

READ ME FILE

Paper title: DSGE Reno: Adding a Housing Block to a Small Open Economy Model

Authors: Christopher G Gibbs, Jonathan Hambur and Gabriela Nodari

Description

This 'read me' file contains details of the code and data included in this archive that were used to generate the results reported in RDP 2018-04. Plotting data for all figures appearing in the RDP can be found in the spreadsheet 'RDP 2018-04 Graph Data.xlsx'.

Data

See Appendix A in the paper for details on the data sources and construction. The data are contained in FourSectorData.xlsx.

Figure data:

Data for all figures are available for external release. Data for Figures 3–5 are used to plot a visual representation of supply and demand and are not actual data.

Code

The results for the paper were constructed using Matlab version 2015b, Dynare version 4.4.0 and Eviews version 9. All files are available for external release.

Foreign Sector:

The code to estimate the foreign sector can be found in the 'Foreign Sector' directory.

This directory contains a number of Dynare files:

- **multisector_foreign.mod:** Run this code to estimate the model. It calls a number of sub-files (below). To run, FourSectorData.xlsx must be in the current working directory.
- **variable_declarations_foreign.mod:** This file declares the model variables and sets the calibrations for the calibrated parameters.
- **model_equations_foreign.mod:** This file contains the model equations.
- **calc_steadystate_foreign.mod:** This file calculates the steady state values of the model.
- **estimate_model_foreign.mod:** This file outlines the priors for the estimated parameters and estimates the model. The FourSectorData.xlsx must be in the current working directory.

The results in **Table 3** can be obtained by running the multisector_foreign.mod code. Existing results are stored in multisector_foreign_results, while the multisector_foreign sub-directory contains details of the Bayesian estimation.

Domestic Sector:

The code to estimate the domestic sector can be found in the 'Domestic Sector' directory.

This directory contains a number of Dynare files:

- **multisector.mod:** Run this code to estimate the model and/or construct impulse response functions. It calls a number of sub-files, outlined below. FourSectorData.xlsx must be in the current working directory.
 - To re-estimate the model, set estimate_mod = 1 in line 13 and define_irfs = 0 in line 16.
 - To produce standard impulse response functions (IRFs), run scenarios or produce decompositions, set estimate_mod = 0 in line 13 and define_irfs = 0 in line 16. The 'multisector_mode' file and

'multisector' sub-directory (with estimated Metropolis Hastings output inside) must be in the current directory.

- To produce Bayesian IRFs, set `estimate_mod = 0` in line 13 and `define_irfs = 1` in line 16. The Bayesian IRFs output will be stored in the `oo_` object. This can be moved into the 'Bayesian IRFs' directory to produce **Figures 9, B1 and C1** (see details below).
- **variable_declarations.mod**: This file declares the model variables.
- **calibrations_baseline.mod**: This file contains the calibration for parameters affecting the steady state of the models well as the measurement errors' standard deviations.
- **dynamic_calibrations_forestimation.mod**: Sets initial values for estimated parameters, and calibrated parameters for investment adjustment costs and foreign sector. Estimated value of the foreign sector parameters should be entered here (posterior mean).
- **model_equations_loglinear.mod**: This file contains the model equations.
- **calc_steadystate_nonlinear.mod**: This file calculates the nonlinear steady-state values of the model. It relies on having `Manualsteadystate.m`, `LMFsolve.m` and `calcss_external.m` in the working directory. To change steady-state shares, make sure to update in `calcss_external.m` (lines 86–95).
- **estimate_model.mod**: This file outlines the priors for the estimated parameters and estimates the model. The `FourSectorData.xlsx` must be in the current working directory.
- **run_model.mod**: This file outlines the priors for the estimated parameters and runs the model using previous estimation results. The `FourSectorData.xlsx` must be in the current working directory. The `multisector_mode` file and `multisector` sub-directory (with estimated Metropolis Hastings output inside) must be in the current directory.
- **run_model_bayes_irfs.mod**: This file outlines the priors for the estimated parameters, runs the model using previous estimation results and produces Bayesian IRFs. The `FourSectorData.xlsx` must be in the current working directory. The `multisector_mode` file and `multisector` sub-directory (with estimated Metropolis Hastings output) must be in the current directory.
- **check_calibration_targets_loglinear.mod**: This code checks the steady state and dynamic predictions of the model against pre-defined allowable values. The values can be changed in the code.

The results in **Tables 2 (last column), 4 and 5** can be obtained by running the `multisector.mod` code with any of the settings. Existing results are stored in `multisector_results`, while the `multisector` sub-directory contains details of the Bayesian estimation.

