A Fortran toolkit for analyzing nonlinear economic dynamic models

Iskander Karibzhanov

Department of Economics
University of Minnesota

March 2, 2009

Outline

Outline

Outline

Outline

Overview of the Fortran toolkit

Available Features

- checks analytic log-linearization vs numerical.
- solves DSGE models using QZ decomposition.
- computes log-likelihood using Kalman Filter.
- performs maximum likelihood estimation.

Missing Features

- simulations
- impulse responses
- first and second moments
- second-order approximation

Overview of the Fortran toolkit

Available Features

- checks analytic log-linearization vs numerical.
- solves DSGE models using QZ decomposition.
- computes log-likelihood using Kalman Filter.
- performs maximum likelihood estimation.

Missing Features

- simulations
- impulse responses
- first and second moments
- second-order approximation

Available Software

Uhlig's toolkit

- requires analytic log-linearization.
- requires MATLAB.
- no Kalman filter – no MLE.

Uribe's toolkit

- requires MATLAB symbolic toolbox.
- very slow.
- no Kalman filter – no MLE.

DYNARE

- requires MATLAB (except DYNARE++).
- little control beyond model file.

Available Software

Uhlig's toolkit

- requires analytic log-linearization.
- requires MATLAB.
- no Kalman filter – no MLE.

Uribe's toolkit

- requires MATLAB symbolic toolbox.
- very slow.
- no Kalman filter – no MLE.

DYNARE

- requires MATLAB (except DYNARE++).
- little control beyond model file.

Available Software

Uhlig's toolkit

- requires analytic log-linearization.
- requires `MATLAB`.
- no Kalman filter – no MLE.

Uribe's toolkit

- requires `MATLAB` symbolic toolbox.
- very slow.
- no Kalman filter – no MLE.

`DYNARE`

- requires `MATLAB` (except `DYNARE++`).
- little control beyond model file.

The general problem

There are n deterministic and m expectational equations:

$$
\begin{aligned}
0 &= Ax_t + Bx_{t-1} + Cy_t + Dz_t \\
0 &= E_t\{Fx_{t+1} + Gx_t + Hx_{t-1} + Jy_{t+1} + Ky_t + Lz_{t+1} + Mz_t\} \\
z_{t+1} &= Nz_t + \varepsilon_{t+1}, \quad \varepsilon_{t+1} \sim N(0, \Sigma)
\end{aligned}
$$

where
x_t is $m \times 1$ vector of endogenous states,
y_t is $n \times 1$ vector of endogenous variables (jump variables),
z_t is $k \times 1$ vector of exogenous stochastic processes.

The model matrix

The general problem can be written in matrix notation

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} I & 0 & 0 \\ 0 & F^{-} & I \end{bmatrix} \begin{bmatrix} A & B & D & C \\ F & 0 & L & J \\ G & H & M & K \end{bmatrix} \begin{bmatrix} x_t \\ x_{t-1} \\ z_t \\ y_t \end{bmatrix}$$

where F^{-} is a diagonal lead operator.
This model matrix is the input for the subroutine SOLVE.

Solution

We are looking for a recursive equilibrium law of motion

$$x_t = Px_{t-1} + Qz_t$$
$$y_t = Rx_{t-1} + Sz_t$$

or written in state space form

$$\begin{bmatrix} x_t \\ z_{t+1} \\ y_t \end{bmatrix} = \begin{bmatrix} P & Q \\ 0 & N \\ R & S \end{bmatrix} \begin{bmatrix} x_{t-1} \\ z_t \end{bmatrix} + \begin{bmatrix} 0 \\ \varepsilon_{t+1} \\ 0 \end{bmatrix}$$

This transition matrix is the output from the subroutine SOLVE.

Algorithm

Assuming C is invertible, we can reduce the problem to

$$0 = E_t\{\hat{F}x_{t+1} + \hat{G}x_t + \hat{H}x_{t-1} + \hat{L}z_{t+1} + \hat{M}z_t\}$$

where

$$\begin{aligned}
\hat{F} &= F - JC^{-1}A \\
\hat{G} &= G - JC^{-1}B - KC^{-1}A \\
\hat{H} &= H - KC^{-1}B \\
\hat{L} &= L - JC^{-1}D \\
\hat{M} &= M - KC^{-1}D
\end{aligned}$$

Solving for P

We need to solve a quadratic matrix equation

$$\hat{F}P^2 + \hat{G}P + \hat{H} = 0$$

Define

$$\hat{A} = \begin{bmatrix} \hat{F} & 0 \\ 0 & I \end{bmatrix} \quad \hat{B} = \begin{bmatrix} -\hat{G} & -\hat{H} \\ I & 0 \end{bmatrix}$$

Then $\hat{B}\begin{bmatrix} P \\ I \end{bmatrix} = \hat{A}\begin{bmatrix} P \\ I \end{bmatrix} P$. So P is a solution iff the columns of $\begin{bmatrix} P \\ I \end{bmatrix}$ span a deflating subspace for the pair $(\hat{B}, \hat{A})$.

