Homework 4

(Due: February 26 by 5 pm)

Note: Study groups *discussing* the problems are strongly encouraged. But write your own answers.

1 Competitive Equilibrium in One-Sector Growth Model under Certainty

1.1 Competitive Equilibrium

Consider a decentralized economy with a large number of firms and a large number of households of total measure one.

The state of the economy is the aggregate capital stock. We distinguish between "big K" and "little k." Households and firms see big K as the aggregate state that is beyond their control. Little k are chosen by firms and households; upon optimization, they do not see the connection between K and k. In equilibrium, $K = k$.

We specify price functions $\tilde{r}(K)$ and $w(K)$ that represent, respectively, the rental price on capital and the wage rate when the aggregate state is K. The law of motion for the aggregate capital is denoted as $K' = G(K)$.

A firm maximizes its profit given the factor prices $(w_t, \tilde{r}_t) = (\tilde{r}(K_t), w(K_t))$:

$$\max_{k_t, n_t} Ak_t^\alpha n_t^{1-\alpha} - w(K_t)n_t - \tilde{r}(K_t)k_t,$$

where $Ak^\alpha n^{1-\alpha}$ is a production function, k_t is the capital input and n_t is the labor input. Note that the firm's first-order conditions and market clearing implies that the equilibrium pricing functions are

$$\begin{aligned} \tilde{r}(K) &= \alpha A K^{\alpha-1} \\ w(K) &= K^\alpha - \tilde{r}(K)K. \end{aligned} \tag{1}$$

Each household possesses homogeneous physical capital, denoted k, and supplies one unit of labor. Since there is measure one of household, total supply of the labor in the economy is one (i.e., $n = 1$). A household problem is:

$$\begin{aligned} max \quad & \sum_{t=0}^{\infty} \beta^t \ln c_t \\ s.t. \quad & c_t + x_t = \tilde{r}(K_t)k_t + w(K_t) \\ & k_{t+1} = (1-\delta)k_t + x_t \\ & k_0 \ given \end{aligned}$$

where c_t is the consumption, x_t is the gross investment, $\tilde{r}_t \equiv \tilde{r}(K_t)$ is the rental rate on capital, $w_t = w(K_t)$ is a competitive wage, and $\delta \in (0,1)$ is a depreciation rate. Given an initial aggregate capital $K_0 = k_0$, the law of motion of aggregate capital, $K_{t+1} = G(K_t)$, determines a sequence of future aggregate capital stock $\{K_t\}_{t=0}^{\infty}$ and hence, through Eq. (1), a sequence of future prices $\{(w_t, \tilde{r}_t)\}_{t=0}^{\infty}$. Each household solves the above maximization problem **given** the price sequence $\{(w_t, \tilde{r}_t)\}_{t=0}^{\infty}$; this household maximization, in turn, determines the law of motion of capital, $k' = g(k, K)$, and hence it determines, through the equilibrium condition $k = K$, another price sequence $\{(w_t, \tilde{r}_t)\}_{t=0}^{\infty}$. This defines a mapping from the space of sequences of prices $\{(w_t, \tilde{r}_t)\}_{t=0}^{\infty}$ into itself. The equilibrium price sequence is the fixed point of this mapping.

Rewrite the household problem in terms of Bellman's equation as follows:

$$\begin{aligned} v(k,K) &= \max_{c,k'} \ln(c) + \beta v(k', G(K)) \\ s.t. \quad & c + k' \leq (1 + \tilde{r}(K) - \delta)k + w(K). \end{aligned} \tag{2}$$

Let the associated policy function for the next period capital choice be $g(k, K)$. Given $g(k, K)$, the budget constraint implies that $c(k, K) = (1 + \tilde{r}(K) - \delta)k + w(K) - g(k, K)$.

Define a *recursive competitive equilibrium* for this economy as a collection of functions: $v(k, K)$, $c(k, K)$, $g(k, K)$, $\tilde{r}(K)$, $w(K)$, $G(K)$, and $C(K)$ such that:

- Given the pricing functions, $\tilde{r}(K)$ and $w(K)$, and aggregate law of motion $G(K)$, the functions $v(k, K)$, $c(k, K)$, and $g(k, K)$ satisfy the household Bellman equation.
- Given price functions $\tilde{r}(K)$ and $w(K)$, firms maximize profits.
- The markets for consumption goods, labor, and capital clear so that $c(K, K) = C(K, K)$, $n = 1$[1], and $g(K, K) = G(K)$.

[1] n is the demand for labor and 1 is the supply of labor.

We may define the steady state of this economy with the steady state aggregate capital level K^* that is the fixed point of the law of motion of aggregate capital , i.e., $K^* = G(K^*) = g(K^*, K^*)$.

In the following, you are asked to numerically solve the recursive competitive equilibrium.

1. Use the parameter values: $\alpha = 0.25$, $A = (\alpha\beta)^{-1}$, and $\delta = 1$. Our discretization for k and K will be $k_i = 0.8 + (i-1)\kappa$ and $K_i = 0.8 + (i-1)\kappa$ for $i = 1, 2, ..., n$. We choose $\kappa = 0.01$ with $n = 41$ so that $k_1 = K_1 = 0.8$ and $k_n = K_n = 1.2$.

   To numerically solve the recursive equilibrium, consider the following algorithm:

   (a) Start from the initial guess of the law of motion of aggregate capital $G^0(K)$. Choose a constant function $G^0(K) = 1$ for every K as an initial guess.

