

README numerical_exercise (MATLAB)

NOTE on FOLDER STRUCTURE: The working directory should include all the files listed below, and folders “Figures” and “Tables” to store the results. These destination folders can be edited from the preamble of the relevant files.

“Tables” contains the output tables and the file “Compare tables.tex” which can be run to compile the output tables

“Figures” contains the output from the “3DGraph” files described below.

main3DGraph, main3DGraphTaxes: produce the level curve plots in figure 3. Panel A and B

CALLS ON:

- target_deviations_K_share: solver to set the adequate parameters to meet the specified targets
- maket_eq_p: computes the equilibrium for given parameters and pairs of taxes for capital and labor
- setParams: function creating a structure with all the parameters. Default can be overridden by inputting a main_par structure with field names corresponding to the parameters to override.
- ramsey_unconstrained_p: solves the unconstrained Ramsey problem

Since these graphs are computationally expensive to produce, we include the .mat files collecting the data that results from the computations. Using these files, the user can reproduce our graphs simply by running lines 203 onwards of main3DGraph and lines 194 onwards of main3DGraphTaxes.

mainAllTables_KSR: produces Tables 1-2 in the main text and Tables A.1-2, A.4-A.8 in the appendix

CALLS ON:

- target_deviations_K_share
- mainTableReforms_KSR: Reform Table 2
- makeTable_KSR: all other tables

makeTable_KSR: creates a table like Table 1 in the text based on input parameters. First, it set parameters so as to target the moments of interest in the competitive equilibrium. It then sets the government expenditure to the revenues collected in the market equilibrium and solves the other planning problems subject to this revenue—and other relevant—constraints.

CALLS ON:

- market_equilibrium_p: market equilibrium computations

- `ramsey_unconstrained_p`: solves the planner problem with the only constraint that a specific revenue must be met
- `ramsey_constrained_p`: solves the planner problem subject to the revenue constraint and a lower bound on labor taxation
- `ramsey_theta_p`: solves for optimal automation θ , when the planner has a revenue constraint and needs to keep taxes fixed at observed 2010 levels
- `ramsey_constrained_capital_p`: solves the planner problem subject to the revenue constraint and an upper bound on capital taxation
- `computeWelfareCEq`: computes consumption-equivalent welfare variations moving from a specified allocation to another
- `computeDisplacedWorkers`: computes the fraction of workers displaced in marginal tasks when moving from a specified allocation to another
- `table2latex_t`: organizes a matlab T into a latex table, given a structure of table options.

Each of the equilibrium sub-routines requires:

- `output_raw_p`: computes equilibrium output for given quantities of capital and labor

`mainTable_K_comp`: produces table A.3 accounting for within-task labor-capital complementarity.

CALLS ON:

- `target_deviations_K_comp`: compute deviation of calibration targets and equilibrium quantities in the competitive equilibrium of the economy
- `market_eq_K_comp`: compute market equilibrium with complementary capital
- `makeTable_K_comp`: produce the table

`makeTable_K_comp`: produces the complementary capital table.

CALLS ON:

- Same functions as in `makeTable_KSR`, with added suffix “_K_comp”

`makeTableReforms_KSR`: Produces the reform Table 2 in the main text.

CALLS ON:

- `compute_rev_tauA`: computes the government revenue for a given set of taxes plus an automation tax τ_A
- `market_eq_p_tauA`: computes the market equilibrium for a given set of taxes and an automation tax of τ_A

`mainTable`: creates locus of points that is elaborated on to produce Figure 2, Panels A and B (to clean)

`mainTimeSeriesFigs`: plots the time series in Figure 1, and Figures A1-A3. Note: the directories `save_fig` and `source_data_path` should be changed to the desired directory to save figures and the R sub-folder containing the data for the figures produced by

the R routines (see below). The data are produced as described below

R README

NOTE on FOLDER STRUCTURE: The working directory should include all the files listed below, and folders “bin”, “out”, “raw” with the files in our replication files. Destination folders can be edited from the preamble of the relevant files. In order to indicate the directory where the R folder is saved, please use the initial lines in the file `setupLines.R`

The files produce the time series for taxes that we use to compute average tax rates, as well as intermediate inputs like interest rates and depreciation series like those plotted in appendix Figures A1. Figures are produced in Matlab based on the output of the sequence of R codes reported below.

In order to reproduce the results, files should be run in the following order:

1. `setupLines`: setting the base folders, and setting up other defaults to be used in the analysis. The required packages are reported in the initial lines. The user can install these packages from the CRAN repository by running `install.packages('packageName')`. The user should edit the object “baseDir” to the folder where our replication kit is saved. Which should have subfolders:
 1. R (with further subfolders “bin”, “raw”, “out”)
 2. `numerical_exercise` (with matlab files)
2. `runTaxSeriesBEA`: computes the average tax Series from BEA data as described in the text
3. `runDividendTaxes`: computes taxes on dividends on capital gains as described in the text
4. `runInvestmentPriceSeries`
5. `computeRatesSteadyState`: generates the depreciation allowance series and the average capital tax rates assuming that interest rates, investment price growth and depreciation are fixed at their historical average. Saves the Data needed to reproduce figure A.1, A.2). **Calls on:** `functionComputeWedges`
6. `runFigures`: exports the data to produce Figures 1 and A.3 with the exception of `laborTaxes.csv`, saved by `runTaxSeriesBEA`)

The structure and contents of the raw and out folders should be preserved in order to correctly run the file `mainTimeSeriesFigs.m` described in the MATLAB README above.

The directory “raw” contains all the input files needed for the analysis. The names are self-explanatory and additional descriptions on the file contents can be found by reading the “import” section of the .R file of interest