Macroeconometrics Problem Set 4: Explanatory Notes

Corti Pietro (3061891), Reichlin Vito (3081010)

11 December 2022

Point 1

We start by considering the following ECM:

$$\Delta y_t = \begin{bmatrix} -0.6 & 0.5 \\ 0.5 & -5/12 \end{bmatrix} y_{t-1} - \begin{bmatrix} 0.4 & 0.5 \\ 0 & 0 \end{bmatrix} \Delta y_{t-1} + \varepsilon_t$$

The general formulation for the ECM can be given by the following equation:

$$\Delta y_t = \xi(L)\Delta y_{t-1} - \Phi(I) y_{t-1} + \varepsilon_t$$

The matrix $\Phi(I)$ is the matrix of coefficients for the VAR computed in I, and it can be written as the product of two matrices B and A (in general both nxk matrices), where A is the matrix of cointegrated vectors.
The matrix $\xi(L)$ is the matrix of coefficients corresponding to the first differences. In our case we have:

$$\Phi(I) = \begin{bmatrix} 0.6 & -0.5 \\ -0.5 & 5/12 \end{bmatrix}$$

and

$$\xi(L) = \begin{bmatrix} -0.4 & -0.5 \\ 0 & 0 \end{bmatrix}$$

In general we can say that $rank(\Phi(I)) = k$ where k = number of cointegrated vectors. So, the first thing to do here is to compute the rank of this matrix to find the number of cointegrations. One easy way to do it is applying Kronecker algorithm, starting by the determinant of this matrix.
With easy computations we find that:

$$det(\Phi(I)) = \frac{6}{10}\frac{5}{12} - \frac{5}{10}\frac{5}{10} = \frac{1}{4} - \frac{1}{4} = 0$$

We easily understand that the two vectors of the matrix are linearly dependent, and so $rank(\Phi(I)) = 1$, i.e. we only have one cointegration in this process.

Point 2

From what we have found before we can derive the cointegrating vector A and the matrix B. To find the cointegrating vector we normalize its first element such that the relationship between the components of y is:

$$y_{1,t} = \gamma y_{2,t} + \varepsilon_t$$

Notice that the matrix $\Phi(I)$ can be decomposed as the product of the matrix of cointegrating vectors A and a matrix B of coefficients. In this case:

$$\Phi(I) = BA' = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \begin{bmatrix} 1 & \gamma \end{bmatrix} = \begin{bmatrix} 0.6 & -0.5 \\ -0.5 & 5/12 \end{bmatrix}$$

from which we can find:

$$B = \begin{bmatrix} \frac{3}{5} \\ -\frac{1}{2} \end{bmatrix} \quad A = \begin{bmatrix} 1 \\ -\frac{5}{6} \end{bmatrix}$$

In general we can define a basis of a generic vector space V as a set of linearly independent vectors $\{x_1, x_2, ..., x_k\} \in V$ that span the space V; in our case the vector A that we have just found is a basis for our cointegration space, given that it is linearly independent with itself, and its liner combination can span the whole space. Of course, the cointegrating vectors are not unique given that any multiple of A can be another cointegrating vector.

Point 3

To derive the VAR in levels it is useful to rewrite the ECM in the following way:

$$y_t - y_{t-1} = \begin{bmatrix} -0.6 & 0.5 \\ 0.5 & -5/12 \end{bmatrix} y_{t-1} - \begin{bmatrix} 0.4 & 0.5 \\ 0 & 0 \end{bmatrix} y_{t-1} + \begin{bmatrix} 0.4 & 0.5 \\ 0 & 0 \end{bmatrix} y_{t-2} + \varepsilon_t$$

rearranging we obtain:

$$y_t = \begin{bmatrix} 0 & 0 \\ 0.5 & 7/12 \end{bmatrix} y_{t-1} + \begin{bmatrix} 0.4 & 0.5 \\ 0 & 0 \end{bmatrix} y_{t-2} + \varepsilon_t$$

which is the VAR in levels we were looking for.
From this representation we can also find the Companion form representation, which will be useful when computing the Impulse Response Functions (IRFs). Let's call the matrices of the VAR in the following way:

$$A_1 = \begin{bmatrix} 0 & 0 \\ 0.5 & 7/12 \end{bmatrix} \quad A_2 = \begin{bmatrix} 0.4 & 0.5 \\ 0 & 0 \end{bmatrix}$$

So the Companion form matrix will be the following:

$$C = \begin{bmatrix} A_1 & A_2 \\ I & 0 \end{bmatrix}$$

where all the elements are 2x2 matrices, and as a consequence C is a 2x2 matrix.

