ECON 6140 - Problem Set # 1

Julien Manuel Neves

February 1, 2018

Kuhn-Tucker conditions

For the Kuhn-Tucker conditions to be satisfied, we need f_0 , f_1 and f_2 to be concave, continuous functions from S (convex) into \mathbb{R} .

By setting $S = \mathbb{R}^2$, we have at least satisfy one condition. Sadly, it is straightforward to see that $f_0(x,y) = x^2$ and $f_1(x,y) = (1-x)^3 - y$ while continuous are not concave.

In fact, this problem KKT conditions yields only necessary conditions for local maximum. KKT:

$$2x = 3\lambda(1-x)^{2}$$

$$\mu = \lambda$$

$$\lambda((1-x)^{3} - y) = 0$$

$$\mu y = 0$$

$$\lambda \ge 0$$

$$\mu > 0$$

One possible candidate could be (0,0), but any point where x < 0 and y = 0 will yield be an improvement. Additionally, if let $x \to \infty$ and y = 0, x^2 will go to infinity and the constraints will be satisfy. Hence, our problem is unbounded.

Optimal equilibrium allocations

Let $u'(0) = \infty$ and $u'(\infty) = 0$. Since there's no capital accumulation, we have $k_1 = k_2 = 0$.

(1) Household constraint:

$$p_1[b-c_1] + p_2[e-rb-c_2] \ge 0$$

where $r = f'(k_0)$.

(2) Since there is no firm, the household's problem is the same as the planner's problem, i.e.

$$\max_{c_1, c_2} u(c_1) + \beta u(c_2)$$
s. t. $b - c_1 \ge 0$

$$e - rb - c_2 \ge 0$$

$$c_1, c_2 \ge 0$$

where $r = f'(k_0)$.

- (3) We need u and the constraints to be continuous, strictly increasing and concave functions.
- (4) Let b be the total amount borrowed and b_H the amount borrowed at price $h'(k_0)$. Then, the household problem is the following:

$$\max_{c_1, c_2} u(c_1) + \beta u(c_2)$$
s. t. $b - c_1 \ge 0$

$$e - h'(k_0)b_H - f'(k_0)(b - b_H) - c_2 \ge 0$$

$$b - b_H \ge 0$$

$$\bar{b} - b \ge 0$$

$$c_1, c_2 \ge 0$$

The Lagrangian is given by:

$$\mathcal{L} = u(c_1) + \beta u(c_2) + \lambda_1[b - c_1] + \lambda_2[e - h'(k_0)b_H - f'(k_0)(b - b_H) - c_2] + \mu_1[b - b_H] + \mu_2[\bar{b} - b] + \psi_1c_1 + \psi_2c_2$$

KKT:

$$u'(c_1) + \psi_1 = \lambda_1$$

$$\beta u'(c_2) + \psi_2 = \lambda_2$$

$$\lambda_1 - f'(k_0)\lambda_2 + \mu_1 - \mu_2 = 0$$

$$\lambda_2[f'(k_0) - h'(k_0)] - \mu_1 = 0$$

$$\lambda_1[b - c_1] = 0$$

$$\lambda_2[e - h'(k_0)b_H - f'(k_0)(b - b_H) - c_2] = 0$$

$$\mu_1[b - b_H] = 0$$

$$\mu_2[\bar{b} - b] = 0$$

$$\psi_1 c_1 = 0$$

$$\psi_2 c_2 = 0$$

$$\lambda_1 \ge 0, \lambda_2 \ge 0$$

$$\mu_1 \ge 0, \mu_2 \ge 0$$

$$\psi_1 > 0, \psi_2 > 0$$

Note that if $u(\cdot)$ is well-behave, we need $\lambda_1, \lambda_2 > 0$ which in turns implies $\mu_1 > 0$, i.e. $b = b_H$.

Additionally, we have $\psi_1 = 0$ and $\psi_2 = 0$.

Then, we have two cases

(i) $\mu_2 = 0$ Then,

$$u'(c_1) = \beta h'(k_0) u'(c_2)$$

where $c_1 = b$ and $c_2 = e - h'(k_0)b$. Hence,

$$h'(\cdot) \uparrow \to c_1 \downarrow, c_2 \uparrow, b \downarrow$$

$$e \uparrow \to c_1 \uparrow, c_2 \uparrow, b \downarrow$$

$$\bar{b} \uparrow \to c_1 -, c_2 -, b -$$

(ii) $\mu_2 > 0$ Then, $u'(c_1) = \beta h'(k_0)u'(c_2) + \mu_2 > \beta h'(k_0)u'(c_2)$ where $c_1 = \bar{b}$ and $c_2 = e - h'(k_0)\bar{b}$. Hence, $h'(\cdot) \uparrow \to c_1 \downarrow, c_2 \uparrow, b \downarrow$

$$h'(\cdot) \uparrow \to c_1 \downarrow, c_2 \uparrow, b \downarrow$$

$$e \uparrow \to c_1 \uparrow, c_2 \uparrow, b \downarrow$$

$$\bar{b} \uparrow \to c_1 \uparrow, c_2 \downarrow, b \uparrow$$

Hence, if the shadow value of \bar{b} is 0, the borrowing limit is not reach the Euler equation holds with equality. If the shadow value of \bar{b} is strictly positive, then we run into some problems because the consumer would like to borrow more than \bar{b} since the marginal utility of consumption in period 1 is higher than its marginal cost. Note that while the marginal utility of consumption is not equal to its marginal cost in the case of $\mu_2 > 0$, the consumer could do better if he could borrow more money.

Thus, is the situation Pareto efficient? Well, there's only one consumer. Hence, it is impossible to make him better off while making no one else worst off since he is alone in this economy and already maximizing according to the constraints.