

Simulating Linear Rational Expectations Systems

David Aadland and Kevin X.D. Huang *

August 2004

Abstract

In this note, we describe an alternative method for solving and simulating artificial data from dynamic linear rational expectations models. The advantage of this solution algorithm over existing ones is its simplicity, ease in handling systems with many leads and lags, and lack of need to distinguish between predetermined and jump variables.

JEL Codes: C63, C88

Keywords: Dynamic, Linear, Rational Expectations, Simulation

PRELIMINARY – DO NOT CITE WITHOUT PERMISSION OF AUTHORS

*David Aadland is an assistant professor of economics at the University of Wyoming, PO Box 3985, Laramie, WY, 82071-3985, USA; 307-766-4931 (phone); 307-766-5090 (fax); aadland@uwyo.edu (e-mail). Kevin X.D. Huang is a senior economist at the Federal Reserve Bank of Kansas City, 925 Grand Boulevard, Kansas City, MO 64198, USA; 816-881-2701 (phone); 816-881-2199 (fax); kevin.huang@kc.frb.org (e-mail). The views expressed herein are those of the authors and do not necessarily reflect the views of the Federal Reserve Bank of Kansas City or the Federal Reserve System.

1 Introduction

This note describes an alternative method for solving and simulating systems of linear rational expectational difference equations. The method improves on existing algorithms (e.g., Blanchard and Kahn, 1980; Farmer 1995; Sims, 2003) because it does not require the user to distinguish between free and jump variables and easily handles systems with multiple expectational leads, such as Taylor's (1980) staggered wage model. The method of Blanchard and Kahn, for example, requires the user to repeatedly apply the solution algorithm in order to sequentially reduce the order of the system.

Another advantage of our method is its transparency. The algorithm simply stacks each block of m expectational difference equations from $t = 1, \dots, T$ in a large matrix, using the sparse nature of the matrices and recent advances in computing speed to determine the equilibrium solution. Initial and terminal conditions pin down the endpoints of the solution while recursions on the main equations determine the body of the solution. The algorithm can generate responses to one-time shocks or ensemble averages of first, second or higher moments produced from simulations to a full set of shocks. Gauss and matlab codes to implement the algorithm are freely available.¹

2 General Model

Consider a system of m linear expectational difference equations of the form

$$A\bar{y}_t + B_t\hat{y}_{t+1} + C\bar{x}_t + D_t\hat{x}_{t+1} = 0 \quad (1)$$

where $t = 1, \dots, T$; \bar{y}_t is a $(\bar{m} \times 1)$ vector of endogenous (either jump or predetermined) variables; and ${}_t\hat{y}_{t+1} = E(\hat{y}_{t+1}|\Omega_t)$ is a $(\hat{m} \times 1)$ vector defined as the mathematical expectation of \hat{y}_{t+1} conditioned on the set Ω_t , which includes all information dated time t and earlier. Lags and further leads of the endogenous variables can readily be incorporated into \bar{y}_t and ${}_t\hat{y}_{t+1}$. For example, if p lags are required then $\bar{m} = m(1 + p)$ and $\bar{y}_t = (y_{t-p}, \dots, y_{t-1}, y_t)'$, where y_t includes m distinct endogenous variables. If q leads are required then $\hat{m} = mq$ and ${}_t\hat{y}_{t+1} = ({}_ty_{t+1}, {}_ty_{t+2}, \dots, {}_ty_{t+q})'$. The coefficient matrix A is of dimension $(m \times \bar{m})$, B is of dimension $(m \times \hat{m})$, C is of dimension $(m \times \bar{n})$ and D is of dimension $(m \times \hat{n})$. The

¹Gauss and matlab codes can be found at www.uwo.edu/aadland/research/businesscycles/.

