[IrisToolbox] for Macroeconomic Modeling

Efficient Triangular Solution to Rational-Expectations Models and Its Forward Expansion

`jaromir.benes@iris-toolbox.com`

Summary

In this article, we show the algorithm implemented in the [IrisToolbox] for calculating a first-order approximate solution to rational-expectations (RE) models such that its transition matrix is upper triangular. The algorithm is efficient in the sense that no extra matrix factorization is needed in addition to the one that produces a more conventional solution. Furthermore, we also calculate the forward expansion of the resulting solution to include the effect of anticipated future shocks.

Introduction

We show an efficient algorithm for computing a first-order solution to rational-expectations (RE) models such that its transition matrix is upper triangular, or quasi triangular. The algorithm is efficient in the sense that no extra matrix factorization is needed in addition to the one that produces a more conventional solution. We argue that the triangularity, achieved by suitably transforming the state vector, is preferable and convenient for a number of reasons. Furthermore, we also calculate the forward expansion of the resulting solution to include the effect of expectations of future exogenous shocks.

Why is a triangular solution superior to the more conventional one with a general transition matrix? There are at least two broad reasons. First, handling systems with triangular transition matrices is computationally more efficient in many contexts; a notable example is solving a Lyapunov equation to obtain unconditional covariance matrices. Second, a triangular solution greatly simplifies the treatment of models with unit roots, such as automated detection of non-stationary variables or initialization of a Kalman filter. We discuss and exemplify this point later in more detail.

The article is organized as follows. Section 2 explains the underlying assumptions and shows a recursive-form solution with a triangular transition matrix and forward expansion. Section 3 illustrates the algorithm on an example unit-root model. Appendix 1 provides details of the solution algorithm.

Linear(ized) Model

We use a version of the algorithm proposed by Klein (2000)[1], which builds upon a generalized Schur (also known as QZ) decomposition when solving out for the expectations of endogenous variables. We though deviate from Klein in three important aspects:

- We transform the vector of predetermined (or backward-looking) variables so as to give rise to a triangular transition matrix in the resulting solution.
- We allow unit roots in the model and treat them correctly as *stable*, or *non-explosive*, roots from the point of view of saddle-path stability (contrary to footnote 8 on page 1410 in Klein, there is nothing problematic about unit roots).
- We provide explicit formulas for the effects of future expectations of exogenous inputs without assuming any specific process for them.

In fact, in the last two aspects, we simply return to the very origins of solving RE models laid by Blanchard and Kahn (1980)[2].

We consider the following form of linear, linearized, or log-linearized rational-expectations models:

$$A\,\mathrm{E}_t \begin{bmatrix} x^b_t \\ x^f_{t+1} \end{bmatrix} + B \begin{bmatrix} x^b_{t-1} \\ x^f_t \end{bmatrix} + C\,e_t = 0, \tag{Model}$$

where an $n_x \times 1$ vector of endogenous variables consists of x^b_t, an $n_b \times 1$ vector of *predetermined*, or backward-looking, variables for which $\mathrm{E}_t[x^b_t] = x^b_t$, with x^b_{t-1} being given, and x^f_t, an

$$n_f \times 1$$

vector of *non-predetermined*, or forward-looking, variables. Futhermore, e_t is an $n_e \times 1$ vector of exogenous processes, and $\mathrm{E}_t[\cdot]$ is a time t conditional expectations operator. The current realisation of the exogenous vector as well as its expectations, e_t and $\mathrm{E}_t[\,e_{t+k}]\$,\$k = 1, 2, \ldots$, are known at time t. We, however, do not introduce any further assumptions about the law of motion for e_t, apart from a stability requirement adopted from BK, condition (1c) on page 1305. Obviously, the first-order difference equation (Model) can easily accommodate systems with lags and leads higher than one by simply augmenting x^b_t and x^f_t with auxiliary, time-shifted, variables.

