

Chapter 2 Dynamic Programming

Bin Wang*

School of Economics, Jinan University

In the last chapter, we use the Lagrangian multiplier method to solve a two-period dynamic problem with equality constraint. What if we have a multiple-period or even an infinite-horizon problem? We may still use the Lagrangian multiplier method to get the first order conditions for these problems. But in macroeconomics we inherit dynamic programming or recursive methods from other disciplines to solve the infinite-horizon problems.

We will cover the following materials in this chapter:

- A multiple-period optimal growth model;
- Dynamic programming
- Practical dynamic programming: global methods
- Practical dynamic programming: local methods

1 A Multiple-Period Optimal Growth Model

Consider an economy in which the social planner allocates resources. At the beginning of Period 0, the economy is endowed with capital k_0 . The planner decides how much to consume c_0 in this period and how much to leave as capital of next period k_1 with production in this period with technology $f(k_0) = Ak_0^\alpha$. In the end of Period 0, k_0 is fully depreciated. In Period 1 to T , the social planner does the same decisions. At the end of Period T , the economy ends. The social planner's problem in this economy is to maximize the sum of

*Email: binwang@jnu.edu.cn

discounted utility $\sum_{t=0}^T \beta^t u(c_t)$ subject to resource constraints.

$$\begin{aligned} \max_{\{c_t\}_{t=0}^T, \{k_t\}_{t=1}^{T+1}} \quad & \sum_{t=0}^T \beta^t u(c_t) \\ \text{subject to: } \quad & c_t + k_{t+1} \leq Ak_t^\alpha, \forall 0 \leq t \leq T \\ & c_t, k_{t+1} \geq 0, \forall 0 \leq t \leq T, \\ & k_0 \quad \text{given.} \end{aligned}$$

We can see that $c_t, 0 \leq t \leq T$ and $k_{t+1}, 0 \leq t \leq T-1$ cannot be zero in the equilibrium.

Proposition 1. At the optimum, $c_t > 0, 0 \leq t \leq T$ and $k_{t+1} > 0, 0 \leq t \leq T-1$.

Basically, the log utility function value goes to negative infinity if consumption goes to zero. If one of the k_{t+1} is zero then the following c, k will be all zero. So $c_t, 0 \leq t \leq T$ and $k_{t+1} > 0, 0 \leq t \leq T-1$ at the equilibrium.

This is a typical optimization problem with inequality constraint. We can prove that in the optimum the first constraint should be binding¹. Then the social planner's problem is simplified as:

$$\begin{aligned} \max_{\{c_t\}_{t=0}^T, \{k_t\}_{t=1}^{T+1}} \quad & \sum_{t=0}^T \beta^t u(c_t) \\ \text{subject to: } \quad & c_t + k_{t+1} = Ak_t^\alpha, \forall 0 \leq t \leq T \\ & k_{T+1} \geq 0, \\ & k_0 \quad \text{given.} \end{aligned}$$

This optimization problem still contains inequality constraint, which is different from the optimization problem with only equality constraint. We can tackle this problem by **Kuhn-Tucker Theorem**. Still, we first form the Lagrangian function:

$$L(c_0, \dots, c_T, k_1, \dots, k_{T+1}) = \sum_{t=0}^T [\beta^t u(c_t) + \lambda_t (Ak_t^\alpha - c_t - k_{t+1})] + \mu_T k_{T+1}$$

¹An inequality constraint is called binding when the constraint is satisfied with equality; an inequality constraint is slack when the constraint is not satisfied with equality.

where $\lambda_t, t = 0, \dots, T; \mu_T$ are associated with constraints $c_t + k_{t+1} = Ak_t^\alpha, \forall 0 \leq t \leq T$ and $k_{T+1} \geq 0$. For the last inequality constraint $k_{T+1} \geq 0$, we have two scenarios:

- $k_{T+1}^* = 0$, the inequality constraint is binding at the optimum. This scenario is the same the familiar Lagrangian multiplier problem;
- $k_{T+1}^* > 0$, the inequality constraint is slack at the optimum. When a constraint is slack, the objective at the optimum is not influenced by this constraint. The optimization problem is now a problem without this constraint $k_{T+1} \geq 0$, or we can consider $\mu_T = 0$ in this scenario and then the Lagrangian function is the one without this constraint.

Then we need to add one more condition for the inequality constraint:

$$\mu_T k_{T+1} = 0, \mu_T \geq 0, k_{T+1} \geq 0$$

where $\mu_T \geq 0$ comes from that the Lagrangian multiplier should be positive for equality condition. It states that either $\mu_T = 0$ or $k_{T+1} = 0$ at the optimum, which includes both scenarios of the inequality constraint. The Kuhn-Tucker conditions are:

$$\begin{aligned} L_{c_t} &= \beta^t u'(c_t) - \lambda_t = 0, \quad t = 0, \dots, T \\ L_{k_{t+1}} &= -\lambda_t + \lambda_{t+1} A \alpha k_{t+1}^{\alpha-1} = 0, \quad t = 1, \dots, T-1 \\ L_{k_{T+1}} &= -\lambda_T + \mu_T = 0 \\ \mu_T &\geq 0, k_{T+1} \geq 0, \mu_T k_{T+1} = 0 \end{aligned}$$

As we see from the first condition $\lambda_t = \beta^t u'(c_t) > 0$, thus $\mu_T = \lambda_T = \beta^T u'(c_T) > 0$. Then $\mu_T k_{T+1} = 0$ implies that $k_{T+1} = 0$. The complementary slackness condition $\mu_T k_{T+1} = 0$ has an important implication in the dynamic optimization. It states that in the end either the marginal utility of capital is zero $\mu_T = 0$ or the capital is wholly consumed in the end $k_{T+1} = 0$. Combine all these conditions:

$$u'(c_t) = \beta A \alpha k_{t+1}^{\alpha-1} u'(c_{t+1}), 0 \leq t \leq T \tag{1}$$

The optimal path $(c_0^*, \dots, c_T^*, k_1^*, \dots, k_{T+1}^*)$ is characterized by four conditions: The four

Euler Condition:	$u'(c_t) = \beta A\alpha k_{t+1}^{\alpha-1} u'(c_{t+1}), 0 \leq t \leq T-1$
Resource Constraint:	$c_t + k_{t+1} = Ak_t^\alpha, \forall 0 \leq t \leq T$
Initial Condition:	k_0 is given
Transversality Condition:	$k_{T+1} = 0$

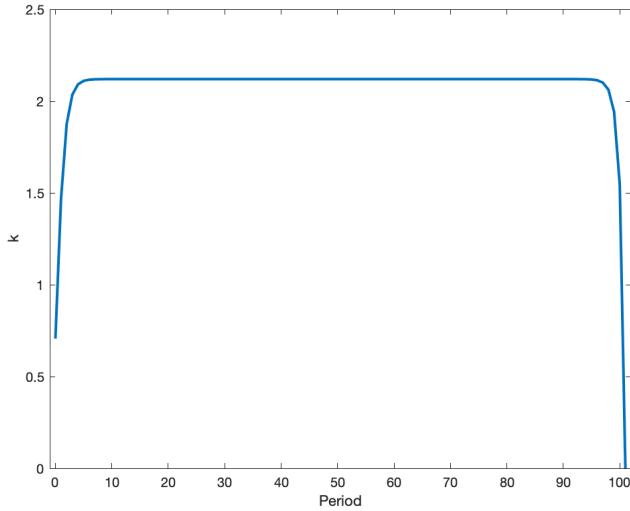
conditions are both necessary and sufficient for the optimizer of the social planner's problem and the transversality condition can pin down the solution of the problem. To see this point, we substitute c_t of the resource constraint to the Euler condition. We have $T + 1$ unknowns k_1, \dots, k_{T+1} for $T + 1$ equations.

$$u'(Ak_t^\alpha - k_{t+1}) = \beta A\alpha k_{t+1}^{\alpha-1} u'(Ak_{t+1}^\alpha - k_{t+2}), t = 0, 1, \dots, T-1 \quad (2)$$

$$k_0, \text{given}; k_{T+1} = 0. \quad (3)$$

We can use several algorithms to find the optimal point by solving the $T + 1$ nonlinear equations. In Matlab, we can use the 'fsolve' function to get the solution. The solution k_1, \dots, k_{T+1} is drafted in Figure 1.

