

Graduate Macro Theory II:

Notes on Solving Linearized Rational Expectations Models

Eric Sims
University of Notre Dame

Spring 2017

1 Introduction

The solution of many discrete time dynamic economic models is a system of non-linear difference equations. One method for approximating the solution to these models is by log-linearizing the system of equations about a point (typically the steady state), thereby translating the system of non-linear difference equations into a system of (approximately) linear difference equations. This note describes how to arrive at the approximate policy functions/decision rules once the system of equations has been transformed into a log-linearized system.

2 Details

Let \mathbf{X}_t be an $(n + m) \times 1$ vector of variables expressed as percentage deviations from steady state (i.e. a typically element of \mathbf{X}_t is $\tilde{x}_{k,t}$ for $k = 1, \dots, n + m$). Let n be the number of “jump” or “forward-looking” variables, while m is the number of states or predetermined variables. In a deterministic growth model, for example, $n = 1$ (consumption) and $m = 1$ (the capital stock), while in the stochastic growth model $n = 1$ (consumption) and $m = 2$ (capital stock and TFP). I’m going to partition the vector of variables into two parts – $x_{1,t}$ is $n \times 1$ vector containing the jump variables while $x_{2,t}$ is $m \times 1$ vector containing the state variables. The linearized solution takes the form:

$$E_t \begin{bmatrix} x_{1,t+1} \\ n \times 1 \\ x_{2,t+1} \\ m \times 1 \\ (n+m) \times 1 \end{bmatrix} = \begin{matrix} \mathbf{M} \\ (n+m) \times (n+m) \end{matrix} \begin{bmatrix} x_{1,t} \\ n \times 1 \\ x_{2,t} \\ m \times 1 \\ (n+m) \times 1 \end{bmatrix}$$

$$E_t \mathbf{X}_{t+1} = \mathbf{M} \mathbf{X}_t \tag{1}$$

Note that I have to put an expectations operator in front of the variables dated $t + 1$ since I'm here allowing the model to have a stochastic element.

We can typically derive a closed form expression for \mathbf{M} in terms of the underlying parameters of the model once it has been log-linearized. But this does *not* mean that we have the solution to the model. \mathbf{M} tells us how the variables in the system will evolve given an initial starting point. But we only have the initial starting point for the state variables – we do not know where to “start” the jump variables. We have to work harder to figure that out, essentially imposing a terminal condition of non-explosion. The rest of what we do in these notes is working out how to find that starting position for the non-predetermined variables.

Recall the definition of eigenvalues and eigenvectors. An eigenvalue is a scalar, λ , and an eigenvector is a vector, \mathbf{v} , which jointly satisfy:

$$\mathbf{M}\mathbf{v} = \lambda\mathbf{v} \tag{2}$$

$$(\mathbf{M} - \lambda\mathbf{I})\mathbf{v} = \mathbf{0} \tag{3}$$

Unless you've screwed up, there will be the same number of distinct eigenvalues as there are rows/columns in the square matrix \mathbf{M} (i.e. in this case there will $n + m$ eigenvalues), some of which may be complex. There will also be the same number of distinct eigenvectors as there are rows/columns of \mathbf{M} . Index these eigenvalues/eigenvectors by $k = 1, \dots, n + m$. The above definition will hold for each $k = 1, \dots, n + m$. In other words:

$$\mathbf{M}\mathbf{v}_1 = \lambda_1\mathbf{v}_1 \quad \dots \quad \mathbf{M}\mathbf{v}_2 = \lambda_2\mathbf{v}_2 \quad \dots \quad \mathbf{M}\mathbf{v}_{n+m} = \lambda_{n+m}\mathbf{v}_{n+m}$$

This means that we can stack these up as follows:

$$\mathbf{M} \begin{bmatrix} v_{1,1} & v_{2,1} & \cdots & v_{n+m,1} \\ v_{1,2} & v_{2,2} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ v_{1,n+m} & v_{2,n+m} & \cdots & v_{n+m,n+m} \end{bmatrix} = \begin{bmatrix} v_{1,1} & v_{2,1} & \cdots & v_{n+m,1} \\ v_{1,2} & v_{2,2} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ v_{1,n+m} & v_{2,n+m} & \cdots & v_{n+m,n+m} \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & \vdots \\ \vdots & 0 & \ddots & \vdots \\ 0 & \cdots & \cdots & \lambda_{n+m} \end{bmatrix}$$

Use the following notation to simplify this a bit:

$$\mathbf{\Gamma} = \begin{bmatrix} v_{1,1} & v_{2,1} & \cdots & v_{n+m,1} \\ v_{1,2} & v_{2,2} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ v_{1,n+m} & v_{2,n+m} & \cdots & v_{n+m,n+m} \end{bmatrix}$$

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & \vdots \\ \vdots & 0 & \ddots & \vdots \\ 0 & \cdots & \cdots & \lambda_{n+m} \end{bmatrix}$$

Using this notation, we have:

$$\mathbf{M}\mathbf{\Gamma} = \mathbf{\Gamma}\mathbf{\Lambda} \quad (4)$$

$$\mathbf{M} = \mathbf{\Gamma}\mathbf{\Lambda}\mathbf{\Gamma}^{-1} \quad (5)$$

It is convenient to note that you can arrange the eigenvalues and eigenvectors in whatever order you want, just so long as the k^{th} column of $\mathbf{\Gamma}$ corresponds with the k^{th} eigenvalue which occupies the (k, k) position of $\mathbf{\Lambda}$. As such, it is helpful to “order” the eigenvalues from smallest to largest (in absolute value . . . if there are complex parts of the eigenvalues, order them by modulus, where the modulus is the square root of the sum of the squared non-complex and complex components; e.g. if $y = x + zi$, then the modulus is $\sqrt{x^2 + z^2}$. If $z = 0$, the modulus is just the absolute value).

