Model Solution Matrices

know_all_about_solution.m

by Jaromir Benes

May 27, 2015

Summary

Describe and retrieve the state-space form of a solved model. IRIS uses a state-space form with two modifications. First, the state-space system is transformed so that the transition matrix is upper triangular (quasi-triangular). Second, the effect of future anticipated shocks can be directly computed upon request, and added to the system stored in the model object.

Contents

1 Clear Workspace . . . 2
2 Load Solved Model Object . . . 2
3 First Order Solution (State Space) . . . 2
4 Transition Matrix . . . 4
5 Variables in State Space Vector . . . 5
6 Forward Expansion of Model Solution . . . 7
7 Help on IRIS Functions Used in This File . . . 8

1 Clear Workspace

Clear workspace, close all graphics figures, clear command window, and check the IRIS version.

```
clear;
close all;
clc;
irisrequired 20140315;
```

2 Load Solved Model Object

Load the solved model object built in `read_model`.

```
load read_model.mat m;
```

3 First Order Solution (State Space)

The function `solve` executed earlier in `read_model.m` computes the first-order accurate state-space representation of the model. IRIS uses a transformed representation that has a number of advantages.

$$
\begin{aligned}
&[x_t^f; \alpha_t] = T\alpha_{t-1} + K + R_1 e_t + R_2 \,\mathrm{E}_t\left[e_{t+1}\right] + \dots \\
&y_t = Z\alpha_t + D + He_t \\
&x_t^b = U\alpha_t \\
&\mathrm{E}[e_t e_t'] = \Omega
\end{aligned}
$$

- x^f non-predetermined (forward-looking) variables;
- x^b predetermined (backward-looking) transition variable;
- e residuals;
- y measurement variables;
- α vector of transformed pre-determined variables;
- T transition matrix; the transformed vector α is set up so that T is upper quasi-triangular – see next section.

```
[T,R,K,Z,H,D,U,Omg] = sspace(m); %#ok<ASGLU>

disp('State-space matrices');

disp('Size of T');
size_of_T = size(T) %#ok<NOPTS>

disp('Size of R');
size_of_R = size(R) %#ok<NOPTS>

disp('Size of K');
size(K)

disp('Size of Z');
size(Z)

disp('Covariance matrix of residuals');
Omg %#ok<NOPTS>
```

```
State-space matrices
Size of T
size_of_T =
    24    13
Size of R
size_of_R =
    24     7
Size of K
ans =
    24     1
Size of Z
ans =
     4    13
Covariance matrix of residuals
Omg =
   1.0e-04 *
         0         0         0         0         0         0         0
         0         0         0         0         0         0         0
         0         0    0.9901         0         0         0         0
         0         0         0    0.9901         0         0         0
         0         0         0         0    0.0100         0         0
         0         0         0         0         0    0.9901         0
         0         0         0         0         0         0    0.9901
```

4 Transition Matrix

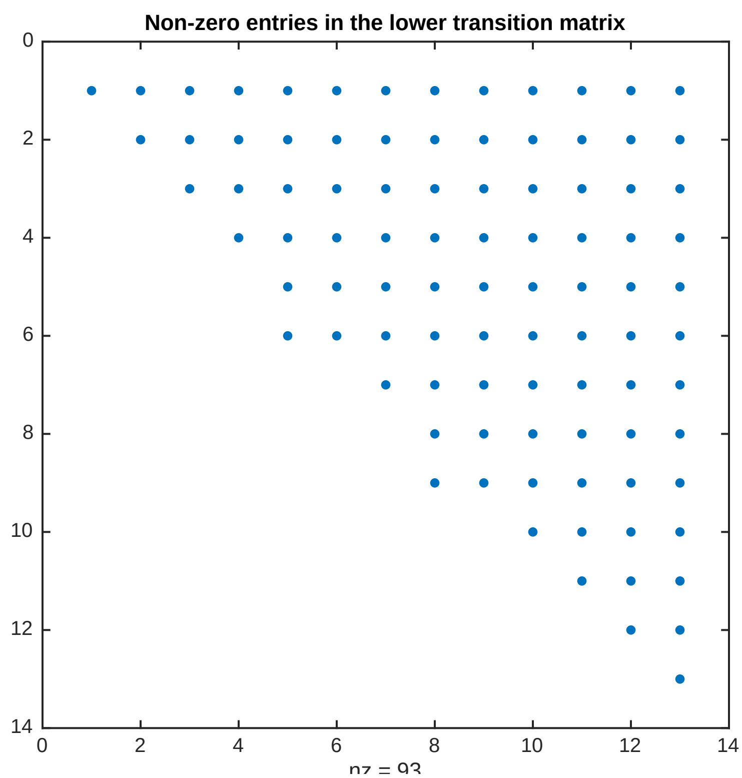
The transition matrix `T` can be divided into the upper part `Tf` (which determines the non-predetermined variables) and the square lower part `Ta` (which determines the vector alpha). The matrix `Tf` is in general rectangular, `nf`-by-|nb|, whereas `Ta` is a sqaure matrix, `nb`-by-|nb|. The dynamics of the model is solely given by `Ta`; the transformation `alpha` is chosen so that `Ta` is always upper quasi-triangular.