exogenous variables are given by the $(\hat{n} \times 1)$ vector ${}_t\hat{x}_{t+1} = E(\hat{x}_{t+1}|\Omega_t)$ and the $(\bar{n} \times 1)$ vector \bar{x}_t , which follows

$$\bar{x}_{t+1} = F\bar{x}_t + s_{t+1}$$

where s_{t+1} is a white-noise vector and F is of dimension $(\bar{n} \times \bar{n})$. Lags and further leads of the exogenous variables can similarly be incorporated into \bar{x}_t and ${}_t\hat{x}_{t+1}$. It is convenient to rewrite (1) as

$$Mz_t = -Nw_t \tag{2}$$

where $M = (A, B)$, $N = (C, D)$, $z_t = (\bar{y}_{t,t} \hat{y}_{t+1})'$ and $w_t = (\bar{x}_{t,t} \hat{x}_{t+1})'$. All variables are expressed as deviations from their steady-state values.

3 Solution Algorithm

3.1 Incorporating the Initial and Terminal Conditions

The solution algorithm relies on the initial conditions $z_t = 0$ for $t \leq 0$ (i.e., prior to the shocks having been turned on) and the terminal conditions $z_t = 0$ for $t \geq T_*$ (i.e., a sufficient period after the shocks have been turned off). It is convenient to write the entire mT equations from (2) in a single matrix equation

$$\bar{M}z = -\bar{N}w + \Delta$$

where $z = (y_1, y_2, \dots, y_T)'$ and $w = (x_1, x_2, \dots, x_T)'$. The expectational errors, Δ , are stacked in a $(T \times 1)$ vector given by

$$\Delta = \begin{bmatrix} \bar{M}(1, \cdot)\delta'_{1,z} + \bar{N}(1, \cdot)\delta'_{1,w} \\ \bar{M}(2, \cdot)\delta'_{2,z} + \bar{N}(2, \cdot)\delta'_{2,w} \\ \vdots \\ \bar{M}(T, \cdot)\delta'_{T,z} + \bar{N}(T, \cdot)\delta'_{T,w} \end{bmatrix}$$

where $\bar{M}(r, \cdot)$ represents row r of \bar{M} , $\delta'_{r,z} \equiv z - E(z|\Omega_r)$ is the conditional expectation error, and similarly for w . The exogenous vector w can be expressed in terms of the shocks

according to $w = \bar{F}^{-1}s$, where

$$\bar{F} = \begin{bmatrix} -I & F & 0 & \cdots & 0 & 0 \\ 0 & -I & F & & 0 & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & 0 & \ddots & F & 0 \\ 0 & 0 & 0 & \cdots & -I & F \end{bmatrix}$$

and $s = (s_1, s_2, \dots, s_T)'$. Substitution for w gives

$$\bar{M}z = -\bar{N}\bar{F}^{-1}s + \Delta. \quad (3)$$

The coefficient matrices \bar{M} and \bar{N} are partitioned into three parts: the initial conditions, main body and terminal conditions

$$\bar{M} = \begin{bmatrix} M_0 \\ M_1 \\ M_T \end{bmatrix} \quad \text{and} \quad \bar{N} = \begin{bmatrix} N_0 \\ N_1 \\ N_T \end{bmatrix}.$$

The coefficient matrix M_0 associated with the initial conditions is of dimension $(mp \times mT)$ and is given by

$$M_0 = \begin{bmatrix} M_{-mp} & 0 & \cdots & 0 \\ M_{-m(p-1)} & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ M_{-m} & 0 & \cdots & 0 \end{bmatrix}$$

where M_{-r} indicates that the first r columns of M have been deleted. The coefficient matrix M_1 is of dimension $(m(T - q - p) \times mT)$ and is given by

$$M_1 = \begin{bmatrix} M & 0 & \cdots & 0 \\ 0 & M & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & M \end{bmatrix}.$$

Lastly, the coefficient matrix M_T associated with the terminal conditions is of dimension $(mq \times mT)$ and is given by

$$M_T = \begin{bmatrix} 0 & \cdots & 0 & I_m & -I_m & 0 & \cdots & 0 \\ 0 & \cdots & 0 & I_m & 0 & -I_m & & 0 \\ \vdots & & \vdots & & \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & I_m & 0 & 0 & \cdots & -I_m \end{bmatrix}.$$

The structure of \bar{N} is analogous.

