# Notes on Replicating "Taxing Top Incomes and Tax Avoidance" by Alessandro Di Nola, Georgi Kocharkov, Almuth Scholl, Anna-Mariia Tkhir and Haomin Wang

January 22, 2025

**Abstract**

These notes serve as a guide to the computer programs used in the quantitative analysis of the paper. A brief summary of the structure of the computer code, the data, and the execution procedures are presented.

## 1 Structure

The folder has these directories:

- Code: Code and relevant folders for the main model.
  - exe: executable files.
  - input: model parameter values.
  - matlab: Matlab code to process Fortran-generated output. Main scripts are: `matlab_plotter`, `compstat_plotter`, `comptran_plotter` and `NoAvoidanceExp_plotter`.
  - output: stores Fortran-generated output.
  - src: Fortran source files.
- Robustness: Code and relevant folders for the robustness check (Fixing Total Pension Benefits). The file-structure is the same as Code.
- Data:

  - IRS data for Section 2
  - moments

# 2 Software Requirements

The code was run on Matlab R2024b and Stata 18 on Windows 11 on a computer with 12th gen Intel-core-i9-12900K CPU 3.20 GHz and 64 GB RAM. The runtime for the Fortran code for computing the steady-state is about 30 seconds, for the transition about 3 minutes. The runtime for the Stata code is just a few seconds. The code is compiled using the Intel `ifx` compiler, Version 2024.0.2. The Fortran code makes use of OpenMP and the MKL libraries provided by Intel.

# 3 Data Sources

Data for the empirical analysis and computation of data moments come from public sources:

- SCF: Raw data file and Summary Extract Public Data are downloaded from `https://www.federalreserve.gov/econres/scf_2013.htm`.
- IRS tax return tables: `https://www.irs.gov/statistics`.
- SUSB tables: `https://www.census.gov/programs-surveys/susb/data/tables.html`
- BDS tables: `https://data.census.gov/table/BDSTIMESERIES.BDSESIZE`
- FRED data: `https://fred.stlouisfed.org/`
- OECD data: `https://data-explorer.oecd.org/`
- TAXSIM: `https://taxsim.nber.org/stata`.

# 4 Model Solution and Experiments

The model is solved in Fortran. The following part contains a *step-by-step* description on how to configure and run the code in order to replicate the benchmark quantitative model.

## 4.1 Benchmark model

To solve the benchmark model in general equilibrium, follow these steps:

1. Set your current directory to `Code`.

2. Set relevant options in `src\main.f90`. In particular, to replicate the benchmark model, set

```
cf_avoidance = 0
cf_occpol = 0
do_run = 0
do_GE = 1
do_benchmark = .true.
do_CEV_decomp  = 0
 file_params = "estim_params_benchmark.txt"
 dir_bench   = trim(savedir)//trim("ss_bench")//trim(slash)
```

3. Compile the Fortran source files using the file `makefile_win`. In your command window, type

```
nmake /f makefile_win
```

   This will generate the executable file `run.exe` in subfolder `exe`.

4. To run the code, type the following

```
exe/run.exe
```

   After the model is solved, model output (in the form of txt files) is saved in subfolder `output\ss_bench`. In subfolder `output` you will see `targets_model_manual.txt` which saves all model parameters and moments.

## 4.2 Re-calibrated equal-tax-treatment model

To solve the re-calibrated equal-tax-treatment model in general equilibrium, set relevant options in `src\main.f90` as follows

```
cf_avoidance = 2
cf_occpol = 0
do_run = 0
do_GE = 1
do_benchmark = .true.
do_CEV_decomp  = 0
file_params = "estim_params_CF2.txt"
dir_bench   = trim(savedir)//trim("CF2_bench")//trim(slash)
```

The steps are the same as those in section 4.1.

## 4.3 Tax reform experiments

1. Follow the above steps to solve the model benchmark
2. Change flags in `src\main.f90` as follows. For the equal tax treatment experiment without fiscal neutrality

   ```
   cf_avoidance = 2
   do_GE = 1
   do_benchmark = .false.
   ```

   For the sole-proprietor only experiment without fiscal neutrality,

   ```
   cf_avoidance = 3
   do_GE = 1
   do_benchmark = .false.
   ```

   The remaining flags are set the same way as in the benchmark model. With fiscal neutrality, set `do_GE = 2`.

## 4.4 Revenue maximization (comparative statics)

To run revenue maximization in the benchmark model, set flags in `src\main.f90` as follows

```
cf_avoidance = 0
cf_occpol = 0
do_run = 2
do_GE = 1
n_tau_h =52
```

```
do_benchmark = .false.
do_CEV_decomp  = 0
file_params = "estim_params_benchmark.txt"
dir_bench   = trim(savedir)//trim("ss_bench")//trim(slash)
```

To run revenue maximization in the equal tax treatment economy, set flags in `src\main.f90` as follows

```
cf_avoidance = 2
cf_occpol = 0
do_run = 2
do_GE = 1
n_tau_h =52
do_benchmark = .false.
do_CEV_decomp  = 0
file_params = "estim_params_CF2.txt"
dir_bench   = trim(savedir)//trim("CF2_bench")//trim(slash)
```

## 4.5 Welfare maximization with transition dynamics

To run welfare maximization with transition in benchmark model, set flags in `src\main.f90` as follows

```
cf_avoidance = 0
cf_occpol = 0
do_run = 3
do_GE = 2
do_benchmark = .true.
do_CEV_decomp  = 1
file_params = "estim_params_benchmark.txt"
dir_bench   = trim(savedir)//trim("ss_bench")//trim(slash)
```

To run welfare maximization with transition in the equal tax treatment economy, set flags in `src\main.f90` as follows

```
cf_avoidance = 2
cf_occpol = 0
do_run = 3
do_GE = 2
do_benchmark = .true.
do_CEV_decomp  = 1
n_tau_h =52
file_params = "estim_params_CF2.txt"
dir_bench   = trim(savedir)//trim("CF2_bench")//trim(slash)
```

### 4.6 Tax Reform Act of 1986 experiment

Change flags in `src\main.f90` as follows.

```
cf_avoidance = 0
cf_occpol = 0
do_run = 1
do_GE = 2
do_benchmark = .false.
n_tau_h =3
```

The remaining flags are set the same way as in the benchmark model (section 4.1).

