DSGE in a Nutshell

BY XIAONAN PENG

Abstract

In this review, we delve into the intricacies of Dynamic Stochastic General Equilibrium (DSGE) models and their associated solution techniques. Specifically, the review is designed to assist newcomers to the field in comprehending the paper titled "Differentiable State-Space Models and Hamiltonian Monte Carlo Estimation." To facilitate this understanding, we provide an in-depth derivation of both the perturbation method and the Bayesian approach, using the Real Business Cycle (RBC) model as a representative example. The goal is to elucidate the conceptual underpinnings of DSGE models and to demonstrate the manual solution process, equipping readers with the knowledge to tackle these models effectively.

1 Introduction

Understanding Dynamic Stochastic General Equilibrium (DSGE) models can be a daunting task for those new to the field of macroeconomics. The complexity arises from the diverse mathematical and computational techniques these models employ. To bridge this gap, this review provides a step-by-step guide to solving DSGE models, aimed specifically at helping beginners grasp the fundamental concepts and methodologies.

The journey begins with the Real Business Cycle (RBC) model, a starting point for understanding DSGE models. We meticulously work through the linearization of the RBC model using a first-order perturbation, which simplifies the model while retaining essential dynamics. This step transforms the model into a state-space form, making it more tractable for further analysis.

With the state-space representation in hand, we turn to the Kalman filter, a tool that allows us to evaluate the likelihood of the model given actual economic data. This step is crucial as it connects theoretical models with real-world observations, providing a means to assess the model's validity.

Once we've established the model's likelihood, parameter estimation comes next. Unlike many references that gloss over this critical step, we dive into the details of Hamiltonian Monte Carlo (HMC), a sophisticated sampling method. HMC improves the efficiency of drawing samples from the posterior distribution of the model's parameters, offering a more nuanced understanding of the model's behavior.

Our goal in this review is not merely to outline the process but to delve into the detailed derivations often omitted in the literature. Many articles on DSGE models tend to be overly abstract, making them difficult for newcomers to follow. We intend to fill this void by providing clear, step-by-step explanations and derivations that are rarely available elsewhere.

The paper is organized as follows:

Section 2 introduces the basics of DSGE models, detailing the foundational neoclassical growth model and its transition into the RBC model, setting the stage for the analysis that follows.

Section 3 describes the transformation of the RBC model into a state-space representation, followed by an explanation of the perturbation method used to solve the model.

Section 4 discusses Bayesian methods for parameter estimation, with a focus on HMC and its application to derive parameter distributions from the model.

Section 5 examines the recursive Epstein-Zin utility function, detailing the perturbation method required to solve it within the DSGE framework.

Section 6 looks at modern methods, such as deep learning, for solving DSGE models.

By the end of this paper, readers should have a clear understanding of the steps involved in building, solving, and estimating DSGE models, equipped with the knowledge to undertake their own economic research.

2 DSGE basics

In macroeconomic analysis, Dynamic Stochastic General Equilibrium (DSGE) models stand as a pivotal tool for understanding complex economic dynamics. These models capture the interactions of agents, such as households and firms, and their response to policy changes and exogenous shocks.

2.1 Neoclassical Growth Model

We start from the neoclassical growth model to introduce the idea and solution methods of DSGE models. Neoclassical growth model is the simplest and fundamental macroeconomic model to describe the growth economy.

Suppose market consists of a representative household and a representative firm.

- The representative household owns capital k_t, labor l_t, the household's problem is to maximize lifetime utility by choosing the consumption c_t, given by:

$$\begin{aligned} &\max_{\{c_t, l_t, k_t\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t U(c_t) \\ s.t. \quad & c_t + i_t \leqslant r_t k_t + w_t l_t \\ & k_{t+1} = (1-\delta)k_t + i_t \\ & k_t, c_t \geqslant 0, 0 \leqslant l_t \leqslant 1 \end{aligned}$$

  The term $\beta^t U(c_t)$ is the discounted utility at period t, β is the discount factor.

  The first condition means that the household earns income from supplying labor and renting capital to the firm and the sum of consumption and investment should be less than the income, where i_t is the investment, w_t is the labor wage, r_t is the rental rate of capital.

  The second condition means the capital accumulation process, δ is the depreciation rate of capital.

- The representative firm's problem is to maximize profits by choosing capital K_t and labor L_t, given by:

$$\begin{aligned} &\max_{\{y_t, K_t, L_t\}} y_t - r_t K_t - w_t L_t \\ s.t. \quad & y_t = F(K_t, L_t) \end{aligned}$$

  where y_t is the production output and F is the production function.

Definition. *A **competitive equilibrium** is a stochastic process for $\{c_t, l_t, k_t, i_t, y_t\}$ and prices $\{w_t, r_t\}$ such that*

1.

$$\begin{aligned} \text{Market clear:} \quad & y_t = c_t + i_t \\ \text{Labor clear:} \quad & l_t = L_t \ (w_t \text{ clears the labor market}) \\ \text{Capital clear:} \quad & k_t = K_t \ (r_t \text{ clears the capital market}) \end{aligned}$$

2. *Household solves their utitlity maximization problem*

3. *Firm solves their profit maximization problem.*

Definition. *An allocation* $\{c_t, k_t, l_t\}$ *is* ***feasible*** *if for all t, we have*

$$F(k_t, l_t) = c_t + k_{t+1} - (1-\delta)k_t, \;\; c_t, k_t \geqslant 0$$

An allocation is ***Pareto optimal*** *if it's feasible and there is no other feasible allocation which can make at least one household better without making anyone else worse.*

We can compute the Pareto optimal allocation by solving the **social planner problem**:

$$\begin{aligned} &\max_{\{c_t\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t U(c_t) \\ s.t. \;\; &F(k_t, l_t) = c_t + k_{t+1} - (1-\delta)k_t \\ &c_t, k_t \geqslant 0 \end{aligned}$$

Theorem. *The first and second fundamental theorems of welfare economics state that*

$$\text{allocation in competitive equilibrium} \Leftrightarrow \text{Pareto optimal allocation}$$

Now we transfer our problem of finding the competitive equilibrium to solving the social planner problem.

2.2 Solution methods

To solve the social planner problems, we typically use methods such as:

- **Bellman Equation:** A recursive formulation representing the value function of an optimization problem, which is used to characterize the optimal policy over time.

- **Euler Equation:** Derived from the first-order conditions of the Bellman equation, it relates the intertemporal choice of consumption and provides conditions that must be satisfied along the optimal path.

In the equilibrium, we can let $l_t = 1$. Define

$$f(k_t) = F(k_t, 1) + (1-\delta)k_t$$

then we have

$$c_t = f(k_t) - k_{t+1} \geqslant 0$$

We can write the social planner problem as

$$V(k_0) = \max_{\{k_t\}_{t=1}^{\infty}, k_{t+1} \leqslant f(k_t)} \sum_{t=0}^{\infty} \beta^t U(f(k_t) - k_{t+1})$$

where V is the value of objective function given a starting point k_0, so we can call $V(k_0)$ as "**value function**". Noticing that the function V can be defined recursively:

$$\begin{aligned}
V(k_0) &= \max_{\{k_t\}_{t=1}^{\infty}, k_{t+1} \leqslant f(k_t)} \sum_{t=0}^{\infty} \beta^t U(f(k_t) - k_{t+1}) \\
&= \max_{\{k_t\}_{t=1}^{\infty}, k_{t+1} \leqslant f(k_t)} U(f(k_0) - k_1) \beta \sum_{t=1}^{\infty} \beta^{t-1} U(f(k_t) - k_{t+1}) \\
&= \max_{k_1 \leqslant f(k_0)} \left\{ U(f(k_0) - k_1) + \beta \max_{\{k_t\}_{t=2}^{\infty}, k_{t+1} \leqslant f(k_t)} \sum_{t=1}^{\infty} \beta^{t-1} U(f(k_t) - k_{t+1}) \right\} \\
&= \max_{k_1 \leqslant f(k_0)} \left\{ U(f(k_0) - k_1) + \beta \max_{\{k_t\}_{t=2}^{\infty}, k_{t+1} \leqslant f(k_t)} \sum_{t=0}^{\infty} \beta^{t} U(f(k_{t+1}) - k_{t+2}) \right\} \\
&= \max_{k_1 \leqslant f(k_0)} U(f(k_0) - k_1) + \beta V(k_1)
\end{aligned}$$

Similarly, we can write

$$V(k_t) = \max_{k_{t+1} \leqslant f(k_t)} \{U(f(k_t) - k_{t+1}) + \beta V(k_{t+1})\}$$

which is called **Bellman equation**.

2.2.1 Bellman equation

More generally, we can summarize the Bellman equation problem as:

$$\begin{aligned}
&\max_{u_t} \sum_{t=0}^{\infty} \beta^t r(x_t, u_t) \\
s.t. \quad & x_{t+1} = g(x_t, u_t), \quad \text{given } x_0
\end{aligned}$$

Then the associated Bellman equation is

$$V(x_t) = \max_{u_t} \{r(x_t, u_t) + \beta V[g(x_t, u_t)]\}$$

where r and g are known functions.

x_t is the state variable, u_t is the control variable. They can be vectors. The state variable summarizes all current information relevant to decision making, and the control variable is the decision to be made. Sometimes it is convenient to define the problem so that the control variable is just the next value of the state variable ($u_t = x_{t+1}$).

The optimal plan can be describe by a policy function. That is, there exists a function h such that the optimal u_t is given by $u_t = h(x_t)$. Then

$$V(x_t) = r(x_t, h(x_t) + \beta V[g(x_t, h(x_t))])$$

To solve the Bellman equation, we have two ways:

1. Recursive methods:

- Suppose we already know the value function. Finding the policy function is a simple univariate (or multivariate if u_t is a vector) optimization problem.

- Suppose we already know the policy function. Finding the value function is simply a matter of plugging the policy function into the objective function.

This suggests a recursive solution method. Suppose we have a reasonable guess of the value function, which we'll call V_0. Any continuous and bounded function is OK here, but a good guess will often speed up the process.

$$V_1(x_t) = \max_{u_t} [r(x_t, u_t) + \beta V_0(g(x_t, u_t))]$$

Define $V_2, V_3, \ldots,$ in the same manner. Under some technical conditions which are met in this case, V_i will converge (uniformly) to the correct value function V. This solution method is called "**value function iteration**."

Alternatively, we could start with an initial guess h_0 for the policy function, then substitute in to

$$V_0 = \sum_{t=0}^{\infty} \beta^t r(x_t, h(x_t)) \quad \text{where } x_{t+1} = g(x_t, h(x_t))$$

Then we can calculate an imporved policy function h_1:

$$h_1(x) = \arg\max_{u} [r(x, u) + V_0(g(x, u))]$$

If we repeat this, h_i will converge (uniformly) to the true policy function and V_i will converge to the true value function. This method is called "**policy function iteration**."

The recursive solution method is very powerful but all of this iteration takes a lot of time to do and there is no closed form for value function and policy function.

2. Euler equation methods

Another solution method is to apply calculus. Under certain conditions, the value function is differentiable, and a solution must satisfy the first order condition as well as a "transversality condition:"

$$\begin{aligned} \frac{\partial r}{\partial u}(x_t, u_t) + \beta V'[g(x_t, u_t)] \frac{\partial g}{\partial u}[x_t, u_t] &= 0 \qquad \text{(FOC)} \\ \lim_{t \to \infty} \beta^t V'(x_t) x_t &= 0 \qquad \text{(TVC)} \end{aligned}$$

transversality condition is closely related to the limit of the complementary slackness condition in the case of finite T. We will ignore it most of the time, but occasionally it's very important. In addition, we can apply a result sometimes called the **Benveniste-Scheinkman theorem**:

$$V'(x_t) = \frac{\partial r}{\partial x}(x_t, u_t) + \beta V'[g(x_t, u_t)] \frac{\partial g}{\partial x}(x_t, u_t)$$

This is kind of a dynamic programming version of the envelope theorem.

2.2.2 Euler equation for neoclassical growth model

Let's apply Euler equation to our model to see how can we get the equations.

$$V(k_t) = \max_{k_{t+1} \leqslant f(k_t)} \{U(f(k_t) - k_{t+1}) + \beta V(k_{t+1})\}$$

We can write the FOC w.r.t. k_{t+1} for the RHS of the Bellman equation

$$0 = -U'(f(k_t) - k_{t+1}) + \beta V'(k_{t+1})$$

If we view k_{t+1} as the policy u_t, and suppose we have true policy function $k_{t+1} = h(k_t)$, then we can write the FOC w.r.t. k_t for the both sides of Bellman equation

$$\begin{aligned} V'(k_t) &= U'(f(k_t) - h(k_t))(f'(k_t) - h'(k_t)) + \beta V'(k_{t+1})h'(k_t) \\ &= U'(f(k_t) - h(k_t))f'(k_t) + [\beta V'(k_{t+1}) - U'(f(k_t) - h(k_t))]h'(k_t) \\ &= U'(f(k_t) - h(k_t))f'(k_t) \end{aligned}$$

where the second term is 0 since the FOC w.r.t. k_{t+1}. Combining two FOCs, we have

$$\begin{aligned} U'(f(k_t) - k_{t+1}) &= \beta U'(f(k_{t+1}) - h(k_{t+1}))f'(k_{t+1}) \\ &= \beta U'(f(k_{t+1}) - k_{t+2})f'(k_{t+1}) \end{aligned}$$

We can substitute $c_t = f(k_t) - k_{t+1}$ to the above equation to get

$$U'(c_t) = \beta U'(c_{t+1})f'(k_{t+1})$$

which is **Euler equation**. It has an economic interpretation: along an optimal path the marginal utility from consumption at any point in time is equal to its opportunity cost.