3.2 A Simple Illustration of the Matrix System

Consider the following illustration with $m = 2$ endogenous variables ($y^{(1)}$ and $y^{(2)}$) and $n = 1$ exogenous variable (x)

$$a_{11}y_{t-1}^{(1)} + a_{12}y_t^{(1)} + b_{11t}y_{t+1}^{(2)} + b_{12t}y_{t+2}^{(2)} + c_1x_t = 0 \quad (4a)$$

$$a_{21}y_t^{(2)} + b_{21t}y_{t+1}^{(1)} + c_2x_{t-1} = 0 \quad (4b)$$

and $x_{t+1} = fx_t + s_{t+1}$ where s_{t+1} is white noise. This can be rewritten in the form (2) by defining the endogenous and exogenous vectors as

$$z'_t = \begin{bmatrix} y_{t-1}^{(1)} & y_{t-1}^{(2)} & y_t^{(1)} & y_t^{(2)} & {}_t y_{t+1}^{(1)} & {}_t y_{t+1}^{(2)} & {}_t y_{t+2}^{(1)} & {}_t y_{t+2}^{(2)} \end{bmatrix} \text{ and } w'_t = \begin{bmatrix} x_{t-1} & x_t \end{bmatrix}$$

and the coefficient matrices as

$$M = \begin{bmatrix} a_{11} & 0 & a_{12} & 0 & 0 & b_{11} & 0 & b_{12} \\ 0 & 0 & 0 & a_{21} & b_{21} & 0 & 0 & 0 \end{bmatrix} \text{ and } N = \begin{bmatrix} 0 & c_1 \\ c_2 & 0 \end{bmatrix}.$$

Assuming $T = 5$, we have

$$\bar{M} = \begin{bmatrix} M_0 \\ M_1 \\ M_T \end{bmatrix} = \begin{bmatrix} a_{12} & 0 & 0 & b_{11} & 0 & b_{12} & 0 & 0 & 0 & 0 \\ 0 & a_{21} & b_{21} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ - & - & - & - & - & - & - & - & - & - \\ a_{11} & 0 & a_{12} & 0 & 0 & b_{11} & 0 & b_{12} & 0 & 0 \\ 0 & 0 & 0 & a_{21} & b_{21} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{11} & 0 & a_{12} & 0 & 0 & b_{11} & 0 & b_{12} \\ 0 & 0 & 0 & 0 & 0 & a_{21} & b_{21} & 0 & 0 & 0 \\ - & - & - & - & - & - & - & - & - & - \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix} \text{ and } z = \begin{bmatrix} y_1^{(1)} \\ y_1^{(2)} \\ y_2^{(1)} \\ y_2^{(2)} \\ y_3^{(1)} \\ y_3^{(2)} \\ y_4^{(1)} \\ y_4^{(2)} \\ y_5^{(1)} \\ y_5^{(2)} \end{bmatrix}.$$

4 Recursive Solution Algorithm

Our recursive solution algorithm works by first determining y_1 , using y_1 to determine y_2 , using y_1 and y_2 to determine y_3 , and so on and so forth. The iterations end when we reach y_T .

