\partial \zeta^1} \zeta_t^1 - \sum_{l=1}^{J-1} \frac{\partial q^1}{\partial \zeta^l} \zeta_t^l \zeta_t^1 - q_t^1, & \frac{\partial q^1}{\partial \eta^2} (1 - \eta_t^2) + \frac{\partial q^1}{\partial \zeta^2} \zeta_t^2 - \sum_{l=1}^{J-1} \frac{\partial q^1}{\partial \zeta^l} \zeta_t^l \zeta_t^2, & \cdots, \\ \frac{\partial q^2}{\partial \eta^1} (1 - \eta_t^1) + \frac{\partial q^2}{\partial \zeta^1} \zeta_t^1 - \sum_{l=1}^{J-1} \frac{\partial q^2}{\partial \zeta^l} \zeta_t^l \zeta_t^1, & \frac{\partial q^2}{\partial \eta^2} (1 - \eta_t^2) + \frac{\partial q^2}{\partial \zeta^2} \zeta_t^2 - \sum_{l=1}^{J-1} \frac{\partial q^2}{\partial \zeta^l} \zeta_t^l \zeta_t^2 - q_t^2, & \cdots, \\ \vdots & \vdots & \cdots, \\ \frac{\partial q^J}{\partial \eta^1} (1 - \eta_t^1) + \frac{\partial q^J}{\partial \zeta^1} \zeta_t^1 - \sum_{l=1}^{J-1} \frac{\partial q^J}{\partial \zeta^l} \zeta_t^l \zeta_t^1, & \frac{\partial q^J}{\partial \eta^2} (1 - \eta_t^2) + \frac{\partial q^J}{\partial \zeta^2} \zeta_t^2 - \sum_{l=1}^{J-1} \frac{\partial q^J}{\partial \zeta^l} \zeta_t^l \zeta_t^2, & \cdots, \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial q^1}{\partial \eta^J} (1 - \eta_t^J) - \sum_{l=1}^{J-1} \frac{\partial q^1}{\partial \zeta^l} \zeta_t^l \zeta_t^J \\ \frac{\partial q^2}{\partial \eta^J} (1 - \eta_t^J) - \sum_{l=1}^{J-1} \frac{\partial q^2}{\partial \zeta^l} \zeta_t^l \zeta_t^J \\ \vdots \\ \frac{\partial q^J}{\partial \eta^J} (1 - \eta_t^J) - \sum_{l=1}^{J-1} \frac{\partial q^J}{\partial \zeta^l} \zeta_t^l \zeta_t^J - q_t^J \end{bmatrix}$$

and

$$\begin{bmatrix} \sum_{l=1}^{J-1} \frac{\partial q^1}{\partial \zeta^l} \zeta_t^l \zeta_t^j - \frac{\partial q^1}{\partial \eta^j} (1 - \eta_t^j) - \frac{\partial q^1}{\partial \zeta^j} \zeta_t^j \\ \sum_{l=1}^{J-1} \frac{\partial q^2}{\partial \zeta^l} \zeta_t^l \zeta_t^j - \frac{\partial q^2}{\partial \eta^j} (1 - \eta_t^j) - \frac{\partial q^2}{\partial \zeta^j} \zeta_t^j \\ \vdots \\ \sum_{l=1}^{J-1} \frac{\partial q^J}{\partial \zeta^l} \zeta_t^l \zeta_t^j - \frac{\partial q^J}{\partial \eta^j} (1 - \eta_t^j) \end{bmatrix},$$

respectively.