Figure 1: Equilibrium Capital



```

1 %=====
2 % Chapter 2
3 % File: ch2-solve-k-finite.m
4 % This program file is to solve the finite optimal growth model
5 % Written: 2020.09.08

```

```

6 % Written by Bin Wang
7 %=====
8
9 clear all;
10 close all;
11 clc;
12
13 %% Parameters
14 A = 5;
15 alpha = 1/3;
16 beta = 0.99;
17 T = 100;
18 kbar = (A*alpha*beta)^(1/(1-alpha));
19 k0 = 1/3*kbar;
20 para = [A,alpha,beta,k0];
21 kt = zeros(T,1);
22 kTplus1 = 0;
23 kt = fsolve(@(k)(finiteRamsey(k,para)),k0*ones(T,1));
24
25 k = [k0;kt;kTplus1];
26
27 figure;
28 plot(0:1:(T+1),k,'LineWidth',2);
29 xlabel('Period');
30 ylabel('k');
31 xlim([-1 T+2]);
32
33 %% function
34 function F=finiteRamsey(k,para)
35 A=para(1);
36 alpha = para(2);
37 beta = para(3);
38 k0 = para(4);
39 kTplus1=0;
40 T = length(k);
41 F = zeros(T,1);
42 F(1) = A*alpha*beta*k(1)^(alpha-1)/(A*k(1)^alpha-k(2))
43 -1/(A*k0^alpha-k(1)); %k0 is given
44 for t=1:T-2
45     F(t+1)= A*alpha*beta*k(t+1)^(alpha-1)/(A*k(t+1)^alpha
46     -k(t+2))-1/(A*k(t)^alpha-k(t+1));
47 end
48 F(T)=A*alpha*beta*k(T)^(alpha-1)/(A*k(T)^alpha-kTplus1)
49 -1/(A*k(T-1)^alpha-k(T));
50 end

```

You may notice that in the middle of all T periods capital k stays the same that $k_t = k_{t+1}$. This is called the **steady state**.

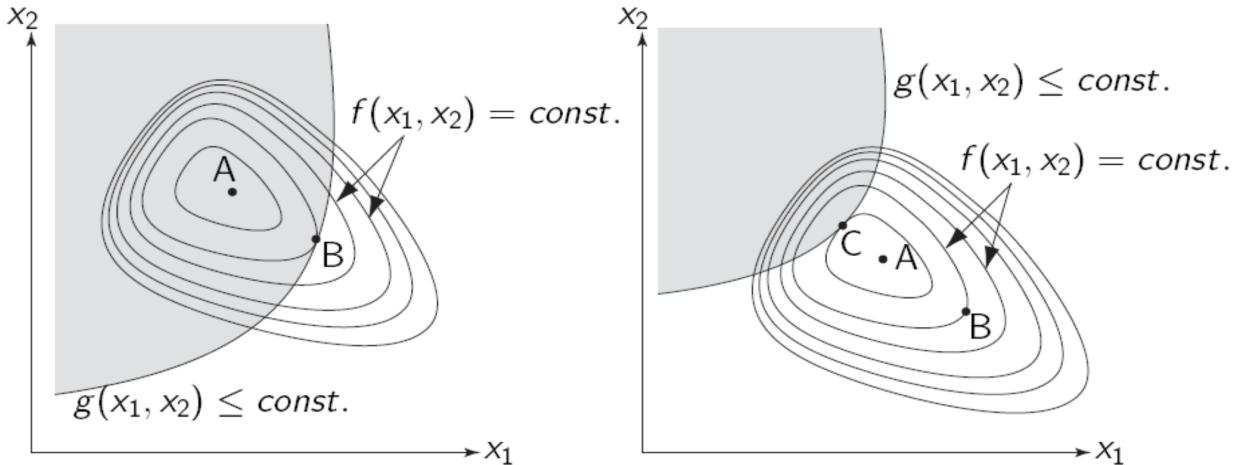
$$k = (A\alpha\beta)^{\frac{1}{1-\alpha}}$$

Kuhn-Tucker Theorem For the optimization problem

$$\begin{aligned} \max_{x_1, x_2} \quad & f(x_1, x_2) \\ \text{s.t.} \quad & g(x_1, x_2) \leq m \end{aligned}$$

The Kuhn-Tucker theorem tries to explore the conditions that the optimal point should satisfy. Then Kuhn-Tucker conditions are basically necessary conditions. Suppose (x_1^*, x_2^*) is the optimal solution that maximizes $f(x_1, x_2)$. There are two scenarios of the optimal solution, either $g(x_1^*, x_2^*) < m$ or $g(x_1^*, x_2^*) = m$. In Figure 2, the shadow region includes all the points that $\{(x_1, x_2) | g(x_1, x_2) \leq m\}$. In both panels, point A achieves the maximum value of $f(x_1, x_2)$ if there is no constraint. The difference between the two panels is whether A is inside or outside the constraint $\{(x_1, x_2) | g(x_1, x_2) \leq m\}$.

Figure 2: Kuhn-Tucker Theorem



- In the left panel, Point A is inside the constraint $\{(x_1, x_2) | g(x_1, x_2) \leq m\}$. In this

scenario, the constrained problem achieves the maximum value at Point A, which is not on the boundary of the constraint. So at the maximal point A, we have

$$g(x_1^*, x_2^*) = g(x_1^A, x_2^A) < m \quad (4)$$

Basically, if we change the constraint, e.g. m, by a bit, the problem still achieves the maximum point at A. It is this sense that we mean the function is not constrained by this constraint $\{(x_1, x_2) | g(x_1, x_2) \leq m\}$ because the maximum point is both A with or without this constraint. Under this scenario, the maximum point A should satisfy the first order conditions.

$$f_{x_1}(x_1^A, x_2^A) = 0, \quad (5)$$

$$f_{x_2}(x_1^A, x_2^A) = 0 \quad (6)$$

$$g(x_1^A, x_2^A) > 0 \quad (7)$$

These three conditions are the first-order conditions of the Lagrangian multiplier method when the Lagrangian multiplier $\lambda^A = 0$.

