1 Calibration

The main structure of the code follows these steps:

1. Start with a set of parameters, define targets.
2. Compute the steady state as the terminal condition.
3. Solve backwards the path from the steady state to 1975.
4. Compute model moments and compare to targets. Update.

These steps are calculated by the following matlab scripts:

1. `Calibration_run.m`
   - This is the code starting the minimization over parameters after defining the targets.
   - All parameters converted to an interval...
2. `Equilibrium.m`
   - The minimization on parameter values uses this one. This code takes the parameters.
   - Computes some relevant variables. Computes some fundamental matrices that define
     - firm position transitions as a result of R&D process
     - transitions for qualities and leadership shares
   - Then it calls `InterestPath.m.`
3. `InterestPath.m`
   - The first thing this one does is calling `Steady.m.`
   - The interest rate guess to be fed into `Steady.m` is read from file if it exists and worked the previous run.
   - `Steady.m` is called several times initialized at different starting points in case convergence not achieved.
4. `Steady.m`
   - Computes the steady state. Spits into `InterestPath.m.`
   - First loads value function solutions from the previous iteration. Uses these as starting point.
   - Checks if, in the previous iteration, guess and the result for steady state values were close enough.
     - If yes, uses a non-linear equation solver built-in in Matlab.
     - If no, uses an iterative user written code `Steady_set_iter.m.`
   - If there is no convergence in first attempt, tries again with non-linear equation solver.
5. `InterestPath.m`
   - Taking the steady state values, this code uses them as the terminal condition to calculate the transition path of the economy. It basically converges on the interest rate path. Provides the sequences of interest rates, average qualities, shares.
   - Starting from terminal condition, first solves R&D decisions at t given values at $t+1$.

- Updates values at t given values at $t+1$ and the implied R&D decisions at t until $t = t_0$.
- Using the series computes the growth sequence and the implied interest rates.
- Compares to the initial interest rate series, repeates until two series are close enough.

6. `Equilibrium.m`
    - Using the time paths of certain variables obtained from `InterestPath.m`, it computes aggregate variables and model analogues of targets. Gives the objective function value to the minimization procedure.
7. `Calibration_run.m`
    - When minimization ends, it provides the minimizing final parameter set.

2 Exercises (Postestimation)

Main files that compute welfare changes and optimal policies are

1. `PostEstimation_optimal_rnd_over_time.m` computes optimal subsidies given tariff rate
2. `PostEstimation_optimal_tariff_over_time.m` computes optimal tariffs given subsidy rate
3. `PostEstimation_optimal_combined_over_time.m` computes joint policy over time
4. `PostEstimation_optimal_over_open.m` computes optimal subsidy over different bilateral tariffs
5. `PostEstimation_trade_elasticity.m` computes trade elasticity in BGP
6. `PostEstimation_trade_spillover.m` computes welfare under different spillover schemes (spillover exercise)

They run on the set of parameters and variables that are computed at the calibrated point and stored to file `Set_postestimation_use.mat` inside `Equilibrium.m` in the calibration folder. The main files call a number of functions that resemble mostly the structure in the calibration workflow:

- `Steady.m`, `Steady_set.m`, `Steady_set_iter.m` are as defined in the calibration procedure.
- `Path_ctrf.m` works similar to `Interest_path.m` defined in the calibration procedure.
- `Trade_ctrf.m` (and `Trade_ctrf_spill.m`) constructs useful variables and matrices used frequently in the computations.
- `postestimation_welfare_trade.m` computes welfare changes.