Data and Replication Material for
"Optimal Monetary Policy in Production Networks"
Jennifer La'O and Alireza Tahbaz-Salehi

This folder contains the replication code and data for "Optimal Monetary Policy in Production Networks," by La'O and Tahbaz-Salehi (*Econometrica*, 2022).

Description of Datasets

The `Raw_Data` folder contains all data sources necessary for replicating the results. The source data are organized in three subfolders.

1. `Raw_Data/BEA_IO_Data`: This subfolder contains the 2019 summary level Make and Use Tables from the Bureau of Economic Analysis.

2. `Raw_Data/Price_Frequency_Data`: This subfolder contains industry-level data on the frequency of price adjustments, constructed by Pastén, Schoenle, and Weber (2020), "The propagation of monetary policy shocks in a heterogeneous production economy." *Journal of Monetary Economics*, 116, 1–22. The original data provided to us by Pastén et. al (2020) is based on 3-digit NAICS codes. Using the concordance between NAICS codes and IOCodes, we constructed the dataset for frequency of price adjustments based on the BEA tables IOCodes. This dataset is saved as `FrequencyIOCode.xlsx`. The original data can be found in `FrequencyNAICS-3d.csv`.

3. `Raw_Data/Price_Frequency_Data`: This subfolder contains the March 2021 release of the BEA/BLS Integrated Industry-level Production Account (ILPA) for the United States.

Cleaning and Merging the Data

The file `Clean_and_Merge_Data.do` reads the raw data from `Raw_Data` folder and merges the BEA input-output data with the price-adjustment data, while excluding industries corresponding to federal, state, and local governments. This results in a matched dataset consisting of 66 industries. The file then merges the resulting dataset with the ILPA data. The output is written in the (initially empty) `Merged_Data` folder.

Replication Code

Running the following files reproduces the results in the paper. These files have to be run after running `Clean_and_Merge_Data.do`, which cleans and merges the various datasets.

– `MainCode.m`: This file reproduces the results in Section 5 of the paper, including Table 1 and Figure 3. It also generates the numbers reported in the text of Section 5. The file calls various auxiliary functions in `functions` folder.

– ExtendedCode.m: This file extends MainCode.m and reproduces the results in Section 5 of the paper as well as the results in Supplementary Appendix C. Specifically, it generates Table 1, Figure 3, and Table C.1, as well as the numbers reported in the text of Section 5. The file calls various auxiliary functions in functions folder.

Auxiliary Functions

The codes that generate the results, MainCode.m and ExtendedCode.m, call the following auxiliary functions. These auxiliary functions are saved in functions folder.

– optimalpolicy.m: This function calculates the sectoral weights under optimal policy (based on Theorem 2 of the paper). The inputs are preference parameters and the intra-industry elasticity of substitution. The output is the optimal policy weights characterized in equation (39) of the paper.

– expected_welfareloss_nonlinear.m: This function calculates the expected welfare loss due to the presence of nominal rigidities for the fully nonlinear model. The inputs are the vector of monetary policy weights assigned to different industries, the preference parameters, and the intra-industry elasticity of substitution. The output is the expected welfare loss as a fraction of steady state consumption. This function calls expost_welfareloss_nonlinear.m.

– expost_welfareloss_nonlinear.m: This function calculates welfare loss due to the presence of nominal rigidities for the fully nonlinear model, given a monetary policy and the realization of productivity shocks. The inputs are the realized productivity shocks and the vector of policy weights. The output is the realized welfare loss as a fraction of the economy's steady state consumption. This function uses SolveEqNominalPricer.m.

– SolveEqNominalPricer.m: This function solves for the equilibrium nominal prices in the fully nonlinear model as a function of policy weights and the realized productivity shocks.

– welfarelossQuadratic.m: This function calculates the expected welfare loss of a policy to a second-order approximation using Proposition 6 in the paper.

– cossim.m: This function calculates the cosine similarity between two vectors.

– sector_aggregate.m: This function aggregates policy weights assigned to different industries to the sectoral level.