The codes are in Fortran. There are two main folders which include all codes.

- **Codes-QE-[Main Codes]**: This folder includes codes to replicate results summarized in "Section 5: Quantitative Results".

- **Codes-QE-[Robustness]**: This folder includes codes to replicate results summarized in "Section 6: Robustness".

Codes-QE-[Main Codes]

Recall that "Section 5: Quantitative Results" reports results for three different levels of ϕ. The codes for each case are located in the following sub-folders:

(1) **QE-[Phi=0.5192]**: Codes for the benchmark case ϕ=0.5192

(2) **QE-[Phi=0.0000]**: Codes for the case in which ϕ=0.0000 (lower bound)

(3) **QE-[Phi=1.0000]**: Codes for the case in which ϕ=1.0000 (upper bound)

This section explains the format of the codes and the procedures that should be followed to replicate the results for the benchmark case (ϕ=0.5192). For the other two cases (ϕ=0 and ϕ=1), the same procedures apply.

First, we will provide a summary and description of the "directory of subfolders" in which codes and results are included. After that, we will provide a detailed description of the procedure that should be followed to replicate the results.

"Directory of Subfolders"

The folder **QE-[Phi=0.5192]** includes 7 subfolders, where

- The first 3 subfolders include (i) the codes that compute the relevant ex-ante and ex-post steady states and (ii) the results.
- The latter 4 subfolders include (i) the codes that compute the relevant transitional dynamics and (ii) the results.

As indicated above, the steady state codes and results are included in 3 subfolders. These are:

(1) **SS - [ante]**: This folder includes codes to calculate the benchmark ex-ante steady state results in which τ_c=0.34. The results are also included in this folder - in a subfolder named **1A - [0.34]**.

(2) **SS - [td=tg]**: This folder includes codes to calculate *several* ex-post steady state results in which τ_c is reduced to several different levels while τ_d and τ_g are raised together by keeping them equal to each other (τ_d=τ_g) to compensate for the lost revenue from lower corporate tax rates. The folder also includes results in subfolders named **2A - [0.XX] - [td=tg]**.

(3) **SS - [td]**: This folder includes codes to calculate *several* ex-post steady state results in which τ_c is reduced to several different levels while only τ_d is raised to compensate for the lost revenue from lower corporate tax rates. The results are also included in this folder - in subfolders named **3A - [0.XX] - [td]**.

As indicated above, the transition codes and results are included in 4 subfolders. These are:

(4) **Trans - [td=tg]**: This folder includes the codes to calculate the transition between the ex-ante steady state (where τ_c=0.34) and *several* ex-post steady states (where τ_c is lowered to different levels while τ_d and τ_g are raised), where corporate tax cuts are treated as unexpected. The results are also included in this folder - in subfolders named **1T - [0.XX] - [td=tg]**.

(5) **Trans - [td=tg] - [Ant]**: This folder includes the codes to calculate the transition between the ex-ante steady state (where τ_c=0.34) and one ex-post steady state (where τ_c is lowered to **zero** and τ_d and τ_g are raised), where corporate tax cuts are assumed to be anticipated either 1 or 2 periods in advance. The results are also included in this folder - in subfolders named **1T - [0.00] - [td=tg] - [Ant-1]** and **1T - [0.00] - [td=tg] - [Ant-2]**

(6) **Trans - [td]**: This folder includes the codes to calculate the transition between the ex-ante steady state (where τ_c=0.34) and *several* ex-post steady states (where τ_c is lowered to different levels while only τ_d is raised), where corporate tax cuts are treated as unexpected. The results are also included in this folder - in subfolders named **2T - [0.XX] - [td]**.

(7) **Trans - [td] - [Ant]**: This folder includes the codes to calculate the transition between the ex-ante steady state (where τ_c=0.34) and one ex-post steady state (where τ_c is lowered to **zero** while only τ_d is raised), where corporate tax cuts are assumed to be anticipated either 1 or 2 period in advance. The results are also included in this folder - in subfolders named **2T - [0.00] - [td] - [Ant-1]** and **2T - [0.00] - [td] - [Ant-2]**

"Procedures to Follow to Replicate Steady State Results"

The steady state results are calculated by 3 Fortran files: in addition to the main file (named OLG.f90), there are two supplementary files for parameters (PARAME.f90) and utilities (Utilities.f90) that are called by the main code.

