README for code associated with calibration, plotting of output, and robustness results of Equilibrium Technology Diffusion, Trade, and Growth by Perla, Tonetti, and Waugh (AER 2020)

All in matlab or python with the associated jupyter notebooks.

Create Calibration Moments

The notebook `../../section_7-1.ipynb` pulls data and computes the empirical moments used in calibration. Most data in that notebook is pulled directly from public online sources like FRED or the BLS. Some data is from the Synthetic Longitudinal Business Database. Data used in the paper was current from these sources as of April 5, 2020. The notebook `../../section_7-1.ipynb` outputs the following `.csv` files to the `src/calibration/data/` folder, which are then used as inputs into the calibration code:

- `data/growth_and_r_moments.csv` are the measured productivity growth and real interest generate
- `data/firm_moments.csv` are the firm transition dynamic moments
- `data/bejk_moments.csv` are the BEJK exporter and relative size moments
- `data/trade_moments.csv` is the trade share moment
- `data/entry_moment.csv` is the entry moment

Master File

The master file `master_calibration.m` produces all calibration and robustness results. In MATLAB, just run

```
>> master_calibration
```

This file runs all other files, prints the results, and generates the appropriate output files used as input when creating figures and numbers that appear in the paper.

Structure of Main Calibration File `calibrate_wrap.m`

The file `calibrate_wrap.m` calibrates the model using code that computes the BGP (balanced growth path) equilibrium. Transition dynamics are computed in Julia and described here.

The file `calibrate_wrap.m` does the following:

1. Reads in the moments that are generated from the python notebook described above in create calibration moments

2. Minimizes the distance between the data moments and model moments. The code that computes the model moments via simulation is organized in the following way:

- `calibrate_growth.m` organizes parameters, passes them to the function to retrieve model moments, then constructs the objective function the calibration is minimizing. The moments are stacked in the following order:
    - g, lambda_{ii}, fraction of exporters, size of exporters vs non exporters, and then the quantile moments (note how these are passed through) into vector form
    - Then line 47 describes the objective function used. This is just the standard GMM quadratic objective with an identity weighting matrix
- `compute_growth_fun_cal.m` takes in parameters, computes the BGP equilibrium, constructs the moments from simulated data, and calculates welfare. The following functions perform these operations:
- `eq_functions/calculate_equilibrium.m` computes the BGP equilibrium
- `eq_functions/selection_gbm_constant_Theta_functions.m` are the equilibrium relationships that map to equations in the paper
- `static_firm_moments.m` Computes the BEJK moments, fraction of exporters, and the relative size of domestic shipments for exporters vs non-exporters
- `markov_chain/generate_transition_matrix.m` constructs the transition matrix across productivity states. **Note** that the code `markov_chain/generate_stencils.m` is required to generate the grid. This is called before the calibration routine since it is kept constant
- `markov_chain/coarsen_to_quintiles_four.m` maps the transition matrix into a four by four matrix over a 5 year horizon to match up with data moments

3. Given the calibration results, the following output is created:

- A file `/parameters/calibration_params.csv` is created with the calibrated parameters
- An associated `.mat` file `cal_params.mat` is generated for use within matlab. The variable `new_cal` has the values in the in the following order: d, theta, kappa, 1/chi, mu, upsilon, sigma
- Comparisons of BGP equilibria associated with different parameter values are presented. To compute BGP results for different parameter values the code `compute_growth_fun_cal` is called with the appropriately changed parameter values

---

Robustness Exercises

In the paper Figures 6 and 7 display how changes to the GBM parameters μ and υ and to the BGP entry rate δ affect key outcomes.

- **Changes in δ** The code that creates the underlying data for Figure 7 is:

    - `robust_no_recalibrate_delta.m`. This reads in `cal_params.mat` generated from the calibration routine and then works through a several loops. First it changes χ by a scale value, then for the given χ it changes delta. The main file it calls is `compute_growth_fun_cal.m`. The output are the following `.mat` files:

    - `output/robust/delta/param_values_delta_xx` where `xx` is the scale value used for the given run of different δs. Each row shows the growth rate, baseline parameter values, then μ, υ, σ, and δ for that given run

    - `output/robust/delta/norecalibrate_values_delta_xx` where `xx` is the scale value for the given run. Each row shows the growth rate, counterfactual growth rate, difference, welfare gain, and the associated δ

- **Changes in μ and υ** The code that creates the underlying data for Figure 6 is:

    - `robust_no_recalibrate_gbm.m`. This reads in `cal_params.mat` generated from the calibration routine and then works through a several loops. First it changes χ by a scale value, then for the given χ it changes μ and υ by a common scale value. The main file it calls is `compute_growth_fun_cal.m`. The output are the following `.mat` files:

    - `output/robust/gbm/param_values_gbm_xx.mat` where `xx` is the scale value used for the given run of different δs. Each row shows the growth rate, baseline parameter values, then μ, υ, σ, and δ for that given run

    - `output/robust/gbm/norecalibrate_values_gbm_xx.mat` where `xx` is the scale value for the given run. Each row shows the growth rate, counterfactual growth rate, difference, welfare gain, and the associated δ

    - `/parameters/calibration_chi_xx.csv`, where `xx` is the value for χ at which the code is run. This file computes the parameter values which, for the given χ, result in a growth rate which is closest to the targeted baseline growth rate of 0.79.

- **Robustness to changes in scaling parameter ζ** This code is exactly the same as `calibrate_wrap.m` with the only difference being that alternative ζs are fixed and a loop over different calibrations is performed for the different ζs. The output is:

    - `/parameters/calibration_zeta_xx.csv` where `xx` is the ζ value for the given run and each `.mat` files contains the parameter values

- **Comparison to Sampson (2016)** This code recalibrates the model to target moments related to Sampson (2016 QJE), in part by setting GBM parameters to zero. The code `calibrate_wrap_sampson.m` performs the calibration. The underlying code and structure is exactly the same as in the main calibration code `calibrate_wrap.m`.

    - `/parameters/calibration_sampson.csv` reports the resulting parameter values

- **No Firm Dynamics** This code recalibrates the model with GBM parameters set to zero. The code `calibrate_wrap_no_firm_dynamics.m` performs the calibration. The underlying code and structure is exactly the same as in the main calibration code `calibrate_wrap.m`.

    - `/parameters/calibration_no_firm_dynamics.csv` reports the resulting parameter values

- **Comparision to Atkeson and Burstein (2010)** The base of this code was downloaded from Ariel Burstein's website at: http://www.econ.ucla.edu/arielb/innovcodes.zip. Starting from that code we set the discount rate, elasticity of substitution, the exit rate of firms, and the aggregate import share to be the same as in our baseline calibration. The code, located in `/src/calibration/ABmodel/Master_ptw.m`, runs the routine and returns the welfare gains from the same reduction in trade costs as studied in the baseline exercise of PTW, inclusive of the transition path.