The first step in the solution algorithm is to solve (3) for z

$$z = -\bar{M}^{-1}\bar{N}\bar{F}^{-1}s + \bar{M}^{-1}\Delta$$

and then take expectations conditional on the information set Ω_1 , which in turn implies that $E(\bar{M}^{-1}\Delta|\Omega_1) = 0$ and $E(y_1|\Omega_1) = y_1$ (since \bar{M} will involve many zeros, the computational efficiency of the solution algorithm will be improved by the use of sparse matrix routines). This pins down the first m elements of z . In the second step, we eliminate the first m rows from (3), move all terms involving y_1 to the right-hand side, and incorporate them by appropriate expansions of \bar{N} and s . The reduced system can then be rewritten in the form of (3), where it is recognized that \bar{M} is now of dimension $((m-1)T \times (m-1)T)$, $z' = (y_2, y_3, \dots, y_T)$, \bar{N} is now of dimension $((m-1)T \times (n+m))$ and $s' = (s_1, s_2, \dots, s_T, y_1)$. By inverting \bar{M} and taking expectations conditional on Ω_2 , we then pin down y_2 . Further

iterations will sequentially determine the values for y_3, y_4, \dots, y_T .

The shocks s can be used to generate impulse response functions (IRFs) or full simulations. IRFs give the responses of the endogenous variables to a one-time unit shock in the disturbances. Unlike the full simulations, only one iteration of the solution algorithm is necessary because $E(y_s|\Omega_1) = y_s$ for $s = 1, \dots, T$. The full simulations can be used to calculate ensemble averages of various moments (means, standard deviations, etc.) for the endogenous variables. The algorithm allows for each element in the multivariate process s_t to have a distinct variance. For both IRFs and full simulations, the most important feature of the algorithm is that they be turned off sufficiently early such that when the terminal conditions are imposed, they no longer have any significant effect on the endogenous variables.

4.1 Example. Taylor's Staggered Wage Model

Next, we present a numerical example in order to contrast our algorithm with other solution techniques. Taylor (1979) proposed a model of staggered wages capable of generating endogenous persistence in unemployment, wages and prices. We calibrate Taylor's model and contrast IRFs from our algorithm with those produced using Farmer's (1995) method.

Taylor's model is comprised of three equations – a wage-setting equation, aggregate demand curve and an aggregate wage index:

$$x_t = b_1 x_{t-1} + b_2 x_{t-2} + d_1 \hat{x}_{t+1} + d_2 \hat{x}_{t+2} + \gamma(y_t + d_1 \hat{y}_{t+1} + d_2 \hat{y}_{t+2}) + \epsilon_t \quad (5)$$

$$y_t = -\beta \omega_t + v_t \quad (6)$$

$$\omega_t = (x_t + x_{t-1} + x_{t-2})/3 \quad (7)$$

where x_t is the contract wage set for three periods, y_t is the output gap, ω_t is the wage index, ϵ_t is an i.i.d. wage shock and v_t is an i.i.d. aggregate demand shock. Hats over variables indicate expectations based on time t information (i.e., $\hat{x}_{t+1} = E[x_{t+1}|\Omega_t]$). All variables are measured in logs and deviations from steady state.

4.1.1 Farmer's Method

For Farmer's method, it will be convenient to simplify the model by substituting (5) and (6) into (7), to get

$$x_t = a_1 x_{t-1} + a_2 x_{t-2} + a_3 \hat{x}_{t+1} + a_4 \hat{x}_{t+2} + a_5 \epsilon_t + a_6 \nu_t,$$

where

$$\begin{aligned} a_1 &= [3b_1 - \gamma\beta(1 + d_1)]/\lambda; \\ a_2 &= [3b_2 - \gamma\beta]/\lambda; \\ a_3 &= [3d_1 - \gamma\beta(d_1 + d_2)]/\lambda; \\ a_4 &= d_2[3 - \gamma\beta]/\lambda; \\ a_5 &= 1/3\lambda; \quad a_6 = \gamma/3\lambda; \\ \lambda &= 3 + \gamma\beta(1 + d_1 + d_2). \end{aligned}$$

Next, we write the model in forward-looking vector autoregressive form, $\Gamma_1 \theta_t = \Gamma_2 \theta_{t+1} + \Gamma_3 s_t$, where

$$\Gamma_1 = \begin{bmatrix} -a_2 & -a_1 & 1 & -a_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad \Gamma_2 = \begin{bmatrix} 0 & 0 & 0 & a_4 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}; \quad \Gamma_3 = \begin{bmatrix} a_5 & a_6 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

and $\theta_t = \begin{bmatrix} x_{t-2} & x_{t-1} & x_t & x_{t+1} \end{bmatrix}$.