Generalized Schur method

Theorem

All solutions of the quadratic matrix equation are of the form

$$P = -Z_{11}Z_{21}^{-1},$$

where

$$\hat{A}Z = QS$$
$$\hat{B}Z = QT$$

is a generalized Schur decomposition with Q and Z unitary and S and T upper-triangular, and where all matrices are partitioned as block 2×2 matrices with $m \times m$ blocks.

Generalized Schur method

Proof.

The first block columns of $\hat{A}Z = QS$ and $\hat{B}Z = QT$ give

$$\hat{F}Z_{11} = Q_{11}S_{11} \tag{1}$$
$$Z_{21} = Q_{21}S_{11} \tag{2}$$
$$-\hat{G}Z_{11} - \hat{H}Z_{21} = Q_{11}T_{11} \tag{3}$$
$$Z_{11} = Q_{21}T_{11} \tag{4}$$

Postmultiplying (3) by Z_{21}^{-1} and using (4) and then (1) and (2) proves that $Z_{11}Z_{21}^{-1}$ is a solution.

$$-\hat{G}Z_{11}Z_{21}^{-1} - \hat{H} = Q_{11}Q_{21}^{-1}Z_{11}Z_{21}^{-1} = \hat{F}Z_{11}Z_{21}^{-1}Z_{11}Z_{21}^{-1}. \qquad \square$$

Solving for Q, R and S

Once we found P, we can find Q by solving

$$\hat{F}QN + (\hat{F}P + \hat{G})Q + (\hat{L}N + \hat{M}) = 0$$

This can be written as a generalized Sylvester equation

$$\hat{F}Q - Z = -\hat{L}$$
$$(\hat{F}P + \hat{G})Q + ZN = -\hat{M}$$

which can be solved by the LAPACK subroutine TGSYL. Solving for R and S is simply

$$R = -C^{-1}(AP + B)$$
$$S = -C^{-1}(AQ + D)$$

Kalman Filter

The state space model consists of the transition (or state) and measurement (or observation) equations:

$$s_{t+1} = a_t + F_t s_t + \eta_t$$
$$y_t^o = b_t + H_t s_t + \epsilon_t$$

where $s_t = (x_{t-1}, z_t)$ and y_t^o is a subset of observed jump variables y_t.

Define $\Phi_t \equiv \begin{bmatrix} F_t \\ H_t \end{bmatrix}$ and $\Omega_t \equiv \text{cov}(\eta_t, \epsilon_t) = \begin{bmatrix} V_t & G_t \\ G_t' & R_t \end{bmatrix}$

In our static case,

$$F_t = \begin{bmatrix} P & Q \\ 0 & N \end{bmatrix} \qquad V_t = \begin{bmatrix} 0 & 0 \\ 0 & \Sigma \end{bmatrix}$$
$$H_t = \begin{bmatrix} R & S \end{bmatrix} \qquad G_t = 0, R_t = 0$$

Kalman Filter

The algorithm is

1. Set $t = 1$, $s_t = (I - F)^{-1} a_t$ and P_t that solves discrete Lyapunov equation $P = FPF' + V$.
2. Calculate

$$\begin{bmatrix} \bar{s}_{t+1} \\ \hat{y}_t \end{bmatrix} = \begin{bmatrix} a_t \\ b_t \end{bmatrix} + \Phi_t s_t$$

$$\begin{bmatrix} \bar{P}_{t+1} & M_t \\ M_t' & D_t \end{bmatrix} = \Phi_t P_t \Phi_t' + \Omega_t$$

$$K_t = M_t D_t^{-1}$$

3. Update

$$\hat{\epsilon}_t = y_t - \hat{y}_t$$

$$s_{t+1} = \bar{s}_{t+1} + K_t \hat{\epsilon}_t$$

$$P_{t+1} = \bar{P}_{t+1} - K_t M_t'$$

$$\log L = \log L + \log |D_t| + \hat{\epsilon}_t' D_t^{-1} \hat{\epsilon}_t$$

4. if $t = T$ stop, else $t = t + 1$ goto (2)

Outlook

- Kalman filter can update the Cholesky factor of state covariance matrix using QR decomposition.
- Stable eigenvalues in the shock transition matrix can be generated using Schur decomposition.
- Use the Control and Systems Library SLICOT (www.slicot.org).