Replication Code for "Monetary Policy, Bounded Rationality, and Incomplete Markets"

Emmanuel Farhi Harvard University

Iván Werning MIT

June 26, 2019

This note documents the Matlab code to replicate the figures in the paper "Monetary Policy, Bounded Rationality, and Incomplete Markets", by Emmanuel Farhi and Iván Werning.

1 Prerequisites

The code relies on the third party `export_fig` package to save the figures in PDF format. The package is included in the replication file, but it requires installed versions of Ghostscript and Xpdf command line tools to work.

2 Instructions to Replicate Figures

i. Unzip the folder `levelk_replication_code`.

ii. Open Matlab (code tested under Matlab R2018a and R2019a) and change current folder to `levelk_replication_code`.

iii. Run `main.m` to add the necessary folders to your path and load the baseline calibration.

iv. Run any `figX.m` script to generate the simulations and draw the graphs.

These scripts were run with 8 workers on parallel using a Linux-based machine Intel(R) Xeon(R) CPU E7-8880 v4 @ 2.20GHz (x176) running RedHat 7.6, with 1.48TB of RAM. Running times were roughly <30m for `fig1.m` and `fig3.m`, ~2h for `fig5.m`, and ~6h for `fig2.m` and `fig4.m`.

3 Documentation

The package is divided in four folders:

- `export_fig`: package by Yair Altman to export Matlab figures to PDF format properly. It can be found on GitHub following https://github.com/altmany/export_fig.
- `figures`: where the figures are stored.
- `simulations`: folder to store the simulated data on which figures are based.
- `src`: contains the source code to solve the incomplete markets models and calculate the response to monetary policy shocks under level-k thinking. It also contains `rouwenhorst.m`, a third party implementation of the Rouwenhorst method by Karen A. Kopecky.

The source code is written using the object-oriented programming (OOP) paradigm. There are eight classes:

- `SteadyStateReducedForm`: calculates the steady state of the *rigid* prices incomplete markets model from **Section 4.2** in the paper.
  - `DynamicsReducedForm`: inherits from its parent class `SteadyStateReducedForm`. Calculates the different level-k responses of the economy to a one-time unexpected monetary policy shock.
- `SteadyStateReducedFormRA` and `DynamicsReducedFormRA`: same as above, with complete markets.
- `SteadyStateUpstream`: calculates the steady state of the *sticky* prices incomplete markets model from **Section 5** in the paper.
  - `DynamicsInfUpstream`: inherits from its parent class `SteadyStateUpstream`. Calculates the different level-k responses of the economy to a one-time unexpected monetary policy shock.
- `SteadyStateUpstreamRA` and `DynamicsInfUpstream`: same as above, with complete markets.

A simulation is just an instance (object) of one of this types (classes).