Quantitative Macroeconomics

Winter 2023/24

Week 8

Willi Mutschler
willi@mutschler.eu

Version: 1.0
Latest version available on: GitHub

Contents

1. Structural Impulse Response Function

Consider the structural VAR(p) model

$$B_0 y_t = B_1 y_{t-1} + \cdots + B_p y_{t-p} + \varepsilon_t$$

where the dimension of B_i, $i = 0, \ldots, p$, is $K \times K$. The $K \times 1$ vector of structural shocks ε_t is assumed to be white noise with covariance matrix $E(\varepsilon_t \varepsilon_t') = I_K$. That is, the elements of ε_t are mutually uncorrelated and also have a clear interpretation in terms on an underlying economic model. The reduced-form VAR(p) model is given by

$$y_t = \underbrace{B_0^{-1} B_1}_{A_1} y_t + \cdots + \underbrace{B_0^{-1} B_p}_{A_p} y_{t-p} + \underbrace{B_0^{-1} \varepsilon_t}_{u_t}$$

where the reduced-form error covariance matrix is $E(u_t u_t') = \Sigma_u = B_0^{-1} B_0^{-1'}$. Going back and forth between the structural and the reduced-form representation requires knowledge of the structural impact matrix B_0^{-1}. For now, assume that B_0^{-1} is a known matrix. We are interested in the response of each element of y_t to a one-time impulse in ε_t:

$$\frac{\partial y_{t+h}}{\partial \varepsilon_t'} = \Theta_h, \quad h = 0, 1, 2, \ldots, H$$

where Θ_h is a $K \times K$ matrix with individual elements $\theta_{jk,h} = \frac{\partial y_{j,t+h}}{\partial \varepsilon_{k,t}}$.

1. Usually the objective is to plot the responses of each variable to each structural shock. How many so-called impulse response functions do we need to plot?

2. Consider the VAR(1) representation of the VAR(p) process, i.e.

$$Y_t = AY_{t-1} + U_t$$

where

$$Y_t = \begin{pmatrix} y_t \\ \vdots \\ y_{t-p+1} \end{pmatrix}, A = \begin{pmatrix} A_1 & A_2 & \cdots & A_{p-1} & A_p \\ I_k & 0 & \cdots & 0 & 0 \\ 0 & I_K & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & I_k & 0 \end{pmatrix}, U_t = \begin{pmatrix} u_t \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Show that

$$y_{t+h} = JA^{h+1} Y_{t-1} + \sum_{j=0}^{h} JA^j J' u_{t+h-j}$$

where $J = [I_K, 0_{K \times K(p-1)}]$.

3. Derive an expression for Θ_h in terms of J, A and B_0^{-1}.

4. How would you infer from the response of the inflation rate, say Δp_t, the implied response of the log price level p_t?

5. Write a function that plots the IRFs given inputs A, B_0^{-1}, number of lags p, maximum horizon H, an indicator vector whether the cumulative sum of the IRFs should be computed, and string arrays for the names of the variables and shocks.

Readings

- Kilian and Lütkepohl (2017, Ch. 4.1)

2. Recursively Identified Models

A popular argument in macroeconomics has been that oil price shocks in particular may act as domestic supply shocks for the U.S. economy. Thus, the question of how oil price shocks affect U.S. real GDP and inflation has a long tradition in macroeconomics. Postulate a VAR(4) model with intercept for the percent changes in the real WTI price of crude oil ($\Delta rpoil_t$), the U.S. GDP deflator inflation rate (Δp_t), and U.S. real GDP growth (Δgdp_t). Consider the dataset given in "USOil.csv". The data are quarterly and the estimation period is 1987q1 to 2013q2.

1. How can you identify the oil price shock statistically and economically?
2. Estimate the reduced-form vector autoregressive model by least-squares or maximum-likelihood to obtain a consistent estimate of the reduced-form error covariance matrix.
3. Estimate the structural impact multiplier matrix B_0^{-1} based on a lower-triangular Cholesky decomposition of the residual covariance matrix.
4. Estimate the structural impact multiplier matrix B_0^{-1} based on solving the system of nonlinear equations that implicitly defines the elements of B_0^{-1} using a nonlinear equation solver that finds the vector x such that $F(x) = 0$, where $F(x)$ denotes a system of nonlinear equations in x. To this end, vectorize the system of equations $B_0^{-1}B_0^{-1'} - \hat{\Sigma}_u$. The objective is to find the unknown elements of B_0^{-1} such that

   $$F_{SR}(B_0^{-1}) = \left[vech\left(B_0^{-1}B_0^{-1'} - \hat{\Sigma}_u\right), \quad b_0^{12}, \quad b_0^{13}, \quad b_0^{23}\right]' = 0$$

   where the vech operator is used to select the lower triangular elements of $B_0^{-1}B_0^{-1'} - \hat{\Sigma}_u$.