Point 4

We can represent the regression with the companion form matrix in the following way:

$$z_t = Cz_{t-1} + v_t$$

where C has been defined above, and the other variables are the following:

$$z_t = \begin{bmatrix} y_t \\ y_{t-1} \end{bmatrix} \quad z_{t-1} = \begin{bmatrix} y_{t-1} \\ y_{t-2} \end{bmatrix} \quad v_t = \begin{bmatrix} \varepsilon_t \\ 0 \end{bmatrix}$$

In this way we can easily derive the VMA representation for this VAR in companion form as follows:

$$z_t = \Psi(L)v_t \quad with \quad \Psi(L)_i = C^i$$

so the process can be written in polynomial form in the following way:

$$z_t = (I + CL + C^2L^2 + ...)v_t$$

this representation is very useful to derive the IRFs for the model. In fact let's suppose to give a unitary shock at time t, and let's observe what happens in the following time periods:

- t: $v_t = 1$, i.e. a unitary shock
- $t+1$: $z_{t+1} = (I + CL + C^2L^2 + ...)v_{t+1}$, so the shock of this period is: $CLv_{t+1} = Cv_t = C$
- $t+2$: $z_{t+2} = (I + CL + C^2L^2 + ...)v_{t+2}$, so here the shock will be: C^2

we can proceed in this way for every period using the same logic.
To represent this in our Matlab file, after having represented the matrices A_1, A_2, C as defined above we have created the IRFs with a simple loop in which at every step we identify the power of the companion form matrix ("called "Cn" in the file) and store its value inside the object "TIRF". To represent it graphically we use similar commands to previous problem sets such as "squeeze" to reduce the dimensionality and represent graphically our object.
The graphical representation is the following:

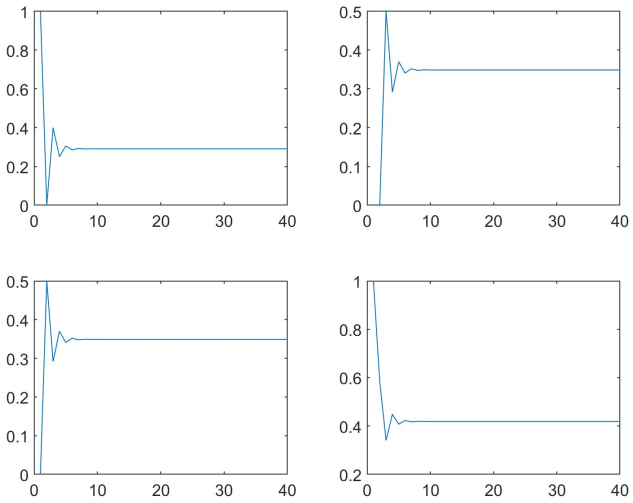

Figure 1: Theoretical IRFs for the model presented

Point 5

a) For this point the first thing we do is to set the parameters values such as the dimension "T", 250, and the number of Monte-Carlo simulations we have "mc", 1000.
After this we start a loop to obtain the effective Monte-Carlo after having identified the storing for the IRFs ("IRF_lev_storing" in the file) that will be used to collect all results and to compute means and confidence intervals.
Then, the data generating process (DGP from now on) is computed following the text: we call the dependent variable "y" and use the matrices from the VAR in level; for the first loops we have used the error term called "eps" with a "randn" command. Then we inverted "y" in "Y" so to get a 250x2 matrix and the split it into "Y1" and "Y2"; the VAR is estimated with a "varm" command using for the moment one lag (but this can be changed easily).
With the stored OLS we compute the IRFs in the usual way with a loop for the VMA representation, and store them in the variables "IRF_lev" and we use the squeeze command for graphical representation as usual.
After having computed the means ("m_IRF_lev") and the confidence intervals ("ci_IRF1_lev") with the storing function used before, we can represent graphically our MC simulation.