   (b) After h iterations, we have the law of motion of capital $G^h(K)$. Then, using the pricing functions (1), we solve a discretized version of the household Bellman equation by backward induction:

   $$v(k, K) = \max_{k'} \ \ln[(1 + \tilde{r}(K) - \delta)k + w(K) - k'] + \beta v(k', G^h(K)). \tag{3}$$

   Let the policy function be $g^h(k, K)$. See below on how to solve (3).

   (c) If $||g^h(K, K) - G^h(K)|| < \epsilon$, then $G^h(K)$ is the fixed point and so stop. If $||g^h(K, K) - G^h(K)|| \geq \epsilon$, then compute a new estimate of $G^{h+1}(K) = \xi G^h(K) + (1 - \xi) g^h(K, K)$ for a fixed "relaxation parameter" $\xi \in (0, 1)$ and go back Step (b).

2. Plot on the graph both the law of motion for the recursive equilibrium, $G(K)$, and the policy function of corresponding *planner's* Bellman equation.[2] Check if they are the same. Hand in program and its output.

How to numerically solve (3): Notice that the state space is now (k, K) rather than k. Thus, the number of states in household problem is n^2.

First, define the utility function $U(k, K, k') \equiv \ln[(1 + \tilde{r}(K) - \delta)k + w(K) - k']$. This utility function can be represented by an "n^2 by n" utility matrix, say $\mathbf{U}$, where an $[(i + (j-1)n), i']$ element equal to $\ln[\max\{(1 + \tilde{r}(K_j) - \delta)k_i + w(K_j) - k_{i'}, eps\}]$.

Second, the law of motion $G^h(K)$ can be represented by an "n by n" Markov matrix, denoted by Q_K, such that the (j, j')th element is 1 if $K_{j'} = G^h(K_j)$ and 0 otherwise. Let $q_{j,j'}$ be the (j, j')th element of the Markov matrix Q_K.

[2] This requires you to numerically solve the corresponding planner's Bellman equation.

Then, the discretized Bellman equation is given by

$$v(i,j) = \max_{i'=1,2,\ldots,n} U(k_i, K_j, k_{i'}) + \beta \sum_{j'=1}^{m} q_{j,j'} v(i',j'). \tag{4}$$

We solve the fixed point of this Bellman equation as follows:

1. Set the initial guess $v^0 = \mathbf{0}_{n,n}$, where $\mathbf{0}_{n,n}$ is an "n by n" matrix where each element is a scalar 0.

2. After l value function iterations, we have a "n by n" matrix representing a value function, v^l. Given v^l, compute

   $$v^{l+1}(i,j) = \max_{i'=1,2,\ldots,n} U(k_i, K_j, k_{i'}) + \beta \sum_{j'=1}^{m} q_{j,j'} v^l(i',j')$$

   , for every $(i,j) \in \{1,2,...,n\} \times \{1,2,...,n\}$.

   One way to do this is to use "do-loop" twice. For $(i,j) = (1,1)$, compute $U[k_1, K_1, k_{i'}] + \beta \sum_{j'=1}^{m} q_{1,j'} v^l(i',j')$ for $i' = 1,2,...,n$ and store their values in an "n by 1" vector. Then, find a max among those values. If the $i^*(1,1)$th element is the max, then $v^{l+1}(1,1) = U[k_1, K_1, k_{i^*(1,1)}] + \beta \sum_{j'=1}^{m} q_{1,j'} v^l(i^*(1,1), j')$. This is an "inner" loop. Repeat this for every pair of $(i,j) \in \{1,2,3,...,n\} \times \{1,2,3,...,n\}$ ("outer loop') to get $\{v^{l+1}(i,j)\}$ and store them in a "n by n" matrix; this is a function $v^{l+1} = T(v^l)$, where T is an operator defined by the right hand side of (4).

   In practice, avoid loops as much as possible; we may "vectorize" the state space. First, compute an n by n matrix, $ev^l \equiv v^l Q_K'$, which represents the next period's value function under the law of motion of aggregate capital Q_K'; the (i',j)th element represents the next period's value function when $k = k_{i'}$ is chosen and *today's* aggregate capital is K_j.

   Second, compute $w^{l+1} = \mathbf{U} + \beta[ev^{l'} \bigotimes \mathbf{1}_n]$, where $\mathbf{1}_n$ is an n by 1 vector with each element equal to 1 and $\bigotimes$ is the Kronecker product. Third, for each *row*, take a max across columns to compute an "n^2 by 1" vector v^{l+1}. You can do this using only one command, *max*.[3] Finally, to get an n by n matrix, use a command $reshape(v^{l+1}, n, n)$.

3. Repeat the recursive computation of Step 2 until $\max_{(i,j)\in\{1,2,...,n\}\times\{1,2...,n\}} |v^l(i,j) - v^{l+1}(i,j)| < \frac{\epsilon}{1-\beta}$, where we set $\epsilon = 0.001$. Let v^* be the computed fixed point.

4. Compute the policy function represented by an "n by n" Markov matrix, denoted by Q_k, such that the (i,i')th element is 1 if $k_{i'} = g(k_i, K_i)$.[4] We may compute $||Q_k - Q_K||$ to check if Q_K is the fixed point or not.

[3] In Matlab, use the command $[v^{l+1}, I] = \max(w, [\,], 2)$.

[4] After $[v_1, I] = \max(w, [], 2)$, $diag(reshape(I,n,n))$ returns an n by 1 vector with the ith is the integer i' such that $k_{i'} = g(k_i, K_i)$.