README FILE FOR REPLICATION MATERIALS

“Policy Effects of International Taxation on Firm Dynamics and Capital Structure”

By Adam Hal Spencer

A Overview

The package accompanying this document is comprised of two folders as follows

- Empirical: this folder contains the Stata do file codes and other miscellaneous data required to replicate the empirical work presented in the paper.
- Model: this folder contains Fortran and Matlab codes required to solve the variants of the model in addition to the Matlab codes required to create figures.

B Data Availability

The data I use come from several sources.

- *Compustat-Capital IQ*. Data are subject to distribution restrictions as they are proprietary. I access these data trough Wharton Research Data Service (WRDS). To request an account and access the data please visit: `https://wrds-www.wharton.upenn.edu/register/`. I use both Compustat North America - daily fundamentals and the historical segments - daily data. These data were downloaded from WRDS on April 21, 2021. I get to the download page for the fundamentals data with the following chain of navigation on the WRDS page.
  - Home,
  - Get Data,
  - Compustat - Capital IQ,
  - Compustat, North America - Daily,
  - Fundamentals Annual.

  From there, I download the data manually as follows.

  - Step 1: choose the date range 1979–2017.

- Step 2: select the "gvkey" and "search the entire database" radio buttons.
- Step 3: choose your variable types. Select the data items check button. Find the variables listed in \Empirical \fundamentals_variables.xls.
- Step 4: select query output of Stata file (*.dta).

I get to the download page for the segments data with the following chain of navigation on the WRDS webpage.

- Home,
- Get Data,
- Compustat - Capital IQ,
- Compustat,
- Historical Segments - Daily,
- Historical Segments.

I then download the data manually as follows.

- Step 1: choose the date range 1979–2017.
- Step 2: apply your company codes: choose the "gvkey" and "search the entire database" radio buttons. Also check all the segment type buttons.
- Step 3: choose query variables. I select all: these are listed in \Empirical \segment_variables.xls.
- Step 4: select query output of Stat file (*.dta).

• *World Bank*. I take the imports of goods and services (% of GDP) series at link `https://data.worldbank.org/indicator/NE.IMP.GNFS.ZS?end=2017&mostrecentyeardesc=true&start=1960`. I then download the series using the EXCEL link on the right. The downloaded data are in \Empirical \world_bank.xls. I take the 2017 number of 15%. I downloaded these data on the 20^{th} of August 2020.

• *Organization for Economic Cooperation and Development (OECD)*. I download the "Taxing Wages - The United States" file at link `https://www.oecd.org/unitedstates/taxing-wages-united-states.pdf`. I then take the labour income tax rate from the 2017 number on the figure entitled "average tax wedge over time for a single worker" on page 2. I include this in the replication package under the name \Empirical \oecd.pdf. I downloaded this document on the 20^{th} of August 2020.

- *Bureau of Economic Analysis (BEA).* To obtain the numbers relating to employment by U.S. multinationals by destination, I first go first to the BEA's data homepage at `https://www.bea.gov/data/intl-trade-investment/activities-us-multinational-enterprises-mnes`. I make use of the following sequence of navigation to download the data manually.

  - Step I: Interactive Data,
  - Step II: Interactive Tables: Activities of Multinational Enterprises,
  - Step III: Click the radio buttons for "Unted States direct investment abroad" and "Data on activities of multinational enterprises",
  - Step IV: All Foreign Affiliates (data for 2009 forward),
  - Step V: Employment,
  - Step VI: By Country Only (Major Countries),
  - Step VII: Select the year "2017", with Columns as "Employees" and Rows as "–All Geographical Areas–",
  - Step VIII: I then download the data in excel format.

  The data are enclosed in \Empirical \BEA_employees.xls. I downloaded these data on the 18^{th} of November 2020. Then to obtain the numbers in relation to sales of multinational affiliates and parents, I follow the same procedure for Step I – Step IV above. I then use the following navigation to obtain total sales of foreign affiliates

  - Total sales,
  - Aggregate Totals Only,
  - Select the year as "All Years" with Aggregate Totals as "Sales" and "All Countries Total",
  - I then download as an excel spreadsheet.

  To obtain the sales for U.S. multinational parents, I return to step III above and then follow

  - All U.S. Parent Companies (data for 2009 and forward),

- Total Sales,
- Aggregate Totals Only,
- Select the year as "All Years" with Aggregate Totals as "Sales" and "All Industries Total",
- I then download as an excel spreadsheet.