For finite time period: $t = 1, \ldots, T$, the boundary condition is easy: $V(k_{T+1}) = 0 \Rightarrow K_{T+1} = 0$. But for the infinite time period, we need to find a limiting boundary condition.

We can multiply β^t to the both sides of Euler equation to get

$$\beta^t U'(c_t) = \beta^{t+1} U'(c_{t+1})f'(k_{t+1})$$

We can view the LHS as the marginal utility in period t, RHS as the marginal utility gain in period $t+1$ by giving up 1 unit of consumption in period t. Then we have

$$\lim_{t \to \infty} \beta^t U'(c_t)f'(k_t)k_t = 0$$

which is the **transversality condition** (TVC). It basically ensures that the present value of the benefits from investing in capital falls over time, preventing unbounded accumulation. Mathematically, it is expressed as the limit of the discounted value of capital approaching zero as time goes to infinity, ensuring that the marginal utility of capital does not outweigh the benefits from consumption.

Therefore, an allcocation $\{k_t\}$ that satisfies the Euler equation and the TVC solves the social planner problem.

Up to know, there is still one left, we don't know the true policy function $h(\cdot)$. We can use some methods such as perturbation to **approximate policy function h under Euler equation and other equilibrium conditions.**

In summary, there are three common solution methods for DSGE include:

- **Value Function Iteration (VFI)**: This involves iterating on the Bellman equation's value function until convergence. It's exact but can be computationally intensive.

- **Perturbation Methods**: These methods linearize the model around its steady state and are faster but may be less accurate if the economy is far from the steady state or if the model is highly non-linear

- **Projection Methods**: These methods approximate the value function or policy function over the state space using basis functions. They are flexible and can handle higher dimensions but require careful selection of basis functions and can be complex to implement.

Each method has its strengths and weaknesses: VFI is precise but slow, perturbation is fast but can be imprecise, and projection methods are flexible but complex. The choice of method often depends on the particular features of the model and the researcher's objectives.

2.3 Real Business Cycle Model

In this section, we will introduce the real business cycle (RBC) model, which is a cornerstone of modern macroeconomics, built on the neoclassical growth framework. It incorporates stochastic shocks to capture the fluctuations in the economy, known as business cycles. The whole notes is talking about how to solve the RBC model. Similar to the neoclassical model, the RBC model features a representative household and a representative firm making intertemporal choices under uncertainty.

The representative household wants to maximize expected utility, choosing consumption c_t and labor supply l_t, subject to a stochastic budget constraint:

$$\begin{aligned} &\max_{\{c_t,l_t,k_t\}} \mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t U(c_t, l_t) \\ s.t. \quad & c_t + i_t \leqslant r_t k_t + w_t l_t \\ & k_{t+1} = (1-\delta)k_t + i_t \\ & k_t, c_t \geqslant 0, 0 \leqslant l_t \leqslant 1 \end{aligned}$$

where $U(c_t, l_t)$ is the household's period utility function, which depends on consumption and leisure (where leisure is $1 - l_t$), β is the discount factor.

The representative firm wants to maximize profit

$$\begin{aligned} &\max_{\{K_t,L_t\}} F(K_t, L_t) - w_t L_t - r_t K_t \\ s.t. \quad & F(k_t, l_t) = e^{z_t} k_t^{\alpha} l_t^{1-\alpha} \end{aligned}$$

where $F(k_t, l_t)$ is a Cobb-Douglas production function, e^{z_t} represents the level of technology (or total factor productivity (TFP)), and z_t follows an AR(1) process:

$$z_t = \rho z_{t-1} + \sigma\varepsilon_t, \varepsilon_t \sim N(0,1)$$

Here, ρ is the persistence parameter, σ is the standard deviation of the technological shock, and ε_t is the stochastic shock term with a standard normal distribution.

So the social planner's problem is

$$\begin{aligned} &\max_{\{c_t,l_t\}} \mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t U(c_t, l_t) \\ s.t. \quad & c_t + k_{t+1} - (1-\delta)k_t = y_t \\ & y_t = e^{z_t} k_t^{\alpha} l_t^{1-\alpha} \\ & z_t = \rho z_{t-1} + \sigma\varepsilon \end{aligned}$$

Then we can write the equilibrium condition as

$$
\begin{aligned}
U_c'(c_t, l_t) &= \beta \mathbb{E}_t U_c'(c_{t+1}, l_{t+1}) f'(k_{t+1}) \\
f(k_t) &= y_t + (1-\delta)k_t \\
c_t + k_{t+1} - (1-\delta)k_t &= y_t \\
y_t &= e^{z_t} k_t^{\alpha} l_t^{1-\alpha} \\
z_t &= \rho z_{t-1} + \sigma \varepsilon_t
\end{aligned}
$$

Assuming we have the state transfer function h, and the policy function g s.t.

$$
\begin{aligned}
k_{t+1} &= h(k_t, z_t) \\
c_t, l_t &= g_1(k_t, z_t), g_2(k_t, z_t)
\end{aligned}
$$

here we view k_t, z_t as state variables and c_t, l_t as control variables.

We can encapsulate the equilibrium conditions of the RBC model using an operator $\mathcal{H}$, which combines the system of equations that must be satisfied for the economy to be in equilibrium:

$\mathcal{H}(h, g) =$

$$
\begin{cases}
U_c'(g_1(k_t, z_t), g_2(k_t, z_t)) - \beta \mathbb{E}_t U_c'(g_1(h(k_t, z_t), z_{t+1}), g_2(h(k_t, z_t), z_{t+1}))(e^{z_{t+1}} \alpha h(k_t, z_t)^{\alpha - 1} + 1 - \delta) \\
g_1(k_t, z_t) + h(k_t, z_t) - (1-\delta)k_t - e^{z_t} k_t^{\alpha} g_2(k_t, z_t)^{1-\alpha} \\
z_{t+1} - \rho z_t - \sigma \varepsilon_{t+1}
\end{cases}
$$

where the operator $\mathcal{H}$ is a functional with arguments: the state transfer function h and the policy functions g. The functions h and g are the true state transition and policy functions if they satisfy

$$
\mathcal{H}(h, g) = 0
$$

Q: Why do we need the operator $\mathcal{H}$?

A: The operator $\mathcal{H}$ provides a systematic way to identify the true state transition function h and policy functions g. By finding the functions that zero out the operator, we ascertain the policy functions that satisfy the model's intertemporal optimization conditions, capturing the economy's response to shocks and its dynamic evolution over time.

3 State-space representation and perturbation

Perturbation algorithm is a common method to solve DSGE models, which build Taylor series approximations to the solution of a DSGE model around its deterministic steady state using implicit function theorem.

3.1 State-space representation

State-space representation is a general representation for DSGE models, this representation can describe any DSGE model. Let x_t be the state variable, y_t be the control variable, z_t be the observables, θ is the parameter of the model. We have

$$
\begin{aligned}
\text{Policy function:} \quad & y_t = g(x_t; \theta) \\
\text{State transfer:} \quad & x_{t+1} = h(x_t; \theta) + \eta \varepsilon_{t+1} \\
\text{Measurement equation:} \quad & z_t = q(x_t, v_t; \theta)
\end{aligned}
$$

where the $\eta\varepsilon_{t+1}$ represents the exogenous shocks affecting the state transition, η is the loading matrix of shocks to each variable, ε_t is the exogenous shocks.

For RBC model we introduce above, the state variable x_t can be capital k_t, and TPF z_t, the control variable y_t can be consumption c_t, labor l_t, and production output q_t (here we use q_t to difference with control variable y_t), the observable can be the real data from economy, such as consumption c_{obs} and investment i_{obs}, where c_{obs} is different with c_t, one is our observed quantity, one is the model prediction.

Q: Why are exogenous shocks represented outside the state transition function $h(\cdot)$?

A: Exogenous shocks are separated from the state transition function to isolate the deterministic component of the model's dynamics, allowing us to analyze the impact of stochastic processes on the system's behavior.

We can write the equilibrium condition as

$$\mathbb{E}_t\mathcal{H}(y', y, x', x, \theta) = 0$$

$$x' = h(x) + \eta\varepsilon', y = g(x)$$

where x' represents a variable at period $t+1$ and x a variable at period t. Here we place the conditional expectation operator outside $\mathcal{H}$: for those equilibrium conditions without expectations, the conditional expectation operator will not have any impact. Moving expectation outside $\mathcal{H}$ will make some of the derivations below easier to follow.

Definition. *If we suppress the stochastic component of the model, we can define the deterministic steady state (DSS) of the model as vector $\bar{x}, \bar{y}$ such that*

$$\mathcal{H}(\bar{y}, \bar{y}, \bar{x}, \bar{x}) = 0$$

representing the state of the model when it is unaffected by stochastic shocks.

Here we use a simplified RBC model as an example to show how to solve the DSGE model by perturbation. We assume $l_t = 1$ and $U(c_t, l_t) = \log c_t$. Then we can write the equilibrium conditions in the operator $\mathcal{H}$ like we did in last section:

$$\mathcal{H} = \begin{cases} \frac{1}{c_t} - \beta\frac{\alpha e^{z_{t+1}} k_{t+1}^{\alpha-1} + (1-\delta)}{c_{t+1}} \\ c_t + k_{t+1} - (1-\delta)k_t - q_t \\ q_t - e^{z_t}k_t^{\alpha} \\ z_{t+1} - \rho z_t - \sigma\varepsilon_{t+1} \end{cases}$$

here we have

$$\begin{aligned} \text{state variable:} \quad & x = [k, z] \\ \text{control variable:} \quad & y = [c, q] \\ \text{parameters:} \quad & p = [\alpha, \beta, \rho, \delta, \sigma] \end{aligned}$$

the steady state can be solved from $\mathcal{H}(\bar{y}, \bar{y}, \bar{x}, \bar{x}) = 0$:

$$\bar{k} = \left(\frac{\frac{1}{\beta} - 1 + \delta}{\alpha}\right)^{\frac{1}{\alpha-1}}; \ \bar{z} = 0; \ \bar{c} = \left(\frac{\frac{1}{\beta} - 1 + \delta}{\alpha}\right)^{\frac{\alpha}{\alpha-1}} - \delta\left(\frac{\frac{1}{\beta} - 1 + \delta}{\alpha}\right)^{\frac{1}{\alpha-1}}; \ \bar{q} = \left(\frac{\frac{1}{\beta} - 1 + \delta}{\alpha}\right)^{\frac{\alpha}{\alpha-1}}$$

3.2 First-order perturbation solution

A general first order perturbation approximate g and h around $(x;\chi)=(\bar{x};0)$, where χ is the perturbation parameter as controlling the standard deviation of the stochastic process. The policy and transfer equation depends on χ

$$y=g(x;\chi),\quad x'=h(x;\chi)+\chi\eta\varepsilon'$$

Noticing that perturbation parameter is not a true parameter in our model, we just introduce it as an imaginary parameter to help us find the perturbation solution. We can let it be 1 in the final result, which not influence the result.

From the definition of DSS, we have

$$\bar{y}=g(\bar{x};0),\bar{x}=h(\bar{x};0)$$

The first order Taylor expansions of g and h around DSS are:

$$\begin{aligned} g(x;\chi) &= g(\bar{x};0)+g_x(\bar{x};0)(x-\bar{x})+g_\chi(\bar{x};0)\chi \\ h(x;\chi) &= h(\bar{x};0)+h_x(\bar{x};0)(x-\bar{x})+h_\chi(\bar{x};0)\chi \end{aligned}$$

We can show that $g_\chi(\bar{x};0)$ and $h_\chi(\bar{x};0)$ both equal 0 (we ignore the proof here), then we can see that the first order approximation is independent of perturbation parameter χ, we have

$$\begin{aligned} g(x) &= g(\bar{x})+g_x(\bar{x})(x-\bar{x}) \\ &= \bar{y}+g_x(\bar{x})(x-\bar{x}) \\ h(x) &= h(\bar{x})+h_x(\bar{x})(x-\bar{x}) \\ &= \bar{x}+h_x(\bar{x})(x-\bar{x}) \\ \Rightarrow\ y-\bar{y} &= g_x(\bar{x})(x-\bar{x}) \\ x'-\bar{x} &= h_x(\bar{x})(x-\bar{x})+\chi\eta\varepsilon' \end{aligned}$$

We define $\hat{x}=x'-\bar{x}$ and $\hat{y}=y-\bar{y}$ be the deviations from the DSS, and let $\chi=1$. Thus, we can write the first-order solution to the model as:

$$\begin{aligned} \hat{y} &= g_x\hat{x} \\ \hat{x} &= h_x\hat{x}+\eta\varepsilon' \end{aligned}$$

where $g_x=g_x(\bar{x}),h_x=h_x(\bar{x})$. For simplicit, we will use x,y to represent $\hat{x},\hat{y}$ in the remaining part of the notes.