$$L(x_1, x_2, \lambda) = f(x_1, x_2) - \lambda(g(x_1, x_2) - m) \quad (8)$$

$$\frac{\partial L}{\partial x_1}(x_1^A, x_2^A, \lambda^A) = f_{x_1}(x_1^A, x_2^A) = 0, \quad (9)$$

$$\frac{\partial L}{\partial x_2}(x_1^A, x_2^A, \lambda^A) = f_{x_2}(x_1^A, x_2^A) = 0 \quad (10)$$

$$g(x_1^A, x_2^A) > 0, \lambda^A = 0 \quad (11)$$

- In the right panel, the point A which has maximal value for $f(x_1, x_2)$ without the constraint is outside the shadow constraint region. As A is not attainable, the maximal point should be on the constraint boundary, e.g. Point C. In this scenario, we have the result:

$$g(x_1^*, x_2^*) = g(x_1^C, x_2^C) = m \quad (12)$$

Imagine, if we change the constraint a bit, e.g. m, then the position of C will change

accordingly. It is this sense that we mean the maximum value is constrained by the constraint. This scenario is the same as the optimization problem with equality constraint. We know that the slope of the isoquent (indifference curve) should be the same as that of the constraint boundary. By using the implicit function theorem, we know that

$$\text{slope} = -\frac{f_{x_1}(x_1^C, x_2^C)}{f_{x_2}(x_1^C, x_2^C)} = -\frac{g_{x_1}(x_1^C, x_2^C)}{g_{x_2}(x_1^C, x_2^C)}$$

Exchange

$$\frac{f_{x_1}(x_1^C, x_2^C)}{g_{x_1}(x_1^C, x_2^C)} = \frac{f_{x_2}(x_1^C, x_2^C)}{g_{x_2}(x_1^C, x_2^C)} = \lambda^C$$

We let this number equals to λ^C . Then the point C should satisfy:

$$\begin{aligned} f_{x_1}(x_1^C, x_2^C) - \lambda^C g_{x_1}(x_1^C, x_2^C) &= 0 \\ f_{x_2}(x_1^C, x_2^C) - \lambda^C g_{x_2}(x_1^C, x_2^C) &= 0 \end{aligned}$$

These two along with the constraint are basically the conditions that the partial derivative of Lagrangian function should be zero at Point C.

$$L(x_1, x_2, \lambda) = f(x_1, x_2) - \lambda(g(x_1, x_2, \lambda)) \quad (13)$$

$$\begin{aligned} \frac{\partial L}{\partial x_1}(x_1^C, x_2^C, \lambda^C) &= f_{x_1}(x_1^C, x_2^C) - \lambda^C g_{x_1}(x_1^C, x_2^C) = 0 \\ \frac{\partial L}{\partial x_2}(x_1^C, x_2^C, \lambda^C) &= f_{x_2}(x_1^C, x_2^C) - \lambda^C g_{x_2}(x_1^C, x_2^C) = 0 \\ \frac{\partial L}{\partial \lambda}(x_1^C, x_2^C, \lambda^C) &= g(x_1^C, x_2^C) - m = 0, \lambda^C > 0 \end{aligned}$$

Combine the above two scenarios and we know that no matter $g(x_1^*, x_2^*) < 0$ or $g(x_1^*, x_2^*) = 0$. The optimal point (x_1^*, x_2^*) should satisfy the **Kuhn-Tucker conditions**.

$$L(x_1, x_2, \lambda) = f(x_1, x_2) - \lambda(g(x_1, x_2, \lambda)) \quad (14)$$

$$\begin{aligned}\frac{\partial L}{\partial x_1}(x_1^*, x_2^*, \lambda^*) &= f_{x_1}(x_1^*, x_2^*) - \lambda^* g_{x_1}(x_1^*, x_2^*) = 0 \\ \frac{\partial L}{\partial x_2}(x_1^*, x_2^*, \lambda^*) &= f_{x_2}(x_1^*, x_2^*) - \lambda^* g_{x_2}(x_1^*, x_2^*) = 0 \\ \frac{\partial L}{\partial \lambda}(x_1^*, x_2^*, \lambda^*) \lambda^* &= (g(x_1^*, x_2^*) - m) \lambda^* = 0, \lambda^* \geq 0, g(x_1^*, x_2^*) - m \geq 0\end{aligned}$$

In the first scenario, the maximum value of the function will not change if we change m . So $\lambda^* = 0$. In the second scenario, the maximum value of the function will change if we change m . So $\lambda^* > 0$. Basically, you should understand this point so that you can get a deep understanding about static and dynamic optimization.

Infinite Horizon The optimal growth model of finite horizon already shows the dynamic intertemporal substitution of consumer's optimal behavior. Why do we need infinite horizon? Or why is infinite-horizon model so popular in macroeconomics? There are several reasons:

- Technically beautiful. We can get beautiful results in infinite-horizon model without losing the main insights of dynamic optimization.
- Since life is limited, finite-horizon model seems more practical. However, nobody knows when she or he dies in the future. Suppose every period the probability to die is $1 - \nu$. In period t , the consumer's expected utility would be $\beta^t \nu^t u(c_t)$. The consumer should maximize the sum of expected utility forever. Here time horizon t is uncertain though life is limited.

$$\max_{c_t, k_{t+1}} (\beta \nu)^t u(c_t)$$

We can redefine the effective discount factor as $\beta' = \nu \beta$ and so there is a reason to justify the infinite-horizon model. From now on, we will focus on infinite-horizon model to illustrate dynamic stochastic macroeconomic problems.

2 Dynamic Programming

2.1 Sequence Problem and Functional Equation

Consider the optimal growth model with infinite-horizon:

$$\text{SP: } v^*(k_0) = \max_{\{c_t\}_{t=0}^{\infty}, \{k_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u(c_t)$$

$$s.t. \quad c_t + k_{t+1} \leq f(k_t)$$

$$c_t, k_{t+1} \geq 0$$

$$k_0, given$$

where $u(c_t) = \ln c_t$ and $f(k_t) = Ak_t^\alpha$. We can substitute c_t with the resource constraint.

$$\text{SP: } v^*(k_0) = \max_{\{k_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u(f(k_t) - k_{t+1})$$

$$s.t. \quad k_{t+1} \geq 0$$

$$k_0, given$$

The solution of this infinite-horizon problem is to find a plan $\{k_{t+1}\}_{t=0}^{\infty}$ that the objective $\sum_{t=0}^{\infty} \beta^t u(f(k_t) - k_{t+1})$ is maximized. We usually call it **sequence problem** (short for **SP**). If we find the optimal plan $\{k_{t+1}^*\}_{t=0}^{\infty}$ to maximize the objective, we can get the indirect utility function or value function of the given capital of period 0 k_0 . At the beginning of Period 0, k_0 is given. We usually call this given variable **state variable** and the variables chosen from the problem **choice variable** such as k_1 . Likewise, at the beginning of period t , the state variable is k_t and the choice variables are k_{t+1} .

The basic idea of dynamic programming is to turn the sequence problem into a functional equation; that is, to transform the problem into one of finding a function rather than a sequence. The relevant **functional equation** (short for **FE**) or **Bellman equation** can be

written as follows.

$$\text{FE: } v(k_t) = \max_{k_{t+1}} u(f(k_t) - k_{t+1}) + \beta v(k_{t+1}), \quad \forall k_t$$

$$s.t. \quad 0 \leq k_{t+1} \leq f(k_t)$$

k_t , given

Intuitively, the FE problem is to find a policy $k_{t+1} = h(k_t)$, which determines what the control variable k_{t+1} should be for a given value of the state variable k_t . Mathematically, this corresponds to maximizing $v(k_t), \forall k_t$ by choosing the optimal k_{t+1} for any k_t . This is the reason that why FE or Bellman equation is called the **recursive formation**.

