Code for "Innovation, Reallocation and Growth"

Daron Acemoglu Ufuk Akcigit Harun Alp
Nicholas Bloom William Kerr

May 14, 2018

This file contains an overview of the code for replicating the results. It is implemented in `MATLAB`. There are 5 folders which refer to different versions of the model. Each folder includes a `MASTER.m` file which produces the results for the corresponding model. Final output is compiled, as they appear in the paper, under the subfolder `output`. A more detailed description of the code is given below.

Parameters

There are 13 parameters of the model, defined in the code as follows:

lam:	Innovation step size
psi:	Exogenous destruction rate
nu:	Transition rate from high-type to low-type
alpha:	Probability of being high-type entrant
phi:	Fixed cost of operation
theta_l:	Innovative capacity of low-type firms
theta_h:	Innovative capacity of high-type firms
theta_e:	Innovative capacity of entrants
eps:	CES parameter
disc:	Discount rate
gamma:	Innovation elasticity for incumbents
sigma:	Inverse of the intertemporal elasticity of substitution
Ls:	Measure of high-skilled workers

Parameters are read from a text file under the subfolder `params`. Global structure ***p*** keeps parameter values currently being used, as well as policy values.

Equilibrium solver

Equilibrium of the model can be characterized as a solution to a system of 6 equations in 6 unknowns: **(i)** wage rate [***ws***], **(ii)** mass of active product lines owned by low type firms [***cactiv(1)***], **(iii)** mass of active product lines owned by high type firms [***cactiv(2)***], **(iv)** expected value to a low type firm of a newly innovated product line [***eyq(1)***], **(v)** expected value to a high type firm of a newly innovated product line [***eyq(2)***] and **(vi)** average quality [***qbar***]. Global structure ***eq*** stores all the relevant equilibrium objects and passes them over to different routines. Initial guess for the solution is logged under subfolder `eqvars`.

The following are the most important files for solving the equilibrium:

- `initalg.m`**:** It creates the global structure ***alg*** that contains tuning parameters, binary switches and file names.

- `solver.m`: It loads in model parameters, and initial guess for equilibrium variables and run `fsolve` on the equilibrium function in `eqfunc.m`.
- `eqfunc.m`: It takes in guessed equilibrium variables and returns the equation errors which are calculated based on the following routines:
  - `innovation.m`: It finds innovation rates ($\boldsymbol{x}$) and the minimum quality of a product line for each type of firms ($\boldsymbol{qmin}$).
  - `qualityDist.m`: It solves the quality distribution and calculates the updated mass of active product lines.
  - `qbarActive.m`: It computes the average quality of active product lines.
  - `calcey.m`: It uses quality distribution to find updated expected product line values.
  - `labordem.m`: Using innovation rates and wage rate, it finds labor demand.

Estimation

Estimation routine is implemented in `smm.m`. This routine searches over the parameter space to minimize the distance between simulated and data moments. The objective function for the estimation is `smmobj.m` which takes the proposed parameter values, solves the model, simulates a panel of firms and calculates various moments of interest. Firm simulation and moment calculations are done in `compMoments.m`, which itself calls the mex file `firmsim.mexmaci64`. This file is compiled from the source file `firmsim.cpp` (in `C++`, located under subfolder `mex`) for a Mac (64-bit) machine. The source file is needed to be recompiled for any other platform. To speed up the simulation, parallelization option is available, which can be controlled by the parameters in `initalg.m`. Finally, `bootstrapSD.m` computes standard errors for the estimated parameters based on bootstrap.

Optimal Policies

The social planners problem is solved in `socplan_opt.m`, which uses `socplan_solver.m` and `socplan_eqfunc.m`. For the baseline model, social planner's choice variables are: (i) minimum quality for low type firms [$\boldsymbol{qmin(1)}$], (ii) minimum quality for high type firms [$\boldsymbol{qmin(2)}$], (iii) innovation rate for low type firms [$\boldsymbol{x(1)}$], and (iv) innovation rate for high type firms [$\boldsymbol{x(2)}$]. `run_pols.m` finds subsidy rate for different policies that corresponds to 1% of GDP. Finally, optimal subsidy policy is run through `policy_opt.m`. It searches over subsidy rates to maximize welfare.