However, we still don't know the value of first order derivatives h_x and g_x. We can use the first order perturabtion to $\mathcal{H}$ to solve the g_x and h_x, the dimensions of g_x and h_x are $n_y\times n_x$ and $n_x\times n_x$. A first order Taylor expansion yields:

$$\mathcal{H}_{y'}y'+\mathcal{H}_y y+\mathcal{H}_{x'}x'+\mathcal{H}_x x=0$$

where $\mathcal{H}_{x,ij}=\frac{\partial\mathcal{H}_i}{\partial x_j},i=1,\dots,n=n_x+n_y,j=1,\dots,n_x;\mathcal{H}_{y,ij}=\frac{\partial\mathcal{H}_i}{\partial y_j},i=1,\dots,n,j=1,\dots,n_y$. All these derivatives are evaluated at steady state. Then

$$\begin{bmatrix}\mathcal{H}_{x'} & \mathcal{H}_{y'}\end{bmatrix}\begin{bmatrix}x'\\y'\end{bmatrix}+\begin{bmatrix}\mathcal{H}_x & \mathcal{H}_y\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix}=0$$

We know that g_x and h_x are in x' and y', so this is a matrix equation for g_x and h_x. To solve this equation, we need to introduce some linear algebra terminologies.

For our model, we let

$$A = [\ \mathcal{H}_{x'}\ \ \mathcal{H}_{y'}\ ], B = [\ \mathcal{H}_x\ \ \mathcal{H}_y\ ]$$

Then the matrix equation becomes

$$A\begin{bmatrix} x' \\ y' \end{bmatrix} + B\begin{bmatrix} x \\ y \end{bmatrix} = 0$$

We want to decompose A and B to simplify our equation.

Q: What decomposition should we use?

A: We can't ensure that A and B are diagonalizable. So we need to use Schur decomposition.

Theorem. *Given any matrix A, we can do* ***Schur decomposition*** *as follows:*

$$U^*AU = T$$

where U is an unitary matrix and T is an upper-triangular matrix.

However, Schur decompostion to A and B can't simplify our equation:

$$U_1SU_1^*\begin{bmatrix} x' \\ y' \end{bmatrix} + U_2TU_2^*\begin{bmatrix} x \\ y \end{bmatrix} = 0$$

U_1 and U_2 are two different unitary matrices, we can't eliminate them. If A and B commutes, they can simultaneously upper-triangularized. But it's not true in general. So we need to use generalized Schur decomposition, which is particularly useful when the matrices do not commute and cannot be simultaneously upper-triangularized.

Definition. *We define the* ***generalized eigenvalue*** *by solve the equation:*

$$Ax = \lambda Bx$$

then we say that λ is the generalized eigenvalue of $A - \lambda B$.

Genralized Schur decomposition can be used to solve the generalized eigenvalue problem.

Theorem. *Given any matrices A and B, we can do* ***generalized Schur decomposition*** *(QZ decomposition) as follows:*

$$Q^*AZ = S, Q^*BZ = T$$

where S and T are upper triangular matrices, Q and Z are unitrary matrices.

Moreover,

$$\lambda_i = \frac{S_{ii}}{T_{ii}}, i = 1, \ldots, n$$

are the generalized eigenvalues of A and B.

Applying the QZ decomposition to our equation:

$$\begin{aligned} 0 &= A\begin{bmatrix} x' \\ y' \end{bmatrix} + B\begin{bmatrix} x \\ y \end{bmatrix} \\ &= QSZ^*\begin{bmatrix} x' \\ y' \end{bmatrix} + QTZ^*\begin{bmatrix} x \\ y \end{bmatrix} \\ \Rightarrow 0 &= SZ^*\begin{bmatrix} x' \\ y' \end{bmatrix} + TZ^*\begin{bmatrix} x \\ y \end{bmatrix} \end{aligned}$$

Therefore, we can write the block-wise form as

$$\begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix}\begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix}^*\begin{bmatrix} x' \\ y' \end{bmatrix} + \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix}\begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix}^*\begin{bmatrix} x \\ y \end{bmatrix} = 0$$

where $S_{11}, Z_{11},$ and T_{11} are the $n_x \times n_x$ matrices, $S_{22}, Z_{22},$ and T_{22} are the $n_y \times n_y$ matrice. We know that in the QZ decomposition, we can arrange the order of generalized eigenvalues by change the columns of Q and Z. We need to order the generalized eigenvalues such that $\lambda_i = \frac{S_{ii}}{T_{ii}}$ satisfies the **Blanchard-Kahn condition**:

$$\left|\frac{S_{11,ii}}{T_{11,ii}}\right| \geqslant 1 \text{ and } \left|\frac{S_{22,jj}}{T_{22,jj}}\right| < 1, \forall i, j$$

This will make the last n_y generalized eigenvalues are stable, which is in the unit circle; first n_x eigenvalues are unstable. The Blanchard-Kahn condition ensure that the converging solution exists and is unique, we will see this in the following.

Next, we can simplify the above equation

$$\begin{aligned} 0 &= \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix}\begin{bmatrix} Z_{11}^* & Z_{21}^* \\ Z_{12}^* & Z_{22}^* \end{bmatrix}\begin{bmatrix} x' \\ y' \end{bmatrix} + \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix}\begin{bmatrix} Z_{11}^* & Z_{21}^* \\ Z_{12}^* & Z_{22}^* \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} \\ &= \begin{bmatrix} S_{11}Z_{11}^* + S_{12}Z_{12}^* & S_{11}Z_{21}^* + S_{12}Z_{22}^* \\ S_{22}Z_{12}^* & S_{22}Z_{22}^* \end{bmatrix}\begin{bmatrix} x' \\ y' \end{bmatrix} + \begin{bmatrix} T_{11}Z_{11}^* + T_{12}Z_{12}^* & T_{11}Z_{21}^* + T_{12}Z_{22}^* \\ T_{22}Z_{12}^* & T_{22}Z_{22}^* \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} \\ &= \begin{bmatrix} (S_{11}Z_{11}^* + S_{12}Z_{12}^*)x' + (S_{11}Z_{21}^* + S_{12}Z_{22}^*)y' \\ S_{22}Z_{12}^*x' + S_{22}Z_{22}^*y' \end{bmatrix} + \begin{bmatrix} (T_{11}Z_{11}^* + T_{12}Z_{12}^*)x + (T_{11}Z_{21}^* + T_{12}Z_{22}^*)y \\ T_{22}Z_{12}^*x + T_{22}Z_{22}^*y \end{bmatrix} \end{aligned}$$

We can write the above as two equations:

$$S_{11}(Z_{11}^*x' + Z_{21}^*y') + S_{12}(Z_{12}^*x' + Z_{22}^*y') = -T_{11}(Z_{11}^*x + Z_{21}^*y) - T_{12}(Z_{12}^*x + Z_{22}^*y)$$

$$Z_{12}^*x' + Z_{22}^*y' = -S_{22}^{-1}T_{22}(Z_{12}^*x + Z_{22}^*y)$$

Noticing that S_{22} and T_{22} are the upper triangular matrix, so we have

$$S_{22}^{-1}T_{22} = \begin{bmatrix} \frac{T_{22,11}}{S_{22,11}} & & * \\ & \ddots & \\ 0 & & \frac{T_{22,n_yn_y}}{S_{22,n_yn_y}} \end{bmatrix}$$

B-K condition $\left|\frac{S_{22,jj}}{T_{22,jj}}\right| < 1$ makes $S_{22}^{-1}T_{22}$ have unstable diagonal entries (out of the unit circel). Then we have $Z_{12}^*x + Z_{22}^*y = 0$, otherwise the system will explode. This is true for any time period, namely, $Z_{12}^*x' + Z_{22}^*y' = 0$.

The above two equations becomes

$$\begin{aligned} S_{11}(Z_{11}^*x' + Z_{21}^*y') = -T_{11}(Z_{11}^*x + Z_{21}^*y) &\Rightarrow (Z_{11}^* + Z_{21}^*g_x)x' = -S_{11}^{-1}T_{11}(Z_{11}^* + Z_{21}^*g_x)x \\ &\Rightarrow x' = (Z_{11}^* + Z_{21}^*g_x)^{-1}S_{11}^{-1}T_{11}(Z_{11}^* + Z_{21}^*g_x)x \\ Z_{12}^*x + Z_{22}^*y = 0 &\Rightarrow y = -(Z_{22}^*)^{-1}Z_{12}^*x \end{aligned}$$

Noticing that another B-K condition $\left|\frac{S_{11,ii}}{T_{11,ii}}\right| \geqslant 1$ will make $S_{11}^{-1}T_{11}$ has stable diagonals and the solution is unique.

Finally, we take the real part as the first order derivatives:

$$\begin{aligned} g_x &= -\text{real}((Z_{22}^*)^{-1}Z_{12}^*) \\ h_x &= -\text{real}\{(Z_{11}^* + Z_{21}^* g_x)^{-1} S_{11}^{-1} T_{11}(Z_{11}^* + Z_{21}^* g_x)\} \end{aligned}$$

3.3 Perturbation solution derivatives

We already have the formula for g_x and h_x, then we can compute the first order state-space model given the parameter θ. Don't forget that g_x and h_x are also depend on the model parameter θ: $g_x = g_x(\theta), h_x = h_x(\theta)$. In order to find the optimal parameter, we need to find the derivative of the perturbation solution.

Firstly, we find the derivative of the DSS. The parameter θ in general will change the DSS values $\bar{x}$, $\bar{y}$. We are interested in $\frac{\partial \bar{x}}{\partial \theta}$ and $\frac{\partial \bar{y}}{\partial \theta}$. Let $F(x;\theta) = \mathbb{E}_t \mathcal{H}(y', y, x', x; \theta)$. We take the total derivatives wrt θ:

$$F_\theta(x;\theta) = \mathcal{H}_x \frac{\partial x}{\partial \theta} + \mathcal{H}_y \frac{\partial y}{\partial \theta} + \mathcal{H}_{x'} \frac{\partial x'}{\partial \theta} + \mathcal{H}_{y'} \frac{\partial y'}{\partial \theta} + \mathcal{H}_\theta = 0$$

where $\mathcal{H}_{\theta;ij} = \frac{\partial \mathcal{H}_i}{\partial \theta_j}, i = 1,\ldots,n, j = 1,\ldots,n_\theta$. We evaluate this function at the DSS $x' = x = \bar{x}, y' = y = \bar{y}$,

$$\begin{bmatrix} \mathcal{H}_y + \mathcal{H}_{y'} & \mathcal{H}_x + \mathcal{H}_{x'} \end{bmatrix} \begin{bmatrix} \frac{\partial \bar{y}}{\partial \theta} \\ \frac{\partial \bar{x}}{\partial \theta} \end{bmatrix} + \mathcal{H}_\theta = 0$$

We can solve this linear equation to get $\frac{\partial \bar{x}}{\partial \theta}$ and $\frac{\partial \bar{y}}{\partial \theta}$.