More generally, let:

$$\mathbf{\Lambda} = \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix}$$

$Q \times Q$ $B \times B$

Here Λ_1 is a $Q \times Q$ diagonal matrix containing the Q stable eigenvalues, while Λ_2 is a $B \times B$ diagonal matrix containing the B unstable eigenvalues (obviously $Q + B = n + m$, but neither n nor m are necessarily guaranteed to equal Q or B , respectively . . . this will be discussed in more depth below).

Using the eigenvalue/eigenvector decomposition of \mathbf{M} , we can rewrite the system as follows now:

$$E_t \mathbf{X}_{t+1} = \mathbf{\Gamma}\mathbf{\Lambda}\mathbf{\Gamma}^{-1} \mathbf{X}_t$$

Pre-multiply each side by $\mathbf{\Gamma}^{-1}$ to get:

$$E_t \mathbf{\Gamma}^{-1} \mathbf{X}_{t+1} = \mathbf{\Lambda} \mathbf{\Gamma}^{-1} \mathbf{X}_t$$

Now define the auxiliary vector \mathbf{Z}_t as follows:

$$\mathbf{Z}_t = \mathbf{\Gamma}^{-1} \mathbf{X}_t$$

Given this renaming, we can write the system as:

$$E_t \mathbf{Z}_{t+1} = \mathbf{\Lambda} \mathbf{Z}_t$$

$$E_t \begin{bmatrix} Z_{1,t+1} \\ Z_{2,t+1} \end{bmatrix} = \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix} \begin{bmatrix} Z_{1,t} \\ Z_{2,t} \end{bmatrix}$$

I've just partitioned \mathbf{Z}_t into two parts – $Z_{1,t}$ is partitioned as the first Q variables in \mathbf{Z}_t while $Z_{2,t}$ is the second B elements of \mathbf{Z}_t . Because we've effectively rewritten this as a VAR(1) process with a diagonal coefficient matrix, $Z_{1,t}$ and $Z_{2,t}$ evolve independently of one another. We can write the expected values updating forward in time as:

$$E_t Z_{1,t+T} = \Lambda_1^T Z_{1,t}$$

$$E_t Z_{2,t+T} = \Lambda_2^T Z_{2,t}$$

Because the eigenvalues in Λ_1 are all stable (absolute value less than one), $\Lambda_1^T \rightarrow 0$ as $T \rightarrow \infty$. The same does not hold true for the second expression, which is partitioned into the explosive eigenvalues. Because the eigenvalues in Λ_2 are all unstable, $E_t Z_{2,t+T} \rightarrow \infty$ as T grows unless $Z_{2,t} = 0$. We cannot let $Z_{2,t+T} \rightarrow \infty$ while simultaneously being consistent with (a) transversality conditions and/or (b) feasibility constraints.

To fix ideas, let me write out in long hand what $\mathbf{\Gamma}^{-1}$ is:

$$\mathbf{\Gamma}^{-1} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}$$

$\begin{matrix} Q \times n & Q \times m \\ B \times n & B \times m \\ (Q+B) \times (n+m) \end{matrix}$

Since $Q + B = n + m$ this is obviously still a square matrix, but the individual partitions need not necessarily be square matrices. Recall that there are Q stable eigenvalues and B unstable ones, while there are n jump variables and m state variables. Let's write out in long hand what the Z 's are:

$$\begin{matrix} Z_{1,t} & = & G_{11} x_{1,t} + G_{12} x_{2,t} \\ Q \times 1 & & Q \times n \times 1 & Q \times m \times 1 \end{matrix}$$

$$\begin{matrix} Z_{2,t} & = & G_{21} x_{1,t} + G_{22} x_{2,t} \\ B \times 1 & & B \times n \times 1 & B \times m \times 1 \end{matrix}$$

As noted above, the transversality/feasibility conditions require that $Z_{2,t} = 0$. We can use this to then solve for the initial position of the jump variables ($x_{1,t}$) in terms of the given initial conditions of the states ($x_{2,t}$):

$$\underset{B \times 1}{0} = G_{21}x_{1,t} + G_{22}x_{2,t}$$

Solving, this yields:

$$G_{21}x_{1,t} = -G_{22}x_{2,t}$$

Provided that G_{21} is a square matrix, we can invert G_{21} and then we can solve this as:

$$x_{1,t} = -G_{21}^{-1}G_{22}x_{2,t}$$

In other words, this is our linearized policy function. For a given state vector (i.e. given values of $x_{2,t}$) this will tell us what the value of the jump variables need to be. Now quickly, what does it mean for G_{21} to be square/invertible? Recall that the dimension of G_{21} is $B \times n$, where B is the number of unstable eigenvalues and n is the number of jump variables. Put differently, we must have an equal number of unstable eigenvalues as we do jump variables – this is the condition for saddle point stability. If we don't have enough unstable eigenvalues, there will be an infinite number of solutions. If we have too many unstable eigenvalues, there will be no solution.