I've enclosed the datasets for both affiliates and parents, as well as the implied ratio, in this replication package under the name \Empirical \BEA_sales.xls. I downloaded these data on the 20^{th} of August 2020. Finally to obtain the financial outflow data, I follow the same procedure for step I and step II above. I then use the following navigation

- Click the radio buttons for "United States direct investment abroad" and "Balance of payments and direct investment position data",
- Click the radio button for "Financial Outflow Transactions Without Current-Cost Adjustment",
- Click the radio button for "By Country Only (All Countries)",
- Choose "All Years", "Financial Outflows" and "All Geographical Areas",
- I then download as an excel spreadsheet.

I dowloaded these data on FDI flows on July 19 2020. They are included in \Empirical \BEA_fdiflows.xls. The first sheet has the raw data and the second converts nominal flows to real flows. Note that the CPI numbers are taken from the \Empirical \CPI_data.dta file. The number quoted in the text is in cell F8 on the *real* sheet.

- *Bureau of Labor Statistics (BLS)*. I use CPI data from this source, predominantly for the purpose of deflating firm sales in estimating the productivity process. I use series CUSR0000SA0L1E called "all items less food and energy in U.S. city average, all urban consumers, seasonally adjusted". This is downloadable at `https://beta.bls.gov/dataViewer/view/timeseries/CUSR0000SA0L1E`. The numbers are included with the replication package in Excel (*.xls) format in \Empirical \CPI.xls. I downloaded these data on the 18^{th} of August 2020. I also enclose a Stata format (*.dta) file with annual CPI numbers that are derived from CPI.xls. This file is generated in the fundamentals.do file.

C Dataset List

Data file	Source	Notes	Provided
\Empirical \CPI.xls	BLS	CPI data	Yes
\Empirical \cpi_data.dta	BLS	This finds a yearly figure for the CPI using CPI.xls (takes the December value for each year). This is generated in the fundamentals.do file	Yes
\Empirical \segments_raw.dta	Compustat	Segment data downloaded from Compustat	No (proprietary)
\Empirical \fundamentals_raw.dta	Compustat	Fundamentals data downloaded from Compustat	No (proprietary)
\Empirical\world_bank.xls	World bank	Gives λ from the data	Yes
\Empirical\BEA_employees.xls	BEA	Employment data from the BEA	Yes
\Empirical\BEA_fdiflows.xls	BEA	FDI outflows data from BEA	Yes
\Empirical\BEA_sales.xls	BEA	Affiliate and parent sales totals from the BEA	Yes
\Empirical \OECD.pdf	OECD	Gives τ^W	Yes

D Computational Requirements

Software

- Stata codes were run on Stata/SE 15.0 for Windows.

- estout is an additional package utilised for saving regression outputs. It was installed using the command

```
ssc install estout, replace
```

- xtable was also used and installed with the command

```
ssc install xtable, replace
```

- Linux: Slurm compute service on the “Augusta” cluster at Nottingham.
- Intel Fortran compiler 2019 version 5.075 (for running Fortran codes on the cluster).
- Visual Studio 2019 with Intel Parallel Studio XE Cluster Edition for Windows v2020 (for running the *closed_economy.f90* Fortran code on the desktop).
- Matlab version R2018b.

Hardware

- Stata, Matlab and *closed_economy.f90* Fortran codes were run on a Windows 10 (64-bit) machine with Intel(R) Core(TM) i3-7100T CPU @ 3.40GHz 3.41 GHz, 8.00GB RAM.
- All other Fortran codes were run on a Linux server with 20 cores (Intel Xeon Gold 6138 20C 2.0GHz CPU) and 192Gb of RAM.

E Description of Programs

Empirical folder

1. segments.do: works with Compustat’s segment data. This file allocates firms according to their foreign engagement status through the procedure described in the paper’s data appendix. This program also undertakes the regression analysis described in the empirical validation section of the paper.
    - Section “create the dataset” extracts information from the raw fundamentals data for the purpose of establishing firms’ counties of nationality. It incorporates this information into the segment dataset and then saves a new (*.dta) file.

- Section "clean the dataset" manipulates the data prior to analysis.
- Section "allocate firm statuses" gives a preliminary assignment of a firm's foreign status, which is subject to further adjustment below.
- Section "aggregate variables of interest across firm segments" combines all the reported information for firm segments to obtain aggregate measures for the overall firm entity.
- Section "adjust for temporary status changes" makes the status change considerations as in Fillat & Garetto (2015). This then saves the final datafile for further use in the program fundamentals.do described below.
- Section "reforms and the fraction of exporting firms" undertakes the analysis in the empirical validation exercise. It allocates a treatment indicator and firm nationality indicator based on the panel and then runs the difference-in-difference regression as well as placebos. This then outputs file *reg_table_out1.xls* and *reg_table_out2.xls*, which give 12 in the main paper. This section then runs the dynamic regression and outputs file *trends.eps*, which is the parallel trends figure 3 in the paper.

