Software for RE Analysis

Bennett T. McCallum

August 23, 2001 (Revised 02-17-04)

This writeup describes a set of Matlab files for formulating, solving, and analyzing linear rational expectations (RE) models. Some guidance to their installation and use is also provided. These are all Matlab files of the .m type, which must be used with the standard Matlab setup. The core file is solvek.m, which is an algorithm for calculating RE solutions, as described in McCallum (1998, 1999). It is a modified version of a program written by Paul Klein, whose more detailed analysis may be found in Klein (2000).

Although solvek.m is the core file, it is not the one that a user will work with. Instead he will spend much of his time writing or modifying model files, which specify the model to be solved by solvek.m. In addition, considerable time will be spent analyzing the properties of the model using files that conduct stochastic simulations and plot impulse response functions, based on a previously-obtained solution. The current package includes files called sim33p.m and impo.m for these purposes. A sample model file is included as well, to provide a starting place for the user. It is called modfile1.m. Some exercises to begin with are described below.

The first step is to install the .m files in the proper Matlab folder. This will differ from one computer to another; the point is that the files must be in the specified path. The following files should be installed: solvek.m, reorder.m, qzswitch.m, sim33p.m, autocor.m, impo.m, and modfile1.m. Also, if the computer's Matlab installation does not include the Control toolbox, it will be necessary to install dlsim.m and dimpulse.m. If the Control toolbox is on the computer, it is best not to install these files. It is important that the files are installed as .m files, not as text (.txt) files. It is suggested that the user print out the files modfile1.m, solvek.m, impo.m, and sim33p.m and read the introductions.

To begin, the user should try to run modfile1.m. To do so he simply types "modfile1" (without the quote marks and without .m) in the Command Window and then hits enter.[1] If the file runs properly, the cursor will move to the next line and give no output. To see if the model has been solved, the user can then enter "m" in the Command Window. A matrix, called m by solvek.m, will be printed out. It is the matrix denoted M in the introduction section of solvek.m. Thus, it is a crucial part of the solution. The other parts are the matrices N, P, and Q, which may be obtained by entering "n", "p", or "q" in the command window. If these steps are successful, the user should modify modfile1.m by changing the value of some parameter and then solving the modified model. For example, the policy-rule parameter mu1 could be changed. After changing it, modfile1.m must be saved, perhaps under a new name so as to also retain the unmodified file. Then enter the new name, print out the matrix m and note that its values have changed. More substantial changes will be discussed below.

Next the user should plot some impulse response functions. After solving a model, enter impo in the command window. If all goes well, a set of impulse response functions will appear on the screen. They are responses to a particular shock, which is specified in an early line of the impo.m file. Look at that file to see which shock is being used, then change it and observe the new impulse responses. In this regard, note that the file impo.m includes a "gtext" statement that causes a string of text to be included on the graph at a location chosen

[1] This is one way to run a .m file: simply enter its name, without the suffix, in the Command Window.

by the user (by positioning a moveable "+" mark with the mouse and then clicking). If this statement is active (i.e., not commented out of the file), this text must be positioned and clicked before the user can to proceed normally. The impo.m file is a very simple, homemade file for generating impulse functions. The user should examine it and learn how to add or change panels, or change which variables are plotted in the existing panels. It is possible to extend uses of the file in various ways, such as plotting responses to different shocks in various panels and even showing plots from two models in each panel.[2]

The file sim33p.m runs stochastic simulations of a solved model. The user must have agreement between it and the model file as to the numbering of the different shocks. Then, the standard deviations of the innovations to the various (AR(1)) shocks must be entered into the sim33p.m file, and also the list and names of the endogenous variables whose variance and covariances are desired. The introductory portion of sim33p.m should be studied carefully to see how this is done and to learn of the many options regarding what output is generated.[3] Note that the AR coefficients of the shock terms are specified in the model file, not in sim33p.m.

After successfully reaching this point, the user will want to begin learning how to specify models. A first step is to modify the model in modfile1.m in more substantial ways than merely changing the value of some parameter. In doing this, the user may find it helpful to refer to Appendix A below, which explains how to generate lagged variables, expectational variables, etc.[4] First, however, he needs to understand what format of the model is presumed by solvek.m. For this, one should study the introductory writeup lines of

[2] One complication is that including a predetermined variable in the list of those whose response is studied requires that simulated values are taken from the X matrix rather than the Y matrix.
[3] Again there are complications if a predetermined variable is in the list to be studied.
[4] Indeed, to understand the file modfile1.m, he will probably need to refer to this appendix.

solvek.m. Basically, the format is $AE_tx_{t+1} = Bx_t + Cz_t$ where x_t is a vector of endogenous variables, $x_t = [y_t' \; k_t']'$, where k_t is predetermined and y_t is non-predetermined. Also, z_t is a vector of exogenous variables generated by a multivariate AR(1) process. To specify a model, the user must decide the ordering[5] of the variables in x_t (and in z_t), specify the dimensions of A, B, and C, and enter their non-zero values. The number of variables in k_t must be correctly specified and also the coefficients of the AR(1) matrix, which is called phi in the program. The format can, despite appearances, accommodate virtually any specification of a linear RE model (by defining new variables that are lagged values of others, etc.).

