README

Monetary Policy, Redistribution, and Risk Premia

Rohan Kekre* Moritz Lenel†

Description of files and programs

- `SolutionAlgorithm.pdf`: detailed description of the solution algorithm
- `src/params`: Matlab script producing .csv files that define the model parameterizations
- `src/fortran`: Fortran codes solving the model
- `src/matlab`: Matlab codes creating tables and figures
- `bin`: compilation scripts for Linux, Windows and macOS PCs as well as Unix-based distributed computing environments
- `output`: collects all figures and tables as presented in the paper and its appendix

Instructions to reproduce tables and figures

The working directory to execute all code and programs is `bin`. The example bash script `linux_run.sh` executes all of the following steps within that folder:

1. Within `bin`, run `src/params/create_param_files.m` to create the .csv files that characterize the nine different calibrations that the numerical results in the paper are based on.
2. Within `bin`, compile the solution program based on the provided Fortran source code. See `linux_run.sh` for an example on the relevant files that need to be linked in the compilation process. After compilation, run the program for every parameterization by providing indexes 1-9 as options to the executable.
3. Once the main program has been run for every parameterization, run `src/matlab/main.m` from within `bin` to produce tables and figures. Then compile `results.tex` in `output` within its own folder.

*Chicago Booth and NBER. Email: rohan.kekre@chicagobooth.edu.
†Princeton. Email: lenel@princeton.edu.

Note on the compilation of the Fortran code: The provided compilation scripts (in particular `linux_run.sh`) serve only as examples. They need to be adapted to any particular computing environment. We compiled the program using the Intel Fortran compiler which is freely available as part of the Intel oneAPI HPC toolkit. We used Matlab version R2021a to create tables and figures. The Fortran program uses the proprietary numerical library of the Numerical Algorithm Group (NAG). Most routines used in the code can be easily replaced by open source alternatives and we might provide an open source version of our code in the future. We are also working on a Julia translation of our numerical solution. This and codes for our related projects can be found at `https://github.com/KekreLenel/MPR`.

List of tables and figures and corresponding output

The Latex file `output/results.tex` collects all figures and tables. The matlab file `src/matlab/main.m` produces all figures and tables by calling various functions and scripts. The following table references the Matlab files ultimately producing the tables and figure files, all of which are located in `src/matlab/`. Figures and tables are saved in `output/figures` and `output/tables` (create those folders before running `main.m`).

Table/Figure #	Program	Output file
Table V	`create_moment_tables.m`	`Targeted_Moments_1.tex`
Table VI	`create_moment_tables.m`	`Untargeted_Moments_1.tex`
Figures 2-4	`plot_irfs.m`	`monetary_fig_split{1-3}.eps`
Table VII	`create_CS_tables.m`	`Campbell_Shiller.tex`
Table VIII/IX	`create_decomp_tables.m`	`decomp_tab_1/2.tex`
Figure A.2	`plot_irfs.m`	`TFP_compact.eps`
Figure A.3	`plot_irfs.m`	`dis_compact.eps`
Table A.IV	`create_moment_tables.m`	`Targeted_Moments_8.tex`
Figure A.5	`plot_irfs.m`	`interm_compact.eps`
Table A.VI	`create_CS_tables.m`	`Campbell_Shiller_interm.tex`
Table A.VI	`create_moment_tables.m`	`Targeted_Moments_6.tex`
Figure A.5	`plot_irfs.m`	`idio_compact.eps`
Table A.VII	`create_CS_tables.m`	`Campbell_Shiller_idio.tex`