Solving DSGE models

Diego Vilán *

Spring 2012

This note seeks to exemplify some of the most common techniques used for solving, calibrating and simulating dynamic general equilibrium models. It covers log-linearization as well as recursive solution methods. A follow-up set of separate notes will cover how to do all of the above procedures using the Matlab add-on called Dynare, as well as known cases with closed form solutions.

1 Introduction

Solutions to dynamic stochastic general equilibrium (DSGE) models are often characterized by a set of simultaneous, non-linear difference equations. To derive the agent's optimal decision rules there are, broadly speaking, two alternatives. The first approach consists of approximating the solution with a (usually) linear polynomial at the steady state. This effectively translates the system of non-linear difference equations into a (approximately) linear one. Once the equilibrium is characterized by a linear system, there exist a variety of methods to solve the system of equations. Popular methods are based on matrix decomposition techniques, where the classic reference is Blanchard and Kahn (1980). SCHUR and QZ decompositions are also commonly used.

???An alternative to the above is to apply the method of undetermined coefficients to the system of linearized equations???.

The second alternative is to apply recursive methods. By not relying on matrix decompositions, the method becomes available even when certain matrix operations are unavailable. A useful reference is Uhlig (1997).

2 Solving Linearized Rational Expectation Models

Suppose the linearized solution to a system of stochastic difference equations takes the form:

$$AE_tX_{t+1} = BX_t + CZ_{t+1}$$

Let X_t be a $(n+m)$ x 1 vector of variables expressed as percentage deviation from steady state. In particular, let n be the number of forward-looking (also referred to as jump or choice) variables, while m the number of backward-looking (also known as states or predetermined) variables. For example, in a deterministic neoclassical growth model the predetermined variables would be the capital stock

*__DISCLAIMER__: I wrote these notes as a study aid for myself. They are work in progress and could be incomplete, inaccurate and even somewhat incorrect. Keep that in mind should you decide to use them. Comments and suggestions welcomed!

($m = 1$), while the forward looking one would be consumption ($n = 1$). In a stochastic growth model however, $n = 1$ (consumption), while $m = 2$ (capital stock and productivity shock).

The linearized solution to a system of stochastic difference equations will take the form:

$$E_t X_{t+1} = M X_t$$

We begin by partitioning the vector of variables in two parts: $x_{1,t}$ is the n x 1 vector containing all choice variables, while $x_{2,t}$ is the m x 1 vector containing the state variables[1]. In turn:

$$E_t \begin{bmatrix} x_{1,t+1} \\ \text{(n x 1)} \\ \\ x_{2,t+1} \\ \text{(m x 1)} \end{bmatrix} = \underbrace{M}_{(n+m)x(n+m)} \begin{bmatrix} x_{1,t} \\ \text{(n x 1)} \\ \\ x_{2,t} \\ \text{(m x 1)} \end{bmatrix}$$

Given that M is a square matrix it will have the same number of distinct eigenvalues as there are rows/columns, some of which may be complex[2]. There will also be the same number of distinct eigenvectors as there are rows/columns of M.

Given that each eigenvalue has a corresponding eigenvector, and that in this case there are n + m distinct ones, one could index these eigenvalues/eigenvectors by $k = 1, ..., n + m$:

$$\begin{gathered} Mv_1 = \lambda_1 v_1 \\ Mv_2 = \lambda_2 v_2 \\ \vdots \\ Mv_{n+m} = \lambda_{n+m} v_{n+m} \end{gathered}$$

Which means we can stack these up in the following way:

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

Define the following as:

[1] Note that $x_{2,0} > 0$ is assumed to be known.

[2] Recall that an eigenvalue is a scalar λ, and an eigenvector is a vector v which **jointly** satisfy:

$$\begin{gathered} Mv = \lambda v \\ (M - \lambda I) = 0 \end{gathered}$$

Note that every eigenvalue has a distinct and **corresponding** eigenvector.