2. fundamentals.do: this file works with Compustat's fundamentals data, as well as the firm status allocations found from running segments.do above, to generate the summary statistics in the paper.

   - Section "clean the fundamentals data and merge with the cleaned segment information" loads the raw fundamentals data before combining with the final dataset generated by running segments.do. It then cleans the data before analysis. The cleaned data are saved before proceeding to the next section.
   - Section "estimate the rho productivity persistence parameter" estimates parameter ρ_θ for use in the model using the Olley & Pakes (1996). It outputs the regression coefficient, which is saved for later export from Stata.
   - Section "generate the investment rate and leverage moments" trims the data and then calculates firm-level summary statistics. These are then saved for later export from Stata.
   - Section "generate the depreciation rate parameter" trims the data and then provides an estimate of parameter δ to be used on the model. This is saved for later export from Stata.

- Sections "generate the fractions moments", "generate the export intensity moment", "generate the transition probability moments" and "generate the size premia moments" generate more summary statistics. The latter finds the size premia in terms of productivity as well as in terms of assets. The data are trimmed and then these moments are saved for later export from Stata.
- Section "output the parameters and moments" tabulates three sets of saved numbers — those for the parameters ρ_θ and δ, those for the moments and another that outputs the number of observations over [1980,2005]. They are put into matrix form and then exported using the putexcel command. The tables are saved as excel spreadsheets called *compustat_parameters.xls*, *compustat_moments.xls* and *compustat_noobs_eqiss.xls* respectively.

3. fundamentals_shortseries.do: this file serves the same purpose as fundamentals.do, except it is adjusted to use a shorter timespan in accordance with the empirical validation exercise. Note then that the sections of the code are the same as in the above descriptions for item 2, to the end of computing moments for the 1979–2004 period.

Model folder

4. baseline.f90: this Fortran file solves for the equilibrium of the baseline model and runs the main simulation experiment in section V.ii. of the paper. This file contains the following modules and subroutines.

   (a) parameters module. This defines key parameters and variables to be used by the code at large. It contains a subroutine called "grids", which outputs the variable grids used in the computations. Note that this subroutine takes text files called *theta_grid_Aug29.txt* and *Q_grid_Aug29.txt* as inputs — these are the discretised values of the productivity variable θ_t and its transition matrix respectively. These are constructed externally using the *addacooper.m* code that accompanies Floden (2008). It is available at `https://martinfloden.net/files/ar1_processes_matlab_code.zip`.

   (b) steady_state subroutine. This finds the steady state corresponding to the baseline model. It calls several other subroutines to find the equilibrium objects. The solution process has two options — one used in finding the calibration steady state and another for the post-policy change (hereafter referred to as the counterfactual) steady state. In the calibration, the wage W_t is fixed to unity and the sunk cost of firm creation f^{HQ} is treated as a free parameter. In this case, the only equilibrium object found iteratively is the aggregate level of production. Each iteration involves setting the new conjecture as the level of supply from that previous. When one solves for the counterfactual steady state, three equilibrium objects are found iteratively — the domestic aggregate production (as in the calibration), the domestic wage and the price of H goods in F. These three objects are updated on each iteration gradually with varying intensities. Note that this subroutine has various outputs, some of which are from the calibration step to be used in the counterfactual step and others are for the purpose of solving for the counterfactual transition path (to be discussed later).

(c) firm_mass subroutine. This subroutine outputs the endogenous steady state measure using the labour market clearing condition.

(d) incumbent_aggregates subroutine. This finds the aggregate variables associated with the equilibrium firm mass found in 4c.

(e) household_ss subroutine. This takes aggregate steady state objects from the firm-side of the model as inputs and then returns the corresponding aggregate level of consumption and utility.

(f) distribution subroutine. This takes the policy functions of firms as an input and finds the steady state equilibrium firm distribution iteratively.

(g) moments subroutine. This takes policy functions from the firm problem and the equilibrium steady state distribution found in 4f to find the implied moments. This is a long subroutine that generates both scaled moments, (such as the average leverage ratio) as well as unscaled averages (such as the average capital stock).

(h) vfun_entrant subroutine. This takes the incumbent firm value function as an input and returns the ex-ante firm entry value as well as entrant firm policy functions.