Assigning the parameters values $b_1 = d_1 = 0.4$, $b_2 = d_2 = 0.1$, $\beta = 0.4$ and $\gamma = 0.2$, the eigenvalues of $\Gamma = \Gamma_1^{-1} \Gamma_2$ are

$$eigs(\Gamma) = (-7.057, 1.581, 0.701, -0.170).$$

This is a "regular" or determinate model because the number of eigenvalues inside the unit circle equals the number of free variables (i.e., x_t and x_{t+1}). Diagonalizing Γ as $Q\Lambda Q^{-1}$, iterating forward on the stable roots, and taking expectations conditional on time

t information, allows us to reduce the order of the system by one and rewrite it as

$$\begin{bmatrix} q_{11} & q_{12} & q_{13} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{t-2} \\ x_{t-1} \\ x_t \end{bmatrix} = \begin{bmatrix} 0 & 0 & -q_{14} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ x_t \\ x_{t+1} \end{bmatrix} + \text{innovations},$$

where q_{ij} represents the (i, j) element of Q^{-1} . Repeating the procedure above produces the final-form system from which IRFs can be calculated.

4.1.2 AH Method

Taylor's model can be put in our framework by defining the following matrices

$$A = \begin{bmatrix} 0 & b_2 & 0 & b_1 & 0 & 0 & 0 & -1 & \gamma \\ 0 & 0 & 0 & 0 & 0 & 0 & \beta & 0 & 1 \\ 0 & 1/3 & 0 & 0 & 1/3 & 0 & -1 & 1/3 & 0 \end{bmatrix};$$

$$B = \begin{bmatrix} 0 & d_1 & \gamma d_1 & 0 & d_2 & \gamma d_2 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}; N = \begin{bmatrix} 0 & 0 \\ 0 & -1 \\ 1 & 0 \end{bmatrix}$$

and

$$z'_t = \begin{bmatrix} \omega_{t-2} & x_{t-2} & y_{t-2} & \omega_{t-1} & x_{t-1} & y_{t-1} & \omega_t & x_t & y_t & \omega_{t+1} & x_{t+1} & y_{t+1} & \omega_{t+2} & x_{t+2} & y_{t+2} \end{bmatrix};$$

$$s'_t = \begin{bmatrix} \epsilon_t & v_t \end{bmatrix}.$$

Unlike Blanchard and Kahn or Farmer's methods, our algorithm is a one-shot game, even though the system involves expectations of variables more than one period ahead. Table 1 compares the IRFs generated from our algorithm and those from Farmer's method. For this example, the algorithms require similar computing time and produce identical IRFs although ours is much simpler to implement.

Table 1. Comparing Impulse Responses to One-Time Contract Shocks

<i>lag</i>	Farmer's Method			AH Method		
	ω	x	y	ω	x	y
1	0.40752113	1.2225634	-0.16300845	0.40752113	1.2225634	-0.16300845
2	0.60761532	0.60028257	-0.24304613	0.60761532	0.60028257	-0.24304613
3	0.74239837	0.40434917	-0.29695935	0.74239837	0.40434917	-0.29695935
4	0.41899554	0.25235487	-0.16759821	0.41899554	0.25235487	-0.16759821
25	0.00002864	0.00001692	-0.00001145	0.00002864	0.00001692	-0.00001146

5 Conclusion

In this note, we describe an alternative method for simulating artificial data from dynamic linear rational expectations models. The advantage of this algorithm over existing ones is its transparency, ease in handling systems with many leads and lags, and lack of need to distinguish between predetermined and jump variables.

References