# QUANTITATIVE REPLICATION OF GLOBAL BANKS AND SYSTEMIC DEBT CRISES

This file explains how to run the codes to replicate the main quantitative tables and figures from the paper. The folder contains all of the codes needed to replicate the results from the baseline model. Extensions and robustness files are inside the "robustness" folder. The version of Julia used is v.0.6, since the project was already quite developed by the time the v.1.x versions of Julia appeared.

The following packages are needed to run these codes: Distributions; Interpolations; Optim; Roots; FastGaussQuadrature; Plots; JLD; DataFrames.

## Folders

**tables**. Folder that has all of the tables presented in the paper. Use file "do_tables_paper.tex" to compile the pdf that collects all figures and tables shown in the paper.

**figures**. Folder that contains figures for the baseline model.

**model_data**. Folder where the solution of the model is saved.

**simulated_moments**. Folder where all moments are saved.

**robustness**. Folder containing model extensions that are relevant for the tables presented in the main part of the paper.

## Baseline Model Codes

**main_run.jl**. This is the main code; it calls all other codes. The option *do_module* allows you to solve the model starting from a guess, simulate the model, and perform other exercises shown in the paper. Note that you do not need to solve the model to simulate, since the structures are already saved in the folder *model_data*. Choose *do_module=1* to solve the model; *do_module=2* to simulate the model and compute moments; *do_module=3* to perform the Boom-Bust episode; and *do_module=4* to conduct the asset pricing exercise (codes are inside the robustness folder).

**00_tauchen.jl**. Tauchen discretization function.

**00_define_types.jl**. Defines types and some auxiliary functions. *CoreModule* has general elements and also defines objects for the economy without aggregate shocks. *AshocksModuleOBC* defines objects for the economy with aggregate shocks.

**00_structure_NAG.jl**. Functions for solving the economy without aggregate shocks.

**00_structure_ASM.jl**. Functions for solving the problem of the borrower and lender in an economy with aggregate shocks.

**00_global_loop.jl**. Functions for simulating the economy and iterating over the projection rule.

**00_simulation_functions.jl**. Set of functions for simulating single countries, a panel of countries, and the market value of net worth. It also has functions for computing moments using simulations, functions for the Lehman episode IRFs, an amplification exercise, elasticity (target), and a liquidity exercise.

**01_solve_model.jl**. Calls functions that solve the model. If *solve_once=true*, it solves the model given a set of parameters *eval_params=true*, a projection rule, and a starting point for the model. If *iterate_coeffs=true*, it also iterates over the projection rule. In either case, if *save_models=true*, it **overwrites** the solution to the model. This is the solution that will be used to compute moments and perform all exercises.

**02_baseline_moments.jl**. Uploads the solution of the model and computes sets of conditional and unconditional moments. If *do_calib_mms=true*, the code computes targeted and untargeted moments as well as the IRFs during the Lehman collapse episode. If *do_fall_omega=true*, it computes the IRF for the liquidity policy exercise (needed for Appendix Figure C4). If *do_amplif_mod=true*, it performs the amplification exercise under low and high exposure (needed for Appendix Figure C3).

**03_boom_bust.jl**. Code that computes the IRFs for the Lehman episode, starting from 2004.

**04_crossecAP_returns.jl**. Code to perform the exercise of cross-sectional asset pricing.

**create_tables.jl**. Code that creates the tables shown in the paper. It uploads data from the baseline model, and the model extensions (inside robustness folder).

## Model Extensions and Robustness

Codes for robustness and model extensions are inside the robustness folder. Each model extension has its own subfolder. In what follows, we briefly review each case.

**FM_diff_process**. Corresponds to the model extension *Measured Income Process* in the paper. Computes the solution under different assumed processes for the idiosyncratic and systemic components of output. The code *main_run.jl* calls functions defined in the baseline and runs the solution of the model. The code *01_solve_model.jl* actually solves the alternative model and provides the relevant simulated moments, saved in the folder *simulated_moments*, to be called for when constructing tables in the root folder.

**FM_low_elast**. Corresponds to the model extension *Alternative Elasticity*. Computes the solution under a lower target elasticity. Simulated moments are saved for tables created in root the folder.

**FM_kappa0_benchmark**. Corresponds to the model extension *Asset Managers*. Solves the model when $\kappa = 0$. Simulated moments are saved for tables created in the root folder.

**FM_high_lvg**. Corresponds to the model extension *High Leverge*. Solves the model for the case of high leverage. Simulated moments are saved for tables created in the root folder.

**SC_phi_smooth**. Corresponds to the model extension *Time-varying* $\phi$. Solves the model for a time-varying $\phi_t$. Simulated moments are saved for tables created in the root folder.

**SC_beta_norecalib**. Corresponds to the model extension *Liquidity Provision Policy*. Solves the model for a time-varying $\beta_t$. Simulated moments are saved for figures created in the root folder.

## Replication of Main Figures and Tables

Given a solution to the baseline model and extensions, the main tables and figures can be easily replicated.

To replicate the tables and figures, set *do_module* = 0 in the *main_run.jl* file. Tables and figures will be created given the saved results from the model and its extensions.

Run the latex file *do_tables_paper.tex* to compile figures and tables.

## Replication of Solution to Baseline Model

To compute the moments for the baseline model, given a solution to the model, set *do_module* = 2 in the *main_run.jl* file, with option *do_calib_mms=true*.

To compute the asset-pricing moments of the baseline model, set *do_module* = 4 in the *main_run.jl* file.

To compute a solution to the baseline model from scratch, set *do_module* = 1 in the *main_run.jl* file, with options *save_models*, *solve_once*, *iterate_coeffs* all equal to *true*.

## Replication of Liquidity Provision Policy

This section explains how to replicate Figure C4 of the Appendix. Follow these steps.

(1) In the main folder, file *main_run.jl*, choose *do_module=2* and *do_fall_omega=true*.

(2) Quit Julia and open a new session in the folder "robustness/SC_beta_norecalib." Run *main_run*.

These steps will create the data for Figure C4. As before, the file *create_tables.jl* will create the plot (you will need to run this in the main folder).

## Replication of Extensions to the Model

There are 5 other model extensions (i.e., aside from *SC_beta_norecalib*) in this replication package:

(1) Alternative Elasticity

(2) Measured Income Process

(3) Asset Managers
(4) High Leverge
(5) Time-varying $\phi$

Inside the folder *robustness* there is a folder for each model extension, as explained above. To replicate each model extension, you must enter each of these subfolders and run the Julia file *main_run.jl*. This file (one per model extension/subfolder) will solve the respective model extensions and simulate the model to provide the moments needed to construct the table presented in the paper. These moments are the ones uploaded by the code *create_tables.jl* from the root location to create the paper tables.

*Example procedure for replication*
In what follows, we provide an example of how to replicate results from the model extension *Alternative Elasticity*.

(1) Enter the subfolder *FM_low_elast* and run the Julia code *main_run.jl*.
(2) Exit Julia. Open it again in the main (root) folder.
(3) In the *main_run.jl* file from the root location, set $do\_module = 0$ and run the code.
(4) Compile Latex file *do_tables_paper.tex* to verify that moments of the model extension *Alternative Elasticity* are replicable.

Follow these steps to replicate all model extensions.