Software to run the script: DYNARE and OCTAVE

Lecture 4

Onno Kuik

MOSAICC

Dynare/Octave

- DYNARE is a modeling language
- GNU OCTAVE is a high-level language for numerical computations

- The model is written in the DYNARE language and solved by OCTAVE
- Both are freely redistributable software

Installation

- We do the installation together

MOSAICC

A look at the script

- We will now look at the script of the Dynare file **faomodel.mod**

Syntax of Dynare

- What is written after the sign "//" is no read during the execution of the program.
- A full section is not taken into account during the execution of the program if it is between the following signs:

  /*

  ..... (not executed)

  */

- Executed commands finish with a ";"
- It is possible to split the arguments of a command on several line (since the end of the command is define by ";")
- Line started by "@#" are part of the Dynare Macro Language.

Explanation of the script

The model is programmed in a file called "**faomodel.mod**".

It can be executed by Octave using one of the following commands:

- **dynare faomodel.mod savemacro**

  This executes the simulation and creates another file ("faomodel-macroexp.mod") which compiles the full script of model by writing explicitly each equation (not using any index as in the original "faomodel.mod"). The execution of "faomodel-macroexp.mod" would thus provide the same simulation. **Warning**: the name must be change since the "-" sign is not allowed in the "dynare" function. For instance, change it to "faomodelmacroexp.mod".

- **dynare faomodel**

  This command does the same as the above one but without compiling the full script of model (the "faomodel-macroexp.mod" file).

Explanation of the script

The script is divided in 5 parts. :

1. VARIABLES DECLARATIONS
2. MODEL SPECIFICATION
3. INITIALISATION OF PARAMETERS AND OF ENDOGENOUS AND EXOGENOUS VARIABLES
4. SIMULATION OF THE MODEL
5. PLOTS & SAVING RESULTS

- **The section of the script that need an input from the user start by the mention "USER INPUT" and ends by the mention "USER INPUT (END)".**
- **The lines of the script that need an input from the user have the mention "INPUT VALUES".**

For the good functioning of the model, we recommend not to modify any other section of the model.

MOSAICC

Test and real versions

- **Test version**: this mode allows for simulating an automatically calibrated model. This is useful to cheek if the numbers of sets are correctly defined. Shock can also be performed to cheek the general property of the model. But the calibration is not realistic. For instance, it assumes that the production of each commodity is produced uniformly across each activity. The data that the user can input are signaled by "INPUT VALUES (TEST)". We recommend to simulate this version before starting the calibration on real data. If the test version does not work, the real data one will not either. The most common failure of the test version is the inconstancy between the definition of the sets and their number (see below for more detail).
- **Real data version**: this mode allows for simulating a realistically calibrated model (on real data). To do so a proper Social accounting matrix (SAM) should first be constructed and used as input. Then the user must correctly initialize the variables or parameters mentioned with "INPUT VALUES (REAL DATA)" (see below for more detail).

User delarations

// Choose the version of model you which to simulate: 1 for expanded dynamic, something else for simple dynamic

@#define expdynamic = 10

// Choose the load data option: 1 for real, something else for automatic test

@#define realdata = 1

Automatic calibration

The original SAM matrix used for Morocco is available in "data.xls", sheet "SAM". Calibrating a variable from the SAM matrix is quite straightforward and part of the procedure is automated. We give here one example:

```
step=0;
@#for c in commodities
        QQ_@{c} = SAM(21+step,42); // ******* INPUT VALUES (REAL DATA)
        QM_@{c} = SAM(34,1+step);  // ******* INPUT VALUES (REAL DATA)
step=step+1;
@#endfor
```

Automatic calibration

The program input the data for the variable QQ_@{c} of the first commodity from the SAM matrix (line 21, column 42). We define a step that is incremented once the data for the first commodity is initialized. For the second commodity, the input comes from the SAM (line 22 (=21+1), column 42), for the third, from SAM (line 23 (=21+2), colomn 42), etc.

Of course the step can be used for incrementing the column (see QM_@{c}). When both line and column should be incremented, two steps are defined:

```
step=0;
@#for c in commodities
    step2=0;
    @#for a in activities
                    QXAC_@{a}_@{c} = SAM(9+step2,21+step)+0.00001;
    step2=step2+1;
    @#endfor
step=step+1;
@#endfor
```

MOSAICC

Automatic calibration

Two variables should not be calibrated to 0: the commodity c produced by activity a (QXAC_@{a}_@{c}) and the production factor f of activity a (QF_@{f}_@{a}). Otherwise the model does not converge. This creates a linear dependence between several equations and the model is not solvable (singularity problem). The user will get one or a combination of the following error messages:

Warning: Matrix is close to singular or badly scaled.

Warning: Matrix is singular to working precision.

Warning: Divide by zero. This warning will be removed in a future release.

Consider using DBSTOP IF NANINF when debugging.

Simulation

The user inputs the shock and the model is simulated with the command "simul". Two types of shocks can be implemented:

- **Constant shock**. This consists in a permanent increase of a parameter or an exogenous variable. The shock is constant over the all simulation period. For example, a permanent increase of the VAT rate of 0.01 or a increase permanent of public spending of 1%:

```
@#for a in activities
        tva_@{a}=tva_@{a} + 0.01;
@#endfor

initval;
@#for c in commodities
        QG_@{c} = QG_@{c}*1.01;
@#endfor
end;
```

MOSAICC

Simulation

- **Variable shock**. This consists in a change over time of an exogenous variable. It is defined by the command "shocks". The sheet "ExoVar" of the data.xls file helps to construct this type of shock. This is the shock we simulated in the empirical Moroccan example.

Plots & saving results

6.5 PLOTS & SAVING RESULTS

Results can be plot directly on Octave (rplot command) or save in a text file (dynatype command):

6.5.1 PLOT ENDOGENOUS VARIABLES

rplot BBB; // Plot the variable BBB (only endogenous variable can be plotted)

6.5.2 SAVE RESULTS IN A TEXT FILE PLOT ENDOGENOUS VARIABLES

dynatype(output); // Save all endogenous variables in the file "output" !! **WARNING !! The output file must not be open by another application such as Excel** (but can be open by a text editor such Crimson Editor)

dynatype(output2) YG YH; // Save the listed endogenous variables (here YG and YH) in the file "output2"

Plots & saving results

6.5.3 Data reorganisation on Excel

The endogenous variables are saved in a text file. They appear consecutively in the same column which is not very user friendly. In order to help data output treatment, an automatic reorganisation of the variables in different column has been programmed on excel (see file data.xls). The procedure goes as follows:

1. Open the text file "output" with excel. This should automatically propose to use the "Text Import Wizard" (accept all the steps).
2. Copy-paste the first column of the created file in the spreadsheet "output" of the data.xls file.
3. In the spreadsheet "Initialization", select the good numbers of simulation periods, of lags and of leads. In the example provided with the SDM, these values are 30, 0, 1 respectively (one lead because Dynare require the use of at least one forward-looking variable). For the EDM, these values are 100, 1, 1 respectively.
4. Once the numbers of simulation periods, lags and leads are correctly initialized, the data can be found arranged in column in the spreadsheet "result". One may have to copy or delete the formula in order to have the right number of period.

MOSAICC

Plots & saving results

Please note that the first and last data are not simulation but the initialisation of the date provided by the user. So for a simulation of 30 periods, the command "Dynatype" will provide 32 data. The first and the last data should not be considered for the interpretation of the results.

Now try it yourself
Good luck!

MOSAICC