To produce the IRFs for **Figures 10, 11, 13, 14, 15, B2, B3, and C2**, run `IRF_construct.m`. This code relies on having the dynare programs discussed above in the current working directory. Set `estimate_mod = 0` in line 13 and `define_irfs = 0` in line 16 of `multisector.mod`. It also requires `multisector_scenario_multiple.m`, `additional_irf_variables.m`, `xlswrite1.m` and `FourSectorData.xlsx`. IRFs for most model variables (including those in the figures) will be saved to `IRF_results.xlsx`.

To produce the Historical Decomposition in **Figure 16**, and the Scenario Results in **Figures 17 and 18**, run `ForcDecompScn.m`. This code relies on having the dynare programs discussed above in the current working directory. Set `estimate_mod = 0` in line 13 and `define_irfs = 0` in line 16 of `multisector.mod`. It also requires `multisector_scenario_multiple.m`, `Generate_forecast_distribution.m`, `Recover_sample_means.m`, `HistoricalDecompositionCode.m`, `nan_avg.m`, `xlswrite1.m` and `FourSectorData.xlsx`. The results are saved in `HDecompNew.xlsx` (for Figure 16), and `Scenarios.xlsx` (for Figures 17 and 18).

Forecasting:

The code to perform the forecast exercise is contained in the 'Forecasting' directory.

This folder contains a number of Dynare files, which are discussed in the 'Domestic Sector' sections above.

To construct the DSGE forecasts, run the Forecast.m code. This recursively estimates the DSGE model and produces forecasts. These are then saved in the Forecasts Point.xlsx and Forecast Means.xlsx files. We use the former for our analysis.

To run, the following files need to be in the working directory: multisector.mod; variable_declarations.mod; calibrations_baseline.mod; dynamic_calibrations_forestimation.mod model_equations_loglinear.mod; calc_steadystate_nonlinear.mod; Manualsteadystate.m; LMFsolve.m; calcss_external.m; xlswrite2.m; and estimate_model.mod. FourSectorData.xlsx must be in the current working directory.

To produce the AR(1), naïve model forecasts, open foursectordata.wf1 in EViews and run AR forecasting.prg. This will produce matrices containing forecasts for the variables of interest.

For the forecast comparison, paste the DSGE and naïve forecasts into the relevant sheets in Forecasts Point assess AR1.xlsx. This file links to the FourSectorData.xlsx file which must be in the current working directory. This will produce the results underlying **Figures 6, 7 and 8**.

Structural VARs:

The code to estimate the structural VAR models and reproduce **Figure 12** can be found in the 'SVAR' directory. This directory contains a number of Matlab files, outlined below.

The VARdataRDP.xlsx file contains the data used in the SVAR models. It must be in the current working directory when running the Matlab codes .

- **run_VARS.m:** This is the main code. Run this code to estimate the four SVAR model specifications, construct impulse response functions and plot Figure 12. It will automatically call the sub-routines below.
- **doVar.m:** This function estimates the SVARs and computes IRFs. It calls olsvar_block_exogenous.m, var_irf.m and kilian_bootstrap_block_exogenous.m.
- **get_dummies.m:** This function creates the dummy variables used in the SVAR models.
- **kilian_bootstrap_block_exogenous.m:** This function calculates the Kilian small sample confidence intervals for SVAR impulse response functions (see L Kilian (1998), 'Small-Sample Confidence Intervals for Impulse Response Functions', *The Review of Economics and Statistics*, 80(2), pp 218–230).
- **olsvar_block_exogenous.m:** This function estimates a block exogenous SVAR by OLS.
- **load_key_varnums.m:** This sub-routine assigns a name to the column numbers of the data, which makes it easier to construct data matrices for the SVAR.
- **Squantile.m:** This function calculates end point of confidence intervals.
- **var_irf.m:** This function produces impulse response functions for a SVAR.

Existing results are stored in VAR_model_1, VAR_model_2, VAR_model_3 and VAR_model_4.

Bayesian IRFs:

The code to construct **Figures 9, B1 and C1** is contained in the 'Bayesian IRFs' directory.

To construct the figures, run IRFdataGraphit.m. This pulls from the bayes_irfs_three_sector and bayes_irfs_four_sector Matlab files.

To update the four sector IRFs, run multisector.mod in the 'Domestic Sector' directory, setting estimate_mod = 0 in line 13 and define_irfs = 1 in line 16. The Bayesian IRF output will be stored in the oo_object.