   To this end:

   - Write a function `fSR.m` that computes F_{SR}. A MATLAB implementation of the vech operator is given in listing A.
   - Set the termination tolerance `TolX=1e-4`, the termination tolerance on the function value `TolFun` to `1e-9`, the maximum number of function evaluations `MaxFunEvals` to `50000`, the maximum number of iterations `MaxIter` to `1000`. Save this set of options:`options = optimset('TolX',TolX,`
     `'TolFun',TolFun,'MaxFunEvals',MaxFunEvals,'MaxIter',MaxIter)`
   - Choose an admissible start value for B_0^{-1} and call the optimization routine `fsolve` to minimize your `fSR` function:
     `[B0inv,fval,exitflag,output]=fsolve('fSR',B0inv0,options,'additional inputs to fSR')`
   - Impose the normalization rule that the diagonal elements of B_0^{-1} are positive.

   Compare this to the Cholesky decomposition. Is there a difference?
5. Plot the impulse response function for $H = 30$ of an oil price shock on the real price of crude oil (in logs), the GDP deflator inflation series and on the level of GDP (in logs) using `irfPlots.m` and interpret it economically.

Readings

- Kilian and Lütkepohl (2017, Ch. 8-9)

3. Non-recursively Identified Models By Short-Run Restrictions

Consider a quarterly model of US monetary policy. Let $y_t = (\Delta p_t, \Delta gnp_t, i_t, \Delta m_t)'$, where p_t refers to the log of the GNP deflator, gnp_t to the log of real GNP, i_t to the federal funds rate, averaged by quarter, and m_t to the log of money aggregate M1. The data is given in `Keating1992.csv`.

The structural shock vector $\varepsilon_t = (\varepsilon_t^{AS}, \varepsilon_t^{IS}, \varepsilon_t^{MS}, \varepsilon_t^{MD})$ includes an aggregate supply shock, an IS shock, a money supply shock, and a money demand shock. The structural model can be written as:

$$\begin{pmatrix} u_t^p \\ u_t^{gnp} \\ u_t^i \\ u_t^m \end{pmatrix} = \begin{pmatrix} \varepsilon_t^{AS} \\ -b_{21,0}u_t^p - b_{23,0}u_t^i - b_{24,0}u_t^m + \varepsilon_t^{IS} \\ -b_{34,0}u_t^m + \varepsilon_t^{MS} \\ -b_{41,0}(u_t^{gnp} + u_t^p) - b_{43,0}u_t^i + \varepsilon_t^{MD} \end{pmatrix}$$

Furthermore, it is assumed that $E(\varepsilon_t \varepsilon_t') = \Sigma_\varepsilon$ is NOT the identity matrix; hence, we will use the normalization rule that the diagonal elements of B_0 equal unity.

1. Estimate the reduced-form covariance matrix of a VAR model with four lags and an intercept with ordinary least squares.

2. Provide intuition behind the structural model and derive the B_0 matrix. Note that we are imposing restrictions on both B_0 and Σ_ε, but not on B_0^{-1}.

3. Estimate B_0 and Σ_ε using the nonlinear equation solver `fsolve`. To this end, first set up a function that computes

$$F_{SR}([B_0, diag(\Sigma_\varepsilon)]) = \begin{bmatrix} vech\left(B_0^{-1}\Sigma_\varepsilon B_0^{-1'} - \hat{\Sigma}_u\right) \\ \text{restrictions on } B_0 \end{bmatrix} = 0$$

   where $diag(\Sigma_\varepsilon)$ denotes only the diagonal elements of Σ_ε. Note that these diagonal elements as well as B_0 are stacked into a matrix or vector which is then used as the input for the numerical optimizer. Use feasible options for the optimizer (e.g. as in the previous exercises).

