Readme file for "Optimal Taxation with Private Insurance" (Manuscript: 26154)

Yongsung Chang and Yena Park

No new data were generated or analysed in support of this research. All the codes related to this research are available on Zenodo at https://doi.org/10.5281/zenodo.4444803.

Codes_MS26154.zip contains two folders — one for the Matlab Codes and the other for the Fortran Codes. In this document, we describe each folder first and summarize the mapping between the codes and figures/tables in the end.

1. Matlab Codes Simulation

Matlab_Excel_MS26154 folder contains the Excel files and Matlab codes that are used to compute the equilibrium under the current US tax system and status quo tax reform. It also includes all the outputs that are generated by the codes. The folder consists of three sub-folders — Equilibrium, StatusQuoReform, and Figures. **Equilibrium** folder contains an Excel file and codes that are used to compute equilibrium under the current tax system. **StatusQuoReform** folder contains codes that compute welfare gain of the status quo tax reform. **Figures** folder contains codes that generate all the figures (except for Figure 1 which is generated by `setup.m` below) in the manuscript.
The products need to run all of the Matlab files are: optimization toolbox, statistics toolbox.

(1) Equilibrium

- `Parameters for Computation.xlsx` summarizes all the calibrated parameters which are used when computing the equilibrium under the current US tax and status quo tax reform (Matlab) and when simulating optimal tax schedule (Fortran). There are 9 model specifications used for the optimal tax simulation and 2 model specifications used for the status quo tax reform.
- `setup.m` generates Figure 1 (hazard rates of wage) and saves the information required to compute a Huggett equilibrium. First, a grid for wage (productivity) is generated. Next, it approximates AR1 process with discrete transition matrix using the Tauchen method.

Then it calibrates the parameters—used for the modification of the transition matrix—to match the hazard rates in the data. The outputs are saved in `environ_1.mat`.

`Benchmark_HSV.m` computes a Huggett equilibrium under the US tax schedule and generates some model statistics in Table 3. Before running this code, one needs to run `setup.m` first to generate `environ_1.mat` in the same folder. The US tax schedule is approximated by the HSV tax function ($T(z) = z - \lambda z^{1-\tau}$: use the estimated τ in Heathcote, Storesletten, and Violante (2017) and calculate λ to generate the government expenditure-GDP ratio $\frac{E}{Y_{US}} = 18.8\%$). The discount factor β and the borrowing limit (a_min = a_minfactor*GDP) are calibrated to generate an equilibrium interest rate $r = 0.04$ and the fraction of credit-constrained households (of 10%) in the steady state. In the optimal tax simulation, we use the constant government expenditure E calibrated under the current US tax schedule. The results are saved in `USresult.mat` to produce the graphs related to the U.S. economy by `Figures.m`.

Sensitivity Analysis with 10 grid points: To compute a Huggett equlibrium with various parameters calibrated for the sensitivity analysis, change the parameters in `setup.m` and `Benchmark_HSV.m` as in `Parameters for Computation.xlsx`.

Sensitivity Analysis with finer grid points: Run `setup_N20.m` or `setup_N30.m` for finer grid points ($N = 20$ or $N = 30$ respectively), and the results are saved in `environ_1.mat`. Then, run `Benchmark_HSV.m` after changing the parameters as in `Parameters for Computation.xlsx`.

- `setup_Aiyagari.m` calibrates parameters (used for the modification of a transition matrix) required to compute an Aiyagari equilibrium with production. The outputs are saved in `environ_1.mat`.

  `Benchmark_AiyagariHSV.m` computes an Aiyagari equilibrium under the US tax schedule approximated by the HSV tax function. Run `setup_Aiyagari.m` before running this code.

- `Benchmark_HSV_transitory.m` calculates average MPC in a Huggett economy with transitory η-shock which is reported in Table 3. We compute the average MPCs with $\sigma_\eta = 1$ and $\sigma_\eta = 2$ and report the mean of them. Run `setup.m` first to generate `environ_1.mat` before running `Benchmark_HSV_transitory.m`.

(2) **StatusQuoReform**

- `Figure_UStaxreform.m` generates Figures A.10, A.11, and A.12 — welfare gain from the status quo tax reform and decomposition of the gain. To run this code, `USreform.mat` and `USreform_Aiyagari.mat` should be in the same folder, which are generated as follows.
- `TaxReform_Huggett_currentUS_new.m` computes the welfare gain from the status quo tax reform in a Huggett economy and generates `USreform.mat`.
  `TaxReform_Aiyagari_currentUS_capitaltax.m` computes welfare gain from the status quo tax reform in an Aiyagari economy and generates `USreform_Aiyagari.mat`

(3) **Figures**

- `Figures.m` (i) reads the results files from the subdirectory `Results` and (ii) generates Figures 2-4 and Figures A1-A14 in the paper.
  - It reads 4 Matlab output files (`environ_1.mat`, `USresult.mat`, `USreform.mat`, and `USreform_Aiyagari.mat`) generated from the Matlab-based simulations above.
  - It reads various files (generated from the Fortran-based simulations described below) from the sub-directories. The name of each sub-directory reflects the model specification (e.g., "Benchmark,'' "Lognormal," "Rho08," etc.)
  - Figure 4 uses the values of (i) Federal Income Tax Schedule (as of 2015) and (ii) the Congress Budget Office's the (median) effective marginal tax rates are from "Effective Marginal Tax Rates Low- and Moderate-Income Workers" (Congressional Budge Office) November 2012, Figure 5. These values are already in `Figures.m`.
- The model statistics in Table 3 are generated by `Benchmark_HSV.m` (from the Matlab-based simulation described above). The execution of the file produces the output file `Table3.txt`.