Solution

We construct a solution that has the following form:

$$\begin{bmatrix} x^f_t \\ \alpha_t \end{bmatrix} = \begin{bmatrix} 0 & M_f \\ 0 & M_\alpha \end{bmatrix} \begin{bmatrix} x^f_{t-1} \\ \alpha_{t-1} \end{bmatrix} + R_0\, e_t + R_1 \mathrm{E}_t[e_{t+1}] + \cdots R_k \mathrm{E}_t[e_{t+k}], x^b_t = U\alpha_t, \quad \text{(Solution)}$$

where α_t is a suitable transformation of the vector of predetermined variables such that M_α, and hence also the overall transition matrix of the system, is upper triangular. Note that the expectations of endogenous variables, $\mathrm{E}_t[x^f_{t+1}]$ translate, in general, into an infinite sum of expectations of exogenous processes, which are known by assumption.

First, we take for granted that the system matrices A and B satisfy a generalized saddle-path condition. In other words, they have exactly n_b generalized eigenvalues inside, or on, the unit circle (called *non-explosive*); recall that these also include unit roots), and exactly n_f eigenvalues outside the unit circle (called *explosive* roots); see section 5.3.1 in Klein (2000), and propositions 1 to 3 in Blanchard and Kahn (1980). Next, we can transform the vector of state variables and factorize the system (Model) using the Schur generalized decomposition as follows:

$$\mathrm{E}_t \begin{bmatrix} x^b_t \\ x^f_{t+1} \end{bmatrix} = Z\, \mathrm{E}_t \begin{bmatrix} s_{t+1} \\ u_{t+1} \end{bmatrix}, \quad \text{or} \quad \begin{bmatrix} x^b_{t-1} \\ x^f_t \end{bmatrix} = Z \begin{bmatrix} s_t \\ u_t \end{bmatrix}, \qquad \text{(Transform)}$$

and

$$S\, \mathrm{E}_t \begin{bmatrix} s_{t+1} \\ u_{t+1} \end{bmatrix} + T \begin{bmatrix} s_t \\ u_t \end{bmatrix} + D\, e_t = 0, \qquad \text{(Schur)}$$

where $D = Q\,C$, and $S = QAZ$ and $T = QBZ$ are both upper triangular (for complex Schur forms) or quasi-triangular (for real Schur forms) matrices, and Q and Z are unitary matrices. The new matrices can be split into blocks conformably with n_b and n_f (by quasi-triangular we mean a matrix with 1×1 and 2×2 blocks along the main diagonal, depending on the occurrence of real and complex eigenvalues, and with zeros below it. Whether we perform a complex or real Schur decomposition irrelevant for constructing the solution. The choice affects only the computational efficiency of the procedure),

$$S = \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix}, \quad T = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix}, \quad Z = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{12} \end{bmatrix}, \quad D = \begin{bmatrix} D_1 \\ D_2 \end{bmatrix},$$

and arranged so that the upper left blocks S_{11} and T_{11} contain only the non-explosive eigenvalues whereas S_{22} and T_{22} have only the explosive ones. Furthermore, if there are unit roots in the system, we concentrate them in the upper-left blocks of S_{11} and T_{11}; the reason for doing so becomes obvious later in the paper.

The procedure now consists of four simple steps:

1. solving the the lower, explosive, part of the transformed vector, u_t, using forward iterations;
2. finding a transformation α_t of the predetermined vector, x_t^b, such that it gives rise to a triangular transition matrix;
3. solving for the upper, non-explosive, part of the transformed vector, α_t, in recursive form;
4. solving for the vector of forward-looking variables, x_t^f.

First, we iterate the lower part of eq. (Schur) forward and get the following solution in which we retain the effect of all future expected residuals,

$$u_t = F e_t + G\,F\,\mathrm{E}_t[e_{t+1}] + G^2 F\,\mathrm{E}_t[e_{t+2}] + \cdots \tag{Unstable}$$

where

$$F = -(T_{22})^{-1} D_2, \quad G = -(T_{22})^{-1} S_{22},$$

cf. eq. (5.5) in Klein. For ease of notation, we introduce a conditional expectations operator, $(\phi_t)^k e_t = \mathrm{E}_t[e_{t+k}]$, and re-write (Unstable) as a polynomial in ϕ_t:

$$u_t = \left[\sum_{k=0}^{\infty} (G\,\phi_t)^k\right] F e_t. \tag{UnstablePolyn}$$

Second, we introduce $\alpha_t = (Z_{11})^{-1} x_t^b$, and denote $U := (Z_{11})^{-1}$ for future reference. The new vector α_t is backward-looking, or predetermined, by construction. We will see shortly that this particular transformation leads to a triangularized transition matrix.