4. Use the estimated structural impact matrix $B_0^{-1}\Sigma_\varepsilon^{1/2}$ to plot the structural impulse response functions using `irfPlots.m`. Interpret the effects of an aggregate supply shock on prices, real GNP, the Federal Funds rate and M1.

Readings

- Kilian and Lütkepohl (2017, Ch. 8-9)

References

Kilian, Lutz and Helmut Lütkepohl (2017). *Structural Vector Autoregressive Analysis*. Themes in Modern Econometrics. Cambridge: Cambridge University Press. ISBN: 978-1-107-19657-5. URL: https://doi.org/10.1017/9781108164818.

A. vech.m

progs/matlab/vech.m

```
function B = vech(A)
% function B = vech(A)
% ---------------------------------------------------------------------------
% This function calculates the vech of a square matrix
% ---------------------------------------------------------------------------
% INPUTS
%   - A: a square matrix
% ---------------------------------------------------------------------------
[m,n] = size(A);
if m ~= n
    error('The input matrix must be square');
end
B      = zeros(n*(n+1)/2,1);
index  = 0;
for i=1:n
    B(1+index:index+n+1-i,1) = A(i:n,i);
    index = index + n+1-i;
end
```

A. Solutions

1 Solution to Structural Impulse Response Function

1. There are K variables and K structural shocks; hence, there are K^2 IRFs each of length $H+1$.

2. Let's derive an expression for Y_{t+h}:

$$
\begin{aligned}
Y_t &= AY_{t-1} + U_t \\
Y_{t+1} &= AY_t + U_{t+1} = A^2Y_{t-1} + AU_t + U_{t+1} \\
Y_{t+2} &= AY_{t+1} + U_{t+2} = A^3Y_{t-1} + A^2U_t + AU_{t+1} + U_{t+2} \\
Y_{t+h} &= A^{h+1}Y_{t-1} + \sum_{j=0}^{h} A^j U_{t+h-j}
\end{aligned}
$$

   Left-multiply by J (also note that $J'J = I$):

$$
JY_{t+h} = y_{t+h} = JA^{h+1}Y_{t-1} + \sum_{j=0}^{h} JA^j J'JU_{t+h-j} = JA^{h+1}Y_{t-1} + \sum_{j=0}^{h} JA^j J' u_{t+h-j}
$$

3. From the previous exercise:

$$
y_{t+h} = JA^{h+1}Y_{t-1} + \sum_{j=0}^{h} JA^j J' \underbrace{u_{t+h-j}}_{B_0^{-1}\varepsilon_{t+h-j}}
$$

   Taking the derivative:

$$
\frac{\partial y_{t+h}}{\partial \varepsilon_t'} = \Theta_h = JA^h J' B_0^{-1}
$$

4. Note the following identity: $p_{t+h} = p_{t-1} + \Delta p_t + \Delta p_{t+1} + \cdots + \Delta p_{t+h}$. That is we simply need to cumulate (`cumsum`) the impulse responses of the inflation rate to get the impulse response for the price level.