All files for data and simulation codes are available at: `https://doi.org/10.5281/zenodo.4444803`.

2. Fortran Codes Simulation

Fortran_MS26154 directory contains the Fortran codes computing the optimal tax schedule. Each sub-folder in Fortran_MS26154 directory contains all the Fortran files needed for the simulation under each specification. The codes were compiled with the Intel Fortran 19.1 Update 2 compiler (environment initialized for: x64) and we do not use other libraries. The codes are designed to run with parallelization using OpenMP (it can also be run serially). The Fortran codes are stand-alone codes with no interdependency in compilation or execution. Each program in each sub-folder needs to be compiled together with the all the subroutines files in each sub-folder.

It consists of four sub-folders— **Optimal Tax Huggett Transition**, **Optimal Tax Huggett Transition 20**, **Optimal Tax Huggett Transition 30**, and **Optimal Tax Aiyagari**. Each sub-folder contains the Fortran project with its source codes. **Optimal Tax Huggett Transition** folder contains the codes that simulate the optimal tax schedule in a Huggett economy under the benchmark and other model specifications (for sensitivity analyses), while the other folders contain the codes for finer grid points ($N = 20$ or $N = 30$) and for the simulation of an Aiyagari economy. Run the codes in the following way and save the output files in the directory described below to generate figures using `Figures.m` described above.

1. Set up the parameters in the global module (`global.f90`) as in `Parameters for Computation.xlsx`.
   - *sigma, gamma, beta, GovE, rho, sigma-ep-adj, delta, mo, epsilon*

2. Run the Fortran codes.
   The computation time depends on the number of threads (and also on the hardware specification) since the codes use the OpenMP parallel.[1]

3. Output files are saved in the *output* folder in each project's directory.

4. Copy and paste the output files in the subfolders with correct specification (benchmark, CRRA1, $\cdots$, etc) located in the directory **Matlab_Excel_MS26154 \Figures \Results**.

Brief Algorithm of the Fortran Project

1. Guess a marginal tax schedule $\{T_i'\}_{i=1,\cdots,N}$ (on each grid point) in the main program file `mainparallel.f90`.

2. Given that initial tax schedule, calculate the lump-sum transfer that satisfies the government budget constraint using the `solvetaxmin` function in `mod_functions.f90`.

3. Given the tax schedule, we solve a steady-state equilibrium by calling subroutine `Aiyagari.f90`.

4. Consider a tax reform of increasing marginal tax rate at each grid point, and compute the new steady state and transition under the tax reform at each grid point (by calling subroutine `Aiyagari_tran.f90`.

[1]Using the server PC with 28 parallel threads working, the program with 20 grid points takes roughly 8 hours.

5. Then, use the tax formula (A.5) to compute the right-hand side (RHS) and thus an alternative marginal tax schedule.

6. Check if $||\frac{T'}{1-T'} - RHS|| < tol$. If not, go to step 3 with an alternative tax schedule. This loop is repeated until a fixed-point optimal tax schedule is found.

The program `mainparallel.f90` calls for the subroutine `TaxNoPrivate.f90` computing an optimal tax schedule without private insurance. It also calls the subroutine `SCPE.f90` computing a Self-Confirming Policy Equilibrium tax schedule.

All files for data and simulation codes are available at: https://doi.org/10.5281/zenodo.4444803.

3. Mapping between the codes and figures/tables

- Figure 1: `setup.m` generates this figure. The specific location is marked as a comment in the file.
- Figures 2-4: `Figures.m` generates **Figures 2-4** in the paper. The specific locations are marked in the file.
- Figures A1-A14: `Figures.m` generates **Figures A1-A14** in the Online Appendix. The specific locations are commented in the file.
- `Figure.m` reads some files (generated from model simulations) from the subdirectory "`Results`" and generate figures.
- Tables 1 and 2 are texts (not numeric numbers).
- Table 3: The model statistics are generated by `Benchmark_HSV.m` which produces the output file `Table3.txt`. The only exception is the average MPC which is computed using `Benchmark_HSV_transitory.m` as explained above. The data statistics are from the two papers (cited in the footnote of the article).
- Here are the names of figure files
  - Figure 1: hazard.eps
  - Figure 2: MTR_v_n.eps
  - Figure 3: decomposition1.eps

- Figure 4: MTRz_USvsOPT.eps
- Figure A1: ATRz_USvsOPT3.eps
- Figure A2: decomp1_SCPE.eps
- Figure A3: MTRn_lognormal.eps
- Figure A4: MTRn_rho08.eps
- Figure A5: MTRn_sigma1VsBench.eps
- Figure A6: Decomposition1_sigma1.eps
- Figure A7: MTRn_Frisch025VsBench.eps
- Figure A8: MTRn_tighten.eps
- Figure A9: MTRn_20.eps and MTRn_30.eps
- Figure A10: USreform_CEV.eps
- Figure A11: USreform_Huggett_decomp.eps
- Figure A12: USreform_Aiyagari_decomp.eps
- Figure A13: MTRn_HuggettvsAiyagari2.eps
- Figure A14: Decomposition1_A.eps

All files for data and simulation codes are available at: https://doi.org/10.5281/zenodo.4444803.