Data and Codes for "Endogenous Disasters"

January 2018

Nicolas Petrosky-Nadeau, Lu Zhang, and Lars-Alexander Kuehn

This folder contains all the data and programs used in Petrosky-Nadeau, Zhang, and Kuehn (2018, "Endogenous disasters") to appear in *American Economic Review*.

Note 1: The Matlab programs for solving the different formulations of the search model in our manuscript use the 2012 version of the CompEcon toolbox developed by Prof. Mario J. Miranda at Ohio State. The latest version of the toolbox can be downloaded at http://compeconworkshop.org/. We run our programs on Matlab R2014a.

The subfolder, Data, contains all the files to compute the data moments.

DES_BarroUrsua.m

> This m-file calculates the results in Tables 1 and 2 as well as Table A1 in the Online Appendix. This file uses the following .mat files: Y_BarroUrsua_update_July2014.mat, which contains the Barro-Ursua historical cross-country panel of output, extended through 2013, C_BarroUrsua_update_July2014.mat, which contains the Barro-Ursua historical cross-country panel of consumption, and AP_GFD_DMS_July2014.mat, which contains our historical cross-country panel of asset prices drawn from Global Financial Data and the DMS dataset, extended through 2013.

UVcurve.m

> This m-file plots Figure 1 and Figure A1 in the Online Appendix. This file uses the unemployment and vacancy rate series stored in US_UVdata_1929_2013.xls. It also uses subroutines nberdates.m, shade2_PK.m, and shadenber.m, as well as nberdates.mat to aid the plotting.

MomentsU.m

> This m-file calculates the unemployment rate moments in the data reported in Table 3. This file uses UV_pnz.mat, which contains the unemployment and vacancy rate series in the U.S. from April 1929 to December 2013.

Disaster_moments_Y.m

This m-file calculates the output disaster moments in the data reported in Table 4. This file uses Y_BarroUrsua_update_July2014.mat, which contains the Barro-Ursua historical cross-country panel of output, extended through 2013.

Disaster_moments_C.m

This m-file calculates the consumption disaster moments in the data reported in Table 4. This file uses C_BarroUrsua_update_July2014.mat, which contains the Barro-Ursua historical cross-country panel of consumption, extended through 2013.

The subfolder, The_Baseline_Model, contains all the files to compute the model moments for the baseline model.

projection.m

This m-file implements the projection algorithm for the baseline model. It calls a few subroutines, including residual_log_utility.m, which defines the error function to be minimized, and rouwTrans.m, which implements the Rouwenhorst (1995) discretization. The optimal solution is saved in output_proj_log_utility.mat (also saved in output_proj_log_utility_benchmark.mat).

momongrid.m

Based on the optimal solution, this m-file plots the residual function, as well as Panels A and B in Figure 4.

StatDistr_continuous_state.m

Based on the optimal solution, this m-file simulates the stationary distribution of unemployment, output, and consumption from the baseline model, and produces Panels A-C in Figure 2.

parsimu.m

Based on the optimal solution, this m-file calculates the model moments reported in Table 3 from the baseline model via 10,000 simulations. It calls a few subroutines, including myAutoCorr.m, hpfilter.m, DiscSimu.m, MultiRegressNw.m, and bartlettw.m, all of which are self-explanatory. Please note that due to sampling variations (despite 10,000 simulations), the output from each run of the code can deviate from the output from the next run. However, the differences are small, mostly in the second digit after the decimal in our experience.

parsimu_disaster.m

> Based on the optimal solution, this m-file calculates the disaster moments reported in Table 4 and Panels A-D in Figure 3 from the baseline model via 10,000 simulations. Please note that due to sampling variations (despite 10,000 simulations), the output from each run of the code can deviate from the output from the next run. However, the differences are small, mostly only in the second digit after the decimal in our experience.

Note 2: The order of running the programs is, first, projection.m, second, momongrid.m, and then followed by any of the other three files.

