Homework 2: Solving Steady State Khan and Thomas
Due On February 8th, 5:00pm

The goal of this homework is to introduce you to numerically solving heterogeneous firm models. We will analyze a simple steady state version of Khan and Thomas' (2008) model with idiosyncratic shocks and fixed capital adjustment costs. You are free to use any numerical package you like for this problem set, but I will write the problem set under the assumption you are using `Matlab`.

The model is as follows. There is a fixed mass 1 of firms $j \in [0, 1]$. Each firm has access to a decreasing returns to scale production function $y_{jt} = \varepsilon_{jt} k_{jt}^{\theta} n_{jt}^{\nu}$, where y_{jt} is output, ε_{jt} is idiosyncratic productivity, k_{jt} is the firm's capital stock, n_{jt} is the firm's labor input, and $\theta + \nu < 1$. Firms accumulate capital according to the accumulation equation $k_{jt+1} = (1 - \delta)k_{jt} + i_{jt}$. If $|\frac{i_{jt}}{k_{jt}}| > 0.05$, the firm must pay a fixed cost ξ in units of labor. There is a representative household with preferences over consumption C_t and labor supply N_t represented by the expected utility function $\mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t (\log C_t - aN_t)$. Output can be used for consumption or investment, so $Y_t = C_t + I_t$.

For numerical simplicity we assume that idiosyncratic shocks follow a simple two-state Markov process. Assume that $\varepsilon \in \{\varepsilon_L, \varepsilon_H\}$ where $\varepsilon_L = 0.8$ and $\varepsilon_H = 1.2$. Assume that $\Pr(\varepsilon' = \varepsilon_H | \varepsilon = \varepsilon_L) = 0.25$ and that $\Pr(\varepsilon' = \varepsilon_L | \varepsilon = \varepsilon_H) = 0.25$. The remaining parameters are $\theta = 0.21$, $\nu = 0.64$, $\delta = 0.1$, $\beta = 0.96$, and $\xi = 0.01$.

1. **Solve Representative Agent Steady State** It will be convenient to compute the steady state equilibrium of the model in which there is a representative firm with the same production function as the heterogeneous firms and productivity equal to the mean value of ε according to its stationary distribution. Assume the representative firm rents capital and labor in competitive input markets as in the RBC model from Topic 0. Compute the steady state capital stock K^*_{rep} and steady state wage w^*_{rep} of the representative agent model. Assume that steady state labor supply is $N^*_{\text{rep}} = 0.6$. (Hint: use the first order conditions for the firm's input decision and plug in $N^*_{\text{rep}} = 0.6$. You do not need to know the value of the parameter a.)

2. **Solve Individual Decision Problem** In the steady state recursive competitive equilibrium of the full model, the Bellman equation describing firm optimization is given by

$$v(\varepsilon, k) = \max_{k', n} \varepsilon k^{\theta} n^{\nu} - w^* n - (k' - (1 - \delta)k) - \mathbb{1}\{k' \neq (1 - \delta)k\}\xi w^* + \beta \mathbb{E}[v(\varepsilon', k')] \quad (1)$$

where w^* is the steady state real wage and $\mathbb{1}$ denotes an indicator function. In this question we will solve this dynamic programming problem using discretized value function

iteration. For now, set $w^* = w^*_{\text{rep}}$; in Part 3 we will solve for the true steady state wage that clears the labor market.

(a) We will approximate the true Bellman equation (1) using a technique called `discretized value function iteration`. In particular, we will assume that the capital stock k takes on values in a discrete grid $\mathbf{k} = (k_1, ..., k_N)$ where $k_i > k_{i-1}$ and N is the total number of grid points. Under this assumption, the value function $v(\varepsilon, k)$ can be represented by a $2 \times N$ matrix.

Create a $1 \times N$ vector $\mathbf{k}$ of grid points. Set $N = 200$. Assume that $k_1 = 0.5 \times k^*_{\text{rep}}$, $k_N = 2 \times k^*_{\text{rep}}$, and that the grid points are evenly spaced between the two. (When solving your own models, you should choose the ends of the grid space to be wide enough so that it captures all regions of the state space that are actually visited in equilibrium, but not so wide that you are wasting resources analyzing regions of the state space that are never visited in equilibrium. We will do this at the end of Part 3.)