- At first glance, you should notice that the solution of the FE is not just a "point" k_{t+1} but a function $v(k_t)$ or $k_{t+1} = h(k_t)$. That is because FE should be satisfied for any given k_t . Imagine you give different values for k_t and you need to find an optimal policy k_{t+1} for each value of k_t to satisfy the FE. Then it's just to say that we are finding a function for the solution of the FE.
- Notice that when we say the solution is a policy $k_{t+1} = h(k_t)$, it is equivalent to say that we find a solution $v(k_t)$. That's because

$$v(k_t) = \max_{k_{t+1}} u(f(k_t) - k_{t+1}) + \beta v(k_{t+1}) = u(f(k_t) - h(k_t)) + \beta v(h(k_t)), \quad \forall k_t$$

Next, we state a theorem that the value function of SP $v^*(k_0)$ is the same as the solution of FE $v(k_0)$.

Theorem 2. *The value function of Sequence Problem $v^*(k_0)$ is the same as the solution of FE $v(k_0)$ for any given k_0 .*

Proof. First we show that $v^*(k_0)$ satisfies the FE for any k_0 .

$$\begin{aligned}
v^*(k_0) &= \max_{\{k_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u(f(k_t) - k_{t+1}) \\
&= \max_{\{k_{t+1}\}_{t=0}^{\infty}} u(f(k_0) - k_1) + \beta u(f(k_1) - k_2) + \beta^2 u(f(k_2) - k_3) + \cdots \\
&= \max_{k_1} [u(f(k_0) - k_1) + \max_{\{k_{t+1}\}_{t=1}^{\infty}} \beta u(f(k_1) - k_2) + \beta^2 u(f(k_2) - k_3) + \cdots] \\
&= \max_{k_1} [u(f(k_0) - k_1) + \beta (\max_{\{k_{t+1}\}_{t=1}^{\infty}} u(f(k_1) - k_2) + \beta u(f(k_2) - k_3) + \cdots)] \\
&= \max_{k_1} u(f(k_0) - k_1) + \beta v^*(k_1)
\end{aligned}$$

So the value function of sequence problem $v^*(k_t)$ satisfies the functional equation for all k_0 . Next we show that $v(k_0)$ satisfies the sequence problem.

$$\begin{aligned}
v(k_0) &= \max_{k_1} u(f(k_0) - k_1) + \beta v(k_1) \\
&= \max_{k_1} u(f(k_0) - k_1) + \beta \max_{k_2} [u(f(k_1) - k_2) + \beta v(k_2)] \\
&= \max_{k_1, k_2} u(f(k_0) - k_1) + \beta u(f(k_1) - k_2) + \beta^2 v(k_2) \\
&= \max_{k_1, k_2, k_3} u(f(k_0) - k_1) + \beta u(f(k_1) - k_2) + \beta^2 u(f(k_2) - k_3) + \beta^3 v(k_3) \\
&= \cdots \\
&= \max_{\{k_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u(f(k_t) - k_{t+1})
\end{aligned}$$

Then $v(k_0)$ also satisfies the SP for any k_0 . We proved that the value function of Sequence Problem $v^*(k_0)$ is the same as the solution of FE $v(k_0)$ for any given k_0 . \square

As $v^*(k_0)$ is equivalent to $v(k_0)$, we can get the optimal sequence $\{k_{t+1}^*\}_{t=0}^{\infty}$ by iterating the optimal policy function $k_{t+1} = h(k_t), \forall k_t$. From now on, we explore how to find the solution of the optimal growth model by the FE. For notational simplicity, we abstract the time period but use prime variable to denote the future period variable.

$$\text{FE: } v(k) = \max_{k'} u(f(k) - k') + \beta v(k'), \quad \forall k$$

$$s.t. \quad 0 \leq k' \leq f(k)$$

k , given

Notice that the right hand side of FE still has the value function which is the solution of FE itself. Basically, the value function $v(k)$ is not known before we solve the problem. But

how can we maximize a objective with unknown function? In most of the cases we will deal with, the FE or Bellman equation satisfies a **contraction mapping theorem**, which implies that:

- There is a unique function $v(\cdot)$ which satisfies the FE. And when the utility function is concave and twice differentiable, the value function is also concave and differentiable.
- If we begin with a initial function $v_0(k)$ and define $v_{i+1}(k)$ by

$$v_{i+1}(k) = \max_{k'} u(f(k) - k') + \beta v_i(k')$$

Iterate this mapping and we can find that $v_j(k)$ converges to the value function of the FE.

We will not prove the existence, uniqueness, and concavity of the value function here. Interested students should consult Chapter 5 of Acemoglu (2009) to get more details. We can get a heuristic sense of why this contraction mapping theorem states that v will converge to the solution $v(k)$.

$$\begin{aligned} v_1(k) &= \max_{k'} u(f(k) - k') + \beta v_0(k') \\ &\geq u(f(k) - g_0(k)) + \beta v_0(g_0(k)) \\ &= v_0(k) \end{aligned}$$

where the last equality comes from the definition of $v_0(k)$. Each iteration of the mapping will increase the function value for all k until the value function converges to the solution $v(k)$.

Once we know that $v(k)$ exists, is concave and differentiable, we can apply the conventional optimization technique to get the first order conditions for optimality.

$$-u'(f(k) - k') + \beta v'(k') = 0 \tag{15}$$

But actually we do not know what $v(k)$ is in the first order condition. Since we know $v(k)$ exist and differentiable, we can use the equivalent of Envelope Theorem for dynamic

programming to get $v'(k')$.

$$v'(k) = \frac{\partial[u(f(k) - k') + \beta v(k')]}{\partial k} = u'(f(k) - k')f'(k) \quad (16)$$

Or you can directly calculate the derivative of $v(k)$:

$$\begin{aligned} v(k) &= \max_{k'} u(f(k) - k') + \beta v(k') = u(f(k) - h(k)) + \beta v(h(k)) \\ v'(k) &= u'(f(k) - k')(f'(k) - h'(k)) + \beta v'(k')h'(k) \\ &= u'(f(k) - k')f'(k) + h'(k)(-u'(f(k) - k') + \beta v'(k')) \\ &= u'(f(k) - k')f'(k) \end{aligned}$$

where $k' = h(k)$ in the optimum and the third equality comes from the first order condition $-u'(f(k) - k') + \beta v'(k') = 0$.

Then we can move the time period forward by one period.

$$v'(k) = u'(f(k') - k'')f'(k') \quad (17)$$

Substitute it into the first order condition:

$$-u'(f(k) - k') + \beta[u'(f(k') - k'')f'(k')] = 0 \quad (18)$$

Or

$$u'(c) = \beta u'(c')f'(k') \quad (19)$$

This is the familiar Euler equation of the two-period model, which states the the marginal cost of present consumption sacrifice should equal to the discounted marginal benefit of future return. Or

$$MRS_{c,c'} = \frac{u'(c)}{\beta u'(c')} = f'(k') \quad (20)$$

The marginal rate of substitution of present consumption for future consumption should be equal to the marginal product of capital.