Note 3: To run the comparative statics for the baseline model (Table 5 and Panels C and D in Figure 4, with b = 0.4), please go to each subfolder with the filename starts with "CompStatic_" and copy the "projection_" m-file and "output_proj_log_utility_" mat-file into the parent folder named "The_Baseline_Model." Then please follow the same order of execution discussed in Note 2. The projection file changes since we sometimes adjust the grid setup to accommodate the model's new stationary distribution.

The subfolder, "Home_Production," contains all the files to compute the model moments for the home production model.

Projection_home_production.m

> This m-file implements the projection algorithm for the home production model. It calls a few subroutines, including residual_home_production.m, which defines the error function to be minimized, HomeProd_steadystate.m, which provides the steady state conditions (barely used), and rouwTrans.m, which implements the Rouwenhorst (1995) discretization. The optimal solution is saved in output_proj_home_production.mat (also saved in output_proj_home_production _benchmark.mat).

momongrid_home_production.m

> Based on the optimal solution, this m-file plots the residual function from the home production model, and calculates a few key variables on the grid, which are in turn saved in the "moments_theory_home_production.mat" file.

parsimu_home_production.m

Based on the optimal solution, this m-file calculates the model moments reported in Table 6 from the home production model via 10,000 simulations. It calls a few subroutines, including myAutoCorr.m, hpfilter.m, DiscSimu.m, MultiRegressNw.m, and bartlettw.m, all of which are self-explanatory. Please note that due to sampling variations (despite 10,000 simulations), the output from each run of the code can deviate from the output from the next run. However, the differences are small, mostly in the second digit after the decimal in our experience.

parsimu_disaster_home_production.m

Based on the optimal solution, this m-file calculates the disaster moments reported in Table 6 from the home production model via 10,000 simulations. Please note that due to sampling variations (despite 10,000 simulations), the output from each run of the code can deviate from the output from the next run. However, the differences are small, mostly only in the second digit after the decimal in our experience.

Note 4: The order of running the programs is, first, projection_home_production.m, second, momongrid_home_production.m, and then followed by parsimu_home_production.m or parsimu_disaster_home_production.m.

Note 5: To run the comparative statics for the home production model (Table 6), please go to each subfolder with the filename starts with “CompStatic_home_production_” and copy the “projection_home_production_” m-file and “output_proj_home_production_” mat-file into the parent folder named “Home_Production.” Then please follow the same order of execution discussed in Note 4. The projection file changes since we sometimes adjust the grid setup to accommodate the model’s new stationary distribution.

The subfolder, “Capital,” contains all the files to compute the model moments for the capital model.

Projection_capital.m

This m-file implements the projection algorithm for the capital model. It calls a few subroutines, including residual_capital.m, which defines the error function to be minimized, thetass_fn.m, which provides the steady state conditions (barely used), and rouwTrans.m, which implements the Rouwenhorst (1995) discretization. The optimal solution is saved in output_proj_capital.mat (also saved in output_proj_capital _benchmark.mat).

momongrid_capital.m

Based on the optimal solution, this m-file plots the residual function from the capital model, and calculates a few key variables on the grid, which are in turn saved in the “moments_theory_capital.mat” file.

parsimu_capital.m

Based on the optimal solution, this m-file calculates the model moments reported in Table 7 from the capital model via 10,000 simulations. It calls a few subroutines, including myAutoCorr.m, hpfilter.m, DiscSimu.m, MultiRegressNw.m, and bartlettw.m, all of which are self-explanatory. Please note that due to sampling variations (despite 10,000 simulations), the output from each run of the code can deviate from the output from the next run. However, the differences are small, mostly in the second digit (and sometimes in the first digit) after the decimal in our experience. However, the sampling variations do not affect our quantitative conclusions. This file takes about 40 minutes to run on a 16-CPU Dell workstation with 384 GB of RAM.

parsimu_disaster_capital.m

Based on the optimal solution, this m-file calculates the disaster moments reported in Table 7 from the capital model via 10,000 simulations. Please note that due to sampling variations (despite 10,000 simulations), the output from each run of the code can deviate from the output from the next run. However, the differences are small, mostly only in the second digit (and occasionally in the first digit) after the decimal in our experience. However, the sampling variations do not affect our quantitative conclusions.

