Econ 202A Notes: Macroeconomic Theory I

John G. Friedman

Fall 2023

1 Basic Neoclassical Growth Model (Cass-Koopmans)

Economy consists of many identical infinitely lived households – all have the same preferences and endowments.

Several Interpretations

1. Representative Agent (Robinson Crusoe)
2. Social Planner
3. Infinitely Lived Family (dynasty)

One production sector – output produced from capital and labor. Can be consumed or invested. Investment becomes productive capital after one period.

$$
\begin{aligned}
\max \sum_{t=0}^{\infty} \beta^t u(c_t) \\
\text{s.t. } \beta \in (0,1) \\
c_t + i_t \leq y_t = f(k_t, n_t) \\
k_{t+1} \leq (1-\delta)k_t + i_t \\
0 \leq n_t \leq 1 \\
k_0, k_0 > 0
\end{aligned}
$$

Note that no prices are given here!

1.1 Production Function

$F : \mathbb{R}_+^2 \rightarrow \mathbb{R}_+$

1. continuously differentiable
2. homogenous of degree 1 $\Leftrightarrow$ constant returns to scale
3. strictly quasi-concave

4. $F(0, n) = 0$
   (a) implies capital is essential
   (b) F_k = marginal product of capital > 0
   (c) F_n = marginal product of labor > 0
5. $\lim_{k \to 0} F(k, 1) = \infty$
6. $\lim_{k \to \infty} F(k, 1) = 0$

1.2 Utility Function

$u : \mathbb{R}_+ \to \mathbb{R}$

1. bounded - needed for dynamic programming
2. continuously differentiable
3. strictly increasing
4. strictly concave
5. $\lim_{c \to 0} u'(c) = \infty$

Note: We will use functional forms for F and N for most everything we do in this class.

1.3 Simplifying the Planner's Problem

1. $F_N > 0$ and $u'(c) > 0 \Rightarrow n_t = 1 \ \forall \ t$
2. $u'(c) > 0 \Rightarrow$ resource constraint holds with equality - $c_t + i_t = F(k_t, n_t)$
3. $\beta < 1 \Rightarrow$ require positive rate of return to give up a unit of consumption today for consumption tomorrow [actually $mp_{k+1} - \delta$]
4. Let $f(k) = F(k, 1) + (1 - \delta)k$

Problem becomes:

$$\max \sum_{t=0}^{\infty} \beta^t u(f(k_t) - k_{t+1})$$
$$\text{s.t. } 0 \leq k_{t+1} \leq f(k_t)$$
$$k_0 \text{ given}$$

1. Called a sequences problem by Stokey and Lucas
2. Infinite number of choice variables

$\Rightarrow$ **Use recursive methods.**

2 Dynamic Programming