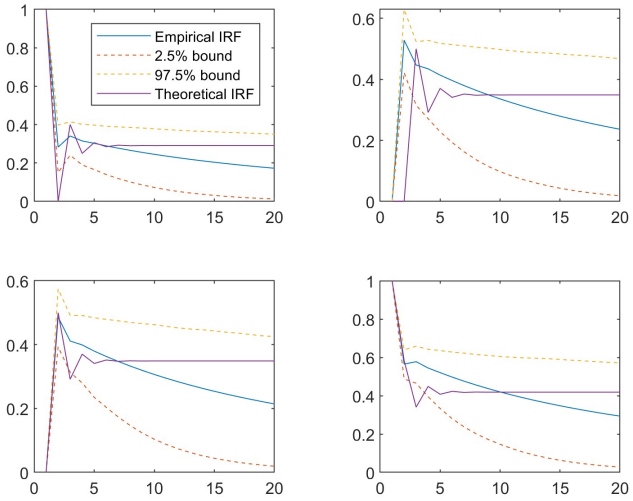

Figure 2: IRFs for the VAR in level

as we can see, the shape of the empirical IRFs and the theoretical ones are similar but of course they do not collide; the spikes represented by the first shocks are present in both of them, but while the theoretical one becomes immediately "flat" as seen above, the empirical one declines slowly, in line with the behavior of a VAR.

b) The construction of a VAR in difference is done in a similar way; however, we repeat the DGP not to overload Matlab and to obtain a more straightforward way to capture the difference. The dependent variable "Y" is obtained in the same way of previous point, but here we have to identify the variable "Z" which is the difference between observations; by doing so we can use again a "varm" to identify the OLS in difference (called "B" in our Matlab file).
The next step is the construction of an automatic way to obtain the companion form, that will be also applied to the next point of the exercise where is asked to increase the number of lags; for the moment we use only one lag and represent it graphically. The automatic command for the companion form basically cumulates on the difference of OLS matrices so to move from the difference representation to the one in levels, and store the final result in the matrix "C" (see the Matlab file for further details).
The following part of the code is the usual one we use for the IRFs representation: we consider a loop for the power of the matrix "C" (called "Cn" in the file), and we store each step in the object called "irf"; we squeeze the results

of the MC simulation and we compute the means and the confidence intervals (respectively "m_IRF" and "ci_IRF" in our file).
The plot of these IRFs is the following:

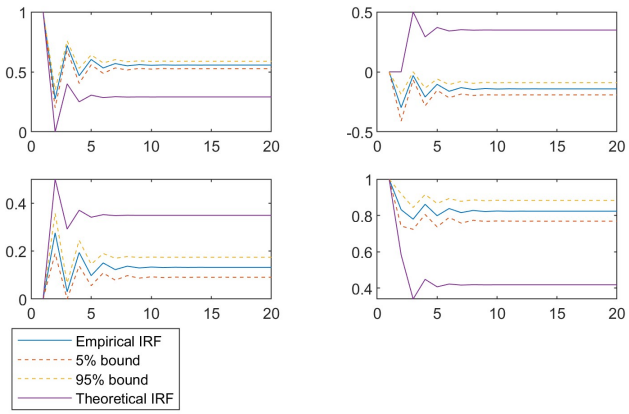

Figure 3: IRFs of the VAR in levels, but estimated from values in first difference

as we can see, they're different from the theoretical one (in purple) and, compared them with the empirical ones, they do not have the slow decline that characterized the previous ones. This is because a system like this one is not correctly specified.

c) The commands for this point are the same of the ones we used in point b, but we repeat them for each lag from 1 to 4. In the Matlab file we build we have two loops: one for the lags, the other for the MC simulation. VAR estimation, OLS matrix and IRFs are computed exactly in the same fashion of before.
The graphical representation is the following (each row corresponds to a lag specification).