Secondly, we are interested in $\frac{\partial g_x}{\partial \theta}$ and $\frac{\partial h_x}{\partial \theta}$, both evaluated at the DSS. The first-order model solution satisfies:

$$F_x(x;\theta) = \mathcal{H}_x + \mathcal{H}_y g_x + \mathcal{H}_{x'} h_x + \mathcal{H}_{y'} g_x h_x = 0$$

For each element θ_i of θ, we take derivative of the above equation:

$$F_{x,\theta_i}(x;\theta) = \frac{d\mathcal{H}_x}{d\theta_i} + \frac{d\mathcal{H}_y}{d\theta_i} g_x + \mathcal{H}_y \frac{\partial g_x}{\partial \theta_i} + \frac{d\mathcal{H}_{x'}}{d\theta_i} h_x + \mathcal{H}_{x'} \frac{\partial h_x}{\partial \theta_i} + \frac{d\mathcal{H}_{y'}}{d\theta_i} g_x h_x + \mathcal{H}_{y'} \frac{\partial g_x}{\partial \theta_i} h_x + \mathcal{H}_{y'} g_x \frac{\partial h_x}{\partial \theta_i} = 0$$

We can apply chain rule to compute the total derivatives of $\mathcal{H}$ w.r.t. θ_i

$$\begin{aligned} \left[\frac{d\,\mathcal{H}_x}{d\theta_i}\right]_j^i &= \mathcal{H}_{xy'} \frac{\partial y'}{\partial \theta_i} + \mathcal{H}_{xy} \frac{\partial y}{\partial \theta_i} + \mathcal{H}_{xx'} \frac{\partial x'}{\partial \theta_i} + \mathcal{H}_{xx} \frac{\partial x}{\partial \theta_i} + \frac{\partial \mathcal{H}_x}{\partial \theta_i} \\ \left[\frac{d\,\mathcal{H}_x}{d\theta_i}\right]_j^i &= [\mathcal{H}_{xy'} + \mathcal{H}_{xy}]_{jk}^i \left[\frac{\partial \bar{y}}{\partial \theta_i}\right]_k + [\mathcal{H}_{xx'} + \mathcal{H}_{xx}]_{jk}^i \left[\frac{\partial \bar{x}}{\partial \theta_i}\right]_k + \left[\frac{\partial \mathcal{H}_x}{\partial \theta_i}\right]_j^i \\ \left[\frac{d\,\mathcal{H}_y}{d\theta_i}\right]_j^i &= [\mathcal{H}_{yy'} + \mathcal{H}_{yy}]_{jk}^i \left[\frac{\partial \bar{y}}{\partial \theta_i}\right]_k + [\mathcal{H}_{yx'} + \mathcal{H}_{yx}]_{jk}^i \left[\frac{\partial \bar{x}}{\partial \theta_i}\right]_k + \left[\frac{\partial \mathcal{H}_y}{\partial \theta_i}\right]_j^i \\ \left[\frac{d\,\mathcal{H}_{x'}}{d\theta_i}\right]_j^i &= [\mathcal{H}_{x'y'} + \mathcal{H}_{x'y}]_{jk}^i \left[\frac{\partial \bar{y}}{\partial \theta_i}\right]_k + [\mathcal{H}_{x'x'} + \mathcal{H}_{x'x}]_{jk}^i \left[\frac{\partial \bar{x}}{\partial \theta_i}\right]_k + \left[\frac{\partial \mathcal{H}_{x'}}{\partial \theta_i}\right]_j^i \\ \left[\frac{d\,\mathcal{H}_{y'}}{d\theta_i}\right]_j^i &= [\mathcal{H}_{y'y'} + \mathcal{H}_{y'y}]_{jk}^i \left[\frac{\partial \bar{y}}{\partial \theta_i}\right]_k + [\mathcal{H}_{y'x'} + \mathcal{H}_{y'x}]_{jk}^i \left[\frac{\partial \bar{x}}{\partial \theta_i}\right]_k + \left[\frac{\partial \mathcal{H}_{y'}}{\partial \theta_i}\right]_j^i \end{aligned}$$

here we use the Einstein notation for summation. The second equality holds since we evaluate the derivatives when $x = x' = \bar{x}, y = y' = \bar{y}$.

Thus, for each element θ_i of θ, we stack the unknowns as: $\begin{bmatrix} \frac{\partial g_x}{\partial \theta_i} \\ \frac{\partial h_x}{\partial \theta_i} \end{bmatrix}$ and solve a Sylvester equation:

$$\begin{bmatrix} \frac{d\mathcal{H}_{y'}}{d\theta_i} & \frac{d\mathcal{H}_y}{d\theta_i} & \frac{d\mathcal{H}_{x'}}{d\theta_i} & \frac{d\mathcal{H}_x}{d\theta_i} \end{bmatrix} \begin{bmatrix} g_x h_x \\ g_x \\ h_x \\ I \end{bmatrix} + \begin{bmatrix} \mathcal{H}_y & \mathcal{H}_{x'} + \mathcal{H}_{y'} g_x \end{bmatrix} \begin{bmatrix} \frac{\partial g_x}{\partial \theta_i} \\ \frac{\partial h_x}{\partial \theta_i} \end{bmatrix} + \begin{bmatrix} \mathcal{H}_{y'} & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial g_x}{\partial \theta_i} \\ \frac{\partial h_x}{\partial \theta_i} \end{bmatrix} h_x = 0$$

For the generalized Sylvester equation, there are two methods to solve

Method 1.

$$E + AX + CXD = 0$$

We define vectorization and Kronecker product as:

$$\text{vec}(X) = [x_1^T x_2^T \cdots x_n^T]^T, x_i \text{ is the ith column of } X, \quad A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}$$

We have an identity:

$$\text{vec}(AXB) = (B^T \otimes A)\,\text{vec}(X)$$

Then we can rewrite the Sylverster equation as

$$\begin{aligned} 0 &= \text{vec}(E) + \text{vec}(AX) + \text{vec}(CXD) \\ &= \text{vec}(E) + (I \otimes A)\text{vec}(X) + (D^T \otimes C)\text{vec}(X) \\ \Rightarrow -\text{vec}(E) &= (I \otimes A + D^T \otimes C)\text{vec}(X) \end{aligned}$$

This is a linear equation which can be uniquely solved when $(I \otimes A + D^T \otimes C)$ is nonsingular, and we can derive X from $\text{vec}(X)$.

Method 2.

We do QZ decomposition to A and C and Schur decomposition to D to get

$$Q_1^* A Z = \mathbb{A}, Q_1^* C Z = \mathbb{C}, Q_2^* D Q_2 = \mathbb{D}$$

Then we substitute these to our generalized Sylvester equation

$$\begin{aligned} E + Q_1 \mathbb{A} Z^* X + Q_1 \mathbb{C} Z^* X Q_2 \mathbb{D} Q_2^* &= 0 \\ \Rightarrow \; Q_1^* E Q_2 + \mathbb{A} Z^* X Q_2 + \mathbb{C} Z^* X Q_2 \mathbb{D} &= 0 \\ \Rightarrow \mathbb{A} Y + \mathbb{C} Y \mathbb{D} &= -F \end{aligned}$$

where $Y = Z^* X Q_2, F = Q_1^* E Q_2$. Now we have a new Sylvster equation whose coefficient matrices are all upper triangular, then we can solve the equation iteratively.

We write the equation as elementwise form

$$\begin{aligned} -F &= \mathbb{A}[\, y_1 \;\cdots\; y_k \;\cdots\; y_n \,] + \mathbb{C}[\, y_1 \;\cdots\; y_k \;\cdots\; y_n \,] \begin{bmatrix} d_{11} & & & & d_{1n} \\ & \ddots & & & \\ & & d_{kk} & & \\ & & & \ddots & \\ & & & & d_{nn} \end{bmatrix} \\ &= \begin{bmatrix} \mathbb{A}y_1 + \mathbb{C}y_1 d_{11} & \cdots & \mathbb{A}y_k + \mathbb{C}\sum_{i=1}^{k} d_{ik} y_i & \cdots & \mathbb{A}y_k + \mathbb{C}\sum_{i=1}^{n} d_{in} y_i \end{bmatrix} \end{aligned}$$

Thus, we transform the matrix equation to n linear systems:

$$\begin{aligned} -f_1 &= \mathbb{A}y_1 + \mathbb{C}y_1 d_{11} \\ -f_2 &= \mathbb{A}y_2 + \mathbb{C}(y_1 d_{12} + y_2 d_{22}) \\ &\vdots \\ -f_n &= \mathbb{A}y_n + \mathbb{C}\sum_{i=1}^{n} d_{in} y_i \end{aligned}$$

We can solve the first one to get y_1 and then substitute y_1 to the second equation to get y_2, by this procedure, we can compute y_1 to y_n.

$$k-\text{th linear system: } (\mathbb{A} + d_{kk}\mathbb{C})y_k = -f_k - \mathbb{C}[\ y_1 \ \cdots \ y_{k-1}\ ] \begin{bmatrix} d_{1k} \\ \vdots \\ d_{k-1,k} \end{bmatrix}$$

For most cases, we will choose method 2 since the numerical stability. The Hessian matrices H_{ij}, $i, j = x, y, x', y'$ are sparse, so the resulting matrix $(I \otimes A + D^T \otimes C)$ is nearly singular, for large dimension problem, method 1 is very unstable, the numerical error is large.

3.4 The Kalman Filter

Up to now, we have the linear state space model, we can compute the state and estimate the observable in any time period. There is still a question for us, we don't know the true model paramter, how can we find the optimal parameter given the observations. The strightfordward method is maximum likelihood estimation:

$$\theta^* = \arg\max_{\theta} p(z^T|\theta)$$

where $z^T = \{z_1, \ldots, z_T\}$ is our obseravations in all time periods. We can use many methods to solve this optimization problem. But we need to find the formula for the likelihood firstly. We need Kalman filter to do this.

We can write the likelihood as

$$\begin{aligned} p(z^T|\theta) &= p(z_T|z^{T-1}, \theta)p(z^{T-1}|\theta) \\ &= p(z_1|\theta)\prod_{t=2}^{T} p(z_t|z^{t-1}, \theta) \\ &= \int p(z_1|x_1, \theta)p(x_1|\theta)dx_1 \prod_{t=2}^{T} \int p(z_t|x_t, z^{t-1}, \theta)p(x_t|z^{t-1}, \theta)dx_t \\ &= \int p(z_1|x_1, \theta)p(x_1|\theta)dx_1 \prod_{t=2}^{T} \int p(z_t|x_t, \theta)p(x_t|z^{t-1}, \theta)dx_t \end{aligned}$$

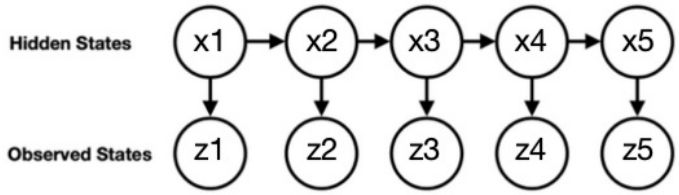
where we used the Markov property of the model:

$$p(z_t|x_t, z^{t-1}, \theta) = p(z_t|x_t, \theta)$$

The obserable z_t only depends on the current state x_t. There is another Markov assumption

$$p(x_{t+1}|x_t, z^t, x^{t-1}, \theta) = p(x_{t+1}|x_t, \theta)$$

The next state x_{t+1} only depends on the current state x_t. We call a model as linear Gaussian state-space models which satisfies the two Markov assumptions. The following is a figure for linear Gaussian state-space model.



Now, we back to our likelihood, $p(x_1|\theta)$ is the prior information which is given, $p(z_t|x_t,\theta)$ is the observation function which is known. The only unknown is perdiction term $p(x_t|z^{t-1},\theta)$, which uses the history obseravations to predict the next time state.

We can write

$$p(x_{t+1}|z^t,\theta) = \int p(x_{t+1}|x_t,\theta)p(x_t|z^t,\theta)$$

where $p(x_{t+1}|x_t,\theta)$ is the state transfer equation which is given, $p(x_t|z^t,\theta)$ is called filtering, which is our desired.

We can use Bayes theorem and Markov property to get

$$p(x_t|z^t,\theta) \propto p(z_t|x_t,\theta)p(x_t|z^{t-1},\theta)$$

Noticing that we can find $p(x_t|z^t,\theta)$ recursively:

Algorithm

Given $p(x_{t-1}|z^{t-1},\theta)$ computed as last time period,

1. Prediction: using last time period filtering to predict next state

$$p(x_t|z^{t-1},\theta) = \int p(x_t|x_{t-1},\theta)p(x_{t-1}|z^{t-1},\theta)$$

2. Correction: using the observation at this time period z_t to correct our prediction

$$p(x_t|z^t,\theta) \propto p(z_t|x_t,\theta)p(x_t|z^{t-1},\theta)$$

This is the general algorithm for filtering, the integral is hard to compute in general. But if our model is linear Gaussian model, we can handle all of the integral, this is Kalman filter's idea.

For linear Gaussian state-space models, we can evaluate the series of posterior distributions of the latent state and the marginal likelihood with the Kalman filter.

The state space model is

$$\hat{y}_t = g_x\hat{x}_t \quad , \quad \hat{x}_{t+1} = h_x\hat{x}_t + \eta\varepsilon_{t+1} \text{ , where } \hat{x} = x - \bar{x}, \hat{y} = y - \bar{y}$$

$$\text{where } g_x = g_x(\theta), h_x = h_x(\theta), \bar{x} = \bar{x}(\theta), \bar{y} = \bar{y}(\theta), \varepsilon \sim N(0,\Sigma)$$

For a general Kalman filter, there are two equations for state x_t and observations z_t

$$\begin{array}{rll} \text{state transfer equation:} & x_t = F_t x_{t-1} + B_t u_t + w_t & w_t \sim N(0,Q) \\ \text{observation equation:} & z_t = H_t x_t + v_t & v_t \sim N(0,R) \end{array}$$

where F_t is the state transition matrix, B_t is the control matrix, u_t is the control variable, w_t is the process noise, H_t is the observation matrix, v_t is the observation noise.