It will be useful later on to construct the following two $2 \times N$ matrices. Create the matrix $\mathbf{E}$ which has ε_L in the first row, repeated in every column, and ε_H in the second row, repeated in every column. Create the matrix $\mathbf{K}$ which has k_i in the i^{th} column, repeated in every row. (Hint: in `Matlab`, use the `repmat` command.)

(b) We will compute the value function by iterating on the Bellman equation (1). To do so, it will be useful to construct a matrix $\mathbf{\Pi}$ whose (i, j) entry is

$$\pi(\varepsilon_i, k_j) = \max_n \varepsilon_i k_j^\theta n^\nu - w^* n. \tag{2}$$

To solve the maximization problem, compute the optimal choice of employment given (ε, k) by taking the first order condition and plug that in to objective function to get a closed-form expression for $\pi(\varepsilon_i, k_j)$.

You should construct the matrix Π using matrix operations – *do not use a for loop*! Loops in `Matlab` are much slower than the corresponding matrix operations. (Hint: use elementwise operations (e.g., `.*` or `./`) to construct the matrix. The matrices $\mathbf{E}$ and $\mathbf{K}$ will come in useful.)

(c) Next we will write a `Matlab` function which takes as inputs a matrix $\mathbf{V}^n$ and outputs two matrices: $\mathbf{V}^{n+1}$, the value function, and $\mathbf{G}^{n+1}$, the policy function.

i. Create the `.m` file for this function.

ii. In the `.m` file, create a $2 \times N \times N$ array whose $(i, j, k)^{th}$ element is equal to

$$\mathbf{\Pi}_{ij} - (k_k - (1-\delta)k_j) - \mathbb{1}\{k_k \neq (1-\delta)k_j\}\xi w^* + \beta \sum_{l=L,H} \Pr(\varepsilon = \varepsilon_l | \varepsilon = \varepsilon_i)\mathbf{V}^n_{lk}.$$

The first dimension of this array corresponds to current productivity ε, the second dimension to current capital k, and the third dimension to possible choices for future capital k'.

iii. Now we will solve the maximization problem at each point in the state space using `Matlab`'s `max` command. This gives the value function $v(\varepsilon, k)$ associated with the right hand side and the policy function $k' = h(\varepsilon, k)$. (Hint: Use the `reshape` command to transform $\mathbf{U}$ into a $2N \times N$ matrix in which the rows index the current state of ε and k pairs and the columns index the possible future choices k'. Then use the command `[V,I] = max(U,2)` to solve the maximization problem. Finally, reshape $\mathbf{V}$ and $\mathbf{I}$ into $2 \times N$ matrices. These correspond to the updated value function $\mathbf{V}^{n+1}$ and the indices of the optimal choices $\mathbf{H}^{n+1}$.)

(d) Now we will write a while loop which executes the value function iteration.

i. Initialize the matrix $\mathbf{V}^0 = \mathbf{\Pi}$ and `err` $= 100$.

ii. Using the `while` syntax, write a loop which at each iteration `n` computes $\mathbf{V}^{n+1}$ and $\mathbf{G}^{n+1}$. Terminate the loop when `err` $= \max |\mathbf{V}^{n+1} - \mathbf{V}^n| < 10^{-6}$.

(e) Finally, plot your results!

i. Plot $v(\varepsilon_L, k)$ and $v(\varepsilon_H, k)$ as a function of k on the same graph.

ii. Plot $k' = h(\varepsilon_L, k)$ and $k' = h(\varepsilon_H, k)$ as a function of k on the same graph.

3. **Compute stationary distribution** We will now compute the stationary distribution associated with the decision rules you just computed in Part 2. Following the discretization strategy above, we will approximate the stationary distribution with a probability mass function over the discretized state space $g(\varepsilon_i, k_j)$. This function must satisfy the law of motion

$$g(\varepsilon_k, k_l) = \sum_{i=L,H} \sum_{j=1}^{N} \Pr(\varepsilon' = \varepsilon_k | \varepsilon = \varepsilon_i) \mathbb{1}\{\mathbf{H}_{ij} = k_l\} g(\varepsilon_i, k_j), \tag{3}$$

where $\mathbf{H}$ is the policy rule from Part 1. Let $\mathbf{G}$ be a $2 \times N$ matrix with (i, j) entry $g(\varepsilon_i, k_j)$. Let $\mathbf{g}$ be the $2N \times 1$ vector representation of this matrix, i.e., $\mathbf{g} = \texttt{reshape}(\mathbf{G}, 2N, 1)$.

