Problem Set 3
Due 01. October, 10:15

Exercise 3.1: Overlapping generations and bubbles

Consider an endowment economy with overlapping generations where agents live for two periods. Agents have preferences over consumption when young, c_t^y, and old, c_t^o, and do not discount the future

$$U_t = \log\left(c_t^y\right) + \log\left(c_{t+1}^o\right).$$

Agent's endowment when young, ω^y is higher then the one in the old age, $\omega^o < \omega^y$. There are two assets in this economy. A private bond b_t that yields a safe interest r_t and is available in zero net supply, and a console bond (an infinitely-lived bond) denoted by a_t with price q_t which pays a dividend $d \geq 0$ in every period and is available in unit supply.

(a) Derive the equations that characterize the competitive equilibirum of this economy.

(b) Show that the price of the console bond satisfies the pricing rule

$$q_t = \frac{q_{t+1} + d}{1 + r_{t+1}}. \tag{1}$$

(c) A **rational asset bubble** is an asset whose value is larger than the present value of its dividends. Assuming that $d = 0$, show that there exist two stationary equilibria (equilibria where the interest rate, r_t, remains constant) - one with a rational asset bubble and one without. What is the equilibrium interest rate in each of these steady states?

(d) Suppose the economy is in a stationary equilibrium with a rational asset bubble (the price of the console bond is strictly positive). What happens to the price of the console bond and the consumption of the young and the old if the beliefs of the young agents shift suddenly so they expect the bond will have zero value the next period?

(e) Based on your answer above, would you say the bubble equilibrium is Pareto superior to the equilibrium without the bubble?

(f) Consider now an economy with strictly positive dividends from the console bond, $d > 0$. How many stationary equilibria are there in this economy? Characterize the stationary equilibrium interest rate.

Exercise 3.2: Value function iteration with paper and pencil

Consider the following version of the neoclassical growth model in discrete time

$$V(k) = \max_{c \geq 0, k' \geq 0} \log(c) + \beta V(k') \text{ s.t. } c + k' \leq k^\alpha + (1-\delta)k,$$

where $V(k)$ is the value function, c denotes consumption, k the physical capital stock, δ is the depreciation rate of capital, and α is the capital income share of the economy. Variables with primes denote on period forward variables (for example, if $k = k_t$ then $k' = k_{t+1}$), this notation reflects that fact that the Bellman equation is independent of calendar time.

(a) Assume that depreciation is 100% and that the resource constraint holds with equality. Start with the guess that the value function is $V^0(k) = 0$. Update your guess by iterating on the the Bellman equation

$$V^{j+1}(k) = \max_{0 \leq k' \leq k^\alpha} \log(k^\alpha - k') + \beta V^j(k'), \quad j = 0, 1, \ldots,$$

and show that the value function converges to $V^{j+1}(k) = A + B\log(k)$ as $j \to \infty$. Pin down the value of B.

Exercise 3.3: Value function iteration with Matlab

Consider the following version of the neoclassical growth model in discrete time

$$V(k) = \max_{c \geq 0, k' \geq 0} \frac{c^{1-\sigma}}{1-\sigma} + \beta V(k') \text{ s.t. } c + k' \leq k^\alpha + (1-\delta)k,$$

where $\alpha = .36$, $\beta = .95$, $\delta = .1$, $\sigma = 2$. As preferences are not logarithmic and depreciation is less than 100%, there is no known closed form solution for the value function and the policy functions, so we will perform the value function iteration with Matlab to find a numerical approximation.

(a) Work through the supplementary set of slides ("A Matlab Primer") that were posted along with this problem set and set up a code in Matlab that approximates the value function, $V(k)$, and the policy functions, $k' = g(k)$ and $c(k)$, using numerical value function iteration (hint: you can use the code that is given in the slides. Your main challenge will be to get the files running on your computer and learn some basic Matlab commands that we will also use in future problem sets).

(b) Implement Howard's improvement (policy function iteration) as explained in Exercise 3.3 on Slide 24. That is, after value function iteration j - which gives you the value function $V^{j,0}(k)$ and the policy function $g^j(k)$ - you iterate n_h more times on the Bellman equation for the given policy function $g^j(k)$ to yield $V^{j,n_h}(k)$ according to

$$V^{j,h+1}(k) = \log(k^\alpha + (1-\delta)k - g^j(k)) + \beta V^{j,h}(g^j(k)), \quad h = 0, 1, ...n_h,$$

before going to the next value function iteration $j+1$. By how much does it speed up your code if you do $n_h = 10$ policy function iterations, each time before you start another value function iteration? Why can you do even better with j policy function iterations (n_h increases with j) in every value function iteration?

Send me your code that includes Howard's improvement per email before the due date.