3 Example: The Deterministic Growth Model

The non-stochastic neoclassical growth model with CRRA preferences, Cobb-Douglas production, and the level of technology normalized to unity can be reduced to a system of non-linear difference equations:

$$\begin{aligned} c_t^{-\sigma} &= \beta c_{t+1}^{-\sigma} (\alpha k_{t+1}^{\alpha-1} + (1 - \delta)) \\ k_{t+1} &= k_t^\alpha - c_t + (1 - \delta)k_t \end{aligned}$$

Log-linearization of these equations about the steady state yields:

$$\begin{aligned} -\sigma \tilde{c}_t &= -\sigma \tilde{c}_{t+1} + \beta(\alpha - 1)R^* \tilde{k}_{t+1} \\ \tilde{k}_{t+1} &= \frac{1}{\beta} \tilde{k}_t - \frac{c^*}{k^*} \tilde{c}_t \end{aligned}$$

Above $R^* = \alpha k^{*\alpha-1}$ (i.e. the steady state marginal product of capital) and $\frac{c^*}{k^*}$ is the steady state ratio of consumption to capital, both of which are function of underlying parameters of the model. This can be re-arranged into the VAR(1) form as:

$$E_t \begin{bmatrix} \tilde{c}_{t+1} \\ \tilde{k}_{t+1} \end{bmatrix} = \begin{bmatrix} 1 - \frac{c^*}{k^*} \frac{\beta(\alpha-1)R^*}{\sigma} & \frac{(\alpha-1)R^*}{\sigma} \\ -\frac{c^*}{k^*} & \frac{1}{\beta} \end{bmatrix} \begin{bmatrix} \tilde{c}_t \\ \tilde{k}_t \end{bmatrix}$$

I assign the following values to the parameters: $\sigma = 1$ (log utility), $\beta = 0.95$, $\delta = 0.1$, and $\alpha = 0.33$. These values imply $k^* = 3.16$ and $c^* = 1.146$. The numerical values of this matrix are easily seen to be:

$$\mathbf{M} = \begin{bmatrix} 1.0352 & -0.1023 \\ -0.3625 & 1.0526 \end{bmatrix}$$

The Matlab function “[lam,V,j] = eig_order(M);” will produce a diagonal matrix of eigenvalues ordered from smallest to largest (this is the output matrix “lam”) and the matrix of eigenvectors corresponding with these eigenvalues (the output matrix “V” will be the matrix of eigenvectors). The output “j” is the index of the the first unstable eigenvalue. The eigenvalues of \mathbf{M} come out to be 0.85 and 1.24, so the conditions for saddle path stability are satisfied (i.e. one explosive root, one stable root). I find that $\mathbf{\Gamma}^{-1}$ is:

$$\mathbf{\Gamma}^{-1} = \begin{bmatrix} -1.0759 & -0.5462 \\ 1.0547 & -0.5861 \end{bmatrix}$$

My Z_t matrix then takes the following form:

$$\begin{aligned} Z_{1,t} &= -1.0759\tilde{c}_t - 0.5462\tilde{k}_t \\ Z_{2,t} &= 1.0547\tilde{c}_t - 0.5861\tilde{k}_t \end{aligned}$$

Using the eigenvalue decomposition, we know that (with only two variables the diagonal matrices of eigenvalues are just scalars):

$$\begin{aligned} Z_{1,t+T} &= \lambda_1^T Z_{1,t} \\ Z_{2,t+T} &= \lambda_2^T Z_{2,t} \end{aligned}$$

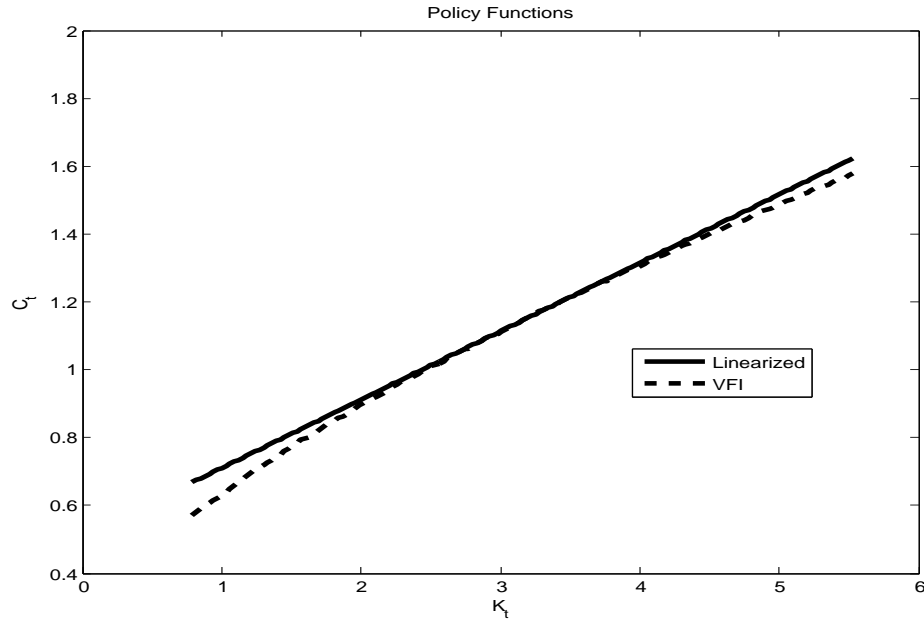
Satisfaction of the transversality and feasibility conditions requires that $Z_{2,t} = 0$. This means our linearized policy function is:

$$\tilde{c}_t = \frac{0.5861}{1.0547} \tilde{k}_t = 0.5557 \tilde{k}_t$$

Note that this is in log-deviations from steady state. To recover the policy function in levels we have to fool around a little bit:

$$\begin{aligned}\frac{c_t - c^*}{c^*} &= 0.5557 \frac{k_t - k^*}{k^*} \\ c_t - c^* &= 0.5557 \frac{c^*}{k^*} k_t - 0.5557 c^* \\ c_t &= 0.4443 c^* + 0.5557 \frac{c^*}{k^*} k_t\end{aligned}$$

Below is a plot of this linearized policy function vs. the policy function I obtain for the same parameterization of the model using value function iteration:



As one can see, the fit is pretty good, especially near the steady state. The approximation grows worse as σ gets bigger (the policy function becomes more concave).

Another Example: The Stochastic Growth Model

This example is identical to above, except I allow for stochastic TFP shocks. The non-linear system of difference equations can be written:

$$\begin{aligned}c_t^{-\sigma} &= \beta E_t c_{t+1}^{-\sigma} (\alpha a_{t+1} k_{t+1}^{\alpha-1} + (1 - \delta)) \\ k_{t+1} &= a_t k_t^\alpha - c_t + (1 - \delta) k_t \\ \ln a_t &= \rho \ln a_{t-1} + e_t\end{aligned}$$

e_t is a white noise process. I assume that $a^* = 1$, which means that the mean of the log of technology is zero. One can show that the log-linearized equations are (note that in the log-

linearization you don't need to first take logs of the technology process, since it is already in log form):

$$\begin{aligned}\tilde{c}_{t+1} - \frac{\beta R^*}{\sigma} \tilde{a}_{t+1} - \frac{\beta(\alpha-1)R^*}{\sigma} \tilde{k}_{t+1} &= \tilde{c}_t \\ \tilde{k}_{t+1} &= k^{*\alpha-1} \tilde{a}_t - \frac{c^*}{k^*} \tilde{c}_t + \frac{1}{\beta} \tilde{k}_t \\ \tilde{a}_t &= \rho \tilde{a}_{t-1} + e_t\end{aligned}$$

Above, note that $e_t = e_t - e^* = e_t$. In other words, it is an absolute deviation from “steady state”, which is zero by assumption of it being white noise. You obviously can't express it as a percentage deviation about a steady state, because any deviation from zero would be infinite. If we re-write this isolating the $t+1$ variables on the right hand side, we get:

$$E_t \begin{bmatrix} \tilde{c}_{t+1} \\ \tilde{k}_{t+1} \\ \tilde{a}_{t+1} \end{bmatrix} = \begin{bmatrix} 1 - \frac{\beta(\alpha-1)R^*}{\sigma} \frac{c^*}{k^*} & \frac{(\alpha-1)R^*}{\sigma} & \frac{\beta R^*(\rho + (\alpha-1)k^{*\alpha-1})}{\sigma} \\ -\frac{c^*}{k^*} & \frac{1}{\beta} & k^{*\alpha-1} \\ 0 & 0 & \rho \end{bmatrix} \begin{bmatrix} \tilde{c}_t \\ \tilde{k}_t \\ \tilde{a}_t \end{bmatrix}$$

Using the same parameterization as in the deterministic example with $\rho = 0.95$, I find that the eigenvalues of this matrix are 0.8512, 0.95, and 1.2367, so the conditions for saddle points stability are satisfied. The inverse of the matrix of eigenvectors are seen to be:

$$\mathbf{\Gamma}^{-1} = \begin{bmatrix} -1.0759 & -0.5462 & 3.5671 \\ 0 & 0 & 3.4172 \\ 1.0547 & -0.5861 & -0.6041 \end{bmatrix}$$

The components of the Z matrixes are then:

