Log-Linear Approximation and Model Solution

David N. DeJong
University of Pittsburgh

Spring 2008, Revised Spring 2010

Last time, we sketched the process of converting model environments into non-linear first-order systems of expectational difference equations. Generically, a given system can be expressed as

$$\Gamma \left(E_t z_{t+1}, z_t, \upsilon_{t+1} \right) = 0, \tag{1}$$

where it is understood that z_t is an $n \times 1$ vector of stationary variables, and υ_t is an $m \times 1$ vector of structural shocks.

Rewriting forecasted variables as the composition of ex post realizations and forecast errors, we introduce expectations errors into the system, which becomes

$$\Gamma\left(z_{t+1}, z_t, v_{t+1}, \eta_{t+1}\right) = 0, \tag{2}$$

where η_t is an $r \times 1$ vector of expectations errors. Note that $\eta_t = f\left(v_t\right)$; i.e., expectations errors arise from the realization of shocks.

Overview of the Approximation/Solution Process

Step 1: Calculate the steady state value of z_t, denoted $\bar{z}$ (if it exists). The steady state solves

$$\Gamma\left(\bar{z}, \bar{z}, 0\right) = 0.$$

Overview, cont.

Step 2 (Approximation): Convert $\Gamma\left(z_{t+1}, z_t, \upsilon_{t+1}, \eta_{t+1}\right) = 0$ into a linear system of the form

$$A x_{t+1} = B x_t + C \upsilon_{t+1} + D \eta_{t+1}, \tag{3}$$

where x_t represents a deviation of z_t from $\bar{z}$. Using linear approximation,

$$x_{it} = z_{it} - \overline{z_i};$$

using log-linear approximation,

$$x_{it} = \ln \frac{z_{it}}{\overline{z_i}}.$$

Overview, cont.

Step 3 (Solution): Obtain a solution of the linear system of the form

$$x_{t+1} = Fx_t + Gv_{t+1}. \tag{4}$$

Notes

- Introduction of the observation errors u_t is postponed until the solution process is completed.
- Higher-order systems can be accommodated by converting to first-order form. For example, the p^{th}-order equation

$$\omega_{t+1} = \rho_1\omega_t + \rho_2\omega_{t-1} + ... + \rho_p\omega_{t-p+1}$$

may be written in first-order form as

$$\begin{bmatrix} \omega_{t+1} \\ \omega_t \\ \vdots \\ \omega_{t-p+2} \end{bmatrix} - \begin{bmatrix} \rho_1 & \rho_2 & \cdots & \cdots & \rho_p \\ 1 & 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & \cdots & \cdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \begin{bmatrix} \omega_t \\ \omega_{t-1} \\ \vdots \\ \omega_{t-p+1} \end{bmatrix} = 0,$$

or more compactly, as

$$x_{t+1} - \Pi x_t = 0, \qquad x_{t+1} = [\omega_{t+1}, \omega_t, ..., \omega_{t-p+2}]'.$$

Approximation

Textbook reference: Ch. 2.1, pp. 11-16.

We begin the approximation step by focusing on the deterministic behavior embodied in (1). Specifically, we set $v_t = 0 \; \forall t$, and focus on the system expressed as

$$\Psi\left(z_{t+1}, z_t\right) = 0, \tag{5}$$

where note that E_t has been dropped, in anticipation of the exploitation of Leibniz' Rule. Doing so will yield the matricies A and B. As we shall see, the matricies C and D are typically constructed trivially by hand.

We will achieve approximation using Taylor Series expansions.

Linearization in Levels

The Taylor Series expansion of (5) about $\overline{z}$ is given by

$$0 \approx \Psi\left(\overline{z}\right) + \frac{\partial \Psi}{\partial z_t}(\overline{z}) \times \left(z_t - \overline{z}\right) + \frac{\partial \Psi}{\partial z_{t+1}}(\overline{z}) \times \left(z_{t+1} - \overline{z}\right).$$

If z_t is univariate, $\frac{\partial \Psi}{\partial z_t}$ is a single value; in general, $\frac{\partial \Psi}{\partial z_t}$ is the $n \times n$ Jacobian matrix of $\Psi\left(z_{t+1}, z_t\right)$ with respect to z_t, evaluated at $\overline{z}$. That is, the $(i,j)^{th}$ element of $\frac{\partial \Psi}{\partial z_t}(\overline{z})$ is the derivative of the i^{th} equation in (5) with respect to the j^{th} element of z_t.

From (3), note that $A = \frac{\partial \Psi}{\partial z_{t+1}}(\overline{z})$ and $B = -\frac{\partial \Psi}{\partial z_t}(\overline{z})$.