We use the notations:

$$x_{t|k} = \mathbb{E}[p(x_t|z^k)], P_{t|k} = \text{Cov}(x_t|z^k) \Rightarrow x_t|z^k \sim N(x_{t|k}, P_{t|k}), \text{where } k \leqslant t$$

The algorithm has two parts: predict and update:

- Predict:

$$\begin{aligned} x_{t|t-1} &= F_t\, x_{t-1|t-1} + B_t u_t \\ P_{t|t-1} &= F_t P_{t-1|t-1} F_t^T + Q \end{aligned}$$

- Update:

$$\begin{aligned} y_t &= z_t - H_t x_{t|t-1} \\ S_t &= H_t P_{t|t-1} H_t^T + R \\ K_t &= P_{t|t-1} H_t^T S_t^{-1} \\ x_{t|t} &= x_{t|t-1} + K_t y_t \\ P_{t|t} &= (I - K_t H_t) P_{t|t-1} \end{aligned}$$

Now, we can derive the Kalman filter for our state space model, there is no control variable u, and $F_t = h_x, H_t = g_x, Q = \eta\Sigma\eta^T, R = 0$.

$$\begin{aligned} \hat{x}_{t|t-1} &= h_x \hat{x}_{t-1|t-1} \\ P_{t|t-1} &= h_x P_{t-1|t-1} h_x^T + \eta\Sigma\eta^T \\ K_t &= P_{t|t-1} g_x^T (g_x P_{t|t-1} g_x^T)^{-1} \\ \hat{x}_{t|t} &= \hat{x}_{t|t-1} + K_t(\hat{y}_t - g_x\, \hat{x}_{t|t-1}) \\ P_{t|t} &= (I - K_t g_x) P_{t|t-1} \end{aligned}$$

3.5 Calibration of Kalman filter

For state-space representation DSGE model:

Recall that we have three equations in general:

1. State transfer equation: $x_{t+1} = h(x_t; \theta) + \eta\varepsilon_{t+1}$

   where η is the load matrix of exogenous schocks, ε_{t+1} is the exogenous shocks

2. Policy function: $y_t = g(x_t; \theta)$

3. Observation equation: $z_t = q(x_t; \theta) + v_t$

   where v_t is the measurement error

For 1st-order perturbation, we have the following equation:

$$\hat{y}_t = g_x \hat{x}_t \quad , \quad \hat{x}_{t+1} = h_x \hat{x}_t + \eta\varepsilon_{t+1} \text{ , where } \hat{x} = x - \bar{x}, \hat{y} = y - \bar{y}$$

$$\text{where } g_x = g_x(\theta), h_x = h_x(\theta), \bar{x} = \bar{x}(\theta), \bar{y} = \bar{y}(\theta)$$

We already know that we can use Kalman filter to inference the marginal likelihood: $p(\hat{y}^T|\theta)$

But the data is given in z^T, how can we use the data to calibrate our model?

In general, we have:

$$z_t = G\,\hat{x}_t + H\bar{u}, \bar{u} = \begin{bmatrix} \bar{y} \\ \bar{x} \end{bmatrix}, G = Q \begin{bmatrix} g_x \\ I \end{bmatrix}$$

For RBC model, we can find G and H:

$$\begin{aligned}
z_t = \begin{bmatrix} c_{\text{obs}} \\ i_{\text{obs}} \end{bmatrix}_t &= Q\begin{bmatrix} g_x \\ I \end{bmatrix}\hat{x}_t + H\begin{bmatrix} \bar{y} \\ \bar{x} \end{bmatrix} \\
\begin{bmatrix} c_{\text{obs}} \\ i_{\text{obs}} \end{bmatrix}_t &= \begin{bmatrix} c_t - \bar{c} \\ i_t - \bar{i} \end{bmatrix} = \begin{bmatrix} c_t - \bar{c} \\ q_t - \bar{q} \end{bmatrix} - \begin{bmatrix} 0 \\ c_t - \bar{c} \end{bmatrix} \\
\text{Notice that: } \begin{bmatrix} c_t - \bar{c} \\ q_t - \bar{q} \end{bmatrix} &= \hat{y}_t = g_x\begin{bmatrix} k_t - \bar{k} \\ z_t - \bar{z} \end{bmatrix} = g_x\hat{x}_t \\
&= \begin{bmatrix} I & 0 \end{bmatrix}\begin{bmatrix} g_x \\ I \end{bmatrix}\hat{x}_t \\
\begin{bmatrix} 0 \\ c_t - \bar{c} \end{bmatrix} &= \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}\hat{y}_t = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} g_x\hat{x}_t \\
\Rightarrow z_t &= \begin{bmatrix} I & 0 \end{bmatrix}\begin{bmatrix} g_x \\ I \end{bmatrix}\hat{x}_t - \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} g_x\hat{x}_t \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} g_x \\ I \end{bmatrix}\hat{x}_t \\
\Rightarrow Q &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \end{bmatrix} \\
H &= 0
\end{aligned}$$

Thus, we can write:

$$z_t = Ag_x x_t + v_t, v_t \sim N(0, e^2), \text{where } A = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}, e \text{ is the measurement error}$$

Then, we can write the new state space model for observation $z = [c, i]$:

$$\begin{aligned}
&\text{State transfer equation:} \quad \hat{x}_{t+1} = h_x\hat{x}_t + \eta\varepsilon_{t+1} \quad , \varepsilon_t \sim N(0, \Sigma) \\
&\text{Measurement equation:} \quad z_t = Ag_x\hat{x}_t + v_t \quad , v_t \sim N(0, e^2 I)
\end{aligned}$$

Then the Kalman filter algorithm as:

Algorithm

Given $\hat{x}_{0|0}$ and $P_{0|0}$, loop over $t = 1:T$

$$\begin{aligned}
\hat{x}_{t|t-1} &= h_x\hat{x}_{t-1|t-1} \\
P_{t|t-1} &= h_x P_{t-1|t-1}h_x^T + \eta\Sigma\eta^T \\
K_t &= P_{t|t-1}(Ag_x)^T(Ag_x P_{t|t-1}(Ag_x)^T + e^2 I)^{-1} \\
\hat{x}_{t|t} &= x_{t|t-1} + K_t(z_t - Ag_x x_{t|t-1}) \\
P_{t|t} &= (I - K_t Ag_x)P_{t|t-1}
\end{aligned}$$

For the initial condition $\hat{x}_{0|0}$ and $P_0 = P_{0|0}$, we set the mean $\hat{x}_{0|0} = 0$ means that we start from the steady state. For prior covariance matrix P_0, a natural choice is the solution to the discrete time Lyapunov equation:

$$h_x P_0 h_x^T - P_0 + \eta\Sigma\eta^T = 0$$

we can solve this using some existing package such as scipy.linalg.solve_discrete_lyapunov.

4 Bayesian Inference

To estimate the optimal parameters of our DSGE model, we employ Bayesian inference, which combines prior beliefs with the information from data. While methods like Maximum Likelihood Estimation and the Generalized Method of Moments are alternatives, they can struggle with non-linear models and may not efficiently incorporate prior information.

The Bayesian approach is advantageous due to its comprehensive framework for parameter uncertainty and its ability to merge prior knowledge with observed data through the posterior distribution. For practical implementation, we utilize Markov Chain Monte Carlo (MCMC) techniques to draw samples from complex posterior distributions.

Among MCMC methods, Hamiltonian Monte Carlo (HMC) is particularly effective for DSGE models. HMC leverages gradient information to navigate the parameter space more efficiently, which is beneficial given the high dimensionality and potential non-linearity of these models.

In essence, Bayesian methods with MCMC, and specifically HMC, offer a robust approach for parameter estimation in DSGE modeling, providing insights into the underlying economic mechanisms and their uncertainties.

4.1 Marginal likelihood

For Marginal likelihood: $p(\theta|z^T)$, we have

$$\ln p(\theta|z^T) \;=\; \ln p(\theta) + \ln p(z^T|\theta) + C$$

the first part is the log-prior, the second part is the log-likelihood, which can be evaluated by Kalman filter, the third part is the integral constant $C = -\ln p(z^T)$ which can be ignored since it not contain the parameter θ.

Then we can compute the gradient of the log-posterior

$$\nabla_\theta \ln p(\theta|z^T) \;=\; \nabla_\theta \ln p(\theta) + \nabla_\theta \ln p(z^T|\theta)$$

The first part is the gradient of log-prior which is easy to compute since we know the pdf of prior. For the second part, the gradient is hard to evaluate since θ also depends on the perturbation coefficients g_x and h_x. We need to use chain rule to compute the gradient, we write the gradient as elementwise total derivative:

$$\begin{aligned}
\frac{d \ln p(z^T|\theta)}{d\theta_i} &= \frac{\partial \ln p(z^T|\theta)}{\partial g_x}\frac{\partial g_x}{\partial \theta_i} + \frac{\partial \ln p(z^T|\theta)}{\partial h_x}\frac{\partial h_x}{\partial \theta_i} + \frac{\partial \ln p(z^T|\theta)}{\partial P_0}\frac{\partial P_0}{\partial \theta_i} + \frac{\partial \ln p(z^T|\theta)}{\partial \theta_i} \\
&= \left[\frac{\partial \ln p(z^T|\theta)}{\partial g_x}\right]_{ij}\left[\frac{\partial g_x}{\partial \theta_i}\right]_{ij} + \left[\frac{\partial \ln p(z^T|\theta)}{\partial h_x}\right]_{ij}\left[\frac{\partial h_x}{\partial \theta_i}\right]_{ij} + \left[\frac{\partial \ln p(z^T|\theta)}{\partial P_0}\right]_{ij}\left[\frac{\partial P_0}{\partial \theta_i}\right]_{ij}
\end{aligned}$$

where the second equality since the parameter is implicitly in g_x and h_x which leads partial derivative $\frac{\partial \ln p(z^T|\theta)}{\partial \theta}$ is zero and we use Einstein's summation convention again.

For $\frac{\partial \ln p(z^T|\theta)}{\partial g_x}$, $\frac{\partial \ln p(z^T|\theta)}{\partial h_x}$ and $\frac{\partial \ln p(z^T|\theta)}{\partial P_0}$, we can use auto-differential package such as Tensorflow to compute. For $\frac{\partial g_x}{\partial \theta}$ and $\frac{\partial h_x}{\partial \theta}$, we already have the result by solving a generalized Sylvster equation. For $\frac{\partial P_0}{\partial \theta_i}$, we can compute it from its definition, by differentiating the Lyapunov equation, we have

$$h_x \frac{\partial P_0}{\partial \theta_i} h_x^T - \frac{\partial P_0}{\partial \theta_i} + \left(\frac{\partial h_x}{\partial \theta_i} P_0 h_x^T + h_x P_0 \frac{\partial h_x^T}{\partial \theta_i}\right) = 0$$

which is amother Lyapunov equation and we assumen that Σ does'n involve any parameter.

Now, we have the gradient of the log-posterior, then we can use HMC to sample θ.

For RBC problem, our paremeter $\theta = \alpha, \beta, \rho$, and we set the pseudotrue values $\alpha = 0.3, \beta = 0.998$, $\rho = 0.9$. Fixed parameter $\delta = 0.025, \sigma = 0.01$. The state variable $x = [k, z]$, control variable $y = [c, q]$ and observable $z = [c_{\text{obs}}, i_{\text{obs}}]$.