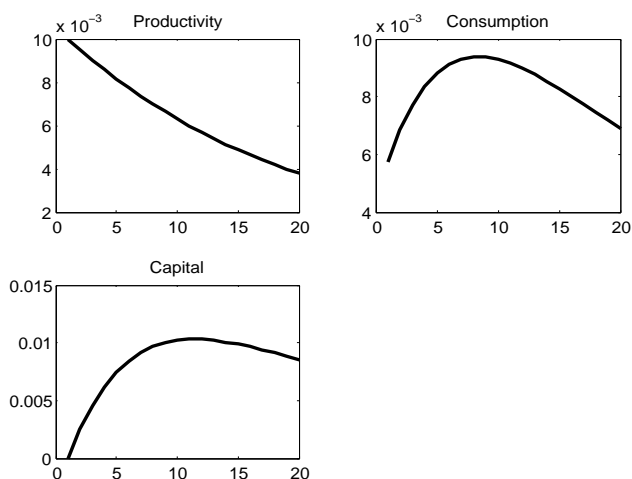
$$\begin{aligned}Z_{1,t} &= \begin{bmatrix} -1.0759 \\ 0 \end{bmatrix} \tilde{c}_t + \begin{bmatrix} -0.5462 & 3.5671 \\ 0 & 3.4172 \end{bmatrix} \begin{bmatrix} \tilde{k}_t \\ \tilde{a}_t \end{bmatrix} \\ Z_{2,t} &= 1.0547 \tilde{c}_t + \begin{bmatrix} -0.5861 & -0.6041 \end{bmatrix} \begin{bmatrix} \tilde{k}_t \\ \tilde{a}_t \end{bmatrix}\end{aligned}$$

Stability requires that $Z_{2,t} = 0$ since that is associated with the explosive eigenvalue. We can then solve for the policy function as:

$$\begin{aligned}\tilde{c}_t &= \frac{-1}{1.0547} \begin{bmatrix} -0.5861 & -0.6041 \end{bmatrix} \begin{bmatrix} \tilde{k}_t \\ \tilde{a}_t \end{bmatrix} \\ \tilde{c}_t &= 0.5557 \tilde{k}_t + 0.5728 \tilde{a}_t\end{aligned}$$

Given this policy function and an initial condition for \tilde{k}_t , I can shock \tilde{a}_t and then let the system

play out. Below are the impulse responses to a one unit shock to technology:



Dealing with “Static” Variables

“Static” variables are defined as variables in the model which only show up at date t – i.e. they are not explicitly forward-looking (jump variables) or explicitly backward-looking (state variables), though these variables often are *implicitly* forward-looking through their dependence of jump variables (like consumption). The basic strategy is to simply solve for the static variables in terms of the jump and state variables. You proceed by finding the policy functions for the jump variables just like you did above.

Some kinds of “static” variables are easier to deal with than others. The easy ones are variables which are essentially just log-linear combinations of the jump variable and state variables. An example is output:

$$\begin{aligned} y_t &= a_t k_t^\alpha \\ \tilde{y} &= \tilde{a}_t + \alpha \tilde{k}_t \end{aligned}$$

Another easy one is investment. From the aggregate resource constraint, log-linearized we know that:

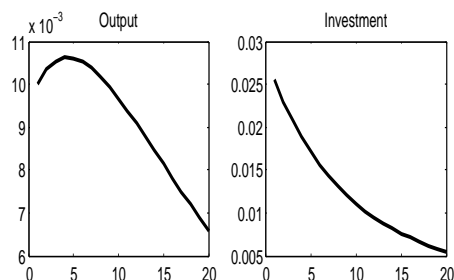
$$\tilde{y}_t = \frac{c^*}{y^*} \tilde{c}_t + \frac{I^*}{y^*} \tilde{I}_t$$

Here $I^* = \delta k^*$. We can then back out investment as:

$$\tilde{I}_t = \frac{y^*}{I^*} \tilde{y}_t - \frac{c^*}{I^*} \tilde{c}_t$$

Put differently, once we know the two states, we have \tilde{y} . It is this reason that sometimes these

variables are called “redundant” variables, because they are simply linear combinations of jump variables and state variables. Below I show impulse responses of output and investment to the productivity shock using the same parameterization as above.



Some static variables present more headaches. An example of one is from a model in which there is variable labor. As an example, suppose that the within period utility function takes the following form:

$$u(c_t, n_t) = \frac{c_t^{1-\sigma}}{1-\sigma} - \psi \frac{n_t^{1+\theta}}{1+\theta}$$

The production function for this economy would take the form:

$$y_t = a_t k_t^\alpha n_t^{1-\alpha}$$

One can show that the consumption Euler and capital accumulation equations takes the following form:

$$\begin{aligned} c_t^{-\sigma} &= \beta E_t c_{t+1}^{-\sigma} (\alpha a_{t+1} k_{t+1}^{\alpha-1} n_t^{1-\alpha} + (1-\delta)) \\ k_{t+1} &= a_t k_t^\alpha n_t^{1-\alpha} - c_t + (1-\delta)k_t \end{aligned}$$

The question is: how do we deal with the n_t ? The answer is that there is also a first order condition for optimal labor supply. You can show that it takes the following form:

$$\psi n_t^\theta = c_t^{-\sigma} (1-\alpha) a_t k_t^\alpha n_t^{-\alpha}$$

In words, this says that the marginal disutility of work equals the marginal utility of consumption times the real wage (the marginal product of capital). What you can do is to solve for n_t in terms of the jump and state variables:

$$n_t^{\theta+\alpha} = \frac{1}{\psi} c_t^{-\sigma} (1-\alpha) a_t k_t^\alpha$$

$$n_t = \left(\frac{1}{\psi} c_t^{-\sigma} (1-\alpha) a_t k_t^\alpha \right)^{\frac{1}{\theta+\alpha}}$$

Given this, you can substitute this wherever n_t shows up in the first order conditions and you're back to the kind of system we had before, though it is admittedly more complicated. In practice, the easier thing to do is often to log-linearize all the equations first, and then eliminate the log-linearized \tilde{n}_t .