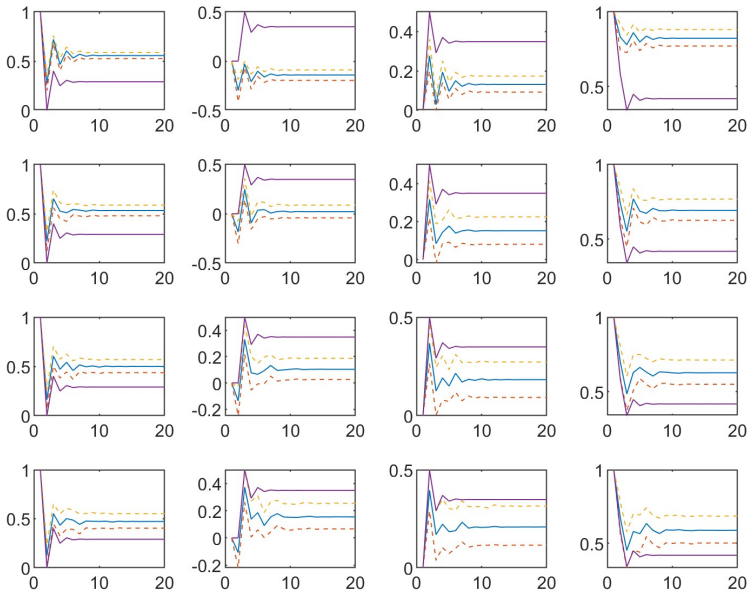

Figure 4: IRFs for the VAR estimated with differences. Each row represent a different lag specification. Top row is 1 lag and last row is 4 lags.

The graph shows that as we move progressively from 1 to 4 lags we obtain better specifications for the curve, for each shock; in particular we notice that increasing the number of lags, the empirical IRF becomes closer and closer to the theoretical one, which means that increasing the number of lags gives better and more sophisticated representations of the shocks.

d) After having cleared all the variables, we write down again the DGP for the model given and the theoretical IRFs , exactly in the same way of the other points.
For the following steps we follow the Johansen procedure as described in the lecture notes.
Firstly, let's remember that any p-order $VAR(p)$ can be written in the ECM as follows:

$$\Delta y_t = \alpha + \xi_1 \Delta y_{t-1} + ... + \xi_{p-1} \Delta y_{t-p+1} + \xi_0 y_{t-1} + \varepsilon_t$$

If the y_{it}s are $I(1)$, then we know that we can write $\xi_0 = -BA'$. We now follow the steps.

Auxiliary regressions

The first step is to estimate the following auxiliary regressions:

$$\Delta y_t = \hat{\Pi}_0 + \hat{\Pi}_1 \Delta y_{t-1} + ... + \hat{\Pi}_{p-1} \Delta_{t-p+1} + \hat{u}_t$$

$$y_{t-1} = \hat{\theta} + \hat{\Xi}_1 \Delta y_{t-1} + ... + \hat{\Xi}_{p-1} y_{t-p+1} + \hat{v}_t$$

In our Matlab file this is done by identifying firstly the regressors in difference, stored in the variable "Z", and then by computing the respective OLS for the two regressions. In particular the one of the first regression (the one with Π) are stored in the variable "OLS_d", while the other (the one with Ξ) is stored in "OLS_l". Of course given that we are considering a MC simulation each OLS is in reality an object collecting 1000 OLS matrices, one for each step in the loop.

Canonical correlations

Here we compute the variances and covariances as described by the Johansen approach. In particular we have:

$$\hat{\Sigma}_{VV} = (1/T) \sum_{t=1}^{T} \hat{v}_t \hat{v}_t'$$

$$\hat{\Sigma}_{UU} = (1/T) \sum_{t=1}^{T} \hat{u}_t \hat{u}_t'$$

$$\hat{\Sigma}_{UV} = (1/T) \sum_{t=1}^{T} \hat{u}_t \hat{v}_t'$$

$$\hat{\Sigma}_{VU} = \hat{\Sigma}_{UV}'$$

Once we collected these values, we find the eigenvalues ($\hat{\lambda}_i$) and eigenvectors ($\hat{a}_i$) of the following matrix:

$$\hat{\Sigma}_{VV}^{-1} \hat{\Sigma}_{VU} \hat{\Sigma}_{UU}^{-1} \hat{\Sigma}_{UV}$$

In our Matlab file this is done by using the previous OLS and estimating the respective error terms for the two regressions, saved as "u" and "v". The Σs are instead called "uu", "vv" and "uv", while the matrix just identified above is called "S". The eigenvalues and eigenvectors were estimated by using the command "eig" an stored in the matrices "V" and "D" respectively. The next thing to do is to order the eigenvalues from the largest to the smallest and to normalize the eigenvectors as follows:

$$\hat{a}_i = \tilde{a}_i \sqrt{\tilde{a}_i' \hat{\Sigma}_{VV} \tilde{a}_i}$$

In the file this is done by the variable "V_hat" and the following loop for the ordering of the eigenvectors (see the file for further details).

