Algorithm

Nicolas Fernandez-Arias

May 2, 2018

1 Outline

At the highest level, the algorithm for computing the equilibrium of the economy consists of the following iterative scheme.

1. Guess R&D labor allocation L^{RD}.
   (a) Using L^{RD}, compute equilibrium objects $L^F, L^I, \bar{w}, \pi$
2. Guess R&D wage $w(m)$
3. Solve for firm values V, W
   (a) Nash equilibrium of a game played between entrants and incumbents
   (b) Originally, plan was to iterate on $z^E(m)$ aggregate entrant innovation depending on state of good.
      - Guess $z^E(m)$.
      - Solve for V. How exactly?
   (c) Now I am thinking it would be better to iterate on M directly, assuming that $z^E(m)$ is constrained to be $\xi \min(m, M)$ (as it is in equilibrium)
      - Guess M
      - Solve incumbent HJB with boundary condition $V'(M) = 0$ (need endogenous grid)
      - Check if $V(0)$ satisfies free entry condition at M.
      - If more entry needed, raise M guess.
      - Continue until convergence

- Now have M and can compute W, assuming individual policies consistent with aggregate policy
- Finally, check worker indifference condition. I don't see why this wouldn't converge. Maybe try this first, I don't know.

(d) Another possibility that may work is to iterate on z^E and on M

- We know that there will be a free-entry mass in equilibrium
- So let's just guess zE, and then we'll use the boundary condition $V'(m_{max}) = 0$, which is just the end of the grid.
- No need for endogenous grid
- How do we check free entry? next step!

(e) **Solve HJB for W:**

- **Integrate backwards starting at boundary condition is $W(m_{max}) = 0$, imposing optimality**
- **Now we are truly solving a game between incumbents and entrants, and I have faith that it will converge. It will be fine. If it's too slow, try other stuff.**
- Do we use optimality or do we assume the policy that gives rise to the assumed aggregate policy?
- The problem with using optimality is that my guess of $w(m)$ is not an equilibrium guess. So, there is no guarantee that the policies that will arise will
- Suppose we do the latter. Then we get some W which is the result of following that strategy. And we keep doing this until we find an M such that

4. Check that $\bar{w} \approx w(m) + \nu W(m)$. If not, go back to step 2 with a new guess $\tilde{w}(m)$ in between $w(m)$ and $\bar{w} - \nu W(m)$.

5. Use policies to solve KF equation and get stationary distribution $\mu(m)$

6. Using policies and stationary distribution $\mu(m)$, compute R&D labor demand $\hat{L}^{RD}$. If off, return to Step 1 with a new guess for L^{RD}.

2 Static equilibrium objects

In equilibrium, it can be shown that:

$$L^F = \frac{\beta(1 - L^{RD})}{\beta + (1-\beta)^2} \tag{1}$$

$$L^I = \frac{(1-\beta)^2}{\beta} L^F \tag{2}$$

$$\bar{w} = \beta^\beta (1-\beta)^{2-2\beta} \tag{3}$$

$$\pi = \beta(1-\beta)^{\frac{2-\beta}{\beta}} \left(\frac{\bar{q}}{w}\right) L_F q_j \tag{4}$$

3 Finite difference solution of HJBs

3.1 Incumbent HJB

The incumbent's HJB is:

$$\begin{aligned}(\rho + \hat{\chi}\xi m \eta(\xi m)) V(m) = \pi &+ \nu \xi m V'(m) \\ &+ \max_z \Big\{ \chi z \phi(z) \Big(\lambda V(0) - V(m) \Big) \\ &- z(w(m) - \nu V'(m)) \Big\}\end{aligned} \tag{5}$$

with boundary condition $V'(m_{max}) = 0$.

Numerical solution

1. Take as given guess zE
2. Integrate backwards HJB

Solve numerically by finite difference. Replace $V(m)$ with V_i and $V'(m)$ with the forward approximation, $\frac{V_{i+1} - V_i}{\Delta_i^m}$.

$$\begin{aligned}(\rho + \hat{\chi}\xi m_i \eta(\xi m_I)) V_i = \pi &+ \nu \xi m_i \frac{V_{i+1} - V_i}{\Delta_i^m} \\ &+ \max_z \Big\{ \chi z \phi(z) \Big(\lambda V_1 - V_i \Big) \\ &- z(w_i - \nu \frac{V_{i+1} - V_i}{\Delta_i^m}) \Big\}\end{aligned} \tag{6}$$

Rearrange to obtain an expression for V_i in terms of V_{i+1}.

