ECON 5110 Class Notes
Solution Techniques for Dynamic Rational Expectation Economies

1 Introduction

This section provides a very, very, very brief introduction to solving dynamic RE models. Good references include books by Stokey, Lucas and Prescott (1989) and Adda and Cooper (2003).

1.1 The Setup

Begin with the following stochastic dynamic programming (DP) problem

$$\max W_T(z_0, x_0, D_T) = E_0 \sum_{t=0}^{T} \beta^t r(z_t, c_t) \tag{1}$$

subject to

$$\begin{aligned} & r(z_t, c_t) \text{ (return function)} \\ z_{t+1} &= f(z_t, x_t, c_t) \text{ (law of motion for the state variable)} \\ c_t &= d_t(z_t, x_t) \text{ (decision rule for control variable)} \\ x_t &= \rho x_{t-1} + \epsilon_t \text{ (exogenous shock with Markov process)} \\ D_T &= (d_0, d_1, ..., d_T) \text{ (sequence of decision rules or a policy)} \\ & z_0, x_0 \text{ given.} \end{aligned}$$

Many problems we will encounter will result in stationary programming problems (i.e., $d_t(z_t, x_t) = d(z_t, x_t)$). The DP problem is to choose an optimal sequence of decisions D_T^* that maximizes W_T.

If $r(\cdot)$ and $f(\cdot)$ are continuous, bounded and strictly concave in c_t and the expectation operator is bounded and continuous in z_t, then an optimal policy exists (see Stokey, Lucas and Prescott, 1989). The value function is

$$V_T(z_0, x_0) = E_0 \sum_{t=0}^{T} \beta^t r(z_t, d_t^*(z_t, x_t)).$$

Given the stationarity of the problem, this can alternatively be written using the Bellman equation, as

$$V(z, x) = \max \{ r(z, d) + \beta E[V(f(z, x, d), x')|x, z] \} . \tag{2}$$

Bellman's expression is useful because it reduces a dynamic problem to a sequence of static problems. Another useful result is Bellman's principle of optimality, which states that if

$$D_T^* = (d_0^*, d_1^*, ..., d_T^*)$$

is an optimal policy then after s periods

$$(d_s^*, d_{s+1}^*, ..., d_T^*)$$

will still be optimal. This is also known as time consistency.

1.2 The Solution Algorithms.

1.2.1 Finite Horizon Case.

In the case of a finite T, the DP problem is solved via backward induction.

1.2.2 Infinite Horizon Case.

In the case of an infinite T, backward induction will not work. Also, because there are always an infinite periods to go, the environment is stationary and will result in a time-invariant decision rule, $c = d(z_t, x_t)$. This also implies that the Bellman function (2) can be written without reference to time.

Method #1. Value Function Iteration. This method uses the Bellman equation directly and iterates on the value function starting from an initial guess. It relies on finding the fixed point of the operator $\Psi : \nu \to \nu$, where

$$\Psi\left[v(z,x)\right] = \max\left\{r(z,d) + \beta E[v(f(z,x,d),x')|x,z]\right\}.$$

Here are the steps associated with Method #1:

1. **Choose functional forms for the return function**. There is no need to choose a functional form for $v(\cdot)$, only for $r(\cdot)$.

2. **Break the state space into a grid**. Obviously, the computer cannot handle a continuum of values for the state variable z. There is a tradeoff between accuracy (finer grid) and computational time (coarser grid).

3. **Iterate on value function**. We start with an initial guess for $v_0(z,x)$, where lower case values v represent candidate functions for V in (2). Because $\Psi(\cdot)$ is a contraction mapping, the initial guess should not influence the result. A common first guess is $v_0(z,x) = 0$. At each step in the iterations, we look at all values on the state-space grid. The iterations stop when $|v_{j+1}(z,x) - v_j(z,x)|$ is sufficiently small. In the end, we get a mapping from all possible values of the state variable to the value function. This is a discrete mapping and can be made continuous with interpolation methods.

4. **Evaluate policy functions**. The policy function $c = d(z,x)$ can be found by collecting all optimal decision values associated with the different grid values in the state space.

Method #2. Linear-Quadratic Approximation Around Steady State. This method also works with the value function $V(z,x)$ but uses a quadratic approximation to the return function with linear constraints. In this case, it is possible to solve for an explicit linear policy function $d(z_t, x_t)$, which when substituted into the state equation will generate a linear law of motion for the state variables.

Letting $w = (z, x, d)'$ and Q be a symmetric matrix of approximation terms, the LQ dynamic programming problem can be written as

$$V(z,x) = \max \left\{ (w^T Q w) + \beta E[V(z', x')|x, z] \right\}.$$

Through similar methods as described above, this will lead to a policy function which is linear in the state variables

$$d = az + bx.$$

Method #3. Direct Attack on the Euler Equations. One example is direct attack is the method we used in class, whereby we substitute the constraints directly into the objective function and linearize the resulting Euler equations around the stationary steady state. This linear system can then be solved for linear policy functions using the method of Blanchard and Kahn (1980).