ML estimates

The last thing we do before computing the IRFs for the ECM is the estimation of its parameters. In particular from the procedure we have:

$$\hat{A} = [\hat{a}_1 \hat{a}_2 ... \hat{a}_k]$$

$$\hat{\xi}_0 = \hat{\Sigma}_{UV} \hat{A} \hat{A}'$$

$$\xi_1 = \hat{\Pi}_1 - \hat{\xi}_0 \hat{\Xi}_1$$

in our file the estimated coefficients are collect respectively "xi_zero" and "xi_one". The last thing we compute is the variance for this model:

$$\hat{\Omega} = (1/T) \sum_{t=1}^{T} [(\hat{u}_t - \hat{\xi}_0 \hat{v}_t)'(\hat{u}_t - \hat{\xi}_0 \hat{v}_t)]$$

which is called "omega" in the file.
Lastly, to represent graphically the IRF as requested we have created storing values for them, called "jIRF"s that are filled with empirical IRFs computed in the same way of before, i.e. creating a loop with the companion form of the ECM we have just estimated.
Using "squeeze" and computing means and confidence intervals, the plot is the following:

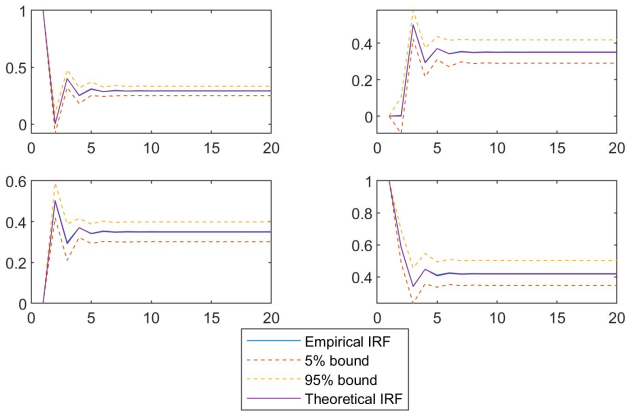

Figure 5: IRFs for the estimated ECM with Johansen procedure

as we can see, the model seems to be correctly estimated given that the theoretical IRF and the one found with ECM are almost overlapping. As a

final remark, we notice that the empirical IRFs computed through different estimation methods differ. When data is estimated using a VAR in level we have OLS suffering from a downward bias since the process contains unit roots, this causes the empirical IRFs to converge toward zero. In the case of the VAR in differences, the estimates are actually not consistent but approximate the theoretical (true) IRFs when more lags are introduced. Lastly, the ECM approach estimates are very precise as they resemble the Data Generating Process, in particular ECMs perform well when cointegrators are present in a non-stationary process.

References

- Bruno, Giovanni (2022), "Econometric Models: Lecture Notes for 2021/2022 ESS"
- Ghysels, Eric Marcellino, Massimiliano (2018), "Applied Economic Forecasting Using Time Series Methods", Oxford University Press
- Enders, Walter (1995), "Applied Econometric Time Series", Wiley Series in Probability and Mathematical Statistics, John Wiley Sons
- Johansen, S. (1988), "Statistical analysis of cointegration vectors", Journal of Economic Dynamics and Control, Vol. 12, Issues 2–3, pp. 231-254, ISSN 0165-1889, https://doi.org/10.1016/0165-1889(88)90041-3.
- Johansen, S. (1991), "Estimation and Hypothesis Testing of Cointegration Vectors in Gaussian Vector Autoregressive Models", Econometrica, vol. 59, n. 6, pp. 1551-1580, URL: https://www.jstor.org/stable/2938278
- Hamilton, James D. (1994), "Time Series Analysis", Princeton University Press
- Sala, Luca (2022), "Notes on Time Series Analysis"