Third, noting that from (Transform)

$$\mathrm{E}_t[s_{t+1}] = \alpha_t - U Z_{12}\,\mathrm{E}_t[u_{t+1}], \quad \text{or} \quad s_t = \alpha_{t-1} - U Z_{12}\,u_t,$$

we can re-write the upper part of eq. (Schur) as

$$S_{11}\alpha_t + (S_{12} - U Z_{12})\,\mathrm{E}_t[u_{t+1}] + T_{11}\alpha_{t-1} + (T_{12} - U Z_{12})\,u_t + D_1 e_t = 0.$$

After substituting for u_t and $\mathrm{E}_t[u_{t+1}]$ from (UnstablePolyn), we obtain the following process for α_t:

$$\alpha_t = M_\alpha\,\alpha_{t-1} + R_\alpha(\phi_t)\,e_t,$$

where $M_\alpha = -(S_{11})T_{11}$ is upper triangular (or quasi-triangular) by construction, and the coefficient matrices $R_{\alpha 0}$, $R_{\alpha 1}$, $R_{\alpha 2}$, $\ldots$ of the infinite polynomial

$$R_\alpha(\phi_t) = R_{\alpha,0} + R_{\alpha,1}\,\phi_t + R_{\alpha,2}\,(\phi_t)^2 + \cdots$$

can be easily calculated by evaluating the following polynominal expression up to any desired order:

$$R_\alpha(\phi_t) = -S_{11}^{-1} D_1 + \\ S_{11}^{-1} \left[(T_{11} U Z_{12} - T_{12}) + (S_{11} U Z_{12} - S_{12})\, \phi_t \right] \left[\sum_{k=0}^{\infty} (G\, \phi_t)^k \right] F.$$

We provide the formulas for the coefficient matrices below.

Fourth, we solve for the vector of forward-looking variables, x_t^f. Using (Transform), we get

$$x_t^f = Z_{21} s_t + Z_{22} u_t = Z_{21} \alpha_{t-1} + (Z_{22} - Z_{21} U Z_{12})\, u_t.$$

Denoting

$$M_f = Z_{21}, R_f(\phi_t) = (Z_{22} - Z_{21} U Z_{12}) \left[\sum_{k=0}^{\infty} (G\, \phi_t)^k \right] F,$$

we can now summarise the resulting dynamics of the model (Model) as follows:

$$\begin{bmatrix} x_t^f \\ \alpha_t \end{bmatrix} = M \begin{bmatrix} x_{t-1}^f \\ \alpha_{t-1} \end{bmatrix} + R(\phi_t)\, e_t x_t^b = Z_{11} \alpha_t,$$

where

$$M = \begin{bmatrix} 0 & M_f \\ 0 & M_\alpha \end{bmatrix}, \quad \text{and} \quad R(\cdot) = \begin{bmatrix} R_f(\phi_t) \\ R_\alpha(\phi_t) \end{bmatrix}.$$

Details of Coefficient Matrices

The coefficient matrices for the polynomial $R_\alpha(\phi_t)$ are given by

$$R_{\alpha,0} = -S_{11}^{-1} D_1 + HF, R_{\alpha,1} = (HG + J)F, \vdots R_{\alpha,k} = (HG + J) G^{k-1} F,$$

where $H := (S_{11})^{-1}(T_{11} U Z_{12} - T_{12})$ and $J := (S_{11})^{-1}(S_{11} U Z_{12} - S_{12})$, whereas the coefficient matrices for $R_f(\phi_t)$ are

$$R_{f,0} = KF, R_{f,1} = KGF, \vdots R_{f,k} = K G^k F,$$

with $K := Z_{22} - Z_{21} U Z_{12}$.

We have now completely described all matrices in the triangular (Solution).

Simulations with Anticipated Shocks

To be completed

References

Blanchard and Kahn (1980). The Solution of Linear Difference Models under Rational Expectations. Econometrica, 48(5):1305–12.

Klein (2000). Using the Generalized Schur Form to Solve a Multivariate Linear Rational Expectations Model. Journal of Economic Dynamics and Control, 24(10):1405–23.

1. ↩
2. ↩