Since true value of $\beta = 0.998$ which is close to 1. We need to transform β as defandine

$$\beta_{\text{draw}} = 100\left(\frac{1}{\beta} - 1\right), \text{then } \beta = \frac{1}{\frac{\beta_{\text{draw}}}{100} + 1}$$

Now we will sample β_{draw} instead of β, we define the new log posterior as

$$\begin{aligned}
\ln p(\alpha, \beta_{\text{draw}}, \rho | y^T) &= \ln p(\alpha, \beta_{\text{draw}}, \rho) + \ln p(y^T | \alpha, \beta(\beta_{\text{draw}}), \rho) + \text{Const.} \\
&= \ln p_\alpha(\alpha) + \ln p_{\beta_{\text{draw}}}(\beta_{\text{draw}}) + \ln p_\rho(\rho) + \ln p\left(y^T \middle| \alpha, \frac{1}{\frac{\beta_{\text{draw}}}{100} + 1}, \rho \right) + \text{Const.}
\end{aligned}$$

When we calculate the gradient of $\ln p(\alpha, \beta_{\text{draw}}, \rho | y^T)$, we have:

$$\nabla_{\theta_i} \ln p(\alpha, \beta_{\text{draw}}, \rho | y^T) = \nabla_{\theta_i} \ln p_\alpha(\alpha) + \nabla_{\theta_i} \ln p\left(y^T \middle| \alpha, \frac{1}{\frac{\beta_{\text{draw}}}{100} + 1}, \rho \right), \text{when } \theta_i = \alpha, \rho$$

$$\begin{aligned}
\nabla_{\beta_{\text{draw}}} \ln p(\alpha, \beta_{\text{draw}}, \rho | y^T) &= \nabla_{\beta_{\text{draw}}} \ln p_{\beta_{\text{draw}}}(\beta_{\text{draw}}) + \nabla_{\beta_{\text{draw}}} \ln p\left(y^T \middle| \alpha, \frac{1}{\frac{\beta_{\text{draw}}}{100} + 1}, \rho \right) \\
&= \nabla_{\beta_{\text{draw}}} \ln p_{\beta_{\text{draw}}}(\beta_{\text{draw}}) + \nabla_\beta \ln p(y^T | \alpha, \beta, \rho) \frac{\partial \beta}{\partial \beta_{\text{draw}}} \\
&= \nabla_{\beta_{\text{draw}}} \ln p_{\beta_{\text{draw}}}(\beta_{\text{draw}}) + \nabla_\beta \ln p(y^T | \alpha, \beta, \rho) \frac{-1}{100\left(\frac{\beta_{\text{draw}}}{100} + 1\right)^2} \\
&= \nabla_{\beta_{\text{draw}}} \ln p_{\beta_{\text{draw}}}(\beta_{\text{draw}}) + \nabla_\beta \ln p(y^T | \alpha, \beta, \rho) \frac{-\beta^2}{100}
\end{aligned}$$

4.2 Joint likelihood

Now, we consider to sample from the joint likelihood $p(\theta, x^T | z^T)$.

$$\begin{aligned}
\ln p(\theta, x^T | z^T) &= \ln p(\theta, x^T) + \ln p(z^T | x^T, \theta) + C \\
&= \ln \left[p(\theta) p(x^T | \theta) \right] + \ln \left[p(z_T | z^{T-1}, x^T, \theta) p(z^{T-1} | x^T, \theta) \right] + C \\
&= \ln p(\theta) + \ln p(x^T | \theta) + \ln p(z_T | x_T, \theta) + \ln p(z^{T-1} | x^T, \theta) + C
\end{aligned}$$

where we used Bayesian theorem in the second equality, and Markov assumption in the second equality. For the last term, we need to prove that z^{T-1} is independent of future time state

$$\begin{aligned}
p(z^{T-1} | x^T, \theta) &= \frac{p(x_T, z^{T-1} | x^{T-1}, \theta)}{p(x_T | x^{T-1}, \theta)} \\
&= \frac{p(x_T | z^{T-1}, x^{T-1}, \theta) p(z^{T-1} | x^{T-1}, \theta)}{p(x_T | x^{T-1}, \theta)} \\
&= \frac{p(x_T | x^{T-1}, \theta) p(z^{T-1} | x^{T-1}, \theta)}{p(x_T | x^{T-1}, \theta)} \\
&= p(z^{T-1} | x^{T-1}, \theta)
\end{aligned}$$

Thus, we have

$$\ln p(z^T|x^T,\theta) = \sum_{t=1}^{T} \ln p(z_t|x_t,\theta)$$

For $\ln p(x^T|\theta)$, we can do the similar decomposition

$$\begin{aligned}
\ln p(x^T|\theta) &= \ln p(x_T|x^{T-1},\theta) + \ln p(x^{T-1}|\theta) \\
&= \sum_{t=1}^{T} \ln p(x_t|x^{t-1},\theta) + \ln p\left(x_0|\theta\right) \\
&= \sum_{t=1}^{T} \ln p(x_t|x_{t-1},\theta) + \ln p\left(x_0|\theta\right)
\end{aligned}$$

Then we can write

$$\ln p(\theta, x^T|z^T) = \ln p\left(\theta\right) + \sum_{t=1}^{T} \ln p(z_t|x_t,\theta) + \sum_{t=1}^{T} \ln p(x_t|x_{t-1},\theta) + \ln p(x_0|\theta) + C$$

We compute the log-posterior not only depends on the data $\{z_t\}_{t=1}^T$, but also depends on the state chain $\{x_t\}_{t=0}^T$. For simplicity, we can use schocks $\{\varepsilon_t\}_{t=1}^T$ and initial state x_0 to express the all the states:

$$\begin{aligned}
\ln p(z_t|x_t,\theta) &= \ln p(z_t|\varepsilon^t, x_0, \theta) \\
\ln p(x_t|x_{t-1},\theta) &= \ln p(\varepsilon_t|\theta)
\end{aligned}$$

Finally, we can write our log-joint-posterior as

$$\ln p(\theta, x^T|z^T) = \ln p(\theta) + \sum_{t=1}^{T} \ln p(z_t|\varepsilon^t, x_0, \theta) + \sum_{t=1}^{T} \ln p(\varepsilon_t|\theta) + \ln p(x_0|\theta) + C$$

Recall our state-space first order DSGE model is

$$x_{t+1} = h_x x_t + \eta\varepsilon_{t+1}, \quad z_t = A g_x x_t + v_t$$

here we don't write hat for simplicity, do remember that our state x_t is the deviation to the steady state, then we have

$$\begin{aligned}
\ln p(z_t|\varepsilon^t, x_0, \theta) &= \ln \mathcal{N}(z_t - A g_x x_t|0,\Omega), \text{where } \Omega = \mathrm{Var}(v_t) \\
\ln p(\varepsilon_t|\theta) &= \ln \mathcal{N}(\eta\varepsilon_t|0, \eta\Sigma\eta^T), \text{where } \Sigma = \mathrm{Var}(\varepsilon_t) \\
\ln p(x_0|\theta) &= 0 \quad \text{since we set } x_0 = 0
\end{aligned}$$

where the pdf of multivariate normal is

$$\ln \mathcal{N}(x|0,\Sigma) = -\frac{1}{2}(k\ln 2\pi + \ln|\Sigma|) - \frac{1}{2}x^T\Sigma^{-1}x$$

Therefore, we can write the joint likelihood as a function of θ and ε^T given z^T:

$$\ln p(\theta, \varepsilon^T|z^T) = \ln p(\theta) + \sum_{t=1}^{T} \ln \mathcal{N}(z_t - A g_x x_t|0,\Omega) + \sum_{t=1}^{T} \ln \mathcal{N}(\eta\varepsilon_t|0, \eta\Sigma\eta^T) + C$$

This means that the joint likelihood computation is filter-free, as we do not have to infer the latent state series x^T or threir posterior, but rather retrive a series of x^T from the shock ε^T.

Consequently, insteady of sampling just θ to compute a marginal likelihood $\ln p(\theta|z^T)$, we will sample θ and ε^T to compute the joint likelihood $\ln p(\theta, x^T|z^T)$, which is equivalent to $\ln p(\theta, \varepsilon^T|z^T)$. The marginal likelihood can be obtained from the joint likelihood by integrating over ε^T:

$$p(\theta|z^T) = \int p(\theta, x^T|z^T)dx^T = \int p(\theta, \varepsilon^T|z^T)d\varepsilon^T$$

Then we can write an algorithm to compute the joint posterior

Algorithm

Given z^T and ε^T, set $L = \ln p(\theta), x_0 = 0$, loop over $t = 1:T$

$$\begin{aligned} x_t &= h_x x_{t-1} + \eta\varepsilon_t, \\ L &= L + \ln\mathcal{N}(z_t - A g_x x_t|0, \Omega) + \ln\mathcal{N}(\eta\varepsilon_t|0, \eta\Sigma\eta^T) \end{aligned}$$

As for the gradient, now we need to calculate the gradient to θ and ε^T. Firstly, we compute the gradient to θ, we can use chain rule to compute like we did in marginal likelihood, noticing that θ only be implicit in the perturabtion coefficients g_x and h_x, so we can write

$$\begin{aligned} \nabla_\theta \ln p(\theta, \varepsilon^T|z^T) &= \nabla_\theta \ln p(\theta) + \nabla_\theta \sum_{t=1}^{T} \ln\mathcal{N}(z_t - A g_x x_t|0, \Omega) + \nabla_\theta \sum_{t=1}^{T} \ln\mathcal{N}(\eta\varepsilon_t|0, \eta\Sigma\eta^T) \\ &= \nabla_\theta \ln p(\theta) + \nabla_\theta \ln p(z^T|\varepsilon^T, \theta) + 0 \end{aligned}$$

For the second term $\nabla_\theta \ln p(z^T|\varepsilon^T, \theta)$, we have

$$\begin{aligned} \frac{d \ln p(z^T|\varepsilon^T, \theta)}{d\theta_i} &= \frac{\partial \ln p(z^T|\varepsilon^T, \theta)}{\partial g_x}\frac{\partial g_x}{\partial \theta_i} + \frac{\partial \ln p(z^T|\varepsilon^T, \theta)}{\partial h_x}\frac{\partial h_x}{\partial \theta_i} + \frac{\partial \ln p(z^T|\varepsilon^T, \theta)}{\partial \theta_i} \\ &= \left[\frac{\partial \ln p(z^T|\varepsilon^T, \theta)}{\partial g_x}\right]_{ij}\left[\frac{\partial g_x}{\partial \theta_i}\right]_{ij} + \left[\frac{\partial \ln p(z^T|\varepsilon^T, \theta)}{\partial h_x}\right]_{ij}\left[\frac{\partial h_x}{\partial \theta_i}\right]_{ij} \end{aligned}$$

where the second equality since the parameter is implicitly in g_x and h_x which leads partial derivative $\frac{\partial \ln p(z^T|\varepsilon^T, \theta)}{\partial \theta}$ is zero and we use Einstein's summation convention again.

For $\frac{\partial \ln p(z^T|\varepsilon^T, \theta)}{\partial g_x}, \frac{\partial \ln p(z^T|\varepsilon^T, \theta)}{\partial h_x}$, we can use auto-differential package such as Tensorflow to compute. For $\frac{\partial g_x}{\partial \theta}$ and $\frac{\partial h_x}{\partial \theta}$, we already have the result by solving a generalized Sylvster equation.

Secondly we compute the gradient to ε^T, since g_x and h_x are independent of ε^T, we can use auto-differentiation to compute the gradient here.

4.3 Markov Chain Monte Carlo (MCMC)

Up to now, we have computed the posterior $p(\theta|z^T)$. z^T is the observed data we can obtain from the real ecomony, θ is the parameter to control our DSGE model. We can use MAP(maximum a posterior) to find the optimal θ,

$$\hat{\theta} = \arg\max_\theta p(\theta|z^T)$$

For the complex posterior, we use MCMC to sample parameter from it. A well-known MCMC algorithm is Metropolis-Hastings algorithm:

Algorithm

- **Step 0 (Initialization):** Set $i = 0$ and an initial θ_i. Solve the model for θ_i and build the state-space representation. Evaluate $p(\theta_i)$ and $p(z^T|\theta_i)$. Set $i = i + 1$.

- **Step 1 (proposal draw):** Get a draw θ_i^* from a proposal density $q(\theta_i^*|\theta_{i-1})$.
- **Step 2 (solving the model):** Solve the model for θ_i^* and build the state-space representation.
- **Step 3 (evaluating the proposal):** Evaluate $p(\theta_i^*)$ and $p(z^T|\theta_i^*)$.
- **Step 4 (accept/reject):** Draw $u_i \sim U(0,1)$, if $u_i \leqslant \frac{p(z^T|\theta_i^*)p(\theta_i^*)q(\theta_{i-1},\theta_i^*)}{p(z^T|\theta_{i-1})p(\theta_{i-1})q(\theta_i^*,\theta_{i-1})}$, set $\theta_i = \theta_i^*$; otherwise set $\theta_i = \theta_{i-1}$
- **Step 5 (iteration):** If $i < I$, set $i = i+1$ and go to step 1. Otherwise stop.

For the posal density $q()$, can be a random walk:

$$\theta_i^* = \theta_{i-2} + ck_i, k_i \sim N(0, \Sigma_k)$$

We call this algorithm as Random Walk Metropolis Hastings(RWMH) algorithm.

4.4 Hamiltonian Monte Carlo (HMC)

Definition. *For $\varepsilon > 0$ and $\forall I$, we define the typical set A_ε^I with respect to target posterior $p(\theta|z^T)$ as*

$$\left\{(\theta_0,\ldots,\theta_I): \left| -\frac{1}{I+1}\sum_{i=0}^{I} \log p(\theta_i|z^T) - b(\theta) \right| \leqslant \varepsilon \right\}$$

where $b(\theta) = -\int p(\theta_i|z^T)\log p(\theta_i|z^T)d\theta_i$ is the differnential entropy of the parameter w.r.t. posterior

HMC use the extra information: gradient of posterior(velocity) $\nabla_\theta p(\theta|z^T)$ to find a chain to sample.

Let $\theta \in \mathbb{R}^d$, with an auxiliray momentum vector $p \in \mathbb{R}^d$

$$p \sim N(0, M)$$

The Hamiltonian of the system is

$$\mathcal{H}(\theta, \boldsymbol{p}) = -\log p(\theta|z^T) + \frac{1}{2}\log((2\pi)^d|M|) + \frac{1}{2}\boldsymbol{p}^T M^{-1}\boldsymbol{p}$$

where the first term is the analog of a potential energy, the second term is a scaled mass matrix, the third term is the analog of a kintic energy. We want to sample a Markov chain $\{\theta_i, \boldsymbol{p}_i\}_{i=1}^I$ drown from the Hamiltonian system.