- The utilities file includes supporting functions that are used in the execution stage. This file does not need any editing.
- Model parameters and several key target levels are included in the parameter file. This file needs to be edited -together with the main file- based on the scenario considered.

1 - Procedures: Ex-Ante Steady State (Benchmark)

- **Set**: In parameters file, we set τ_c=0.34 and τ_g=0.20 - pre-reform tax levels in benchmark.
- **Set**: In the main file, we set upper bound (`td_high`) and lower bound (`td_low`) on search for dividend tax to 0.20 so that we set pre-reform dividend tax to its pre-reform level (no search).

- **Set**: In the main file, we set the upper and lower bounds on wages (`wu/wd`) and interest rates (`rhigh/rlow`) slack enough for the equilibrium search not to hit the boundaries.

  After the parameterization, codes can be compiled. After that stage, one can run the exe file to get the benchmark steady state results. The code will generate several text files including different set of results.

  As indicated above, we have also included the results of this procedure for each case, i.e. the executable .exe file and the steady state results (.txt files) that are generated by running the .exe file. If you prefer to skip the editing/compiling stages, you can check these results in the .txt files directly. Alternatively, in case you want to reproduce the results, you can use/execute the corresponding .exe file in a separate folder (that does not include the .txt files).

2 - Procedures: Ex-Post Steady State (td=tg)

- **Set**: In parameters file, set government expenditure level (gov) to the benchmark level implied by the "Ex-Ante Steady State" results.
- **Set**: In parameters file, set τ_c to the level considered.
- **Set**: In main file, we set upper and lower bounds on dividend tax rate τ_d (`td_high/td_low`) slack enough for the equilibrium search not to hit the boundaries. The code automatically equates $\tau_g = \tau_d$.
- **Set**: In the main file, we set the upper and lower bounds on wages (`wu/wd`) and interest rates (`rhigh/rlow`) slack enough for the equilibrium search not to hit the boundaries.

  After the parameterization, codes can be compiled and the exe file can be executed.

3 - Procedures: Ex-Post Steady State (td)

- The procedure in this setup is similar to the one above in which we assume $\tau_g = \tau_d$. The only different set is, in parameters file, set τ_g=0.20 (in other words, set it to its pre-reform level).

"Procedures to Follow to Replicate Transition Results"

Similarly to the steady state, the transition results are calculated using 3 Fortran files: in addition to the main file (named OLG.f90), there are two supplementary files for parameters (PARAME.f90) and utilities (Utilities.f90) that are called by the main code.

1 - Procedures: Transition (td=tg) - Unexpected

- **Set**: In parameters file, we set initial level of corporate tax rate (`tci`) to 0.34 (pre-reform level) and the post-reform tax rate (`tcf`) to the level considered.
- **Set**: In parameters file, set government expenditure level (gov) to the benchmark level implied by the "Ex-Ante Steady State" results.

- **Include**: The initial and final steady state results must be provided to the transition codes. Accordingly, all text files generated by the corresponding steady state codes must be included in the same folder with the executable, so that the executable can reach these files.
    - Initial steady state results text files should be renamed with the convention of **XXX_I,** where I refers to "Initial" (steady states codes that we use -by default- record the results with an "_F" extension). There are 15 steady state .txt files that need to be renamed and, then, included in the same folder with the transition code executable:
        - Agg1_I.txt Agg2_I.txt Agg3_I.txt Dist_I.txt DistC_I.txt
        - Div_I.txt OptCons_I.txt OptK_I.txt OptMULT.txt _I OptQQQ_I.txt
        - PVal_I.txt OptShare_I.txt Seq_I.txt Val_I.txt VC_I.txt
    - Final steady state results text files already has the correct naming convention of **XXX_F,** where F refers to "Final". There are 15 steady state .txt files that need to be renamed and, then, included in the same folder with the transition code executable:
        - Agg1_F.txt Agg2_F.txt Agg3_F.txt Dist_F.txt DistC_I.txt
        - Div_F.txt OptCons_F.txt OptK_F.txt OptMULT.txt _F OptQQQ_F.txt
        - PVal_F.txt OptShare_F.txt Seq_F.txt Val_F.txt VC_F.txt

  After the setup, the codes can be compiled. After that stage, one can run the .exe file to get the transition results. The code will generate several additional text files including different sets of results.