Linearization in Logs

Here we begin by expressing $\Psi\left(z_{t+1}, z_t\right) = 0$ in terms of logged values of z.

For illustrative purposes, suppose the system is univariate and given by

$$z_{t+1} = f(z_t).$$

Taking logs and noting $z_t = e^{\ln z_t}$, the system becomes

$$\ln z_{t+1} = \ln\left[f(e^{\ln z_t})\right].$$

Then approximating,

$$\ln z_{t+1} \approx \ln\left[f(\bar{z})\right] + \frac{f'(\bar{z})\bar{z}}{f(\bar{z})}\left(\ln(z_t) - \ln(\bar{z})\right),$$

or since $\ln\left[f(\bar{z})\right] = \ln \bar{z}$,

$$\ln\left(\frac{z_{t+1}}{\bar{z}}\right) \approx \frac{f'(\bar{z})\bar{z}}{f(\bar{z})}\left(\ln(\frac{z_t}{\bar{z}})\right).$$

Log-Lin., cont.

More generally, begin by reexpressing $\Psi\left(z_{t+1}, z_t\right) = 0$ as

$$\Psi_1(z_{t+1}, z_t) = \Psi_2(z_{t+1}, z_t), \tag{6}$$

and again using the identity $z_t = e^{\ln z_t}$, taking logs of (6) and rearranging yields

$$\ln \Psi_1(e^{\ln z_{t+1}}, e^{\ln z_t}) - \ln \Psi_2(e^{\ln z_{t+1}}, e^{\ln z_t}) = 0. \tag{7}$$

The first-order Taylor Series approximation of this converted system yields the log-linear approximation we seek.

Log-Lin., cont.

The approximation for the first term:

$$\begin{aligned}\ln \Psi_1(z_{t+1}, z_t) &\approx \ln\left[\Psi_1(\bar{z})\right] + \frac{\partial \ln\left[\Psi_1\right]}{\partial \ln(z_t)}(\bar{z}) \times \left[\ln(\frac{z_t}{\bar{z}})\right] \\ &+ \frac{\partial \ln\left[\Psi_1\right]}{\partial \ln(z_{t+1})}(\bar{z}) \times \left[\ln(\frac{z_{t+1}}{\bar{z}})\right],\end{aligned}$$

where $\frac{\partial \ln[\Psi_1]}{\partial \ln(z_t)}(\bar{z})$ and $\frac{\partial \ln[\Psi_1]}{\partial \ln(z_{t+1})}(\bar{z})$ are $n \times n$ Jacobian matrices. Likewise for $\ln \Psi_2(z_{t+1}, z_t)$.

Log-Lin., cont.

Given these approximations,

$$\begin{aligned} A &= \left[\frac{\partial \ln\left[\Psi_1\right]}{\partial \ln(z_{t+1})}(\bar{z}) - \frac{\partial \log\left[\Psi_2\right]}{\partial \log(z_{t+1})}(\bar{z})\right], \\ x_t &= \ln(\frac{z_{t+1}}{\bar{z}}). \end{aligned}$$

Example

Consider a three-equation subsystem of the RBC model:

$$\begin{aligned} y_t &= c_t + i_t \\ y_t &= z_t k_t^{\alpha} n_t^{1-\alpha} \\ \ln z_{t+1} &= (1-\rho)\ln z_0 + \rho \ln z_t + \varepsilon_t . \end{aligned}$$

Converting into the form of (7),

$$\begin{aligned} \ln y_t - \ln\left[\exp(\ln c_t) + \exp(\ln i_t)\right] &= 0 \\ \ln y_t - \ln z_t - \alpha \ln k_t - (1-\alpha)\ln n_t &= 0 \\ \ln z_{t+1} - (1-\rho)\ln z_0 - \rho \ln z_t &= 0. \end{aligned}$$

Example, cont.

Defining

$$x_t = \left[\ln \frac{y_t}{\bar{y}} \;\; \ln \frac{c_t}{\bar{c}} \;\; \ln \frac{i_t}{\bar{i}} \;\; \ln \frac{n_t}{\bar{n}} \;\; \ln \frac{k_t}{\bar{k}} \;\; \ln \frac{z_t}{\bar{z}} \right]',$$

the corresponding rows of A, B are

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$B = \begin{bmatrix} 1 & \frac{-\bar{c}}{\bar{y}} & \frac{-\bar{i}}{\bar{y}} & 0 & 0 & 0 \\ 1 & 0 & 0 & -(1-\alpha) & -\alpha & -1 \\ 0 & 0 & 0 & 0 & 0 & \rho \end{bmatrix}.$$

Gradient Procedures

As an alternative to constructing A and B by hand, as in the above example, consider the use of a numerical gradient procedure. In GAUSS, the relevant command reference is `gradp`. This command computes the gradient vector (for a single-value function) or Jacobian (for a vector-value function) defined in a procedure.