Algorithm

- **Step 1 (Initialization):** Draw $\boldsymbol{p}_{i+1} \sim p(\boldsymbol{p}_{i+1}|\theta_i) = \mathcal{N}(0, M)$
- **Step 2 (solving the Hamiltonian equation):** Draw $\theta_{i+1} \sim p(\theta_{i+1}|\boldsymbol{p}_{i+1})$. We set initial condition as $\boldsymbol{p}(0) = \boldsymbol{p}_{i+1}, \theta(0) = \theta_i$ and run the Leap-forg numerical scheme to solve the Hamiltonian equation L steps to get $(\theta^*, \boldsymbol{p}^*)$

$$\begin{aligned}
\boldsymbol{p}\Big(\tau + \frac{\varepsilon}{2}\Big) &= p(\tau) + \varepsilon\nabla_\theta \log p(\theta(\tau)|z^T) \\
\theta(\tau+\varepsilon) &= \theta(\tau) + \varepsilon M^{-1}\boldsymbol{p}\Big(\tau + \frac{\varepsilon}{2}\Big) \\
\boldsymbol{p}(\tau+\varepsilon) &= \boldsymbol{p}\Big(\tau + \frac{\varepsilon}{2}\Big) + \varepsilon\nabla_\theta \log(\theta(\tau+\varepsilon))/2
\end{aligned}$$

- **Step 3 (Metreopolis-Hastings step):** Accept $(\theta^*, \boldsymbol{p}^*)$ with

$$\text{prob} = \min\left(1, \exp(\mathcal{H}(\theta_i, \boldsymbol{p}_{i+1}) - \mathcal{H}(\theta^*, \boldsymbol{p}^*))\right)$$

Q: Why we use the difference of Hamiltonian to determine whether accept the new sample or not?

A: Noticing that

$$\begin{aligned}\exp(-\mathcal{H}(\theta,\boldsymbol{p})) &= \frac{p(\theta|z^T)}{(2\pi)^{\frac{d}{2}}|M|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}\boldsymbol{p}^T M^{-1}\boldsymbol{p}\right)\\ \Rightarrow\ p(\theta,\boldsymbol{p}) &= p(\theta|z^T)\mathcal{N}(P|0,M)\end{aligned}$$

this means that $\exp(-\mathcal{H}(\theta,\boldsymbol{p}))$ is the joint distribution of $(\theta,\boldsymbol{p})$, so we can view

$$\exp(\mathcal{H}(\theta_i,\boldsymbol{p}_{i+1})-\mathcal{H}(\theta^*,\boldsymbol{p}^*))=\frac{p(\theta^*,\boldsymbol{p}^*)}{p(\theta_i,\boldsymbol{p}_{i+1})}$$

as the acceptance rate of MH.

Remark. $\exp(\mathcal{H}(\theta_i,\boldsymbol{p}_{i+1})-\mathcal{H}(\theta^*,\boldsymbol{p}^*))$ should be 0 since the conservation of energy, but numerical scheme always has error so that $\mathcal{H}$ is not conservative.

5 Recursive utility

This section is just a review of *Computing DSGE Models with Recursive Preferences and Stochastic Volatility*.

5.1 Recursive Utility

We start from the classical CRRA(Constant Relative Risk Aversion) utility function

$$U(c_t)=\begin{cases}\frac{c_t^{1-\gamma}}{1-\gamma}, & \text{for } \gamma\neq 1\\ \log(c_t), & \text{for } \gamma=1\end{cases}$$

where γ is the coefficient of risk aversion, $\gamma>1$ represents risk aversion, $\gamma<1$ represents risk preference, $\gamma=1$ represents risk neutral.

The utility can be extend to incorporate labor supply

$$U(c_t,l_t)=\frac{c_t^{1-\gamma}}{1-\gamma}-\phi\frac{l_t^{1+\eta}}{1+\eta}$$

where l_t is the labor supplied by the individual at time t that represents disutility or the cost of work in terms of leisure foregone; η is a parameter that represents the elasticity of labor supply, also known as the Frisch elasticity; ϕ is a parameter reflecting the disutility of labor. It scales the "cost" of labor in the utility function.

There is another version of utility written as Cobb-Douglas utility:

$$U(c_t,l_t)=(c_t^v(1-l_t)^{1-v})^{1-\gamma}$$

where v is the weight given to consumption, $1-v$ is the weight given to leisure.

The first recursive utility formulation has a separate term for the disutility of labor, while the second combines consumption and leisure in complementarity within the same term. The first formulation suggests labor directly reduces utility, whereas the second suggests a trade-off between consumption and leisure.

Now, let's move to the recursive utility formulations. Recursive utility models allow for intertemporal consistency in the consumer's decision-making by incorporating future utility into the present decision framework, we use the Cobb-Douglas form to write the recursive utility as:

$$U_t = \max_{c_t, l_t} \Big[(1-\beta)(c_t^v (1-l_t)^{1-v})^{\frac{1-\gamma}{\theta}} + \beta (\mathbb{E}_t U_{t+1}^{1-\gamma})^{\frac{1}{\theta}} \Big]^{\frac{\theta}{1-\gamma}}$$

where β is the discount factor, $\theta = \frac{1-\gamma}{1-1/\psi}$, ψ is the EIS(Elasticity of Intertemporal Substitution) which measures the willingness of an individual to substitute consumption over time in response to changes in the interest rate. A higher ψ means that the individual is more willing to change their consumption pattern due to interest rate fluctuations.

The parameter θ is an index of the deviation with respect to the benchmark CRRA utility function(when $\theta = 1$, we are back in that CRRA case where the inverse of the EIS and risk aversion coincide).

The household's budget constraint is given by

$$c_t + i_t + \frac{b_{t+1}}{R_t^f} = w_t l_t + r_t k_t + b_t$$

where i_t is investment, R_t^f is the risk-free gross interest rate, b_t is the holding of an uncontingent bond that pay 1 unit of consumption good at time $t+1$, w_t is the wage, l_t is labor, r_t is the rental rate of capital, and k_t is capital.

Households accumulate capital according to the law of motion

$$k_{t+1} = (1-\delta)k_t + i_t$$

The output and productivity level as follows

$$y_t = e^{z_t} k_t^{\zeta} l_t^{1-\zeta}, \;\; z_t = \lambda z_{t-1} + e^{\sigma_t} \varepsilon_t, \varepsilon_t \sim N(0,1)$$

The innovation ε_t is scaled by a SV level σ_t, which evolevs as:

$$\sigma_t = (1-\rho)\bar{\sigma} + \rho \sigma_{t-1} + \eta \omega_t, w_t \sim \mathcal{N}(0,1)$$

where $\bar{\sigma}$ is the unconditional mean level of σ_t, ρ is the persistence of the processes, and η is the std of the innovations to σ_t.

We can write the value function representation of the social planner's problem in terms of its three state variables

$$V(k_t, z_t, \sigma_t) = \max_{c_t, l_t} \Big[(1-\beta)(c_t^v (1-l_t)^{1-v})^{\frac{1-\gamma}{\theta}} + \beta (\mathbb{E}_t V^{1-\gamma}(k_{t+1}, z_{t+1}, \sigma_{t+1}))^{\frac{1}{\theta}} \Big]^{\frac{\theta}{1-\gamma}}$$

$$s.t. \quad c_t + k_{t+1} = e^{z_t} k_t^{\zeta} l_t^{1-\zeta} + (1-\delta)k_t$$

$$z_t = \lambda z_{t-1} + e^{\sigma_t} \varepsilon_t, \;\; \varepsilon_t \sim \mathcal{N}(0,1)$$

$$\sigma_t = (1-\rho)\bar{\sigma} + \rho \sigma_{t-1} + \eta w_t, \;\; w_t \sim \mathcal{N}(0,1)$$

Then, we can find the pricing kernel of the economy

$$m_{t+1} = \frac{\partial V_t / \partial c_{t+1}}{\partial V_t / \partial c_t}$$

Now, note that:

$$\frac{\partial V_t}{\partial c_t} = (1-\beta) V_t^{1-\frac{1-\gamma}{\theta}} v \frac{(c_t^{\gamma}(1-l_t)^{1-v})^{\frac{1-\gamma}{\theta}}}{c_t}$$

and we can define

$$m_{t+1}\frac{\partial V_t/\partial c_{t+1}}{\partial V_t/\partial c_t}=\beta\left(\frac{c_{t+1}}{c_t}\right)^{\frac{v(1-\gamma)}{\theta}-1}\left(\frac{1-l_{t+1}}{1-l_t}\right)^{\frac{(1-\gamma)(1-v)}{\theta}}\left(\frac{V_{t+1}^{1-\gamma}}{\mathbb{E}_t V_{t+1}^{1-\gamma}}\right)^{1-\frac{1}{\theta}}$$

This equation shows how the pricing kernel is a§ected by the presence of recursive preferences.

If $\theta=1,$ the last term is equal to 1 and we get back the pricing kernel of the standard CRRA case.

5.2 Perturbation

First, we rewrite the exogenous processes in terms of a perturbation parameter χ,

$$\begin{aligned} z_t &= \lambda z_{t-1}+\chi e^{\sigma_t}\varepsilon_t \\ \sigma_t &= (1-\rho)\bar{\sigma}+\rho\sigma_{t-1}+\chi\eta\omega_t \end{aligned}$$

When $\chi=1$, which is just a normalization, we are dealing with the stochastic version of the model. When $\chi=0$, we are dealing with the deterministic case with steady state $k_{ss}, z_{ss}=0,$ and $\sigma_{ss}=\bar{\sigma}$.

For convenience, we define

$$s_t=(k_t-k_{ss}, z_{t-1}, \varepsilon_t, \sigma_{t-1}-\sigma_{ss}, \omega_t, \chi)$$

Then we can write the social planner's value function, $V(s_t)$, and the decision rules for consumption, $c(s_t)$, investment, $i(s_t)$, capital, $k(s_t)$, and labor, $l(s_t)$, as a function of s_t.

Second, the second-order Taylor approximation of the value function around $s_t=0$ is

$$V(s_t)\approx V_{ss}+V_{i,ss}s_t^i+\frac{1}{2}V_{ij,ss}s_t^i s_t^j$$

where:

1. $V_{ss}=V(0), V_{i,ss}=V_i(0),$ and $V_{ij,ss}=V_{ij}(0)$ are the zeroth to second order derivatives evaluated at steady state.

2. We use the tensor $V_{i,ss}s_t^i=\sum_{i=1}^{6}V_{i,ss}s_{i,t}$ follows Einstein's summation convention

Noticing that the steady-state value function is just $V_{ss}=c_{ss}^v(1-l_{ss})^{1-v}$, which implies that:

$$c_{ss}^v(1-l_{ss})^{1-v}+\frac{1}{2}V_{66,ss}=((1-\tau)c_{ss})^v(1-l_{ss})^{1-v}\Rightarrow V_{ss}+\frac{1}{2}V_{66,ss}=(1-\tau)^v V_{ss}$$

Then:

$$\tau=1-\left[1+\frac{1}{2}\frac{V_{66,ss}}{V_{ss}}\right]^{\frac{1}{v}}$$

The second-order approximation of the decision rule for the consumption is, under differentiability assumptions:

$$c(s_t)\approx c_{ss}+c_{i,ss}s_t^i+\frac{1}{2}c_{ij,ss}s_t^i s_t^j$$

following the same steps, we can derive the decision rules for labor, investment, and capital.

- ECP: Equilibrium conditions perturbation

- VFP: Value function perturbation

We take the first-order conditions of the social planner. First, with respect to consumption today:

$$\frac{\partial V_t}{\partial c_t} - \mu_t = 0$$

where μ_t is the Lagrangian multiplier associated with the resource constraint. Second, with erspect to capital:

$$-\mu_t + \mathbb{E}_t \mu_{t+1}(\zeta e^{z_{t+1}} k_{t+1}^{\zeta-1} l_{t+1}^{1-\zeta} + 1 - \delta) = 0$$

Third, with respecct to labor:

$$\frac{1-v}{v}\frac{c_t}{1-l_t} = (1-\zeta)e^{z_t} k_t^{\zeta} l_t^{1-\zeta}$$

Then, we have

$$\mathbb{E}_t m_{t+1}(\zeta e^{z_t} k_{t+1}^{\zeta-1} l_{t+1}^{1-\zeta} + 1 - \delta) = 1$$

We then substitute for the pricing kernel to get the augmented equilibrium conditions:

$$V_t - \left[(1-\beta)(c_t^v(1-l_t)^{1-v})^{\frac{1-\gamma}{\theta}} + \beta(\mathbb{E}_t V^{1-\gamma}(k_{t+1}, z_{t+1}))^{\frac{1}{\theta}}\right]^{\frac{\theta}{1-\gamma}} = 0$$

$$\mathbb{E}_t\left[\beta\left(\frac{c_{t+1}}{c_t}\right)^{\frac{1-\gamma}{\theta}-1}\left(\frac{V_{t+1}^{1-\gamma}}{\mathbb{E}_t V_{t+1}^{1-\gamma}}\right)^{1-\frac{1}{\theta}}(\zeta e^{z_t} k_{t+1}^{\zeta-1} l_{t+1}^{1-\zeta} + 1 - \delta)\right] - 1 = 0$$

$$\frac{1-v}{v}\frac{c_t}{1-l_t} - (1-\zeta)e^{z_t} k_t^{\zeta} l_t^{1-\zeta} = 0$$

$$c_t + i_t - e^{z_t} k_t^{\zeta} l_t^{1-\zeta} = 0$$

$$k_{t+1} - i_t - (1-\delta)k_t = 0$$

In steady state, $m_{ss} = \beta$ and the set of equilibrium conditions simplifies to:

$$V_{ss} = c_{ss}^v (1-l_{ss})^{1-v}$$

$$\zeta k_{ss}^{\zeta-1} l_{ss}^{1-\zeta} + 1 - \delta = \frac{1}{\beta}$$

$$\frac{1-v}{v}\frac{c_{ss}}{1-l_{ss}} = (1-\zeta)k_{ss}^{\zeta} l_{ss}^{-\zeta}$$

$$R_{ss}^f = \frac{1}{\beta}$$

$$c_{ss} + i_{ss} = k_{ss}^{\zeta} l_{ss}^{1-\zeta}$$

$$i_{ss} = \delta k_{ss}$$

The six unknowns steady state can be derived from the six equations.