(i) vfun_iteration subroutine. This finds the incumbent firm value function associated with the state space. It uses the value function iteration procedure, augmented to leverage Howard's improvement algorithm. It calls several other subroutines that evaluate firm values for a given foreign engagement status. It also outputs firms' policy functions. It contains two internal subroutines for the purspose of evaluating firms' discrete choices.

i. vfun_iteration_dc_optimise subroutine. The value function iteration procedure calls this to weigh-up all of the possible discrete choices on optimisation iterations (as per Howard's algorithm).

ii. vfun_iteration_dc_nooptimise subroutine. The value function iteration procedure calls this to make the firms' discrete choice on *non*-optimisation iterations (as per Howard's algorithm).

(j) vfun_exit subroutine. This outputs the values for a firm exiting associated with the state space.

(k) vfun_domestic subroutine. This outputs the values from being a purely domestic firm. It also outputs the associated policy functions.

(l) vfun_exporter subroutine. This outputs the values associated with the exporting decision and its policy functions.

(m) vfun_multinational subroutine. This outputs the values for the multinational discrete choice and the associated policy functions.

(n) vfun_offshoring subroutine. This returns the values for being an offshoring multinational and its policy functions.

(o) foreign_demand subroutine. This takes the foreign price of H goods as an input and returns the associated level of aggregate demand. Notice that this is only used in the counterfactual step of the steady_state subroutine 4b.

(p) total_ave_labour_demand subroutine. This outputs the level of aggregate labour demand associated with firms' choices. It produces output that can be used with the firm_mass subroutine 4c to get the overall measure of firms.

(q) government_bc subroutine. This takes firms' aggregates as inputs and then returns the aggregate level of taxation as per the H government's budget constraint. This output is used in conjunction with the household_ss subroutine 4e to find welfare.

(r) Q_star_calibration subroutine. This fixes the aggregate level of F demand to unity in the calibration procedure of the steady_state subroutine 4b. It is only used when calibrating the steady state.

(s) transition subroutine. This solves for the counterfactual transition path of the model. It takes as inputs data from solving for the calibrated and counterfactual steady states, as these serve the role of boundaries for the algorithm. This subroutine returns the overall consumption equivalent variation effect of the policy reform, the NPV effect on H tax collections and the time paths for consumption, the trade balance, the measure of entrants, measure of exporters and measure of multinationals. This subroutine utilises the shooting algorithm to solve for the transition. Time paths for aggregate equilibrium objects are conjectured, based on which the firm problems are solved by shooting backwards from the new steady state. Then the firm measure is found by shooting forwards on its law of motion, while leveraging the firm policy functions. The household problem is solved by shooting forwards on their Euler equation. Then at the end of a given iteration, the aggregate objects are slowly updated. This process continues until all the equilibrium conditions are satisfied at any point in time. This subroutine contains several internal subroutines as follows.

   i. aggregates_paths subroutine. This takes the boundary steady state conditions and conjectures an initial time path for the required equilibrium objects.

   ii. vfun_transition subroutine. This takes the period ahead firm value function and time paths for aggregate objects as inputs and then returns the current period value function. It feeds transition inputs to the vfun_iteration subroutine 4i and makes bilateral comparisons to find the firms' equilibrium discrete choices for the period. It also outputs policy functions for the period.

   iii. measure_transition subroutine. This finds the period ahead overall measure of firms. It takes the current period measure and firm policy functions as inputs.

   iv. aggregates_transition subroutine. This takes the period policy functions and firm measure as inputs and returns the endogenous aggregate objects.

   v. household_transition subroutine. This takes income items for the household as inputs and then finds the current period level of consumption and period ahead demand for riskless bonds.

vi. summary_transition subroutine. This finds the consumption equivalent variation welfare effect of the policy counterfactual along the whole transition and new steady state and NPV of the tax revenue effect.

vii. government_bc_transition subroutine. This takes aggregate objects as inputs and then returns the aggregate level of H government tax collections as an output.

5. FEIM.f90: this Fortran file solves for the equilibrium of the free equity issuance model (FEIM) and runs the main simulation experiment in section VI.ii. of the paper. This file contains subroutines that serve the same purpose as in baseline.f90 (see 4 above). That is, there are versions of 4a–4s subroutines in FEIM.f90, albeit with slight differences in things such as updating parameters.

