Algorithm notes

Nicolas Fernandez-Arias

May 16, 2018

1 High level overview

1. Guess L^{RD}
2. Compute $\bar{w}, \pi$ (and L^F, L^I) for $\bar{q} = 1$ using static production firm optimization
3. Solve HJBs to compute incumbent and entrant innovation effort.
4. Solve KF equation to compute stationary distribution over m
5. Check that policy functions aggregate to $\tilde{L} = L^{RD}$.
   - If $\tilde{L} > L^{RD}$, guess higher L^{RD}.
   - If $\tilde{L} < L^{RD}$, guess lower L^{RD}.
   - Convergence: viewing $\tilde{L}$ as a function of the guess L^{RD}, we see that $\tilde{L}(0) > 0$, $\tilde{L}(1) = 0$ (since static profits are zero in that case), $\tilde{L}$ is monotonic and continuous.[1]

2 Details

The difficult part of the algorithm is to find wages such that the policies and HJBs are consistent with wages. Define

$$\tau(m) = \chi_S z_S(m)\eta(z_S(m) + z_E(m)) + \chi_E z_E(m)\eta(z_S(m) + z_E(m))$$

This consists of finding $V(\cdot), z_I(\cdot), W(\cdot), w(\cdot), \bar{z}, M_s$ such that

[1] Haven't shown continuity but probably holds.

1. $V(\cdot), z_I(\cdot)$ solves incumbent HJB with boundary condition $V'(M_s) = 0$ and $z_S(m) = \xi \min(m, M_s)$ and $z_E(m) = \max(0, \bar{z} - z_S(m))$,

$$(\rho + \tau(m))V(m) = \pi + \nu(z_S(m) + z_E(m))V'(m) + \max_{z_I \geq 0} \left\{ z_I \Big(\chi_I \phi(z_I) \Big(\lambda V(0) - V(m) \Big) - (w(m) + V'(m)) \Big) \right\}$$

2. $W(\cdot)$ solves potential spinout HJB with policy $z^*(m) = z_S(m) = \xi \min(m, M_s)$ and boundary condition $W(M_s) = 0$ and the same assumptions about aggregate innovation effort z_S, z_E as in 1,

$$(\rho + \chi_I z_I(m)\phi(z_I(m)) + \tau(m))W(m) = \nu(z_S(m) + z_E(m) + z_I(m))W'(m) + \max_{0 \leq z \leq \xi} \left\{ z \Big(\chi_S \eta(z_S(m) + z_E(m))\lambda V(0) - w(m) \Big) \right\}$$

(Note that (1) spinout R&D is individually CRS, (2) spinouts do not take into account their own effect on the rate at which they are displaced $\tau(m)$, so there is no $-W(m)$ term in the Poisson arrival term, and (3) the above implies that $z_S = \xi$ whenever $\chi_S \eta(z_s(m) + z_E(m))\lambda V(0) > w(m)$)

3. Non-spinout free entry: $\chi_E \eta(\bar{z}) V(0) = \bar{w}$

4. Spinout free entry: $\chi_S \eta(\xi M_s) V(0) = \bar{w}$

5. Worker indifference condition: $w(m) = \bar{w} - \nu W(m)$

2.1 Brute force approach

1. Guess wage $w(\cdot)$

2. Iterate on aggregate spinout effort $z_S(\cdot)$ and $\bar{z}$ to find equilibrium of game which does not impose the indifference condition. Interpretation: partial equilibrium, without considering labor supply decisions.

3. Iterate on $w(\cdot)$ until we find one such that $W(\cdot)$ from previous step that is consistent with indifference condition

2.1.1 Implementation

How do we implement the second step above? Given our initial guesses $z_S(\cdot)$ and $\bar{z}$, we can define the boundary conditions for the HJBs simply by setting $V'(\bar{m}) = W(\bar{m}) = 0$

and simply integrating backwards the HJBs. This returns two objects which can be checked against the guesses for convergence / used to update the guess for the next iteration:

1. An optimal policy for R&D effort of spinouts, which can be compared to the original guess $z_S(\cdot)$. We pick a new guess that is the average of the two.

2. We can check the free entry condition $\chi_E \eta(\bar{z}) V(0) = \bar{w}$. If LHS > RHS, $\bar{z}$ is too low, and we should increase our guess.

The idea is that eventually this finds $z_S(\cdot)$ and $\bar{z}$ that are consistent with individual optimality, given the partial equilibrium R&D wage guess $w(\cdot)$.

2.1.2 Implementing the Implicit Method

Incumbents Have

$$\begin{aligned}\frac{1}{\Delta_t}\Big(V_i^{n+1} - V_i^n\Big) + \rho V_i^{n+1} &= \pi - \tau_i V_i^{n+1} + \nu(z_i^S + z_i^E)\frac{V_{i+1}^{n+1} - V_i^{n+1}}{\Delta_m} \\ &\quad + z_i^{I,n}\Big(\chi_I \phi(z_i^{I,n})\Big(\lambda V_1^{n+1} - V_i^{n+1}\Big) - \Big(w_i - \nu\frac{V_{i+1}^{n+1} - V_i^{n+1}}{\Delta_m}\Big)\Big) \\ z_i^{I,n} &= \arg\max_{z \geq 0}\Big\{z\Big(\chi_I \phi(z)\Big(\lambda V_1^n - V_i^n\Big) - \Big(w_i - \nu\frac{V_{i+1}^n - V_i^n}{\Delta_m}\Big)\Big)\Big\}\end{aligned}$$