The following example code demonstrates how to construct A and B in this alternative manner.

Grad. Procs, cont.

- Predefined variables: number of variables `nvars`, vector of parameters `p`, logged steady state values `xbar`, and indicator variable `tme`. In the proc, y, c, etc. are understood to represent logged values of y, c, etc.
- Both time-$(t+1)$ and time-t variables are set to logged ss values. When tme $== 0$, derivatives are taken with respect to $t+1$ variables; when tme $== 1$, derivatives are taken with respect to t variables.
- In the code, TFP z is referenced as a, and z references the vector-value function.

Grad. Procs, cont.

```
proc loglin(x);
local y, ylag, c, clag, i, ilag, n, nlag, k, klag, a, alag, alp, rh, z;
    alp = p[1]; rh = p[5];
    if tme ==0;     //Build A
        y = x[1]; c = x[2]; i = x[3]; n = x[4]; k = x[5]; a = x[6];
        ylag = xbar[1]; clag = xbar[2]; ilag = xbar[3]; nlag = xbar[4];
        klag = xbar[5]; alag = xbar[6];
    else;     //Build B
        y = xbar[1]; c = xbar[2]; i = xbar[3]; n = xbar[4];
        k = xbar[4]; a = xbar[5];
        ylag = x[1]; clag = x[2]; ilag = x[3]; nlag = x[4]; klag = x[5]; alag = x[6];
    endif;
    z = zeros(nvars,1);
    z[1] = y - ln( exp(c) + exp(i) );
    z[2] = y - a - alp*k - (1-alp)*n;
    z[3] = a - rh*alag;
    retp(z);
endp;
```

Grad. Procs, cont.

To call the proc:

```
tme = 0;
amat = gradp(&loglin,xbar);
tme = 1;
bmat = gradp(&loglin,xbar);
bmat = -1*bmat;
```

Exercise: Extend `loglin(x)` for the fully specified RBC model.

Solution

Textbook reference: Ch. 2.2, pp. 17-30.

There are many alternative approaches to obtaining linear approximations to systems of non-linear expectational difference equations. Here we will focus on the method developed by Sims (2001, *Computational Economics*).

Assuming the existence of saddle-path equilibria, solution methods in general work by ‘decoupling’ the system into stable and unstable components. Restrictions are then imposed on the unstable components so that their influence on the dynamics of the system is eliminated. This amounts to the imposition of saddle-path restrictions.

Solution, cont.

To gain intuition behind how decoupling works, it is useful to work with a deterministic version of the linearized model:

$$Ax_{t+1} = Bx_t.$$

(Details on the fully specified model are given in Section 2.2; refer to the posted Errata for two critical corrections.)

Solution, cont.

Decoupling begins with the execution of the QZ factorization of $(A, \quad B)$ (`GAUSS` code has been developed for this purpose by Paul Soderlind). This involves the calulation of matricies $(Q, \quad Z)$ such that

$$\begin{aligned} A &= Q'\Lambda Z', \qquad B = Q'\Omega Z' \\ QQ' &= ZZ' = I, \end{aligned}$$

with $(\Lambda, \quad \Omega)$ upper triangular. The matricies $(\Lambda, \quad \Omega)$ contain the generalized eigenvalues of $(A, \quad B)$ along their horizontal axes. They are organized such that the eigenvalues are increasing in moving from left to right.

Solution, cont.

Given the existence of a unique saddle-path equilibrium, the number of explosive eigenvalues contained in $(\Lambda, \quad \Omega)$ will equal the number of control variables included in the system. Let n_c denote the number of controls, and n_s the number of state variables, so that $n_s + n_c = n$.

Since $(\Lambda, \quad \Omega)$ are upper-triangular, they can be partitioned into non-explosive and explosive blocks as

$$\begin{aligned} \Lambda &= \begin{bmatrix} \Lambda_{11} & \Lambda_{12} \\ 0 & \Lambda_{22} \end{bmatrix} \\ \Omega &= \begin{bmatrix} \Omega_{11} & \Omega_{12} \\ 0 & \Omega_{22} \end{bmatrix}, \end{aligned}$$

where Λ_{11} is $n_s \times n_s$, Λ_{12} is $n_s \times n_c$, 0 is $n_c \times n_s$, and Λ_{22} is $n_c \times n_c$. Likewise for Ω.

Solution, cont.