Both modfile1.m and Appendix A use a programming trick to keep track of the location of variables in the x_t vector. Each variable is given a name that the user can remember, and then the position number in x_t for the variable is called by this name but with an i attached at its start. For example, the price level might be thought of as p in which case the statement ip = 3 would make p the third variable in x_t. The advantage is that the user then does not have to remember the index of the variables when entering the model's equations. He does this by specifying the non-zero coefficients in the A, B, and C matrices. He will usually work on one equation at a time so can easily remember the row index to be entered. Then for the column index, he does not enter a number directly but rather the "name" such as ip, which has been assigned a number before. Examination of the file modfile1.m should make this practice understandable.

---

[5] The ordering chosen for the x_t variables is irrelevant except that all y_t variables (non-predetermined) must come before the k_t (predetermined) variables.

It may be useful to write out, in macroeconomic notation, the structural equations of the model that are represented by modfile1.m. These are similar to those in the simplest model reported in McCallum (2001), but are not identical. The equations are:

(IS) $y_t = b_0 + b_1[R_t - E_t\Delta p_{t+1}] + E_t y_{t+1} + v_t$

(AS) $\Delta p_t = 0.5[E_t\Delta p_{t+1} + \Delta p_{t-1}] + \alpha \tilde{y}_t + u_t$

(MP) $R_t = (1 - \mu_3)[\mu_1 \Delta p_t + \mu_2 E_{t-1} \tilde{y}_t] + \mu_3 R_{t-1} + e_t$

(Def) $\tilde{y}_t = y_t - \bar{y}_t$

(exog) $\bar{y}_t = 0.95 \bar{y}_{t-1} + \varepsilon_t$

with v_t, u_t, e_t, and ε_t being white noise shocks. Parameter values are $b_0 = 0$, $b_1 = -0.5$, $\alpha = 0.02$, $\mu_0 = 0$, $\mu_1 = 1.99$, $\mu_2 = 0$, $\mu_3 = 0.8$.

References

Klein, Paul, "Using the generalized Schur form to solve a multivariate linear rational expectations model," Journal of Economic Dynamics and Control 24 (September 2000), 1405-1423.

McCallum, Bennett T., "Solutions to linear rational expectations models: a compact exposition," Economics Letters 61 (November 1998), 143-147.

__________________, "Role of the minimal state variable criterion in rational expectations models," International Tax and Public Finance 6 (November 1999), 621-639. Also in International Finance and Financial Crises: Essays in Honor of Robert P. Flood, Jr., edited by Peter Isard, Assaf Razin, and Andrew K. Rose, Kluwer Academic Publishing, 1999.

_________________, "Should monetary policy respond strongly to output gaps?" American

Economic Review Papers and Proceedings 91 (May 2001), 258-262.

Appendix A

Definition of variables for use in Matlab program solvek.m

B. T. McCallum, 2-15-99

Notation: $E_{t-j}x_{t+k}$ is denoted ejxk , except that 0 is not entered for j or k. Also, x_{t-j} is written as xlagj, except that xlag1 is written xlag.

1. Define x_{t-1} from x_t (xlag)

 A(r,ixlag) = 1; and specify that xlag is a predetermined variable
 B(r,ix) = 1;

2. Define E_tx_{t+1} (ex1)

 A(r,ix) = 1;
 B(r,iex1) = 1;

3. Define $E_{t-1}x_t$ (e1x) by lagging E_tx_{t+1}

 A(r,ie1x) = 1; and specify that e1x is predetermined
 B(r,iex1) = 1;

3a. Define $E_{t-1}x_t$ in one step (e1x)

 A(r,ie1x) = 1; e1x predetermined
 A(r,ix) = -1;

4. Define E_tx_{t+2} (ex2)

 A(r,iex1) = 1; this uses definition in 2 and law of iterated expectations
 B(r,iex2) = 1;

5. Define $E_{t-1}x_{t+1}$ (e1x1) by lagging E_tx_{t+2}

 A(r,ie1x1) = 1; and e1x1 predetermined
 B(r,iex2) = 1;

5a. Define $E_{t-1}x_{t+1}$ in one step (with E_tx_{t+1} already defined)

 A(r,ie1x1) = 1; e1x1 predetermined
 A(r,iex1) = -1;