The number of non-predetermined (forward-looking) variables and the number of predetermined (backward-looking) variables (which equals the size of the vector α) can be derived from the size of the matrix `T`.

```
nx = size(T,1);
nb = size(T,2);
nf = nx - nb;

disp('Size of the transition matrix T');
size_of_T %#ok<NOPTS>

disp('Length of the vector x');
nx %#ok<NOPTS>

disp('Length of the vector xf')
nf %#ok<NOPTS>

disp('Length of the vector xb (and of the vector alpha)')
nb %#ok<NOPTS>

Tf = T(1:nf,:);
Ta = T(nf+1:end,:);

figure();
spy(Ta);
title('Non-zero entries in the lower transition matrix');

disp('Unit roots in the model solution');
unit_roots = get(m,'unitRoots') %#ok<NOPTS>

nunit = length(unit_roots);
Ta(1:nunit,1:nunit)
```

```
Size of the transition matrix T
size_of_T =
    24    13
Length of the vector x
nx =
```

```
    24
Length of the vector xf
nf =
    11
Length of the vector xb (and of the vector alpha)
nb =
    13
Unit roots in the model solution
unit_roots =
    1.0000    1.0000
ans =
    1.0000    0.0000
         0    1.0000
```



5 Variables in State Space Vector

Find out the order in which the individual variables occur in the rows and columns of the state-space matrices. The vector of measurement variables and the vector of shocks are straightforward – they are ordered as they are declared in the model code (with the measurement shocks preceding the transition shocks). The vector of transition variables contain also all auxiliary lags and leads.

```
disp('Vector of transition variables (x)');
xvector = get(m,'xVector') %#ok<NOPTS>

disp('Vector of measurement variables (y)');
yvector = get(m,'yVector') %#ok<NOPTS>

disp('Vector of shocks (e)');
evector = get(m,'eVector') %#ok<NOPTS>
```

```
Vector of transition variables (x)
xvector =
    'log(dP{3})'
    'log(dP{2})'
    'log(dP{1})'
    'log(N)'
    'log(Q)'
    'log(H)'
    'log(Pk)'
    'log(Rk)'
    'log(Lambda)'
    'log(d4P)'
    'log(RMC)'
    'log(Y)'
    'log(W)'
    'log(A)'
    'log(P)'
    'log(R)'
    'log(dP)'
    'log(dW)'
    'log(Y{-1})'
    'log(W{-1})'
    'log(A{-1})'
    'log(P{-1})'
    'log(P{-2})'
    'log(P{-3})'
Vector of measurement variables (y)
yvector =
    'Short'
    'Infl'
    'Growth'
    'Wage'
```

```
Vector of shocks (e)
evector =
    'Mp'
    'Mw'
    'Ey'
    'Ep'
    'Ea'
    'Er'
    'Ew'
```

6 Forward Expansion of Model Solution

Forward expansion of the solution is needed in simulations or forecasts with future anticipated shocks. Use the function `expand` to calculate and store the expansion in the model object. Alternatively, if not available, the expansion is automatically added whenever the functions `simulate` or `jforecast` are executed with future anticipated shocks.

```
k = get(m,'forward');

disp('Solution is now expanded t+k periods forward');
k %#ok<NOPTS>

m = expand(m,2);

display('Solution is now expanded t+k periods forward');
k = get(m,'forward') %#ok<NOPTS>

[T,R,K,Z,H,D,U,Omg] = sspace(m);

disp('Size of the matrix R before expansion');
size_of_R %#ok<NOPTS>

disp('Size of the matrix R after expansion');
size_of_R_exp = size(R) %#ok<NOPTS>
```

```
Solution is now expanded t+k periods forward
k =
     0
Solution is now expanded t+k periods forward
k =
     2
Size of the matrix R before expansion
size_of_R =
```

```
    24     7
Size of the matrix R after expansion
size_of_R_exp =
    24    21
```

7 Help on IRIS Functions Used in This File

Use either `help` to display help in the command window, or `idoc` to display help in an HTML browser window.

```
help model/get
help model/sspace
```