Alternatively, explicitly including the time arguments, the Euler equation can be written as

$$-u'(f(k_t) - k_{t+1}) + \beta[u'(f(k_{t+1}) - k_{t+2})f'(k_{t+1})] = 0 \quad (21)$$

However, Euler equation (21) is not sufficient for optimality. Additionally, we need the transversality condition. The transversality condition is essential in infinite-horizon problems, because it ensures that there are no beneficial simultaneous changes in an infinite number of choice variables. As we see in the finite optimal growth model in Section 1, we need the end point condition that $\mu_T k_{T+1} = 0$ that either the marginal value of capital is zero or the capital of the end is zero. In general, the transversality condition takes the form:

$$\lim_{t \rightarrow \infty} \beta^t v'(k_t) k_t = \beta^t u'(f(k_t) - k_{t+1}) f'(k_t) k_t = 0 \quad (22)$$

It states that the discounted value of the capital should be asymptotically zero. This corresponds to the end point condition of finite problem $\mu_T k_{T+1} = 0$ that the value of the end period capital should be zero.

Theorem 3. *Let $\{k_t^*\}_{t=0}^\infty$ be a sequence of capital with $k_0^* = k_0$. The sequence is optimal for sequence problem if and only if it satisfies both the Euler equation (21) and transversality condition (22).*

Proof. (Sufficiency). We know that the utility function is concave $u(k_t, k_{t+1})$. Then

$$u(k_t, k_{t+1}) - u(k_t^*, k_{t+1}^*) \leq u'(f(k_t^*) - k_{t+1}^*) f'(k_t^*)(k_t - k_t^*) + [-u'(f(k_t^*) - k_{t+1}^*)](k_{t+1} - k_{t+1}^*), \forall k_t$$

Let

$$\begin{aligned}
\Delta &= \lim_{T \rightarrow \infty} \sum_{t=0}^T \beta^t [u(f(k_t^*) - k_{t+1}^*) - u(f(k_t) - k_{t+1})] \\
&\geq \lim_{T \rightarrow \infty} \sum_{t=0}^T \beta^t [u'(f(k_t^*) - k_{t+1}^*) f'(k_t^*)(k_t^* - k_t) + [-u'(f(k_t^*) - k_{t+1}^*)](k_{t+1}^* - k_{t+1})] \\
&= \lim_{T \rightarrow \infty} \sum_{t=0}^T \beta^t [-u'(f(k_t^*) - k_{t+1}^*) + \beta u'(f(k_{t+1}^*) - k_{t+2}^*) f'(k_{t+1}^*)(k_{t+1}^* - k_{t+1})] \\
&\quad - \lim_{T \rightarrow \infty} \beta^{T+1} u'(f(k_{T+1}^*) - k_{T+2}^*) f'(k_{T+1}^*)(k_{T+1}^* - k_{T+1}) \\
&= \lim_{T \rightarrow \infty} \sum_{t=0}^T \beta^t [-u'(f(k_t^*) - k_{t+1}^*) + \beta u'(f(k_{t+1}^*) - k_{t+2}^*) f'(k_{t+1}^*)(k_{t+1}^* - k_{t+1})] \\
&\quad - \lim_{T \rightarrow \infty} \beta^{T+1} u'(f(k_{T+1}^*) - k_{T+2}^*) f'(k_{T+1}^*) k_{T+1} + \lim_{T \rightarrow \infty} \beta^{T+1} u'(f(k_{T+1}^*) - k_{T+2}^*) f'(k_{T+1}^*) k_{T+1} \\
&= \lim_{T \rightarrow \infty} \beta^{T+1} u'(f(k_{T+1}^*) - k_{T+2}^*) f'(k_{T+1}^*) k_{T+1} \geq 0
\end{aligned}$$

where the first inequality comes from the property of concavity stated above, the second and third equality comes from rearranging the terms, the last equality comes from application of the Euler equation and the transversality condition. Then the sequence that satisfies the Euler equation and transversality condition is the solution of the sequence problem.

$$\sum_{t=0}^{\infty} \beta^t [u(f(k_t^*) - k_{t+1}^*)] \geq \sum_{t=0}^{\infty} \beta^t [u(f(k_t) - k_{t+1})] \tag{23}$$

□

We omit the necessary part of this theorem since it's a little complicated. You should notice that the two conditions of Euler equation and transversality condition are both important for the optimal solution. Remember in the finite case, if the Euler equation is satisfied but $k_{T+1} > 0$ in the end, this kind of plan can't be the solution.

We can also solve the FE **without substituting** c_t .

$$v(k) = \max_{c, k'} u(c) + \beta v(k'),$$

$$s.t. \quad c + k' = f(k)$$

k , given

Basically, this is an optimization problem with equality constraint and so the Lagrangian multiplier method can be applied. Form the Lagrangian function:

$$L(c, k', \lambda) = u(c) + \beta v(k') + \lambda(f(k) - c - k')$$

FOCs:

$$L_c = u'(c) - \lambda = 0$$

$$L_{k'} = \beta v'(k')\lambda = 0$$

$$L_\lambda = f(k) - c - k' = 0$$

Combine these conditions:

$$\beta v'(k') = u'(c)$$

$$c = f(k) - k'$$

Still we have one unknown $v'(k)$. Remember $v(k)$ is the value function of the maximization problem and its derivative is equal to the partial derivative of the Lagrangian function with respect to the state variable, which is the envelope theorem.

$$v'(k) = \frac{\partial L}{\partial k} = \lambda f'(k)$$

Substitute it into the FOCs:

$$u'(c) = \beta u'(c')f'(k')$$

$$c = f(k) - k'$$

Then we can get the same Euler equation as the above. And the transversality condition is the same:

$$\lim_{t \rightarrow \infty} \beta^t v'(k_t) k_t = 0$$

The optimal path $\{c_t^*, k_{t+1}^*\}_{t=0}^\infty$ is characterized by four conditions: The four conditions characterizing the optimal solution is similar to those in the finite-horizon problem. The

Euler Condition:	$u'(c_t) = \beta A\alpha k_{t+1}^{\alpha-1} u'(c_{t+1})$
Resource Constraint:	$c_t + k_{t+1} = Ak_t^\alpha$
Initial Condition:	k_0 is given
Transversality Condition:	$\lim_{t \rightarrow \infty} \beta^t u'(c_t) f'(k_t) k_t = 0$

only difference lies on the transversality condition.

Pareto Optimum The PO of this economy is defined as an allocation $\{c_t^*, k_{t+1}^*\}_{t=0}^\infty$ which solves the social planner's problem. And this allocation $\{c_t^*, k_{t+1}^*\}_{t=0}^\infty$ is characterized by the above four conditions.

Steady State of the Solution We now consider the long-run equilibrium properties of the optimal growth model. The long-run equilibrium is a static solution, implying that in the deterministic scenario without stochastic shocks, consumption and capital will be constant in the long run. Thus $c_t = \bar{c}, k_t = \bar{k}, t \geq T_0$ for some T_0 . The steady state should also satisfy the first order conditions:

$$u'(\bar{c}) = \beta A\alpha \bar{k}^{\alpha-1} u'(\bar{c})$$

$$\bar{c} + \bar{k} = A\bar{k}^\alpha$$

We can get the steady state of the optimal growth model:

$$\bar{k} = (A\alpha\beta)^{\frac{1}{1-\alpha}}, \bar{c} = A\bar{k}^\alpha - \bar{k}$$

It is not necessary that the steady state is unique. In some models, there are multiple steady states.