6. optimiser.f90. This Fortran code conducts a search over the parameter space for the model. It contains subroutines that serve the same purpose as 4a–4r. It also includes subroutines coming from the *SIMANN* simmulated annealing package of Goffe (1996), which optimise over the parameter space. This package is downloadable at `https://jblevins.org/mirror/amiller/`. The additional subroutines from this package are *EXPREP, RMARIN, ranmar, PRT1, PRT2, PRT3, PRT4, PRT6, PRT7, PRT8, PRT9, PRT10, PRTVEC* and *SA*. The reader is directed to Goffe (1996) for more details on these.

7. optimal_tau.f90. This Fortran code performs runs to the end of finding the optimal steady state repatriation tax in the context of the baseline model. It contains subroutines that serve the same purpose as 4a–4r above. It takes 21 values of the tax rate over the interval [0,0.2] and solves for the associated steady state for each. This file outputs a table of results from moving from the observed to the optimal rate of 5%, as reported in the main body of the text in section VI.i.

8. emp_validation.f90. This Fortran code undertakes the simulation experiment for the empirical validation exercise in section V.i. of the paper. This code contains versions of the subroutines listed in 4a–4r above. The exercise solves first for the calibration steady state with only 85% of exporters' earnings being taxable by the U.S. Government. It then runs a simulation where this fraction is changed to 100% and solves for the new steady state. The calibration step reports a table of moments, including persistence of the X status, the transition (D, X) and the fraction of exporters. The

counterfactual step reports the latter three moments, as were reported in the main body of the text.

9. closed_economy.f90. This Fortran code solves the closed economy exercise presented in section VI.iii. in the paper. The calibration step reports tables of parameters and moments. This code contains the following modules and subroutines.
   (a) parameters module. This serves the same purpose as 4a above.
   (b) closed_economy.f90 subroutine. This solves for the steady state of the closed economy model. It takes inputs such as an indicator for whether the calibration or counterfactual step is being run. Similarly to 4b above, in the case of the calibration, the fixed cost f^{HQ} is treated as a free parameter and the model iterates only on aggregate production. In the counterfactual case, it iterates on this variable as well as the wage. This subroutine contains several internal subroutines as follows.
      i. vfun_closed subroutine. This finds the value function for incumbent firms for the closed economy model. It finds separately the value of continuation and for exit and then compares the two via the firm's discrete choice. It also outputs the associated policy functions. This is the closed economy analogue of 4i above.
      ii. grids_closed subroutine. This finds the grids for firm state variables for the closed economy model.
      iii. vfun_entrant_closed subroutine. This finds the ex-ante value of entry for the closed economy model. It takes as an input the incumbent firm value function. It also outputs the entrant policy functions. This is the closed economy analogue of 4h above.
      iv. distribution_closed subroutine. This finds the distribution of firms for the closed economy model via iteration. This is the closed economy analogue of 4f above.
      v. moments_closed. This finds moments associated with the closed economy model. It takes firms' policy functions and the distribution as inputs. This is the closed economy analogue of 4g above.
      vi. total_ave_labour_demand_closed subroutine. This finds the aggregate labour demand associated with firm choices in the closed economy model. This is

the closed economy analogue of 4p above.

vii. incumbent_aggregates_closed subroutine. This finds aggregate variables associated with the equilibrium firm mass. This is the closed economy analogue of 4d above.

viii. government_bc subroutine. This takes aggregates as inputs and outputs aggregate government tax collections. This is the closed economy analogue of 4q above.

ix. firm_mass subroutine. This uses the domestic labour market clearing condition to find the equilibrium firm mass. This is the closed economy analogue of 4c above.

x. household_ss_closed subroutine. This takes aggregate variables as inputs and then outputs household consumption and welfare. This is the closed economy analogue of 4e above.

10. TCJA_full.f90. This Fortran code runs the simulation experiment where the domestic government both removes the repatriation tax and reduces the domestic corporate tax rate to 21% simultaneously. It solves for the steady state in the counterfactual scenario. This is referred to in the last paragraph of section V.ii. in the main body of the paper. This program contains subroutines that serve the same purpose as 4a–4r.

11. static_asymmetric.m. This Matlab file solves the static model presented in section VI.i. of the main paper. The main section of the code first declares the parameters (below comment "parameters"), then runs the policy counterfactual where the repatriation tax is removed, (below comment "counterfactual"). Then it solves the model several times to the end of finding the optimal static repatriation tax (below comment "optimal taxation"). The optimal tax exercise solves for the equilibrium for 74 values of the tax rate between 0% and 73%; with each run, the welfare level is saved in an array. After all runs, the maximal welfare level is found from the aforementioned array. This Matlab code takes a text file as an input: *theta_static_grid_Aug29.txt*, which contains the discretised values of the θ_t variable. These values are also found using the code that accompanies Floden (2008) discussed above. The code contains several functions deferred to the end of the script as follows.

