ECON 5110 Class Notes
Solution Techniques for Dynamic Rational Expectation Economies

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

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

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

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.

2 Solution Algorithms.

2.1 Finite Horizon Case.

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

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.

2.2.1 Method #1. Extended Path.

The extended path method is a numerical procedure used to simulate the optimal path for the DP problem above. The procedure works by first solving for the Euler equations. The system of equations – constraints and Euler equations – are then partitioned into three groups and stacked together in a large matrix. The first set of equations represents the steady-state initial conditions before the shocks have been turned on. The second set of equations represents the period where the shocks have been turned and work through the system. The shocks are then turned off and the system is allowed to return to steady state. This is the extended path. The third set of equations imposes the terminal steady-state condition, after a sufficient number of periods have passed and the system returns to steady state.

Recursions on the full matrix system then recover the simulated solution to the DP problem. We used this procedure to simulate the RBC model. You can read more about the procedure by clicking on the Solution Method link from the ECON 5010 website.

2.2.2 Method #2. Direct Attack on the Euler Equations.

This method directly attacks the Euler equations to solve for the linear policy function $d(z_t, x_t)$. First, 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).

AN EXAMPLE. FARMER'S MODEL

Start with Farmer's system of equations:

$$
\begin{aligned}
k_{t+1} &= y_t + (1-\delta)k_t - c_t \\
y_t &= s_t k_t^{\mu}(\gamma_t l_t)^{\nu} \\
n y_t &= c_t l_t^{1+\chi} \\
\frac{1}{c_t} &= \beta E_t\left[\frac{1}{c_{t+1}}(1-\delta+m\frac{y_{t+1}}{k_{t+1}})\right] \\
s_t &= s_{t-1}^{\rho} v_t.
\end{aligned}
$$

Calculate the steady state and linearize the system, writing all variables as proportional deviations from their steady states. Using the production function and labor first-order condition, we can eliminate l_t and y_t. The linearized system can then be written as

$$\hat{c}_t = E_t\hat{c}_{t+1} + a_2 E_t\hat{k}_{t+1} + a_3 E_t\hat{s}_{t+1} \tag{3}$$

$$\hat{k}_{t+1} = a_4\hat{k}_t + a_5\hat{s}_t + a_6\hat{c}_t \tag{4}$$

$$\hat{s}_{t+1} = \rho\hat{s}_t + \epsilon_{t+1}. \tag{5}$$

STANDARD FORM

The next step is to write the system in standard matrix form, where we define the conditional expectation of $\hat{c}_{t+1}$ to be $E_t(\hat{c}_{t+1}) = \hat{c}_{t+1} + w_{t+1}^c$. Similarly for $\hat{k}_{t+1}$ and $\hat{s}_{t+1}$. The system in matrix form is

$$
\begin{bmatrix} 1 & 0 & 0 \\ a_6 & a_4 & a_5 \\ 0 & 0 & \rho \end{bmatrix}
\begin{bmatrix} \hat{c}_t \\ \hat{k}_t \\ \hat{s}_t \end{bmatrix}
=
\begin{bmatrix} 1 & a_2 & a_3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} \hat{c}_{t+1} \\ \hat{k}_{t+1} \\ \hat{s}_{t+1} \end{bmatrix}
+
\begin{bmatrix} 0 & 1 & a_2 & a_3 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}
\begin{bmatrix} \epsilon_{t+1} \\ w_{t+1}^c \\ w_{t+1}^k \\ w_{t+1}^s \end{bmatrix}
$$

or written more compactly as

$$Y_t = JY_{t+1} + BV_{t+1} \tag{6}$$

where $Y_t = (\hat{c}_t, \hat{k}_t, \hat{s}_t)'$, $V_{t+1} = (\epsilon_{t+1}, w_{t+1}^c, w_{t+1}^k, w_{t+1}^s)'$,

$$
J = \begin{bmatrix} 1 & 0 & 0 \\ a_6 & a_4 & a_5 \\ 0 & 0 & \rho \end{bmatrix}^{-1}
\begin{bmatrix} 1 & a_2 & a_3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
$$

and

$$B = \begin{bmatrix} 1 & 0 & 0 \\ a_6 & a_4 & a_5 \\ 0 & 0 & \rho \end{bmatrix}^{-1} \begin{bmatrix} 0 & 1 & a_2 & a_3 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}.$$