  Once more, as indicated above, we have also included the results of this procedure for each case, i.e. the executable .exe, initial and final steady state input text files and the transition results (.txt files) that are generated by running the .exe file. If you prefer to skip the editing/compiling stages, you can check these results in the .txt files directly. Alternatively, in case you want to reproduce the results, you can use/execute the corresponding .exe file in a separate folder (that does not include the transition results .txt files but does include the initial and final steady state input text files).

2 - Procedures: Transition (td) - Unexpected

- The procedure in this setup is the same as the one described above.

3 - Procedures: Transition (td=tg) – Anticipated

- The procedure in this setup is the same as the one described above.

4 - Procedures: Transition (td) - Anticipated

- The procedure in this setup is the same as the one described above.

Codes-QE-[Robustness]

Recall that "Section 6: Robustness" reports results for two robustness checks. The codes for each case are located in the following sub-folders:

(1) **Adjustment Costs**: The effect of lower capital adjustment costs on key steady state results for the benchmark calibration (ϕ=0.5192)

(2) **Debt Financing**: The effect of incorporating debt financing on key steady state results for the benchmark calibration (ϕ=0.5192)

First, we will provide a summary and description of the "directory of subfolders" in which codes and results are included. After that, we will provide a detailed description of the procedures that should be followed to replicate the results.

Adjustment Costs - "Directory of Subfolders" & "Procedures"

The steady state codes and results for the low adjustment cost case are included in 3 subfolders. This set of Fortran codes are parallelized and, accordingly, compiled and executed on a cluster. Therefore, each folder includes two extra files -build.sh and submit.sh- that are required for the parallelization.

(1) **1A-mp-HV-[td=tg]-[0.34]**: This folder includes codes to calculate the new benchmark ex-ante steady state results in which τ_c=0.34. Results are also included in the same folder.

(2) **1A-mp-HV-[td=tg]-[0.00]**: This folder includes the codes to calculate the new ex-post steady state results in which τ_c is reduced to **zero**, τ_d and τ_g are raised together by keeping them equal to each other (τ_d=τ_g) to compensate for the lost revenue from lower corporate tax rates. The results are also included in the same folder.

(3) **1A-mp-HV-[td]-[0.00]**: This folder includes the codes to calculate the new ex-post steady state results in which τ_c is reduced to **zero**, only τ_d is raised to compensate for the lost revenue from lower corporate tax rates. The results are also included in the same folder.

For each case, setup and procedures that needs to be followed to generate the results are similar to the ones described above for the main results reported in "Section 5: Quantitative Results".

Debt Financing - "Directory of Subfolders" & "Procedures"

The steady state codes and results are included in 3 subfolders. Similarly, this set of Fortran codes are parallelized and, accordingly, compiled and executed on a cluster. To summarize

(1) **1A-mp-BK-[tc=0.34]-[td=tg]**: This folder includes codes to calculate the new benchmark ex-ante steady state results in which τ_c=0.34. Simulation results are included in the same folder.

(2) **1A-mp-BK-[tc=0.00]-[td=tg]**: This folder includes codes to calculate the new ex-post steady state results in which τ_c is reduced to **zero**, τ_d and τ_g are raised together by keeping them equal to each other (τ_d=τ_g) to compensate for the lost revenue from lower corporate tax rates. The simulation results are included in the same folder.

(3) **1A-mp-BK-[tc=0.34]-[td]**: This folder includes the codes to calculate the new ex-post steady state results in which τ_c is reduced to **zero**, only τ_d is raised to compensate for the lost revenue from lower corporate tax rates. The simulation results are included in the same folder.

For each case, setup and procedures that needs to be followed to generate the results are similar to the ones described above for the main results reported in “Section 5: Quantitative Results”.