5. A possible implementation:

progs/matlab/irfPlots.m

```
function irfPoint  = irfPlots(Acomp,impactMatrix,nSteps,cumsumIndicator,variableNames,shockNames,noPlot)
% irfPoint  = irfPlots(Acomp,impactMatrix,nSteps,cumsumIndicator,variableNames,shockNames,noPlot)
% -----------------------------------------------------------------------
% Compute impulse response functions for a SVAR model
% B0*y_t = B_1*y(-1) + B_2*y(-2) + ... + B_p*y(-p) + e_t with e_t ~ iid(0,I)
% given Acompnion matrix Acomp of the reduced-form VAR and impactMatrixrix inv(B0)
% -----------------------------------------------------------------------
% INPUT
%   - Acomp:            [(nvars*(nlags-1)) x (nvars*nlags)]  companion matrix of reduced-form VAR
%   - impactMatrix:     [nvars x nvars]                      impact matrix of SVAR model, either inv(B0) or inv(B0)*sqrt(E[epsilon_t*epsilon_t'])
%   - cumsumIndicator:  [nvars x 1]                          boolean vector, 1 indicates for which variable to compute the cumulative sum in IRFs
```

```
%   - variableNames:    [nvars x 1]                          vector of strings with variable names
%   - shockNames:       [nvars x 1]                          vector of strings with structural shock names
%   - noPlot            boolean                              1: turn off displaying of plots (useful for bootstrapping)
% -------------------------------------------------------------------------
% OUTPUT
%   - irfPoint(j,k,h)  [nvars,nvars,nSteps+1]                array with h =0,1,...,nSteps+1 steps, containing the IRF of the 'j' variable to the 'k' shock
% ---------------------------------------------------------------------------
% Willi Mutschler, November 27, 2023
% willi@mutschler.eu
% ---------------------------------------------------------------------------

nvars = size(impactMatrix,1);
nlag  = size(Acomp,2)/nvars;

% set default options if not specified
if nargin < 4 || isempty(cumsumIndicator)
    cumsumIndicator = false(1,nvars);
end
if nargin < 5 || isempty(variableNames)
    variableNames = "y" + string(1:nvars);
end
if nargin < 6 || isempty(shockNames)
    shockNames = "e" + string(1:nvars);
end
if nargin < 7 || isempty(noPlot)
    noPlot = false;
end

% initialize variables
irfPoint = nan(nvars,nvars,nSteps+1);
J = [eye(nvars) zeros(nvars,nvars*(nlag-1))];

% compute the impulse response function using the companion matrix
Ah = eye(size(Acomp)); % initialize A^h at h=0
JtB0inv = J'*impactMatrix;
for h=1:(nSteps+1)
    irfPoint(:,:,h) = J*Ah*JtB0inv;
    Ah = Ah*Acomp; %A^h = A^(h-1)*A
end

% use cumsum to get response of level variables from original variables in differences
for ivar = 1:nvars
    if cumsumIndicator(ivar) == true
        irfPoint(ivar,:,:) = cumsum(irfPoint(ivar,:,:),3);
    end
end
```

```
% plot
if ~noPlot
    % define a timeline
    steps = 0:1:nSteps;
    x_axis = zeros(1,nSteps+1);
    figure('units','normalized','outerposition',[0 0.1 1 0.9]);
    count = 1;
    for ivars=1:nvars % index for variables
        for ishocks=1:nvars % index for shocks
            irfs = squeeze(irfPoint(ivars,ishocks,:));
            subplot(nvars,nvars,count);
            plot(steps,irfs,steps,x_axis,'k','LineWidth',2);
            xlim([0 nSteps]);
            title(shockNames(ishocks), 'FontWeight','bold','FontSize',10);
            ylabel(variableNames(ivars), 'FontWeight','bold','FontSize',10);
            count = count+1;
            set(gca,'FontSize',16);
        end
    end
end
```

2 Solution to Recursively Identified Models

1. The model is identified recursively with the real price of oil ordered first:

$$y_t = \begin{pmatrix} \Delta rpoil_t \\ \Delta p_t \\ \Delta gdp_t \end{pmatrix}$$

   such that the real price of oil is **predetermined** with respect to the U.S. economy. In other words, only the structural oil price shock (ordered first) has an immediate effect on the real price of oil, the other two structural shocks affect the real price of oil with a delay and not on impact. The ordering is thus very important here, as our focus is on the effect of an **unanticipated (exogenous)** increase in the real price of oil. Moreover, as we are only interested in one shock, the model is only partially identified in that only the oil price shock can be given an economic interpretation.

2./3./4./5.