3.2 Entrant HJB

Taking as given the incumbent's policy z^I, the entrant solves

$$\begin{aligned}(\rho + \hat{\chi}\xi m\eta(\xi m) + \chi z^I(m)\phi(z^I(m)))W(m) = \nu(\xi m + z^I)V'(m) \\ + \max_{z\leq\xi}\Big\{\hat{\chi}z\eta(\xi m)\Big(\lambda V(0) - W(m)\Big) \\ - zw(m)\Big\}\end{aligned} \tag{7}$$

Numerical solution

1. Have M from solution to incumbent HJB
2.

We discretize this as

$$\begin{aligned}(\rho + \hat{\chi}\xi m\eta(\xi m) + \chi z^I(m)\phi(z_i^I))W_i = \nu(\xi m + z^I)\frac{W_{i+1} - W_i}{\Delta_i^m} \\ + \max_{z\leq\xi}\Big\{\hat{\chi}z\eta(\xi m)\Big(\lambda V_0 - W_i\Big) \\ - zw_i\Big\}\end{aligned} \tag{8}$$

4 Computing the stationary distribution

4.1 Simulation

4.2 Kolmogorov Forward Equations

The Kolmogorov Forward equation is (need to prove)

$$\begin{aligned}0 = -\partial_q\Big(a^q(q,m)\mu(q,m)\Big) - \partial_m\Big(a^m(q,m)\mu(q,m)\Big) \\ - s(q,m)\mu(q,m) \\ + \mathbb{1}_{\{m=0\}}(1+\lambda)^{-1}\int_0^\infty s((1+\lambda)^{-1}q, m')\mu(q,m')dm'\end{aligned}$$

where $a^q(q,m), a^m(q,m)$ is the drift in the q,m direction, respectively, and $s(q,m)$ is the aggregate innovation rate for a product in state (q,m). Have

$$\begin{aligned}
a^q(q,m) &= -gq \\
a^m(q,m) &= \nu(z^E(q,m) + (1 - x(q,m)z^I(q,m))) \\
s(q,m) &= (\chi_I z^I(q,m) + \chi_E z^E(q,m))\phi(\tau(q,m)) \\
\zeta(q,m) &= z^E(q,m) + (1 - x(q,m))z^I(q,m)
\end{aligned}$$

Following Moll's HACT numerical appendix, discretize the HJB as

$$\begin{aligned}
0 = &-\frac{-gq_{i+1}\mu_{i+1,j} + gq_i\mu_{i,j}}{\Delta^q_{i+1}} - \frac{\nu\zeta_{i,j-1}\mu_{i,j-1} - \nu\zeta_{i,j}\mu_{i,j}}{\Delta^m_{i-1}} \\
&- s_{i,j}\mu_{i,j} + \mathbb{1}_{\{j=1\}} \sum_{j'=1}^{J} s_{i-m,j'}\mu_{i-m,j'}
\end{aligned}$$

Finally, collecting terms, and using $\zeta_{i,j} = z^E_{i,j} + (1 - x_{i,j})z^I_{i,j}$, have

$$\begin{aligned}
0 = &-\nu\zeta_{i,j-1}\mu_{i,j-1} + gq_{i+1}\mu_{i+1,j} \\
&+ (-gq_i + \nu\zeta_{i,j} - s_{i,j})\mu_{i,j} \\
&+ \mathbb{1}_{\{j=1\}} \sum_{j'=1}^{J} s_{i-m,j'}\mu_{i-m,j'}
\end{aligned}$$

Following Moll, when the subscripts are not in the state space, $\mu = 0$ so we just ignore those terms.

To put these equations in matrix form, we cannot use the matrix A as in Moll because $(V^{n+1})^+_{i,j}$ is computed by extrapolation. But I can use the matrix C^n defined by removing the contribution from the extrapolation. We can compute C in the same way as A but setting

$$\left(V^{n+1}\right)^+_{i,j} = \begin{cases} V^{n+1}_{i+m,0} & i < I - (m-1) \\ 0 & i >= I - (m-1) \end{cases}$$

As argued in Moll, if we use the matrix C from above, the stationary distribution is given by the solution to the eigenvalue problem

$$C^T\mu = 0$$

In Moll, the eigenvalue is normalized so that it sums to one. We are ignoring some points that are actually in the state space. We can, in theory, compute this mass as well, simply

assuming that

5 Update rules for $g, L^F, w(q,m), z^E(q,m)$

5.1 Rule for $z^E(q,m)$

Naive approach is to simply set

$$\tilde{z}^E(q,m) = \mathbb{1}_{z^e(q,m)>0}\xi m$$

and then update according to

$$z_1^E(q,m) = \alpha z_0^E(q,m) + (1-\alpha)\tilde{z}_0^E(q,m)$$

But this has problems converging, because the policy jumps around too much, leading to instability. Ideally one would want to vary the rate of updating in some way to compensate for this. I have tried one way, which works with $h(q) = 1$ but has had some issues for non-trivial $h(q)$.