Define

$$z_t = Z' x_t,$$

so that the elements of z_t are linear combinations of the elements of x_t. Then

$$\begin{aligned} A x_{t+1} &= Q' \Lambda Z' x_{t+1} = Q' \Lambda z_{t+1}, \\ B x_t &= Q' \Omega Z' x_t = Q' \Omega z_t, \\ Q' \Lambda z_{t+1} &= Q' \Omega z_t. \end{aligned}$$

Since $QQ' = I$, premultiplying by Q yields

$$\Lambda z_{t+1} = \Omega z_t.$$

Solution, cont.

The partitioned system is

$$\left[\begin{array}{cc} \Lambda_{11} & \Lambda_{12} \\ 0 & \Lambda_{22} \end{array}\right]\left[\begin{array}{c} z_{1t+1} \\ z_{2t+1} \end{array}\right]=\left[\begin{array}{cc} \Omega_{11} & \Omega_{12} \\ 0 & \Omega_{22} \end{array}\right]\left[\begin{array}{c} z_{1t} \\ z_{2t} \end{array}\right],$$

with z_{1t} $n_s \times 1$ and z_{2t} $n_c \times 1$.

Saddle-path stability therefore requires

$$z_{2t}=0 \quad \forall t,$$

which amounts to the satisfaction of n_c equations through the choice of n_c controls.

Solution, cont.

Given satisfaction of the saddle-path conditions, the system evolves as

$$\Lambda_{11} z_{1t+1} = \Omega_{11} z_{1t}.$$

Inverting Λ_{11}, we have

$$z_{1t+1} = \Lambda_{11}^{-1} \Omega_{11} z_{1t}.$$

Defining $Z_{.1}$ as the $n \times n_s$ matrix containing the first through $n_s th$ columns of Z, the system may be written as

$$Z_{.1}' x_{t+1} = \Lambda_{11}^{-1} \Omega_{11} Z_{.1}' x_t.$$

Premultiplying by $Z_{.1}$, and recalling that $Z_{.1} Z_{.1}' = I$, we have

$$\begin{aligned} x_{t+1} &= Z_{.1} \Lambda_{11}^{-1} \Omega_{11} Z_{.1}' x_t \\ &= F x_t. \end{aligned}$$

Solution, cont.

Note #1: For the generalized case in which variables have non-zero steady states, so that

$$Ax_{t+1} = Bx_t + E + C\upsilon_{t+1} + D\eta_t,$$

the saddle-path restriction becomes

$$\begin{aligned} z_{2t} &\equiv z_2 = -\sum_{i=0}^{\infty} M^i \Omega_{22}^{-1} Q_2 E_2, \\ M &= \Omega_{22}^{-1} \Lambda_{22}, \end{aligned}$$

as in (2.43) in the text. When $E = 0$, $z_{2t} = 0$, as above.

Solution, cont.

Note #2: When stochastic uncertainty is introduced in the model, beyond the imposition of the saddle-path restriction

$$z_{2t} \equiv z_2 = -\sum_{i=0}^{\infty} M^i \Omega_{22}^{-1} Q_2 E_2,$$

we must impose a restriction between the stochastic innovations $\{\upsilon_t\}$ and the expectational errors $\{\eta_t\}$. This amounts to the calculation of Φ in the expression

$$Q_1 D = \Phi Q_2 D,$$

as in (2.45) in the text.

Solution, cont.

Exercise #1: Using the extended version of `loglin(x)` completed in the previous exercise, generate the matricies $(A, \quad B, \quad C, \quad D)$ for the fully specified RBC model, and then implement Sims' solution method to obtain the approximate solution

$$\begin{aligned} x_{t+1} &= Fx_t + G\upsilon_t \\ &= Fx_t + e_t. \end{aligned}$$

Exercise #2: Using the approximated solution, construct impulse response functions tracing the reaction over time of each variable in response to an innovation to TFP. Examine the sensitivity of these functions to alternative parameterizations of the model. [Reference: Chapter 4, Sections 1 and 2, pp. 55-79.]

Solution, cont.

Exercise #3: Consider the following specialization of the One Tree Model:

$$
\begin{aligned}
p_t &= \beta E_t \left(p_{t+1} + d_{t+1} \right) \\
d_{t+1} &= \rho d_t + \varepsilon_{t+1}.
\end{aligned}
$$

Defining $x_t = \begin{bmatrix} p_t & d_t \end{bmatrix}'$, map the model into the form

$$A x_{t+1} = B x_t + E + C \upsilon_{t+1} + D \eta_t.$$

Solution, cont.

Then implementing Sims' solution method, show that the saddle-path stability requirement $z_{2t} = 0$ amounts to the restriction

$$\begin{aligned} p_t &= \theta d_t, \\ \theta &= \frac{\rho\beta}{1-\rho\beta}, \end{aligned}$$

and that

$$\Phi = -\frac{\theta}{\rho},$$

as we have derived analytically.