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

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

Using this notation we have that we can rewrite the above as:

$$\begin{aligned} M\Gamma &= \Gamma\Lambda \\ &\Rightarrow \quad M = \Gamma\Lambda\Gamma^{-1} \end{aligned}$$

It should be noted that one can rearrange the eigenvalues and eigenvectors in any order so long as the k^{th} column of Γ corresponds with the k^th eigenvalue which occupies the (k,k) position in Γ. With this in mind, it is often helpful to reorganize the eigenvalues from smallest to largest in absolute value (if there are complex parts to the eigenvalues they should be ordered by modulus). In turn, let:

$$P = \begin{bmatrix} \underbrace{\Lambda_1}_{QxQ} & 0 \\ 0 & \underbrace{\Lambda_2}_{BxB} \end{bmatrix}$$

Here Λ_1 is a QxQ diagonal matrix containing the Q stable eigenvalues of M, while Λ_2 is a BxB diagonal matrix containing the B unstable eigenvalues. Naturally $Q + B = n + m$, but neither n are guaranteed to be equal to Q, nor m to B.

Using the above mentioned decomposition of M we can rewrite the original system

$$E_t X_{t+1} = M X_t$$

as:

$$E_t X_{t+1} = \Gamma P \Gamma^{-1} X_t$$

Pre-multiply each side by Γ^{-1} to get:

$$E_t \Gamma^{-1} X_{t+1} = P \Gamma^{-1} X_t$$

Define $Z_t = \Gamma^{-1} X_t$ and write the system as:

$$E_t Z_{t+1} = P Z_t$$

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

where Z_t has been partitioned in two parts: $Z_{1,t}$ as the first Q variables in Z_t while $Z_{2,t}$ as the second B elements. Because we've effectively rewritten this as a VAR(1) process with a diagonal matrix, $Z_{1,t}$ and $Z_{2,t}$ evolve independently of one another. We can write the expected values updating forward in time as:

$$E_t Z_{1,t+T} = \Lambda_1^T Z_{1,t}$$
$$E_t Z_{2,t+T} = \Lambda_2^T Z_{2,t}$$

Given that all eigenvalues in Λ_1 are less than one in absolute value we know the AR process is stable and as such

$$\lim_{T \to 0} \Lambda_1^T \to 0$$

The same does not hold for the second expression, which is partitioned into the unstable eigenvalues. Because the eigenvalues in Λ_2 all lie outside the unit circle, $E_t Z_{2,t+T} \to \infty$ as T grows larger, which would be incompatible with a stationary solution. This is when the transversality conditions become useful. To see this consider what Γ^{-1} is:

$$\Gamma^{-1} = \underbrace{\begin{bmatrix} \underbrace{G_{11}}_{(Qxn)} & \underbrace{G_{12}}_{(Qxm)} \\ \underbrace{G_{21}}_{(Bxn)} & \underbrace{G_{22}}_{(Bxm)} \end{bmatrix}}_{(Q+B)x(n+m)}$$

Since $Q + B = n + m$ this obviously is still a square matrix, even when 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 and m state variables. In long hand, each Z would be:

$$\begin{aligned} \underbrace{Z_{1,t}}_{(Qx1)} &= \underbrace{G_{11}}_{(Qxn)} \underbrace{x_{1,t}}_{(nx1)} + \underbrace{G_{12}}_{(Qxm)} \underbrace{x_{2,t}}_{(mx1)} \\ \underbrace{Z_{2,t}}_{(Bx1)} &= \underbrace{G_{21}}_{(Bxn)} \underbrace{x_{1,t}}_{(nx1)} + \underbrace{G_{22}}_{(Bxm)} \underbrace{x_{2,t}}_{(mx1)} \end{aligned}$$

where G_{11}, G_{12} are both stable and G_{21}, G_{22} both unstable.

SEGUIR con notes DE ERIC SIMS

3 Recursive Methods

4 Appendix

4.1 Log Linearization