Getting the Dynamics Right

Suppose that you want to construct impulse responses or simulate data from the linearized model. As an example, suppose that we take the deterministic growth model and want to compute what happens in expectation if the capital stock starts out below steady state. The simple thing to do would be to start at some \tilde{k}_0 , set $\tilde{c}_0 = -G_{2,1}^{-1} G_{2,2} \tilde{k}_0$, and then trace out expected future dynamics as:

$$\begin{bmatrix} \tilde{c}_t \\ \tilde{k}_t \end{bmatrix} = \mathbf{M}^t \begin{bmatrix} \tilde{c}_0 \\ \tilde{k}_0 \end{bmatrix} = \mathbf{M}^t \begin{bmatrix} -G_{2,1}^{-1} G_{2,2} \tilde{k}_0 \\ \tilde{k}_0 \end{bmatrix}$$

This is analytically correct, but is prone to numerical problems. Why? Recall the whole idea of saddle point stability. If you are at all off the policy function, even by a very small amount, the system eventually explodes (this is driven by the presence of unstable eigenvalues/roots in \mathbf{M}). In practice, there will be small numerical errors in the policy function, $-G_{2,1}^{-1} G_{2,2}$. We're talking numerical errors to several decimal places, but the system still can't tolerate these. Particularly at longer horizons. If you do the exercise I'm talking about above, everything will look great for about 100 periods . . . but out at longer horizons the system starts to explode.

There is a straightforward way of dealing with this and avoiding the potential for explosion that results from small numerical errors. Consider the general case. Decompose \mathbf{M} into blocks:

$$\mathbf{M} = \begin{bmatrix} M_{1,1} & M_{1,2} \\ M_{2,1} & M_{2,2} \\ n \times n & n \times m \\ m \times n & m \times m \end{bmatrix}$$

Write the original system out in "long hand" using this notation:

$$E_t x_{1,t+1} = M_{1,1} x_{1,t} + M_{1,2} x_{2,t}$$

$$E_t x_{2,t+1} = M_{2,1} x_{1,t} + M_{2,2} x_{2,t}$$

Now, plug in the policy function to eliminate $x_{1,t}$ in both expressions:

$$E_t x_{1,t+1} = \left(-M_{1,1} G_{2,1}^{-1} G_{2,2} + M_{1,2} \right) x_{2,t}$$

$$E_t x_{2,t+1} = \left(-M_{2,1} G_{2,1}^{-1} G_{2,2} + M_{2,2} \right) x_{2,t}$$

Define a new matrix, \mathbf{A} , as follows:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{n \times n} & -M_{1,1} G_{2,1}^{-1} G_{2,2} + M_{1,2} \\ \mathbf{0}_{m \times n} & -M_{2,1} G_{2,1}^{-1} G_{2,2} + M_{2,2} \end{bmatrix}$$

Then write the system as:

$$E_t \begin{bmatrix} x_{1,t+1} \\ x_{2,t+1} \end{bmatrix} = \mathbf{A} \begin{bmatrix} x_{1,t} \\ x_{2,t} \end{bmatrix}$$

Effectively what this does is imposes the policy function so that you can write the AR coefficient matrix with only coefficients on $x_{2,t}$, the vector of states. This turns out to eliminate the problem. You can then proceed as follows – you can “start” the system at some arbitrary value of the state, start the controls at the appropriate place given the policy function, and then iterate forward using the new \mathbf{A} instead of \mathbf{M} .

4 Dealing with Static and Redundant Variables in an Easier Way

Above I briefly discussed how static (variables that are optimally chosen but are not inherently forward-looking, like labor supply) and redundant (variables that are defined in terms of other variables, like output or investment) variables need to be “eliminated” to solve for the linearized policy functions using the approach laid out here. This is correct and can be done by hand, but it is algebraically intensive and can get annoying. Below I discuss a way in which to do this in a way that just involves manipulation of a few matrixes.

Suppose our model is composed of the following (non-linearized) equations:

$$\frac{1}{C_t} = \beta E_t \left(\frac{1}{C_{t+1}} (R_{t+1} + (1 - \delta)) \right)$$

$$K_{t+1} = I_t + (1 - \delta) K_t$$

$$Y_t = A_t K_t^\alpha N_t^{1-\alpha}$$

$$R_t = \alpha A_t K_t^{\alpha-1} N_t^{1-\alpha}$$

$$w_t = (1 - \alpha) A_t K_t^\alpha N_t^{-\alpha}$$

$$\theta N_t^\chi = \frac{1}{C_t} w_t$$

$$Y_t = C_t + I_t$$

$$\ln A_t = \rho \ln A_{t-1} + \varepsilon_t$$

This is 8 equations and 8 variables: C_t , R_t , w_t , A_t , K_t , Y_t , N_t , and I_t . We could recover a simple stochastic growth model if $\chi \rightarrow \infty$, which would imply that N_t is constant. We could then pick θ to normalize $N_t = 1$, which would allow us to eliminate it altogether. Y_t and I_t would then just be redundant – they would be combinations of A_t , K_t , and C_t . w_t and R_t would be static – they come out of optimization, but once N_t is fixed they are again just functions of A_t and K_t . We could eliminate these, leaving a system of just C_t (the jump), K_t (endogenous state), and A_t (exogenous state), and proceed as above).