    (a) value_functions. This function takes aggregate objects as inputs and then returns both the incumbent and ex-ante entry value functions. It also outputs the associated policy functions. This is the static model analogue of 4i above.

(b) distribution. This function finds the distribution of firms across the productivity state given their exit/continue policy function and the exogenous distribution vector. This is the static model analogue of 4f above.

(c) moments. This function finds the moments (both scaled and unscaled) of the economy given the equilibrium distribution of firms and policy functions. This is the static model analogue of 4g above.

(d) total_labour. This function finds the aggregate labour demand based-on the choices of firms. This is the static model analogue of 4p above.

(e) household. This function outputs the level of consumption and utility given aggregate variable inputs. This is the static model analogue of 4e above.

(f) government. This function finds the aggregate level of government tax collections, based on aggregate variables, for distribution to the household. This is the static model analogue of 4q above.

(g) firm_mass. This takes the aggregate labour demand as an input and outputs the endogenous measure of firms. This is the static model analogue of 4c above.

(h) aggregates. This function takes the unscaled averages and firm measure as inputs and then outputs aggregate variables. This is the static model analogue of 4d above.

(i) foreign_demand. This takes the price of H goods in F and then outputs the aggregate level of demand given the exogenous demand curve. This is the static model analogue of 4o above.

(j) equilibrium. This finds the steady state equilibrium associated with the parameter specification. This is used in conjunction with Matlab's *fsolve* command in order to find the equilibrium aggregate objects. It takes an indicator function as an input, which signals whether the calibration or counterfactual is being solved for. In the calibration step, the wage is fixed and entry cost treated as a free parameter and the aggregate domestic production is solved for. In the counterfactual step, the aggregate wage, foreign price and domestic production are found. This is the static model analogue of 4b above.

(k) equilibrium_outputs. This function performs the same computations as the function in 11j, but additionally outputs the moments associated with the calibration.

12. static_deferrabiliy.m. This Matlab code undertakes the deferrability exercise in section VI.iii. of the main paper. Versions of functions 11a–11k are all present in this code as well and serve the same role.

13. static_symmetric.m. This Matlab code solves the symmetric country static model discussed in section VI.iii. of the main paper. This code has the same basic structure as *static_asymmetric.m* in 11 above. It has analogous functions to 11a–11k, but notice that the structure of these functions in terms of inputs/outputs and the computations differ considerably, since they solve for heterogeneous distributions of firms from both H and F.

14. make_figures.m. This Matlab code creates the four pictures presented in the paper. It takes the text files *measureE_baseline.txt*, *measureX_baseline.txt*, *measureM_baseline.txt*, *C_baseline.txt*, *TB_baseline.txt*, *measureE_FEIM.txt*, *measureX_FEIM.txt*, *measureM_FEIM.txt*, *C_FEIM.txt* and *TB_FEIM.txt* as inputs. It then outputs the four figures in the (*.eps) format.

F Instructions to Replicators

Empirical folder details. The *segment.do* file must be run before running *fundamentals.do* or *fundamentals_shortseries.do* files. The first creates an intermediate dataset required for analysis in the latter two.

Model folder details. Each of these codes can be run completely independently of the others. The Fortran codes were all compiled and run on Nottingham's HPC "Augusta" cluster, with the exception of *closed_economy.f90* (this was run on the desktop mentioned earlier with only a single core). In each case, I'd compile and run using Intel Fortran in conjunction with OpenMP. I'd use a single server with 20 cores. Below I list the command used to compile each of the .f90 files. I write this once to economise on space, for a fictitious file called *example.f90*, where notice any of the .f90 files listed in section E can replace it.

```
module load intel/compiler/2019/5.075
ifort example.f90 -mcmodel=medium -Mlarge_arrays -o new -qopenmp -O2 -g -
    traceback -heap-arrays
```

where *new* is the name of the executable file to run in the cluster environment. I then send a submit file to the cluster using a script that contains the command

```
export OMP_NUM_THREADS=20
```

G Computational Speed Estimates

This section gives an estimated run time for each of the codes listed in section E.