(a) We will pursue an iterative strategy to compute $\mathbf{g}$. A key input into this strategy will be a $2N \times 2N$ transition matrix $\mathbf{T}$ such that the stationary distribution $\mathbf{g}$ satisfies

$$\mathbf{g} = \mathbf{T}\mathbf{g}.$$

Hence, the entries in $\mathbf{T}$ correspond to the terms in the summation of 3. We will now construct this matrix.

i. First, compute a $2\times N\times 2\times N$ array whose (i,j,k,l) entry is an indicator variable for whether $\mathbf{H}_{k,l} = k_j$. Again, you should do this using matrix operations and logical indexing – avoid loops! (Hint: use `reshape` and `repmat` liberally if needed).

ii. Second, compute a $2\times N\times 2\times N$ array whose (i,j,k,l) entry is $\Pr(\varepsilon' = \varepsilon_i | \varepsilon = \varepsilon_k)$.

iii. Finally, reshape both of these arrays into $2N \times 2N$ matrices and multiply them together elementwise (i.e., using the `.*` command, not *). This is the transition matrix $\mathbf{T}$.

NB: Although the matrix $\mathbf{T}$ is conceptually straightforward, the matrix operations needed to construct it are quite complicated. This is a significant downside to using `Matlab`, which is optimized for matrix operations and extremely slow for loops. In a lower-level language like `Fortran` or `C`, you would be able to directly construct $\mathbf{T}$ using loops.

(b) Compute the stationary distribution $\mathbf{g}$ by iterating on

$$\mathbf{g}^{n+1} = \mathbf{T}\mathbf{g}^n.$$

Set the initial guess $\mathbf{g}^0$ to be a vector in which each element is equal to $\frac{1}{2N}$.

(c) Plot $g(\varepsilon_L, k)$ and $g(\varepsilon_H, k)$ as a function of k on the same graph. If there is a positive mass of firms at either end of the state space, this indicates the grid bounds are too narrow. If there is empty space at either end of the state space, this indicates that the grid bounds are too wide. Adjust the grid bounds appropriately.

4. **Compute Steady State Equilibrium** So far we have been using w^*_{rep} as a guess of the wage. We will now solve for the equilibrium value for the wage using a version of the Hopenhayn-Rogerson algorithm discussed in class. Let $N^{\text{d}}(w)$ denote aggregate labor demand in the stationary distribution given a wage w and let $N^{\text{s}}(w)$ denote aggregate labor supply from the household. The equilibrium wage must satisfy $N^{\text{d}}(w^*) = N^{\text{s}}(w^*)$.

As we discussed in the representative agent RBC model in Topic 0, a common way to calibrate business cycle models is to choose the disutility of labor supply, a, so that steady state equilibrium labor supply is $N^{\text{s}}(w^*) = 0.6$. This calibration strategy will simplify the computation of the equilibrium wage w^*. First, we will compute the wage w^* such that aggregate labor demand $N^{\text{d}}(w^*) = 0.6$. Second, given this value for the wage, we will use the household's first order condition to back out the parameter a which ensures $N^{\text{s}}(w^*) = 0.6$.

(a) Write a `Matlab` function that takes as input a guess of the steady state wage w and outputs the aggregate labor demand. This function should perform the following steps.

i. Given the wage w, compute the individual decision rules as in Part 1.

ii. Given the decision rules, compute the stationary distribution as in Part 2.

iii. Given the stationary distribution, compute aggregate labor demand. Let $n(\varepsilon_i, k_j)$ denote the labor demand of a firm with individual state $n(\varepsilon_i, k_j)$ (note that this is the sum of labor used in production as in 2 as well as the labor used to pay the fixed cost). With this notation, aggregate labor demand is $N = \sum_{i=L,H} \sum_{j=1}^{N} n(\varepsilon_i, k_j) g(\varepsilon_i, k_j)$.

(b) The steady state wage must satisfy the equation $N^{\mathrm{d}}(w^*) - 0.6 = 0$. Solve this equation. (Hint: use `Matlab`'s `fsolve` routine with w^*_{rep} as an initial guess). What is the equilibrium wage w^*, and how does it compare to w^*_{rep}?

(c) The household's first order condition for labor is $w^* C = a$, where C is aggregate consumption. Compute aggregate consumption using the resource constraint $C = Y - I$ and using the firm's output and investment decision integrated against the stationary distribution. What is the implied value of a?