Note 6: The order of running the programs is, first, projection_capital.m, second, momongrid_capital.m, and then followed by parsimu_capital.m or parsimu_disaster_capital.m.

Note 7: To run the comparative statics for the capital model (Table 7), please go to each subfolder with the filename starts with “CompStatic_capital_” and copy the “projection_capital_” m-file and “output_proj_capital_” mat-file into the parent folder named “Capital.” Then follow the same order of execution discussed in Note 6. The projection file changes since we sometimes adjust the grid setup to accommodate the model’s new stationary distribution.

The subfolder, “Recursive_Utility,” contains all the files to compute the model moments for the recursive utility model.

Projection_recursive_utility.m

This m-file implements the projection algorithm for the recursive utility model. It calls a few subroutines, such as residual_ recursive_utility.m, which defines the error function to be minimized, and rouwTrans.m, which implements the Rouwenhorst (1995) discretization. The optimal solution is saved in output_proj_recursive_utility.mat (also saved in output_proj_recursive_utility_benchmark.mat).

momongrid_ recursive_utility.m

Based on the optimal solution, this m-file plots the residual function from the recursive utility model, and calculates a few key variables on the grid, which are in turn saved in the “moments_theory_ recursive_utility.mat” file.

parsimu_ recursive_utility.m

Based on the optimal solution, this m-file calculates the model (other than disaster) moments reported in Table 8 from the recursive utility model via 10,000 simulations. It calls a few subroutines, including myAutoCorr.m, hpfilter.m, DiscSimu.m, MultiRegressNw.m, and bartlettw.m, all of which are self-explanatory. Please note that due to sampling variations (despite 10,000 simulations), the output from each run of the code can deviate from the output from the next run. However, the differences are small, mostly in the second digit after the decimal in our experience. However, the sampling variations do not affect our quantitative conclusions.

parsimu_disaster_ recursive_utility.m

Based on the optimal solution, this m-file calculates the disaster moments reported in Table 8 from the recursive utility model via 10,000 simulations. Please note that due to sampling variations (despite 10,000 simulations), the output from each run of the code can deviate from the output from the next run. However, the differences are small, mostly only in the second digit after the decimal in our experience. However, the sampling variations do not affect our quantitative conclusions.

Note 8: The order of running the programs is, first, projection_ recursive_utility.m, second, momongrid_ recursive_utility.m, and then followed by parsimu_ recursive_utility.m or parsimu_disaster_ recursive_utility.m.

Note 9: To run the comparative statics for the capital model (Table 7), please go to each subfolder with the filename starts with “CompStatic_RU_” and copy the “projection_recursive_utility_” m-file and “output_proj_recursive_utility_” mat-file into the parent folder named “Recursive_Utility.” Then follow the same order of execution discussed in Note 8. The projection file changes since we sometimes adjust the grid setup to accommodate the model's new stationary distribution.

Note 10: The exception to Note 9 is for the recursive utility model with gamma = psi = 1 (the log utility). For this comparative static experiment, go to the subfolder "CompStatic_RU_log_utility." The order of running the programs is, first, projection_RU_log_utility.m, second, momongrid_RU_log_utility.m, and then followed by parsimu_RU_log_utility.m or parsimu_disaster_RU_log_utility.m. The reason for this exception is that there is no indirect utility function (denoted J) to be solved with log utility. As such, the structure is different from the other comparative statics for the recursive utility model. In addition, this comparative static is a bit different from the baseline model, which also has log utility. The reason is that some parameter values, such as the time discount factor, are different in this comparative static for the recursive utility model. As such, we opt to put the codes together in a separate folder for clarity.

Note 11: Please address all the inquiries about the data/codes to zhanglu@fisher.osu.edu