Code	**Run time (hours)**
segments.do	0.09
fundamentals.do	0.09
fundamentals_shorteries.do	0.09
baseline.f90	8
FEIM.f90	10
optimiser.f90	24
optimal_tau.f90	24
emp_validation.f90	2
closed_economy.f90	0.09
TCJA_full.f90	2
static_asymmetric.m	0.05
static_deferrability.m	0.05
static_symmetric.m	0.05
make_figures.m	0.01

H List of Tables and Programs

Figure/Table	**Program**	**Line**	**Output File**	**Note**
Table 1	No program	N/A	No data	N/A
Table 2	No program	N/A	No data	N/A
Table 3	\Model \baseline.f90	943–960	parameters_outside_baseline.txt	This prints the table from within the code that solves the model

	No program	N/A	\Empirical \world_-bank.xls	This gives λ from the data. Take the figure for 2017.
	No program	N/A	\Empirical \BEA_-employees.xls	This gives employment levels by country for construction of $\tau^{\Pi*}$
	No program	N/A	\Empirical \OECD.pdf	This gives τ^{W}. Take the 2017 number
	\Empirical \fundamentals.do	513–516	\Empirical \compustat_parameters.xls	Outputs δ, ρ_θ
Table 4	\Model \baseline.f90	981–999	parameters_inside_-baseline.txt	This prints the parameters from inside the code that solves the model
	\Model \optimiser.f90	255–271	parameters_optimiser.txt	This code runs an optimiser to find these exact parameters. The line reference is where the final parameters are outputted
Table 5	\Model \baseline.f90	1026–1048	moments_baseline.txt	This prints the model moments from inside the code that solves the model
	\Empirical \fundemantals.do	519–524	compustat_moments.xls	This outputs the Compustat data moments that are calculated earlier in the script

	N/A	C42	\Empirical \BEA_sales.xls	This contains the affiliate sales intensity data. The line reference outputs the average used in the paper
Figure 1	\Model \baseline.f90	2206–2222	measureX_baseline.txt, measureM_baseline.txt, measureE_baseline.txt	This outputs the firm measures data along the transition: the numbers to be used to make figures in Matlab
	\Model\make_figures.m	17–26	Measures_bl.eps	This generates the figure using the above data for the baseline
	\Model \FEIM.f90	2167–2181	measureX_FEIM.txt, measureM_FEIM.txt, measureE_FEIM.txt	This outputs the firm measures data along the transition for the FEIM to be used to make the figure
	\Model\make_figures.m	42–52	Measures_FEIM.eps	This generates the figure using the above data
Figure 2	\Model \baseline.f90	2224–2234	C_baseline.txt, TB_baseline.txt,	This generates the consumption and trade balance time paths data along the transition for the baseline; to be used to make the figure in Matlab

	\Model\make_figures.m	65–76	C_bl.eps	This generates the figure using the above data for the baseline
	\Model \FEIM.f90	2182–2191	C_FEIM.txt, TB_FEIM.txt	This generates the consumption and trade balance time paths data along the transition; for production of the figure in Matlab
	\Model\make_figures.m	86–96	C_FEIM.eps	This generates the figure for the FEIM using the above data
Table 6	\Model \baseline.f90	12,093–12,110	cf_results_ss_baseline.txt	This outputs column 1 for the baseline steady state results
		2200–2203	cf_results_trans_-baseline.txt	This generates the transition aggregates for the baseline in column 1
	\Model\Static_asymmetric.m	125–127	cf_results_static_asymmetric.txt	This generates the static model column results
	\Model \FEIM.f90	12,085–12,102	cf_results_ss_FEIM.txt	This outputs column 3 for the FEIM steady state results

		2160–2163	cf_results_trans_FEIM.txt	This generates the transition aggregates for the baseline in column 3
Table 7	No program	N/A	No data	N/A
Table 8	\Model\closed_economy.f90	509–519	parameters_closed_economy.txt	This outputs the parameter values for the closed economy model
		521–530	moments_closed_economy.txt	This outputs the moments for the closed economy model
Table 9	\Model\closed_economy.f90	237–247	cf_results_closed_economy.txt	This outputs the simulation results table for the closed economy model
Table 10	\Model \FEIM.f90	945–961	parameters_inside_FEIM.txt	This outputs the parameters for the FEIM on the table
	\Model \FEIM.f90	981–995	moments_FEIM.txt	This outputs the moments for the FEIM on the table
Table 11	\Model \TCJA.f90	9118–9134	cf_results_ss_TCJA.txt	This outputs the full TCJA steady state results
Table 12	\Empirical \segments.do	261–284	reg_table_out1.xls, reg_table_out2.xls	This runs and outputs the regression diff-in-diff results/placebo results

