Notes on Replicating **"The Aggregate Consequences of Tax Evasion"** by Alessandro Di Nola, Georgi Kocharkov, Almuth Scholl and Anna-Mariia Tkhir

September 21, 2020

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 `DKST` has three directories:

- Fortran
- Matlab
- Stata

which contain the programs used in the analysis. Folder Fortran contains Fortran source files used in the generation of an executable file.[1] This executable file is used in the solutions of the model, as will be explained below. Folder Matlab contains the basic code to solve the model. The Matlab codes in turn call Fortran to perform the more time intensive computations. Folder Stata contains the do-files that treat the data.

2 Software Requirements

The code was run on Matlab R2020a and Stata 13 on Windows 10 on a computer with Intel-core-i7-9700 CPU 3.00 GHz and 16 GB RAM. The runtime for the Matlab code was of about 4-5 minutes for the partial equilibrium version of

[1] We also include a makefile to make the compilation easier. Alternatively, the user can compile the source files from the command line.

the model. The runtime for the Matlab code was of about 22 minutes for the general equilibrium version of the model.
The runtime for the Stata code is of less than 20 seconds.
The Fortran codes make use of OpenMP. No other external libraries are used (the few general-purpose routines that we use are provided in the module mod_baselib.f90). The Matlab codes use the Matlab toolboxes.

3 Data and Empirical Analysis

In order to obtain data moments run the file `Targets.do` in the Stata subfolder. At the top of this file customize the directory `global DataDir` to the path where you save your data. The file calls the data file `PSID_Amerged_new.dta`. You can choose data cleaning criteria and the definition for the entrepreneurs at the beginning of the file. All flags are commented in detail. Setting a flag `DefEntre=2` generates data targets in Table 3 and the first column of Table 12. Setting a flag `DefEntre=1` generates the second column in Table 12. A subfile `exitrate.do` which is called from the main file `Targets.do` calculates the exit rates for workers and self-employed presented in Table 13.

4 Model Solution and Experiments

The model is solved in Matlab. The Matlab code, in turn, calls the executable that is generated from the Fortran source files. First, compile the Fortran code, using the makefile `makefile_win` (written for Windows machines). Call the executable `fortran_OS`. For example, if running on a Windows machine, the executable should be called `fortran_win64.exe`. The other options would be: `fortran_mac` (for Mac) or `fortran_gfortran64` (for Linux 64-bit). This executable should be placed in the subfolder `matlab\exe`.

To solve the model, execute `main.m` in Matlab. This automatically generates the benchmark results. All of the other results can be obtained by changing the options in `main.m`. All matlab files are extensively commented and we refer the user to the notes therein. As an example, to replicate the benchmark model with tax evasion, set `no_evasion=0` and `do_GE=1` (if general equilibrium is desired) in the main file. In order to compute the counterfactual economy with perfect tax enforcement, set `no_evasion=1` and `do_GE=1` or `do_GE=0` (if prices are not allowed to adjust).

The results of the benchmark economy and of the several counterfactual experiments performed in the paper are saved in the subfolder `results\mat`. The mat files corresponding to the different cases are:

- `taxevasion_ge.mat`: benchmark economy with tax evasion,
- `notaxevasion_ge.mat`: counterfactual economy with perfect tax enforcement in general equilibrium,

- `notaxevasion_pe.mat`: counterfactual economy with perfect tax enforcement in partial equilibrium.
- `notaxevasion_ge_lump_sum`: counterfactual economy with perfect tax enforcement and fiscal neutrality with lump-sum redistribution.
- `notaxevasion_ge_cut_tax_all`: counterfactual economy with perfect tax enforcement and fiscal neutrality with tax cut for all.
- `notaxevasion_ge_cut_tax_se`: counterfactual economy with perfect tax enforcement and fiscal neutrality with tax cut for self-employed only.

Welfare Analysis. To replicate the quantitative results in Section 5.4 (Tax Evasion and Welfare), please run the script `welfare_new.m`. This program calls the functions `welfare_computations_ce1`, `g_operator` and does the following: it loads previously computed results from the subfolder `results\mat` and computes consumption equivalent variations for the desired case.

Decompositions. To replicate the quantitative results in Section 5.3 (Quantifying the Channels of Tax Evasion), set the flags `fix_occpol`, `fix_kpol`, `fix_npol` and `fix_lpol` as explained in the file `fun_parameters.m`.

Tax Evasion and Credit Constraints. To calculate the results in Section 5.5, please set the flag which corresponds to the desired value of λ, either 1.2 or 1.8. Specifically, if the case $\lambda = 1.2$ is desired, set `do_lambda_exp` equal to 1. Otherwise set `do_lambda_exp` equal to 2.[2] For the benchmark $\lambda = 1.5$, set `do_lambda_exp` equal to 0.

4.1 Replicating Tables and Figures

- Figures 1-2-3
  - The data to replicate these figures are stored in `results\mat`.
  - Run `make_figures_123.m`. This script loads the relevant data from the subfolder `results\mat`, generates the figures and saves them in `results\Figures`. The figures are saved as eps by default. The user can change this with the flag `ext` in `make_figures_123.m`.
- Figure 4 (Welfare deciles)
  - The data to replicate this figure are stored in `results\mat`.
  - Run `make_figures_4.m`. This script loads the relevant data from the subfolder `results\mat`, generates the figure and saves it in `results\Figures`.
- Figure 5-6 (Fine and Welfare)
  - The data to replicate these figures are stored in `results\mat`.

[2] Please also set `do_GE=0` since β is calibrated to deliver the interest rate at 4%.

  - Run `make_figures_5_6`. This script loads the relevant data from the subfolder `results\mat`, generates the figures and saves them in `results\Figures`.

- Tables 2-3-4-5-6-7
  - The data to replicate these tables are stored in `results\mat`.
  - Run `make_tables_2_to_7.m`. This script loads the relevant data from the subfolder `results\mat`, generates the Tables and saves them as tex files in `results\Tables`.

- Table 8 (Decomposition)
  - The data to replicate these tables are stored in `results\mat`.
  - Run `make_tables_8`. This script loads the relevant data from the subfolder `results\mat`, generates the table and saves it as a tex files in `results\Tables`.

- Table 9 (Welfare)
  - The data to replicate these tables are stored in `results\mat`.
  - Run `make_tables_9`. This script loads the relevant data from the subfolder `results\mat`, generates the table and saves it as a tex files in `results\Tables`.

- Table 10 (Borrowing Limit)
  - The data to replicate these tables are stored in `results\lambda_mat`.
  - Run `make_tables_10.m`. This script loads the relevant data from the subfolder `results\lambda_mat`, generates the table and saves it as a tex files in `results\Tables`.

- Table 11 (Alleviation of Credit Constraints)
  - The data to replicate these tables are stored in `results\lambda_mat`.
  - Run `make_tables_11.m`. This script loads the relevant data from the subfolder `results\lambda_mat`, generates the table and saves it as a tex files in `results\Tables`.

- Table 15-16 (Varying λ)
  - The data to replicate these tables are stored in `results\lambda_mat`.
  - Run `make_tables_15_16.m`. This script loads the relevant data from the subfolder `results\lambda_mat`, generates the table and saves it as a tex files in `results\Tables`.