progs/matlab/USOil.m

```
% -----------------------------------------------------------------------
% Estimates a SVAR(4) model to identify an oil price shock for the US economy
% -----------------------------------------------------------------------
% Willi Mutschler, December 8, 2022
% willi@mutschler.eu
% -----------------------------------------------------------------------
clearvars; clc;close all;

%% data handling
USOil_data = importdata('../../data/USOil.csv');
ENDO = USOil_data.data; % note that it already has correct order dlog(poil), dlog(p), dlog(gdp) for recursive identification

%% estimate reduced-form
nlag = 4;
opt.const = 1;
VAR = VARReducedForm(ENDO,nlag,opt);

%% structural identification with Cholesky decomposition
ENDO = ENDO(:,[1 2 3]); % order dlog(oilprice) first, then dlog(price deflator) and dlog(GDP)
B0inv_chol = chol(VAR.SigmaOLS([1 2 3],[1 2 3]),'lower');
% note that the Cholesky decomposition always yields positive diagonal elements, so no normalization needed
table(B0inv_chol)

%% structural identification with numerical optimization, identification restrictions are put into auxiliary function USOil_fSR
f = str2func('USOil_fSR');
StartValueMethod = 1; %0: Use identity matrix, 1: use square root, 2: use cholesky as starting value
% options for fsolve
TolX            = 1e-4;  % termination tolerance on the current point
TolFun          = 1e-9;  % termination tolerance on the function value
MaxFunEvals     = 50000; % maximum number of function evaluations allowed
MaxIter         = 1000;  % maximum numberof iterations allowed
```

```matlab
OptimAlgorithm = 'trust-region-dogleg'; % algorithm used in fsolve
options = optimset('TolX',TolX,'TolFun',TolFun,'MaxFunEvals',MaxFunEvals,'MaxIter',MaxIter,'Algorithm',OptimAlgorithm);
if StartValueMethod == 0
    B0inv = eye(size(ENDO,2)); % use identity matrix as starting value
elseif StartValueMethod == 1
    B0inv = VAR.SigmaOLS^.5; % use square root of vcov of reduced form as starting value
elseif StartValueMethod == 2
    B0inv = chol(VAR.SigmaOLS,'lower'); % use Cholesky decomposition of vcov of reduced form
end
f(B0inv,VAR.SigmaOLS)' % test whether function works at initial value (should give you no error)

% call optimization routine fsolve to minimize f
[B0inv_opt,fval,exitflag,output] = fsolve(f,B0inv,options,VAR.SigmaOLS);

% normalize sign of B0inv such that diagonal elements are positive
if any(diag(B0inv_opt)<0)
    idx_negative = diag(B0inv_opt)<0;
    B0inv_opt(:,(idx_negative==1)) = -1*B0inv_opt(:,(idx_negative==1)); % flip sign
end
table(B0inv_opt)

%% compute and plot structural impulse response function
nSteps = 30;
cumsumIndicator = [1 0 1]; % the variables in the SVAR are in differences, we want to plot IRFs of the first and last variable by using cumsum
varNames = ["Real Price of Oil","GDP Deflator Inflation","Real GDP"];
epsNames = ["Oil Price Shock", "eps2 Shock", "eps3 Shock"];
IRFpoint_chol = irfPlots(VAR.Acomp,B0inv_chol,nSteps,cumsumIndicator,varNames,epsNames);
IRFpoint_opt = irfPlots(VAR.Acomp,B0inv_opt,nSteps,cumsumIndicator,varNames,epsNames);

%% both approaches are (numerically) equivalent
norm(abs(B0inv_chol-B0inv_opt),'Inf') % max abs error
norm(abs(IRFpoint_chol(:) - IRFpoint_opt(:)),'Inf') % max abs error
```

Here is the helper function for the more general approach:

progs/matlab/USOil_fSR.m

```matlab
function f = USOil_fSR(B0inv,hatSigmaU)
% f = USOil_fSR(B0inv,hatSigmaU)
% -----------------------------------------------------------------------
% Evaluates the system of nonlinear equations
%   vech(hatSigmaU) = vech(B0inv*B0inv')
% subject to specified short-run restrictions
% -----------------------------------------------------------------------
% INPUTS
%   - B0inv     : candidate for short-run impact matrix. [nvars x nvars]
```

```
%   - hatSigmaU : covariance matrix of reduced-form residuals. [nvars x nvars]
% -----------------------------------------------------------------------
% OUTPUTS
%   - f : function value, see below
% -----------------------------------------------------------------------
% Willi Mutschler, December 8, 2022
% willi@mutschler.eu
% -----------------------------------------------------------------------
f = [vech(B0inv*B0inv' - hatSigmaU);
     B0inv(1,2) - 0;
     B0inv(1,3) - 0;
     B0inv(2,3) - 0;
    ];

% % more general way to implement the restrictions
% % nan means unconstrained, 0 (or any other number) restricts to this number
% Rshort = [nan   0   0;
%           nan nan   0;
%           nan nan nan;
%           ];
% selSR = ~isnan(Rshort); % index for short-run restrictions on impact
% f = [vech(B0inv*B0inv'-hatSigmaU);
%      B0inv(selSR) - Rshort(selSR);
%     ];
end
```

6. In the period under consideration, crude oil is a commodity that is considered vulnerable to negative supply shocks due to the political and social volatility of its (Middle East) source. In macroeconomics a negative supply shock is an unexpected event that changes the supply of a product, resulting in an increase in prices and a decrease in output. From the IRFs we see exactly this pattern: an unexpected oil price shock creates inflationary pressure on the GDP deflator and a reduction in real GDP in the US. In this sense, a positive oil price shock (originating e.g. from the oil producing countries) indeed acts like a negative domestic supply shock for the U.S. economy.