Figure 3	\Empirical \segments.do	288–301	trends.eps	This runs the dynamic regression and then saves the coefficients, plots and outputs the figure
Table 13	\Model\emp_validation.f90	944–962	parameters_inside_emp_validation.txt	This outputs the parameters used in the solution of the empirical validation exercise
		981–996	moments_emp_validation.txt	This outputs the moments for the empirical validation
	\Empirical \fundamentals_shortseries.do	262–267	compustat_moments_shortseries.xls	This do file has mostly the same structure as \Empirical \fundamentals.do
Table 14	\Model\Static_asymmetric.m	76–78	parameters_static_asymmetric.txt	This outputs the static model parameters used in the solution of the model
		911–913	moments_static_asymmetric.txt	This outputs the static model moments
Table 15	\Model\Static_deferrability.m	112–114	cf_results_static_deferrability.txt	This outputs the static deferrability exercise results
Table 16	\Model\Static_symmetric.m	119–121	cf_results_static_symmetric.txt	This outputs the symmetric static model results table

I List of In-Text Numbers

The following table describes how to generate the numbers mentioned in the text of the paper but not reported in tables or figures.

Page & Quote	**Program**	**Line**	**Output File**	**Note**
P2, "...24% excess capacity..."	\Model \baseline.f90	12,112–12,121	slackness_baseline.txt	This outputs the slackness in a firm's collateral constraint reported in the introduction. Numbers for all three statuses both pre and post reform are outputted here
P3, "...36% in 2018..."	No program	N/A	BEA_fdi-flows.xls	This gives the FDI outflows change figure in the introduction. Cell F8 is the number reported on the spreadsheet called "Real"
P26, "...134,992 firm-year..."	\Empirical \fundamentals.do	527–534	compustat_noobs_eqiss.xls	This draws on the *foreqiss.dta* file that was saved earlier in the do file after the data were cleaned. It drops observations prior to 1980 and subsequent to 2005.

P29, "...86% U.S. firms..."	\Empirical \segments.do	251–258	frac_us.xls	This outputs the distribution of U.S. and foreign firms in the segment information prior to running the regressions.
P29, "...1.8% drop in..."	\Model\emp_validation.f90	9119–9128	cf_results_ss_emp_validation.txt	This outputs the empirical validation X fraction change. Also reports the (D,X) transition and (X,X) transition that are mentioned
P34, "...by 4%..."	\Model\Static_asymmetric.m	130–135	sizes_static_asymmetric.txt	This reports the average sizes of firms in the static model pre and post reform
P35, "...tax of 41%."	\Model\Static_asymmetric.m	143–181	opt_tau_results_static_asymmetric.txt	This searches for the optimal static repatriation tax and then reports the numbers from moving to it from the observed rate

P35, "...is around 5%"	\Model \optimal_tau.f90	9140–9144, 9165–9173	welfare_ss_opt_tau.txt, cf_results_ss_opt_tau.txt	This code searches for the optimal tax in the baseline steady state. The first set of lines outputs a file that contains consumption/welfare for each run repatriation tax rate. The second set of lines output the variable changes from moving to the optimal rate from the observed rate
P37, "...average foreign affiliate sales intensity..."	\Model \baseline.f90	12,123–12,126	Fint_baseline.txt	This outputs the pre and post reform foreign affiliate sales intensity from the baseline
	\Model \FEIM.f90	12,104–12,107	Fint_FEIM.txt	This outputs the pre and post reform foreign affiliate sales intensity from the FEIM
P38, "...from 8.9% to..."	\Model \baseline.f90	12,112–12,121	slackness_baseline.txt	This outputs the change in constraint slackness across statues
P73, "...comes to be 0.24%..."	\Model \baseline.f90	12,128–12,133	effrate_baseline.txt	This outputs the closed economy pre-reform rate and the closed economy post-reform rate, both based on the moments from the baseline model

P86, "...is around 1%"	\Model \baseline.f90	1050–1052	revshare_ baseline.txt	This outputs the fraction of pre-reform tax collections that were coming from repatriation taxes in the baseline for use in the deferrability exercise

References

Fillat, J. L., & Garetto, S. (2015). Risk, returns, and multinational production. *The Quarterly Journal of Economics*, *130*(4), 2027–2073.

Floden, M. (2008). A note on the accuracy of markov-chain approximations to highly persistent ar(1) processes. *Economics Letters*, *99*(3), 516-520.

Goffe, W. L. (1996). Simann: A global optimization algorithm using simulated annealing. *Studies in Nonlinear Dynamics & Econometrics*, *1*(3).

Olley, G. S., & Pakes, A. (1996). The dynamics of productivity in the telecommunications equipment industry. *Econometrica*, *64*(6), 1263.