## 5 Replicating Figures and Tables

### 5.1 Figures

- Figure 1
  - Figure 1a is busreceiptsLFO, 1b is shareLFO, 1c is wageLFO, 1d is laborshare
  - Run Section2.m in folder Data\IRS data for Section 2\Datawork
  - Data are stored in Data\IRS data for Section 2\Data

The next figures are plotted using code in folder Code.

- Figure 2
  - contains 4 subfigures. 2a is share_entre_quint_inc, 2b is share_entre_quint, 2c is share_lfo_top_quint_inc, 2d is share_lfo_top_quint
  - Run model_fit_plotter.m
  - Model moments are stored in subfolder output\ss_bench, data moments are subfolder input.
- Figures 3-4-5
  - Figure 3 is occpol_prob, Figure 4a is kpol, 4b is npol, Figure 5a is phipol_es and 5b is phipol_ec
  - Run matlab_plotter.m
  - Model data are stored in subfolder output\ss_bench

- Figure 6
  - Figure 6 is cev_z_exp_taxreform
  - Run NoAvoidanceExp_plotter.m
  - Model data are stored in subfolders output\exp2_fn and output\exp5_fn.
- Figure 7
  - Figure 7a is compstat_taxes_inc_corp_div, 7b is compstat_Y, 7c is compstat_inc_share_top1, 7d is compstat_wealth_share_top1
  - Run compstat_plotter.m
  - Model data are stored in subfolders output\compstat and output\compstat_CF2.
- Figure 8
  - Figure 8a is comptran_cev_vec, 8b is comptran_cev_aggcomp_vec, 8c is comptran_cev_distcomp_vec, 8d is comptran_cev_z, 8e is comptran_base_cev_qo, 8f is comptran_CF2_cev_qo
  - Run comptran_plotter.m
  - Model data are stored in subfolders output\comptran_fn and output\comptran_CF2_fn.
- Figure 9 (also called Figure A1)
  - Figure 9a is exp2_tran_r_w, 9b is exp2_tran_Y_K, 9c is exp2_tran_avek_Y_entre, 9d is exp2_tran_taxes, 9e is exp2_tran_share_entre
  - Run NoAvoidanceExp_plotter.m
  - Model data are stored in subfolder output\exp2_fn.

The remaining figures are based on a robustness check "Fixing Total Pension Benefits". Use folder `Robustness` instead of `Code`.

- Figure 10 (also called Figure A2)
  - Figure 10 is cev_z_exp_taxreform
  - Run NoAvoidanceExp_plotter.m
  - Model data are stored in subfolders output\exp2_fn and output\exp5_fn.
- Figure 11 (also called Figure A3)

- Figure 11a is comptran_cev_vec, 11b is comptran_cev_aggcomp_vec, 11c is comptran_cev_distcomp_vec, 11d is comptran_cev_z, 11e is comptran_base_cev_qo, 11f is comptran_CF2_cev_qo
- Run comptran_plotter.m
- Model data are stored in subfolders output\comptran_fn and output\comptran_CF2_fn.

## 5.2 Tables

- Table 1. Manual input from various sources. Authors' calculations explained below:
  - Dividend tax rate: computed using Stata code `Data\moments\do files\main.do`.
  - Social security cap: average labor income taken from `Data\moments\QCEW\allhlcn13.xlsx` cell R2, downloaded from `https://www.bls.gov/cew/downloadable-data-files.htm`. Social security cap is taken from `https://www.ssa.gov/oact/cola/cbb.html`. We take the ratio between the social security cap and average labor income.
- Table 2. Manual input based on `Code\output\targets_model_manual.txt`, an output file from running code to solve the benchmark model (see Section 4.1 above).
- Table 3 and 4. "Model" column: Manual input based on `Code\output\targets_model_manual.txt`, an output file from running code to solve the benchmark model (see Section 4.1 above). "Data" column: Manual input from various sources. Authors' calculations explained below:
  - Moments from data source "SCF" and "SUSB" are computed using Stata code `Data\moments\do files\main.do`. Computation is based on dataset `Data\moments\work\scf2013_clean.dta`.
  - Share of entrepreneurial income declared as wage: calculations done in `Data\moments\IRS\Calculations Wage Share.xlsx`, based on tax return table of S-corporations `13co04s.xlsx` and C-corporations `13co05ccr.xlsx` in the same directory.
- Table 5. Manual input based on model results from running tax reform experiments (see Section 4.3 above).

- Table 6. Manual input based on model results from running the Tax Reform Act of 1986 experiment (see Section 4.6 above).

- Table 7. Manual input based on model results from running the revenue maximization code in the benchmark model and in the equal-tax-treatment model (see Section 4.4 above).

- Tables 8 and 9 in Appendix A.4. Manual input based on `\output\targets_model_manual.txt`, an output file from running code to solve the benchmark and the re-calibrated models (see Sections 4.1 4.2 above).

- Table 10 in Appendix A.5 Manual input based on model results from running tax reform experiments in the robustness check "Fixing Total Pension Benefits" economy (see Section 4.3 above).