Dynamics of the Solution In macroeconomics, we are interested how the state and control evolve from the initial state to the steady state. This dynamic analysis is illustrated in **phase diagram**. In the (k, c) space, we can analyze how each point (k_t, c_t) in the space evolves by calculating $(k_{t+1} - k_t, c_{t+1} - c_t)$. If both are positive, the point (k_t, c_t) moves northeastward. If both are negative, the point (k_t, c_t) moves southwestward. Likewise, you can analyze the other two scenarios.

Before calculating the difference of (k, c) , we introduce one tool of calculus the Taylor series. The Taylor series of a real function $f(x)$ that is infinitely differentiable at a real number a is the power series:

$$f(x) = f(a) + \frac{f'(a)}{1!}(x - a) + \frac{f''(a)}{2!}(x - a)^2 + \frac{f'''(a)}{3!}(x - a)^3 + \dots$$

where $n!$ denotes the factorial of n . Or,

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x - a)^n$$

Apply the Taylor series theorem and omit higher-order terms.

$$u'(c_{t+1}) \approx u'(c_t) + \frac{u''(c_t)}{1!}(c_{t+1} - c_t)$$

$$\begin{aligned} u'(c_t) &= \beta A \alpha k_{t+1}^{\alpha-1} [u'(c_t) + \frac{u''(c_t)}{1!}(c_{t+1} - c_t)] \\ c_{t+1} - c_t &= -\frac{u'}{u''} \left[1 - \frac{1}{\beta A \alpha k_{t+1}^{\alpha-1}} \right] \end{aligned}$$

Then if $c_{t+1} - c_t > 0$,

$$-\frac{u'}{u''} \left[1 - \frac{1}{\beta A \alpha k_{t+1}^{\alpha-1}} \right] > 0$$

$$1 - \frac{1}{\beta A \alpha k_{t+1}^{\alpha-1}} > 0$$

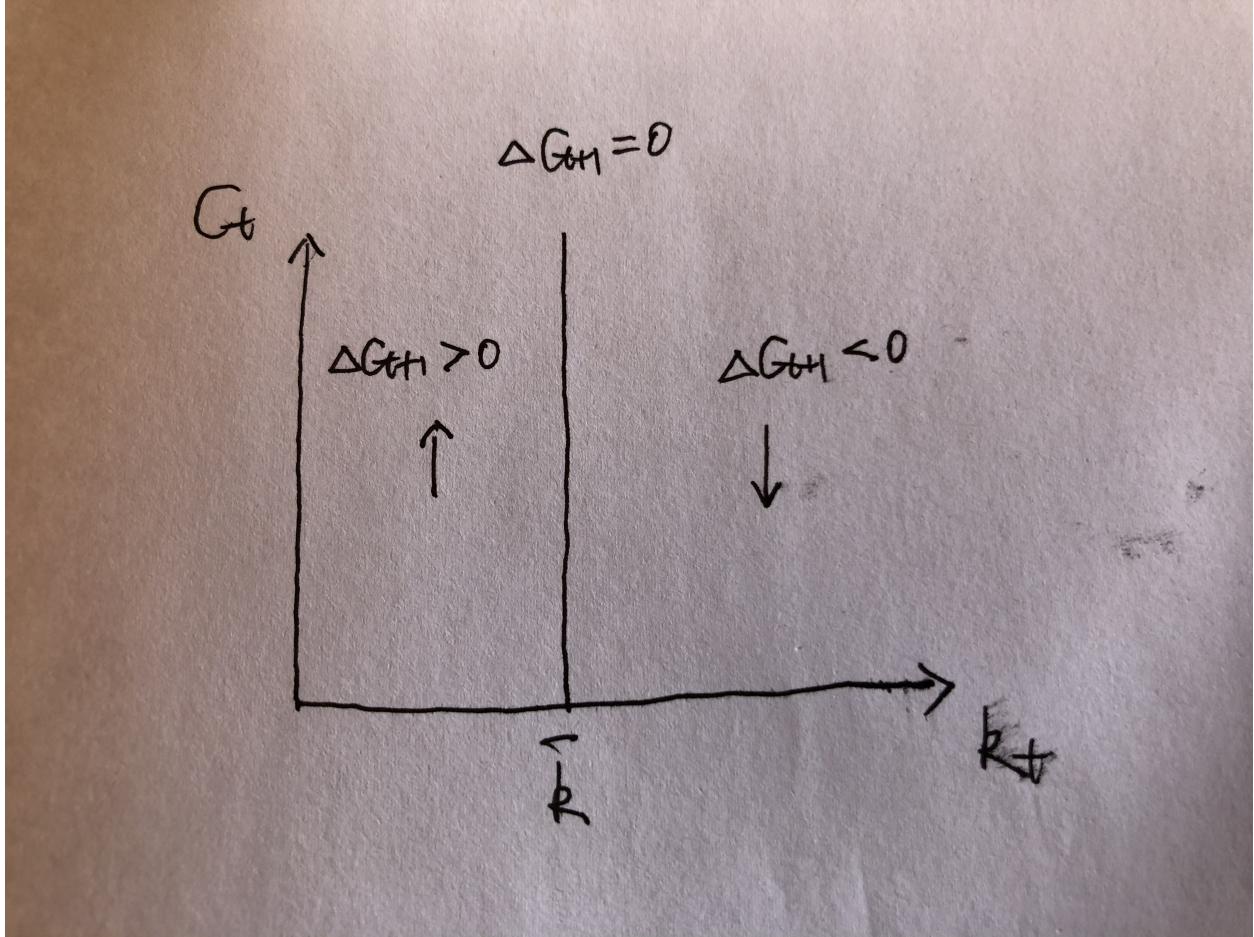
$$A \alpha \beta k_{t+1}^{\alpha-1} > 1$$

$$k_{t+1}^{1-\alpha} < A \alpha \beta$$

$$k_{t+1} < (A \alpha \beta)^{\frac{1}{1-\alpha}} = \bar{k}$$

When $k_{t+1} > \bar{k} = (A \alpha \beta)^{\frac{1}{1-\alpha}}$, $c_{t+1} - c_t < 0$. When $k_{t+1} < \bar{k} = (A \alpha \beta)^{\frac{1}{1-\alpha}}$, $c_{t+1} - c_t > 0$. And when $k_{t+1} = \bar{k} = (A \alpha \beta)^{\frac{1}{1-\alpha}}$, $c_{t+1} - c_t = 0$ and thus c_t stays. If we perturb a little from the steady state (\bar{k}, \bar{c}) , This can be shown in Figure 3. To the left, c_t moves up. To the right of \bar{k} , c_t moves down.

Figure 3: Phase Diagram



Then let's calculate $k_{t+1} - k_t$.