DIAGONALIZATION

The dynamic properties of the model depend critically on the matrix J. It will be convenient to calculate the roots or eigenvalues of the matrix J, (i.e., the λs that solve the following matrix equation $(J-\lambda I)Q = 0$.) The interesting solution to this problem requires that $(J - \lambda I)$ be singular, or in other words, $|J - \lambda I| = 0$. Solving this equation produces the eigenvalues $\lambda_1, \lambda_2, ..., \lambda_n$ where n is the number of rows (or columns) of J. Eigenvalues less than one in absolute value (i.e., inside the unit circle) are called forward stable roots. The associated eigenvectors satisfy $JQ^{(i)} = \lambda_i Q^{(i)}$ for $i = 1, ..., n$. For Farmer's example above, stacking these equations together produces

$$J \begin{bmatrix} Q^{(1)} & Q^{(2)} & Q^{(3)} \end{bmatrix} = \begin{bmatrix} Q^{(1)} & Q^{(2)} & Q^{(3)} \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$

or $JQ = Q\Lambda$ where Q is the matrix of stacked eigenvectors and Λ is the diagonal matrix with the eigenvalues along the main diagonal.

Diagonalization of J involves writing it in the form $J = Q\Lambda Q^{-1}$. We will use this representation of J to write the system as a set of independent equations.

TRANSFORMATION

Begin by taking expectations of (6) conditional on time t information, which produces

$$Y_t = JE_tY_{t+1}.$$

Using the diagonalization of J above, we get

$$Y_t = Q\Lambda Q^{-1}E_tY_{t+1} \Rightarrow Q^{-1}Y_t = \Lambda E_t Q^{-1}Y_{t+1}.$$

If we let $Z_t \equiv Q^{-1}Y_t$, then we can write the matrix system as three independent equations

$$Z_t = \Lambda E_t Z_{t+1} \Rightarrow \begin{bmatrix} z_{1t} \\ z_{2t} \\ z_{3t} \end{bmatrix} = \begin{bmatrix} \lambda_1 E_t z_{1,t+1} \\ \lambda_2 E_t z_{2,t+1} \\ \lambda_3 E_t z_{3,t+1} \end{bmatrix}. \tag{7}$$

LAW OF ITERATED EXPECTATIONS

Equations (7) hold for all $t = 0, ..., \infty$. Using this fact, we can substitute for Z_t on the right side of (7) to get

$$Z_t = \Lambda E_t[\Lambda E_{t+1} Z_{t+2}] = \Lambda^2 E_t Z_{t+2}$$

where we used the law of iterated expectations for the last equality. Repeated substitutions produce

$$Z_t = \Lambda^T E_t Z_{t+T}.$$

For the n_s forward-stable roots, if we let $T \to \infty$ and impose the condition that the $\lim_{T\to\infty} E_t Z_{t+T}$ does not explode too fast, then we have

$$z_{it} = 0$$

for $i = 1, ..., n_s$.

IMPOSE CONSTRAINT

The RBC model ($\mu + \nu = 1$), written in this form, will possess one forward stable root (i.e., $n_s = 1$). Using this fact and the relationship $Z_t \equiv Q^{-1} Y_t$, we now have an additional restriction on the model

$$\hat{c}_t = q^{11} \hat{k}_t + q^{12} \hat{s}_t \tag{8}$$

where the q coefficients are associated with the first (i.e., forward-stable) row of Q^{-1}.

VECTOR AUTOREGRESSION AND THE POLICY FUNCTIONS

Finally, we can substitute (8) into (4) to obtain

$$\begin{aligned} \hat{k}_{t+1} &= a_4 \hat{k}_t + a_5 \hat{s}_t + a_6 (q^{11} \hat{k}_t + q^{12} \hat{s}_t) \\ &= (a_4 + a_6 q^{11}) \hat{k}_t + (a_5 + a_6 q^{12}) \hat{s}_t \\ &= b_1 \hat{k}_t + b_2 \hat{s}_t. \end{aligned}$$

Along with (5), k_0 and s_0, we can describe the evolution of the state variables in vector autoregression (VAR) form as

$$\begin{bmatrix} \hat{k}_{t+1} \\ \hat{s}_{t+1} \end{bmatrix} = \begin{bmatrix} b_1 & b_2 \\ 0 & \rho \end{bmatrix} \begin{bmatrix} \hat{k}_t \\ \hat{s}_t \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \epsilon_{t+1}$$

and the evolution of the remaining variables are determined through the following "policy functions"

$$(\hat{c}_t, \hat{y}_t, \hat{x}_t) = \Pi \begin{bmatrix} \hat{k}_t \\ \hat{s}_t \end{bmatrix}.$$

2.2.3 Method #3. 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

$$d = az + bx.$$

2.2.4 Method #4. 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

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

Here are the steps associated with Method #4:

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.