CW2010-CreditSpreadsMonPol

These codes reproduce the results in:

Cúrdia, V., and M. Woodford (2010)
Credit Spreads and Monetary Policy *Journal of Money, Credit and Banking*, 42 (s1), pp. 3-35.

Technical Appendix

These replication codes are available online at:
https://github.com/vcurdia/CW2010-CreditSpreadsMonPol

Requirements

Matlab (R)

The codes were tested using Matlab (R) R2016b with the following toolboxes

- Symbolic Toolbox
- Optimization Toolbox

LaTeX

LaTeX is used by some tools to compile certain documents.

`epstopdf`, included in most LaTeX releases, is used by some tools.

Additional codes

Codes by Vasco Cúrdia:

- ACR-LQ joint with Filippo Altissimo and Diego Rodriguez Palenzuela, version v1.0.0

Codes by Chris Sims:

- gensys
- optimize

All auxiliary codes included in this repository in subfolders.

Description of Replication Codes

`IntModelFF.m`
Solves the model for a given parameter specification. Code itself has a section describing default parameter configuration (for those optional) and a list of all possible optional arguments. Output is stored in mat files starting with "Output_", concatenated with a specifier of the specific exercise name, generated based on the options used. This mat file needs to be produced before plotting and welfare evaluations can take place. As part of the output display the welfare level for each policy is shown.

`IntModelNoFF.m`
Solves the model with heterogeneous agents, but no spread, for a given parameter specification. Code itself has a section describing default parameter configuration (for those optional) and a list of all possible optional arguments. Output is stored in mat files starting with "Output_", concatenated with a specifier of the specific exercise name, generated based on the options used. This mat file needs to be produced before plotting and welfare evaluations can take place. As part of the output display the welfare level for each policy is shown.

`IntModelRepHH.m`
Solves the model with representative agent (no heterogenous agents and no spread) for a given parameter specification. Code itself has a section describing default parameter configuration (for those optional) and a list of all possible optional arguments. Output is stored in mat files starting with "Output_", concatenated with a specifier of the specific exercise name, generated based on the options used. This mat file needs to be produced before plotting and welfare evaluations can take place. As part of the output display the welfare level for each policy is shown.

`RunAllModels.m`
Runs several times the previous codes (`IntModelFF` , `IntModelNoFF` , `IntModelRepHH`) under alternative parameter configurations. The list of specifications to run can be adjusted at the beginning of the code.

`IRFPlotCompare.m`
Function that plots impulse response functions (IRF) to all the shocks considered (one shock per figure) comparing alternative models under a same policy rule or alternative policies for the same model. The first option is which model(s), specified as a cell array with the suffix of the models desired (`FF` , `NoFF` , and/or `RepHH`). The second option sets the alternative policies to compare (for a list of policies available, can simple load the specification of interest and simply type `IRF` , which will show all policies computed, where `LQ` stands for optimal policy). Cannot set multiple policies and multiple models at the same time. Choose either one model and multiple policies or one policy and multiple models. Can also choose a single model and a single policy. Can choose alternative composition of the panel (different variables to plot), whether to make figure in black and white or color, whether to create eps and many other alternatives. See first few sections of the code for all options. For example, the following command will create a figure comparing outcomes for the `FF` model with baseline calibration under optimal policy, flexible targeting criterion, and the Taylor rule:

```
IRFPlotCompare({'FF'},{'LQ','FlexTarget','Taylor'})
```

`ExerciseZLB.m`
Similar to `IntModelFF.m` but accounting for the ZLB.

`IRFPlotCompareZLB.m`
Plots the IRF to different shocks for the model under alternative policies. In this case the model is `FF` and the policies need to be specified inside the function code.

`TaylorSPAltCoef.m`
Performs a grid search for the optimal response to the spread, in response to a given shock, for a given model specification. Also plots welfare for alternative parameter values. The grid can and should be tweaked to get more or less precision.

`TaylorSPAltCoefFcn.m`
Function called by `TaylorSPAltCoef.m` .

`TaylorBAltCoef.m`
Performs a grid search for the optimal response to credit deviations from steady state, in response to a given shock, for a given model specification. Also plots welfare for alternative parameter values. The grid can and should be tweaked to get more or less precision.

`TaylorBAltCoefFcn.m`
Function called by `TaylorBAltCoef.m` .