3 Solution to Non-recursively Identified Models By Short-Run Restrictions

1. The OLS estimate of the reduced-form covariance matrix Σ_u is

$$\widehat{\Sigma}_u = \begin{pmatrix} 0.061054 & -0.015282 & 0.042398 & 0.0037836 \\ -0.015282 & 0.52305 & 0.07967 & 0.030602 \\ 0.042398 & 0.07967 & 0.71689 & -0.2451 \\ 0.0037836 & 0.030602 & -0.2451 & 1.1093 \end{pmatrix}$$

2. In order to identify the structural shocks from $\Sigma_u = B_0^{-1}\Sigma_\varepsilon B_0^{-1'}$, we require at least $4(4-1)/2 = 6$ additional restrictions on B_0 or B_0^{-1}. In this exercise, we'll do this on B_0 and not its inverse; that is, we can rewrite the equations in matrix notation:

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ b_{21,0} & 1 & b_{23,0} & b_{24,0} \\ 0 & 0 & 1 & b_{34,0} \\ b_{41,0} & b_{41,0} & b_{43,0} & 1 \end{pmatrix}}_{B_0} \underbrace{\begin{pmatrix} u_t^p \\ u_t^{gnp} \\ u_t^i \\ u_t^m \end{pmatrix}}_{u_t} = \underbrace{\begin{pmatrix} \varepsilon_t^{AS} \\ \varepsilon_t^{IS} \\ \varepsilon_t^{MS} \\ \varepsilon_t^{MD} \end{pmatrix}}_{\varepsilon_t}$$

Note that as the diagonal elements of B_0 equal unity, we don't assume that $E(\varepsilon_t\varepsilon_t') = \Sigma_\varepsilon$ is the identity matrix. So this is a different normalization rule as before.

Economically, the above restrictions embody to some extent a baseline IS-LM model:

a) The first equation provides three restrictions by assuming that the price level is predetermined except that producers can respond immediately to aggregate supply shocks (e.g. an unexpected increase in oil or gas prices). This is basically a horizontal aggregate supply (AS) curve; that is why we label the shock with AS.

b) The second equation provides no restrictions as it is assumed that real output responds to all other model variables contemporaneously. This equation can be interpreted as an aggregate demand curve, or an IS curve, that is why we label the shock IS.

c) The third equation provides two restrictions by assuming that the interest rate does not react contemporaneously to aggregate measures of output and prices. This represents a simple money supply function, according to which the central bank adjusts the rate of interest in relation to the money stock and does not immediately observe aggregate output and aggregate prices. Of course, this is in contrast to modern monetary policy theory (and practice).

d) The fourth equation provides one additional restriction by assuming that the first two entries in the last row are identical. The underlying idea is that this equation represents a money demand function in which short-run money holdings rise in proportion to NOMINAL GNP. Moreover, money holdings are allowed to be dependent on the interest rate.

In sum, we have $3+0+2+1 = 6$ restrictions, that is we have an exactly identified SVAR model, identified by short-run exclusion restrictions on B_0. As the structure is not recursive, a Cholesky decomposition is not valid and hence we need to rely on a numerical optimizer (or an algorithm for short-run exclusion restrictions).