In matrix form, have

$$\frac{1}{\Delta_t}(V^{n+1} - V^n) + \rho V^{n+1} = u^n + A^n V^{n+1}$$

where u^n is an I_m-dimensional column vector, such that $u_i^n = \pi - z_i^{I,n} w_i$, and such that A^n is all zeros except for the following. For $i < I_m$,

$$\begin{aligned}A_{i,1} &= z_i^{I,n}\chi_I \phi(z_i^{I,n})\lambda \\ A_{i,i+1} &= \frac{\nu(z_i^S + z_i^E + z_i^{I,n})}{\Delta_m} - \tau_i \\ A_{i,i} &= -\frac{\nu(z_i^S + z_i^E + z_i^{I,n})}{\Delta_m} - z_i^{I,n}\chi_I \phi(z_i^{I,n})\end{aligned}$$

For $i = I_m$, we simply assume that $V'(m) = 0$ for $m > m_i$. Hence we can omit the terms corresponding to derivatives. Therefore, have

$$\begin{aligned}A_{i,1} &= z_i^{I,n}\chi_I \phi(z_i^{I,n})\lambda \\ A_{i,i} &= -z_i^{I,n}\chi_I \phi(z_i^{I,n})\end{aligned}$$

Given all this, we can write the update equation as

$$\begin{aligned} B^n V^{n+1} &= b^n \\ B^n &= \Big(\frac{1}{\Delta_t} + \rho\Big) I - A^n \\ b^n &= u^n + \frac{1}{\Delta_t} V^n \end{aligned}$$

which can then be solved efficiently using MATLAB's sparse matrix solving routines.

Spinouts Let $\sigma_i^n = \tau_i + z_i^{I,n} \chi_I \phi(z_i^{I,n})$. Have

$$\begin{aligned} \frac{1}{\Delta_t}\Big(W_i^{n+1} - W_i^n\Big) + \rho W_i^{n+1} &= -\sigma_i^n W_i^{n+1} + \nu(z_i^S + z_i^E)\frac{W_{i+1}^{n+1} - W_i^{n+1}}{\Delta_m} \\ &\quad + \tilde{z}_i^{S,n}\Big(\chi_S \eta(z_i^S + z_i^E)\lambda V_1^{n+1} - w_i\Big) \\ \tilde{z}_i^{S,n} &= \arg\max_{0 \le z \le \xi} \Big\{ z\Big(\chi_S \eta(z_i^S + z_i^E)\lambda V_1^n - w_i\Big)\Big\} \end{aligned}$$

In matrix form, this is

$$\frac{1}{\Delta_t}(W^{n+1} - W^n) + \rho W^{n+1} = u^n + A^n W^{n+1}$$

where u_n is an I_m-dimensional column vector with

$$u_n^i = \tilde{z}_i^{S,n}\Big(\chi_S \eta(z_i^S + z_i^E)\lambda V_i^{n+1} - w_i\Big)$$

2.1.3 Using equilibrium relationships

The goal would be to impose in the algorithm the fact that in equilibrium

$$\begin{aligned} z_S(m) &= \xi \min(m, M_s) \\ \chi_E \eta(\bar{z}) V(0) &= \bar{w} \\ \chi_S \eta(\xi M_s) V(0) &= \bar{w} \end{aligned}$$

But the problem is that for generic $w(m)$, it may not be the case that the optimum takes the form in the equilibrium. I could search for an optimum within the a certain class of policies...but it's not as clear to me that this would converge. It probably makes sense to do the brute force approach first, and switch to this if that doesn't work.

3 Model parameters

The model parameters are:

- Discount rate ρ
- Labor share (and elasticity of substitution) β
- Incumbent R&D productivity χ_I
- Entrant R&D productivity χ_E
- Spinout R&D productivity χ_S
- Incumbent R&D elasticity γ_I
- Entrant R&D elasticity γ_E
- Spinout R&D elasticity γ_S
- Incumbent quality step λ_I
- Entrant quality step λ_E
- Spinout quality step λ_S
- Learning rate ν
- Entrant size ξ

The baseline calibration will assume that incumbents and entrants have the same R&D technology, but draw from different sources of ideas. All entrants - spinouts or not - draw from the same source of ideas.

- $\chi_I = \chi_S$
- $\chi_E < \chi_S$
- $\chi_S < \chi_E$
- $\lambda_I = \lambda_E$

3.1 Algorithm parameters (grids, etc.)

In the original algorithm I had not realized I could get rid of a state variable by appropriately scaling the exogenous change in the R&D technology with the quality of the good. I now assume that the efficiency of the R&D technology is inversely proportional to the relative quality of the good being improved upon. For goods that are much higher than average quality, it costs more to improve them. This ensures that the R&D policy function of a firm does not depend on its relative quality q. This both simplifies the firm's problem (by eliminating a state variable) and makes the model actually have a BGP.[2]

[2]R&D policy functions vary only along the dimension (m) where there is a stationary distribution. There is no stationary distribution of relative qualities $q/\bar{q}$, but this is not a problem since R&D policy functions do not depend on this variable. Hence we do not need to know the distribution to aggregate.