First, noticing that:

$$\frac{k_{ss}}{l_{ss}} = \left(\frac{1}{\zeta}\left(\frac{1}{\beta} - 1 + \delta\right)\right)^{\frac{1}{\zeta-1}} = \Omega$$

Now, from the leisure-consumption condition:

$$\frac{c_{ss}}{1-l_{ss}} = \frac{v}{1-v}(1-\zeta)\Omega^{\zeta} = \Phi \;\Rightarrow\; c_{ss} = \Phi(1-l_{ss})$$

Then we have

$$c_{ss} + \delta k_{ss} = k_{ss}^{\zeta} l_{ss}^{1-\zeta} = \Omega^{\zeta} l_{ss} \;\Rightarrow\; c_{ss} = (\Omega^{\zeta} - \delta\Omega) l_{ss}$$

and

$$\begin{aligned} \Phi(1 - l_{ss}) &= (\Omega^{\zeta} - \delta\Omega) l_{ss} \\ \Rightarrow\; l_{ss} &= \frac{\Phi}{\Omega^{\zeta} - \delta\Omega + \Phi} \\ k_{ss} &= \frac{\Phi\Omega}{\Omega^{\zeta} - \delta\Omega + \Phi} \end{aligned}$$

This steady state is identical to the steady state of the real business cycle model with a standard CRRA utility function and no SV.

Then we can use the standary second order perturbation to get the quadratic solution.

6 Deep Learning Method

This section is just a review of *Deep learning for solving dynamic economic models*.

6.1 Framework

- Optimization Problem

An exogenous state $m_{t+1} \in \mathbb{R}^{n_m}$ follows a Markov process by an i.i.d. innovation process $\varepsilon_t \in \mathbb{R}^m$ with a transition function M,

$$m_{t+1} = M(m_t, \varepsilon_t)$$

An endogenous state s_{t+1} is driven by the exogenous state m_t and controlled by a choice $x_t \in \mathbb{R}^{n_x}$ according to a transition function S,

$$s_{t+1} = S(m_t, s_t, x_t, m_{t+1})$$

The control x_t saisfies the constraint in the form

$$x_t \in X(m_t, s_t)$$

The state (m_t, s_t) and choice x_t determine the period reward $r(m_t, s_t, x_t)$. The agent maximizes discounted lifetime reward

$$\max_{\{x_t, S_{t+1}\}_{t=0}^{\infty}} \mathbb{E}_0 \left[\sum_{t=0}^{\infty} \beta^t r(m_t, s_t, x_t) \right]$$

where $\beta \in [0, 1)$ is the discount factor and $\mathbb{E}_0[\cdot]$ is an expectation function across future shocks ε_1, $\varepsilon_2, \ldots$ conditional on the initial state (m_0, s_0).

- Decision rule

An optimal decision rule is a function $\varphi: \mathbb{R}^{n_m} \times \mathbb{R}^{n_s} \to \mathbb{R}^{n_x}$ such that $x_t = \varphi(m_t, s_t) \in X(m_t, s_t)$ for all t and the sequence $\{x_t, s_{t+1}\}_{t=0}^{\infty}$ maximizes the lifetime reward for any initial condition (m_0, s_0)

A parametric decision rule is a member of a family of functions $\varphi(\cdot; \theta)$ parameterized by a real vector $\theta \in \Theta$ such that for each θ, we have $\varphi: \mathbb{R}^{n_m} \times \mathbb{R}^{n_s} \to \mathbb{R}^{n_x}$ and $x_t = \varphi(m_t, s_t)$ for all t.

- Value function

Given distribution of shocks $(\varepsilon_1, \ldots, \varepsilon_T)$, value function is define as

$$V(m_0, s_0) = \max_{\{x_t, S_{t+1}\}_{t=0}^{\infty}} \mathbb{E}_{(\varepsilon_1, \ldots, \varepsilon_T)} \left[\sum_{t=0}^{\infty} \beta^t r(m_t, s_t, x_t) \right]$$

For numerical approximation of V, we replace the infinite time problem by a finite time problem and define

$$V^T(m_0, s_0; \theta) = \mathbb{E}_{(\varepsilon_1, \ldots, \varepsilon_T)} \left[\sum_{t=0}^{T} \beta^t r(m_t, s_t, \varphi(m_t, s_t; \theta)) \right]$$

We can find the optimal decision rule by searching θ that maximizes V^T.

For the initial condition (m_0, s_0), it's hard to determine the optimal which maximizes V^T, so we use expectation to define the objective function and can use Monte Carlo to compute

$$\Theta(\theta) = \mathbb{E}_{(m_0, s_0)} \left\{ \mathbb{E}_{(\varepsilon_1, \ldots, \varepsilon_T)} \left[\sum_{t=0}^{T} \beta^t r(m_t, s_t, \varphi(m_t, s_t; \theta)) \right] \right\}$$

This is a nested expectation, which is costly. So we use all-in-one (AiO) expectation operator that combines two expectations into one.

$$\Theta(\theta) = \mathbb{E}_{(m_0, s_0, \varepsilon_1, \ldots, \varepsilon_T)} \left[\sum_{t=0}^{T} \beta^t r(m_t, s_t, \varphi(m_t, s_t; \theta)) \right]$$

- Euler equation

Euler equations are a set of equations written as

$$\mathbb{E}_\varepsilon[f_j(m, s, x, m', s', x')] = 0, j = 1, \ldots, J$$

Euler-residual minimization problem is given by

$$\Theta(\theta) = \mathbb{E}_{(m,s)} \left\{ \sum_{j=1}^{J} v_j (\mathbb{E}_\varepsilon[f_j(m, s, \varphi(m, s; \theta), m', s', \varphi(m', s'; \theta))])^2 \right\}$$

where v_j is the weight on j-th optimality condition.

Now, we cannot directly combine two expectations since there is a squre for $\mathbb{E}_\varepsilon$, we use two independent random draws. We have independent expectation formula:

$$\mathbb{E}_{\varepsilon_1}[f(\varepsilon_1)]\mathbb{E}_{\varepsilon_2}[f(\varepsilon_2)] = \mathbb{E}_{(\varepsilon_1, \varepsilon_2)}[f(\varepsilon_1) f(\varepsilon_2)]$$

Then we can write our Euler-residual as

$$\Theta(\theta) = \mathbb{E}_{(m,s,\varepsilon_1,\varepsilon_2)} \left\{ \sum_{j=1}^{J} v_j [f_j(m, s, x, m', s', x')|_{\varepsilon=\varepsilon_1}][f_j(m, s, x, m', s', x')|\varepsilon = \varepsilon_2] \right\}$$

- Bellman equation

Value function satisfies

$$V(m, s) = \max_{x, s'} \{r(m, s, x) + \beta \mathbb{E}_\varepsilon[V(m', s')]\}$$

Similar to Euler equation method, we can write the optimization as

$$\Theta(\theta) = \mathbb{E}_{(m,s)}\Big[V(m,s) - \max_{x,s'} \{r(m,s,x) + \beta \mathbb{E}_{\varepsilon}[V(m',s')]\} \Big]^2$$

There are three approaches to deal with the maximum operator:

1. FOC: $r_x(m,s,x) + \beta\{\mathbb{E}_{\varepsilon}[V_{s'}(m',s')]\}\frac{\partial s'}{\partial x} = 0$
2. Envelop condition: $r_s(m,s,x) = V_s(m,s)$
3. Direct optimization: $\max_{x,s'}\{r(m,s,x) + \beta\mathbb{E}_{\varepsilon}[V(m',s')]\}$

Here wo focus on the FOC but the other two conditions can be treated in a similar way.

Parametrize value function $V(\cdot;\theta_1)$ and decision rule $x = \varphi(\cdot;\theta_2)$. For given $\theta = (\theta_1, \theta_2)$, Bellman-residual minimization problem is given by

$$\Theta(\theta) = \mathbb{E}_{(m,s)}\{V(m,s;\theta_1) - r(m,s,x) - \beta\mathbb{E}_{\varepsilon}[V(m',s';\theta_1)\}^2 + v\mathbb{E}_{(m,s)}\Big\{ r_x(m,s,x) + \beta\{\mathbb{E}_{\varepsilon}[V_{s'}(m',s';\theta_1)]\}\frac{\partial s'}{\partial x}\Big\}^2$$

where $v > 0$ is a vector of exogenous relative weights of equations in the two objectives.

Define the random variable $\omega = (m, s, \varepsilon_1, \varepsilon_2)$, we can write the above in AiO form:

$$\begin{aligned}
\Theta(\theta) &= \mathbb{E}_w[\xi(\omega;\theta)] \\
&= \mathbb{E}_{(m,s,\varepsilon_1,\varepsilon_2)}\{[V(m,s;\theta_1) - r(m,s,x) - \beta V(m',s';\theta_1)|_{\varepsilon=\varepsilon_1}] \\
&\quad \times [V(m,s;\theta_1) - r(m,s,x) - \beta V(m',s';\theta_1)|_{\varepsilon=\varepsilon_1}\} \\
&\quad + v\Big[r_x(m,s,x) + \beta V_{s'}(m',s';\theta_1)|_{\varepsilon=\varepsilon_1}\frac{\partial s'}{\partial x}\Big]\Big[r_x(m,s,x) + \beta V_{s'}(m',s';\theta_1)|_{\varepsilon=\varepsilon_2}\frac{\partial s'}{\partial x}\Big]
\end{aligned}$$

6.2 Deep learning solution method

We can represent our problem as a optimization as

$$\min_{\theta}\Theta(\theta) = \min_{\theta}\mathbb{E}_w[\xi(\omega;\theta)]$$

In Bayesian framework, we trate ω as random variable, and use the empirical risk

$$\min_{\theta}\Theta^n(\theta) = \min_{\theta}\frac{1}{n}\sum_{i=1}^{n}\xi(\omega_i;\theta)$$

Key idea: Using neural network to approximate the decision rules and value function.

Then we can train a neural network with $\Theta(\theta)$ as the loss function.

References

1. Lecture notes, Economics 808: Macroeconomic Theory, Fall 2004, *Brian Krauth*, SFU

2. Lecture notes, ECON-210C: Business Cycles, Spring 2019, *Johannes Wieland*, UCSD

3. Real Business Cycle Models, *Daniel Vernazza*

4. Lecture notes, Quantitative Macro with Machine Learning, Summer 2023, *Zhigang Feng*, University of Nebraska - Omaha

5. Linear Rational Expectation Model, *Weijie Chen*, University of Helsinki

6. Differentiable State-Space Models and Hamiltonian Monte Carlo Estimation, *David Childers, Jesus Fernandez-Villaverde, etc.*

7. Estimating DSGE Models: Recent Advances and Future Challenges, *Annual Review of Economics, Jesus Fernandez-Villaverde, etc*

8. Solution and Estimation Methods for DSGE Models, *Jesus Fernandez-Villaverde, etc*

9. An Introduction to the Kalman Filter, *Gary Bishop, etc*

10. Eigenvalue and Generalized Eigenvalue Problems: Tutorial, *Benyamin Ghojogh, etc*

11. Using the generalized Schur form to solve a multivariate linear rational expectations model, *Journal of Economic Dynamics & Control, Paul Klein*

12. Computational Methods for Linear Matrix Equations, *V. SIMONCINI*

13. A Conceptual Introduction to Hamiltonian Monte Carlo, *Michael Betancourt*

14. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo, *Journal of Machine Learning Research, Matthew D. Hoffman, etc*

15. Computing DSGE Models with Recursive Preferences and Stochastic Volatility, *Jesus Fernandez-Villaverde, etc*

16. Deep learning for solving dynamic economic models, *Journal of Monetary Economics, Lilia Maliar, etc*