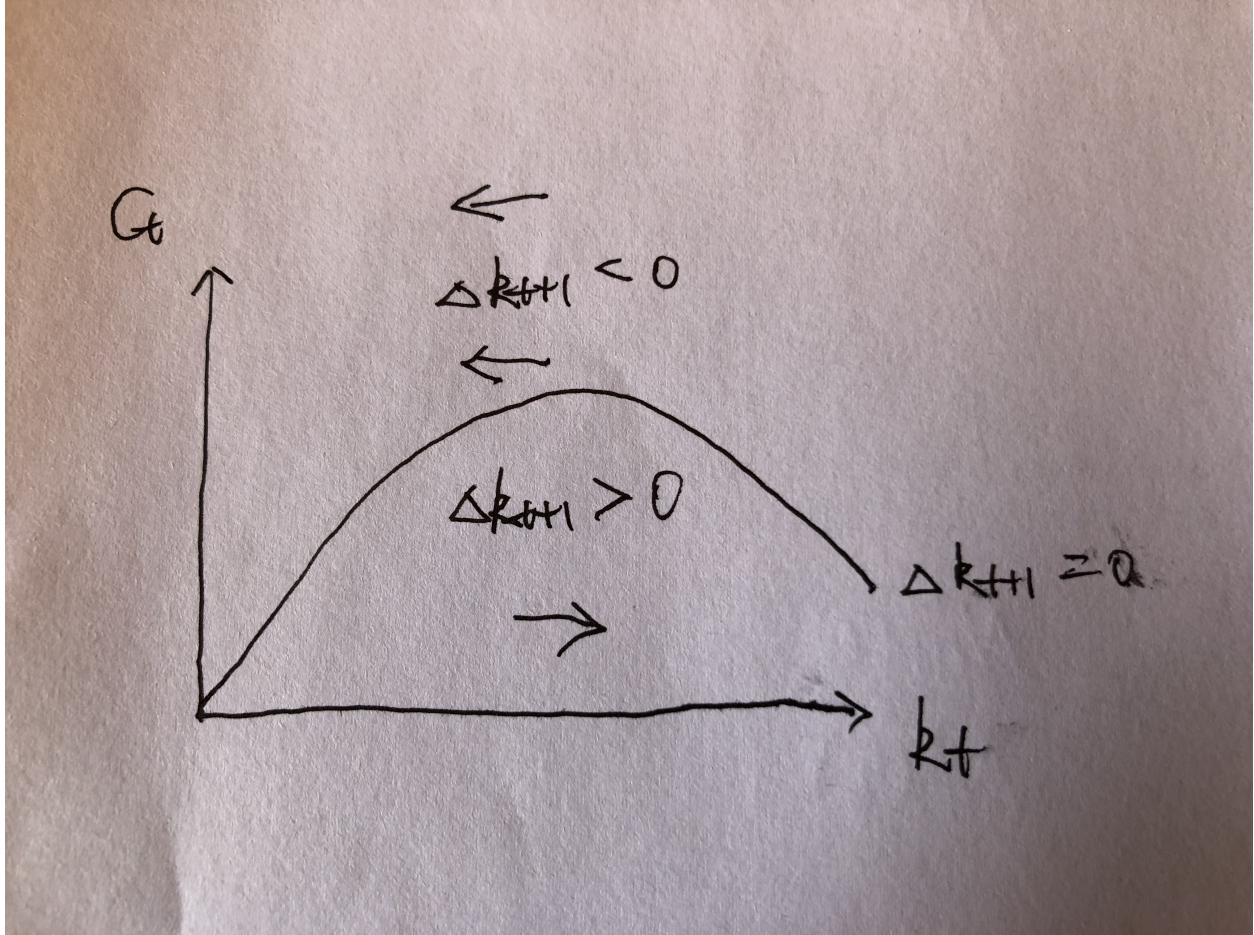
$$c_t + k_{t+1} = Ak_t^\alpha$$

$$k_{t+1} - k_t = Ak_t^\alpha - k_t - c_t$$

When $c_t > Ak_t^\alpha - k_t$, k_t moves down. When $c_t < Ak_t^\alpha - k_t$, k_t moves up. When $c_t = Ak_t^\alpha - k_t$, k_t stays. This is illustrated in Figure 4.

If we combine $(c_{t+1} - c_t, k_{t+1} - k_t)$ together in one diagram, the two curves $k_t = \bar{k}$ and $c_t = Ak_t^\alpha - k_t$ divide the (c, k) space into four regions. The points in region 1 move northwestward. The points in region 2 move southwestward. The points in region 3 move northeastward. The points in region 4 move southeastward. The only stable path goes

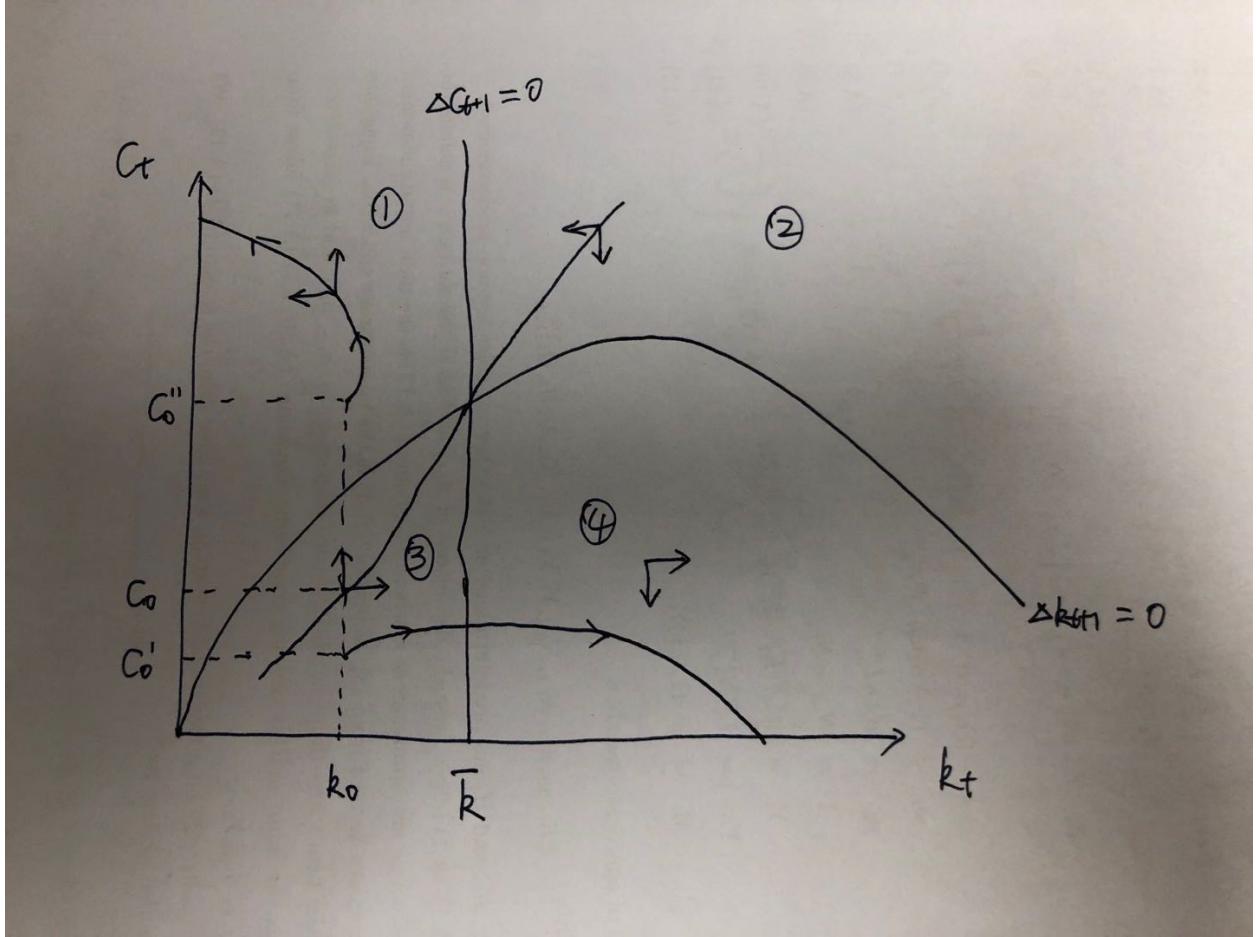
Figure 4: Phase Diagram



through region 2 and 3 which always converges to the steady state. The line through the steady state point is known as the **saddle path**, or **stable manifold**. Only points on this line are attainable. It means that whenever the economy starts at an arbitrary k_0 , $\{c_t, k_{t+1}\}_{t=0}^{\infty}$ will be chosen as what the saddle path indicates. Ask yourself a question, any path off the saddle path is possible for the equilibrium?