3. The code might look like this:

progs/matlab/keatingSR.m

```
% ------------------------------------------------------------------------
% Short-run restrictions in the Keating (1992) model.
% ------------------------------------------------------------------------
% Willi Mutschler, December 27, 2022
% willi@mutschler.eu
```

```
% --------------------------------------------------------------------------
clearvars; clc; close all;

% data handling
Keating1992 = importdata('../../data/Keating1992.csv');
ENDO = Keating1992.data;
[obs_nbr,var_nbr] = size(ENDO);

% estimate reduced-form
nlag = 4;
opt.const = 1;
VAR = VARReducedForm(ENDO,nlag,opt);

% identification restrictions are set up in keatingSR_f_SR
f = str2func('keatingSR_f');

% options for fsolve
TolX            = 1e-4;  % termination tolerance on the current point
TolFun          = 1e-9;  % termination tolerance on the function value
MaxFunEvals     = 50000; % maximum number of function evaluations allowed
MaxIter         = 1000;  % maximum numberof iterations allowed
OptimAlgorithm = 'trust-region-dogleg'; % algorithm used in fsolve
options = optimset('TolX',TolX,'TolFun',TolFun,'MaxFunEvals',MaxFunEvals,'MaxIter',MaxIter,'Algorithm',OptimAlgorithm);

% initital guess, note that we need candidates for both B_0 and diag(SIG_eps) stored into a single candidate matrix
B0_diageps= [eye(var_nbr) ones(var_nbr,1)]; % the first 4 columns belong to B_0, the last column are the diagonal elements of SIG_eps
f(B0_diageps,VAR.SigmaOLS)' % test whether function works at initial value (should give you no error)

% call optimization routine fsolve to minimize f
[B0_diageps,fval,exitflag,output] = fsolve(f,B0_diageps,options,  VAR.SigmaOLS);

% display results
B0 = B0_diageps(:,1:var_nbr); % get B0
SIGeps = B0_diageps(:,var_nbr+1); % these are just the variances
impact = inv(B0)*diag(sqrt(SIGeps)); % compute impact matrix used to plot IRFs, i.e. inv(B0)*sqrt(SigEps)
table(impact)

% compute and plot structural impulse response function
nSteps = 12;
cumsumIndicator = [1 1 0 1];
varNames = ["Deflator", "Real GNP", "Federal Funds Rate", "M1"];
epsNames = ["AS", "IS", "MS", "MD"] + " shock";
irfPoint = irfPlots(VAR.Acomp,impact,nSteps,cumsumIndicator,varNames,epsNames);
```

Here is the helper function to impose the restrictions:

progs/matlab/keatingSR_f.m

```
function f = keatingSR_f(B0_diageps,hatSigmaU)
```

```
% f = keatingSR_f(B0_diageps,hatSigmaU)
% -----------------------------------------------------------------------
% Evaluates the system of nonlinear equations vech(hatSigmaU) =
% vech(B0inv*SIGeps*B0inv') where SIGeps has only values on the diagonal
% given a candidate for B0 and the diagonal elements of SIGeps
% subject to the short-run restrictions in the Keating (1992) model.
% -----------------------------------------------------------------------
% INPUTS
%   - B0_diageps   [nvars x (nvars+1)]  candidate matrix for both short-run impact matrix [nvars x nvars] and diagonal elements of SIGeps (nvar x 1)
%   - hatSigmaU    [nvars x nvars]      covariance matrix of reduced-form residuals
% -----------------------------------------------------------------------
% OUTPUTS
%   - f : function value, see below
% -----------------------------------------------------------------------
% Willi Mutschler, December 14, 2022
% willi@mutschler.eu
% -----------------------------------------------------------------------

nvars = size(hatSigmaU,1);  % number of variables
B0 = B0_diageps(:,1:nvars); % get B0 matrix from candidate matrix
diageps = diag(B0_diageps(1:nvars,nvars+1)); % get diagonal elements of SIG_eps from candidate matrix and make it a full diagonal matrix
B0inv = inv(B0); % compute short-run impact matrix

f = [vech(B0inv*diageps*B0inv'-hatSigmaU);
     B0(1,1) - 1;
     B0(3,1) - 0;
     B0(4,1) - B0(4,2);
     B0(1,2) - 0;
     B0(2,2) - 1;
     B0(3,2) - 0;
     B0(1,3) - 0;
     B0(3,3) - 1;
     B0(1,4) - 0;
     B0(4,4) - 1;
    ];
```

4. An unexpected upward shift of the aggregate supply curve
   - raises the price deflator
   - lowers real GNP
   - raises the federal funds rate
   - lowers the money supply at first, but ultimately raises M1

   This response is not really appropriate for the U.S. economy during the considered sample period, making the identifying restrictions rather questionable.