Additional material for: Minimality of state space solutions of DSGE models and existence conditions for their VAR representation

MASSIMO FRANCHI* AND PAOLO PARUOLO**

ABSTRACT. This file contains MATLAB routines that perform the check in Proposition 4.4. of the paper.

1. DESCRIPTION

This file describes and includes the following MATLAB routines:

- `main.m`: definition of the A, B, C, D matrices and call of prop44
- `prop44.m`: function that checks the rank conditions in Proposition 4.4
- `MatRnkDecSvd.m`: function that performs a matrix rank decomposition, based on the svd function of MATLAB

The scripts are embedded as PDF 'file attachment annotations', which requires PDF version 1.4 or higher (Adobe Acrobat 4.0 or higher). The embedding was generated with the `attachfile` Latex package, see Pakin (2011)

To open the scripts, you need to either

- right click on the icons and choose 'Save Embedded File to Disk...'
- or double-click on the icons .

Note that in Adobe Acrobat, annotations never print unless the Annotations box is checked in the Print dialog.

2. MAIN.M

This script defines the A, B, C, D matrices and runs the other scripts.

Click on to open/save the script as a text file.

```
%% This is the main file; it defines A,B,C and D as in Example 3.2 of the paper
clear;
Phi1=[1, 1 ; -0.5, -0.5]; Phi2=[1, 1 ; 1, 1];
```

Date: October 9, 2014.

* M. Franchi, Sapienza University of Rome, P.le A. Moro 5, 00185 Rome, Italy; e-mail: massimo.franchi@uniroma1.it.

** **Corresponding author:** P. Paruolo, European Commission, Joint Research Centre, Via E.Fermi 2749, I-21027 Ispra (VA), Italy; e-mail: paolo.paruolo@jrc.ec.europa.eu.

```
A = [[Phi1,zeros(2,2)]; [zeros(2,2),eye(2)]];
B = [0.5*eye(2); eye(2)];
C = [Phi1,Phi2];
D = Phi2+0.5*eye(2);
tol = 1e-10;
% Call to prop44 function
[vUnst_lam,vCond,sCond] = prop44(A,B,C,D,tol);
```

3. FUNCTION PROP44

This script checks the orthogonality conditions in Prop 4.4 of the paper.

Click on  to open/save the script as a text file.

```
function [vUnst_lam,vCond,sCond] = prop44(A,B,C,D,tol)
% PURPOSE: check orthogonality conditions in Prop 4.4 of Franchi & Paruolo 2014
%-------------------------------------------------
% USAGE:   [vUnst_lam,vCond,sCond] = prop44(A,B,C,D,tol)
% where:    A = nx x nx matrix
%           B = nx x ny matrix
%           C = nx x nx matrix
%           D = ny x ny matrix
%           tol = tolerance value
%-------------------------------------------------
% RETURNS:   vUnst_lam = s x 1 vector of unstable eigenvalues of F=A-BD^{-1}C
%            vCond = s x 1 logical vector, entries i equal 1 if
%            orthogonality conditions is satisfied for the i-th entry in vUnst_lam
%            sCond = scalar, logical vector, equal 1 if all
%                    orthogonality conditions are satisfied
%-------------------------------------------------

nx=size(A,1);
mF=A-B*mldivide(D,C); mI=eye(size(mF,1));
vLam = eig(mF); vNorm_lam = abs(vLam);     % eigenvalues of F and their norm
vUnst_lam= vLam(vNorm_lam >= 1-tol);       % select unstable eigenvalues of F
num_unst_lam = size(vUnst_lam,1);          % number of unstable eigenvalues of F

if num_unst_lam == 0;
    vCond=1; sCond=1;
```

```
    disp('All eigenvalues of F are stable, and the system is fundamental.');
else
    vCond=zeros(num_unst_lam,1);
     for i = 1:num_unst_lam;
     mP=mF-vUnst_lam(i)*mI;
     [mA,mB,r,mAort,mBort]= MatRnkDecSvd(mP); % rank decomposition of F-lam I = mA*mB'
     rkmAortB=rank(mAort'*B,tol);  rkCmBort=rank(C*mBort,tol); % ranks
     boo1 = rkmAortB < nx-r; boo2 = rkCmBort < nx-r;    % = 1 if orth.condition holds
     vCond(i)= boo1 || boo2;
     end; % for i = 1:num_unst_lam;
     sCond = prod(vCond);                    % = 1 if all orth.conditions hold
     if sCond == 1
         disp('All orth.conditions are satisfied, and the system is fundamental.');
         disp('List of unstable eigenvalues (for which the orth.condition holds):');
         disp(vUnst_lam');
     else
         disp('Some orth. conditions do not hold, and the system is not fundamental.');
         disp('List of unstable eigenvalues for which the orth.condition does not hold:');
         disp(vUnst_lam(not(vCond))');
         disp('List of unstable eigenvalues for which the orth.condition does hold:');
         disp(vUnst_lam(vCond)');
     end; % if sCond ==1 %
end; % if num_unst_lam == 0;%
```

4. FUNCTION MatRnkDecSvd

This script performs a Matrix Rank Decomposition using the SVD.

Click on  to open/save the script as a text file.

```
function [mA,mB,r,mAort,mBort]= MatRnkDecSvd(mP)
% PURPOSE: computes Matrix Rank Decomposition, mP=mA*mB' based on svd
%---------------------------------------------------
% USAGE:      [mA,mB,r,mAort,mBort]= MatRnkDecSvd(mP)
% where:      mP = m x n matrix
%---------------------------------------------------
% RETURNS:    mA = m x r matrix
%             mB = n x r matrix,
%             r = rank(mP)
```

```
%            mAort = m x (m-r) matrix, basis of ort.complement of col(mA)
%            mBort = n x (n-r) matrix, basis of ort.complement of col(mB)
% ------------------------------------------------
[U,S,V] = svd(mP); vs=diag(S); tol = max(size(mP))*eps(max(vs));
r = sum(vs > tol);
mA=U(:,1:r); mAort=U(:,(r+1):end);
mV=V*S'; mB=mV(:,1:r); mBort=V(:,(r+1):end);
```

References

Pakin, S. (2011). The attachfile package,
http://www.ctan.org/tex-archive/macros/latex/contrib/attachfile.