But let's think about the more general case. Let's log-linearize each equation not eliminating any static or redundant variables. The log-linearized equilibrium conditions are (variables without a subscript denote non-stochastic steady state values):

$$-\tilde{C}_t = -E_t \tilde{C}_{t+1} + \beta R E_t \tilde{R}_{t+1}$$

$$\tilde{K}_{t+1} = \delta \tilde{I}_t + (1 - \delta) \tilde{K}_t$$

$$\tilde{A}_t = \rho \tilde{A}_{t-1} + \varepsilon_t$$

$$\chi \tilde{N}_t = -\tilde{C}_t + \tilde{w}_t$$

$$\tilde{Y}_t = \tilde{A}_t + \alpha \tilde{K}_t + (1 - \alpha) \tilde{N}_t$$

$$\tilde{Y}_t = \frac{C}{Y} \tilde{C}_t + \frac{I}{Y} \tilde{I}_t$$

$$\tilde{w}_t = \tilde{A}_t + \alpha \tilde{K}_t - \alpha \tilde{N}_t$$

$$\tilde{R}_t = \tilde{A}_t + (\alpha - 1) \tilde{K}_t + (1 - \alpha) \tilde{N}_t$$

Let's stack all of these up into vectors. Let:

$$X_t = \begin{bmatrix} \tilde{C}_t \\ \tilde{K}_t \\ \tilde{A}_t \\ \tilde{N}_t \\ \tilde{Y}_t \\ \tilde{I}_t \\ \tilde{w}_t \\ \tilde{R}_t \end{bmatrix}$$

Note that I didn't order these randomly – I start with the forward-looking jump variable, then the two states, then followed by the redundant/static variables.

We can write out the log-linearized conditions in matrix form as:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\beta R \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} E_t \begin{bmatrix} \tilde{C}_{t+1} \\ \tilde{K}_{t+1} \\ \tilde{A}_{t+1} \\ \tilde{N}_{t+1} \\ \tilde{Y}_{t+1} \\ \tilde{I}_{t+1} \\ \tilde{w}_{t+1} \\ \tilde{R}_{t+1} \end{bmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & (1-\delta) & 0 & 0 & 0 & \delta & 0 & 0 \\ 0 & 0 & \rho & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & -\chi & 0 & 0 & 1 & 0 \\ 0 & \alpha & 1 & (1-\alpha) & -1 & 0 & 0 & 0 \\ \frac{C}{Y} & 0 & 0 & 0 & -1 & \frac{I}{Y} & 0 & 0 \\ 0 & \alpha & 1 & -\alpha & 0 & 0 & -1 & 0 \\ 0 & (\alpha-1) & 1 & (1-\alpha) & 0 & 0 & 0 & -1 \end{pmatrix} \begin{bmatrix} \tilde{C}_t \\ \tilde{K}_t \\ \tilde{A}_t \\ \tilde{N}_t \\ \tilde{Y}_t \\ \tilde{I}_t \\ \tilde{w}_t \\ \tilde{R}_t \end{bmatrix}$$

Or, more compactly:

$$A_0 E_t X_{t+1} = B_0 X_t$$

There are a bunch of rows of zeros in the A_0 coefficient matrix – these rows correspond to the redundant variables. Note however that we can decompose these matrixes as follows. Let n be the number of jump variables (here it is 1, \tilde{C}_t), m be the number of states (here 2, \tilde{K}_t and \tilde{A}_t), and q be the number of redundant/static variables (here 5). We can write:

$$A_0 = \begin{pmatrix} a_{11} & a_{12} \\ (n+m) \times (n+m) & (n+m) \times q \\ 0 & 0 \\ q \times (n+m) & q \times q \end{pmatrix}$$

$$B_0 = \begin{pmatrix} b_{11} & b_{12} \\ (n+m) \times (n+m) & (n+m) \times q \\ b_{21} & b_{22} \\ q \times (n+m) & q \times q \end{pmatrix}$$

Let y_t be the $(n+m) \times (1)$ vector of jump and state variables, and x_t be the $(q \times 1)$ vector of redundant variables. We can then write this out as:

$$\begin{pmatrix} a_{11} & a_{12} \\ 0 & 0 \end{pmatrix} E_t \begin{bmatrix} y_{t+1} \\ x_{t+1} \end{bmatrix} = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} E_t \begin{bmatrix} y_t \\ x_t \end{bmatrix}$$

From above, we can see that:

$$0 = b_{21}y_t + b_{22}x_t$$

Since b_{22} is square, we can (in principle) invert it, so we have:

$$x_t = -b_{22}^{-1}b_{21}y_t$$

In other words, we can write the vector of redundant variables as a linear combination of the jump and states variables (which we already knew, but this operationalizes it). Note that the

dimension of $b_{22}^{-1}b_{21}$ is $q \times (n + m)$. Hence, we can write:

$$\begin{pmatrix} a_{11} & a_{12} \\ 0 & 0 \end{pmatrix} E_t \begin{bmatrix} y_{t+1} \\ -b_{22}^{-1}b_{21}y_{t+1} \end{bmatrix} = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} E_t \begin{bmatrix} y_t \\ -b_{22}^{-1}b_{21}y_t \end{bmatrix}$$

Or:

$$\begin{pmatrix} a_{11} - a_{12}b_{22}^{-1}b_{21} \\ 0 \end{pmatrix} E_t y_{t+1} = \begin{pmatrix} b_{11} - b_{12}b_{22}^{-1}b_{21} \\ b_{21} \end{pmatrix} y_t$$

Note here that the dimensions work out. a_{11} is $(n + m) \times (n + m)$; a_{12} is $(n + m) \times q$; $b_{22}^{-1}b_{21}$ is $q \times (n + m)$; hence $a_{11} - a_{12}b_{22}^{-1}b_{21}$ is $(n + m) \times (n + m)$. b_{11} is $(n + m) \times (n + m)$; b_{12} is $(n + m) \times q$; and $b_{22}^{-1}b_{21}$ is $q \times (n + m)$; hence $b_{11} - b_{12}b_{22}^{-1}b_{21}$ is $(n + m) \times (n + m)$. Since these are both square, we can invert to form:

$$E_t y_{t+1} = M y_t$$

Where $M = \begin{pmatrix} a_{11} - a_{12}b_{22}^{-1}b_{21} \\ 0 \end{pmatrix}^{-1} \begin{pmatrix} b_{11} - b_{12}b_{22}^{-1}b_{21} \\ b_{21} \end{pmatrix}$. In other words, what we've done here is system reduction – we've reduced the system back to the VAR(1) in only the jumps and states, and given M can solve for the policy function mapping the states into the jump variables exactly as before. Given this new matrix M only in the system of jump and state variables, we can find the policy function just as before, with: $\phi = -G_{21}^{-1}G_{22}$, with $x_{1,t} = \phi x_{2,t}$, where $x_{1,t}$ is the $n \times 1$ vector jump variables and $x_{2,t}$ is the $m \times 1$ vector of states, and the G matrixes correspond to different blocks of the inverse matrix of eigenvectors of M appropriately sorted.

Let's write this out in state space form in terms of the jump and state variables, and then we'll come back to the redundant/static variables. Note that we can write the state as:

$$E_t x_{2,t+1} = M_{21}x_{1,t} + M_{22}x_{2,t}$$

Using the policy function mapping the state into the jumps, we can write this as:

$$E_t x_{2,t+1} = (M_{21}\phi + M_{22})x_{2,t}$$

Now, we may want to write these expression without expectation operators and instead with shocks. We know that:

$$x_{2,t} = (M_{21}\phi + M_{22})x_{2,t-1} + H_0\epsilon_t$$

Where ϵ_t is a $k \times 1$ vector of shocks (in the baseline RBC model it is just $k = 1$, the productivity shock), and H_0 is $m \times k$. In the baseline RBC model if the elements of the states are capital and the productivity variable, we would know that $H_0 = [0 \ 1]$. Since we know that $x_{1,t} = \phi x_{2,t}$, we can write:

$$x_{1,t} = \phi (M_{21}\phi + M_{22})x_{2,t-1} + \phi H_0\epsilon_t$$

We could stack these up to write:

$$\begin{bmatrix} x_{1,t} \\ x_{2,t} \end{bmatrix} = \begin{pmatrix} \phi(M_{21}\phi + M_{22}) \\ M_{21}\phi + M_{22} \end{pmatrix} x_{2,t} + \begin{pmatrix} \phi H_0 \\ H_0 \end{pmatrix} \epsilon_t$$

Now, we need to get the redundant/static variables back in. Recall that we can write:

$$x_t = -b_{22}^{-1}b_{21}y_t$$

Lets define $\psi = -b_{22}^{-1}b_{21}$. This matrix is $q \times (n + m)$. Let's decompose it as follows:

$$\psi = \begin{pmatrix} \psi_{11} & \psi_{12} \\ q \times n & q \times m \end{pmatrix}$$

In other words, we can write the redundant/state variables as:

$$x_t = \psi_{11}x_{1,t} + \psi_{12}x_{2,t}$$

But then using the policy function, we have:

$$x_t = (\psi_{11}\phi + \psi_{12})x_{2,t}$$

Then lagging $x_{2,t}$, we have:

$$x_t = (\psi_{11}\phi + \psi_{12})(M_{21}\phi + M_{22})x_{2,t-1} + (\psi_{11}\phi + \psi_{12})H_0\epsilon_t$$

Hence, we can characterize the solution as:

$$X_t = AX_{t-1} + B\epsilon_t$$

Where:

$$X_t = \begin{bmatrix} x_{1,t} \\ x_{2,t} \\ x_t \end{bmatrix}$$

Where:

$$A = \begin{pmatrix} \phi(M_{21}\phi + M_{22}) \\ M_{21}\phi + M_{22} \\ (\psi_{11}\phi + \psi_{12})(M_{21}\phi + M_{22}) \end{pmatrix}$$

$$B = \begin{pmatrix} \phi H_0 \\ H_0 \\ (\psi_{11}\phi + \psi_{12})H_0 \end{pmatrix}$$

You can then use this formulation to produce impulse response and model simulations, etc.