The answer is no. Why? Think about the Euler equation and transversality condition which are both sufficient and necessary conditions for the optimal solution. Any path on the phase diagram meets the first order conditions since the phase diagram is drawn through Euler equation and resource constraint. However, only the saddle path simultaneously satisfies the transversality condition, which can be shown in Figure 6. Any path off the saddle

Figure 5: Phase Diagram



path violates the transversality condition. Imagine that for k_0 , c_0 is not on the saddle path, for example c_0' or c_0'' on Figure 5. The point either goes to the k axis (the case of c_0') or c (the case of c_0'') axis. Either case will violate the transversality condition in the end since $f'(k_t)$ and $f'(c_t)$ are infinite term in the finite time so that $\beta^t u'(c_t) f'(k_t) k_t = \infty$. The transversality condition determine the choice of c_t for the state k_t , which is the saddle path. We can see this point from the following experiment. We know that ² the saddle path for the consumption in Period 0 is $c_0 = (1 - \alpha\beta)Ak_0^\alpha$. We can check numerically whether other path that $c_0 > (1 - \alpha\beta)Ak_0^\alpha$ or $c_0 < (1 - \alpha\beta)Ak_0^\alpha$ which satisfies the first order conditions satisfy the transversality condition. The transversality conditions do not hold in these two scenarios, shown in Figure 7 and Figure 8.

²This result will be shown in the next subsection.

Figure 6: Saddle Path and Transversality Condition

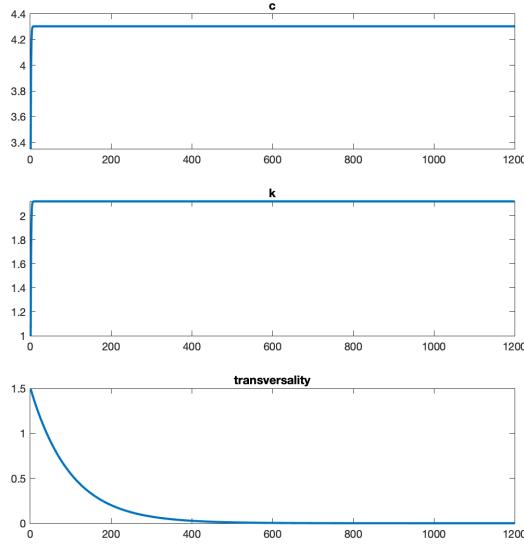
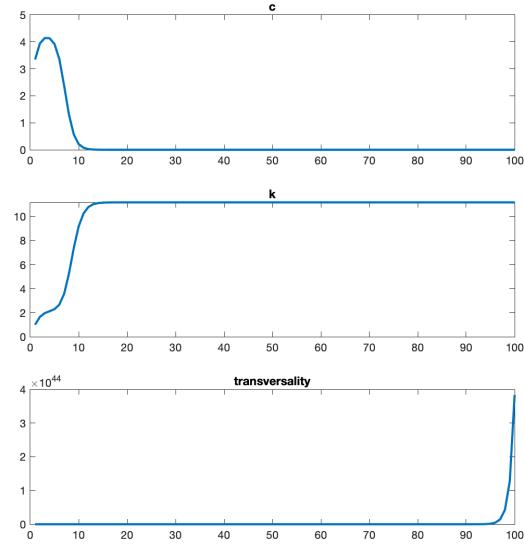
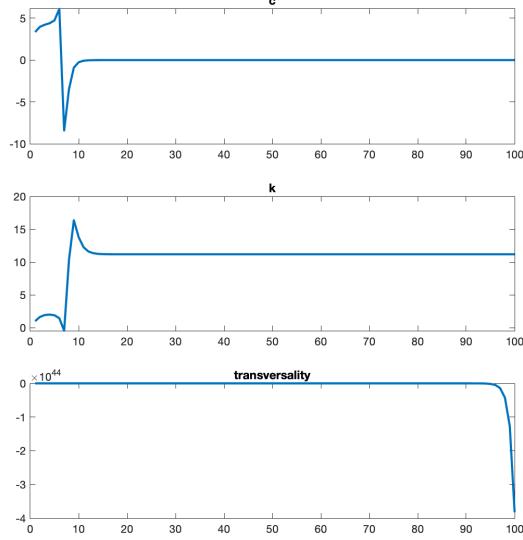


Figure 7: Saddle Path and Transversality Condition



Local Dynamics around the Steady State The above phase diagram is an analysis about the dynamics of all points along the solution. Sometimes, we are interested about the dynamics in the neighbourhood of the steady state. We call it **local dynamics** around the

Figure 8: Saddle Path and Transversality Condition



steady state. Consider the first order conditions:

$$u'(c_t) = \beta u'(c_{t+1}) f'(k_{t+1})$$

$$c_t + k_{t+1} = f(k_t)$$

We can apply the Taylor series theorem to get the first-order Taylor expansion around the steady state to the LHS and RHS of both equations. Let $\tilde{c}_t = c_t - \bar{c}$ and $\tilde{k}_t = k_t - \bar{k}$.

$$\begin{aligned} u'(\bar{c}) + u''(\bar{c})\tilde{c}_t &= \beta(u'(\bar{c}) + u''(\bar{c})\tilde{c}_{t+1})(f'(\bar{k}) + f''(\bar{k})\tilde{k}_{t+1}) \\ \bar{c} + \tilde{c}_t + \bar{k} + \tilde{k}_{t+1} &= f(\bar{k}) + f'(\bar{k})\tilde{k}_t \end{aligned}$$

We know that

$$u'(\bar{c}) = \beta u'(\bar{c}) f'(\bar{k})$$

$$\bar{c} + \bar{k} = f(\bar{k})$$

We omit all second order terms to get the linear approximation here such as the term $\tilde{c}_{t+1}\tilde{k}_{t+1}$.

$$u''(\bar{c})\tilde{c}_t = \beta u''(\bar{c})f'(\bar{k})\tilde{c}_{t+1} + \beta u'(\bar{c})f''(\bar{k})\tilde{k}_{t+1}$$

$$\tilde{c}_t + \tilde{k}_{t+1} = f'(\bar{k})\tilde{k}_t$$

Stack the two equations to the matrix:

$$\begin{bmatrix} \beta u''f' & \beta u'f'' \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{c}_{t+1} \\ \tilde{k}_{t+1} \end{bmatrix} = \begin{bmatrix} u'' & 0 \\ -1 & f' \end{bmatrix} \begin{bmatrix} \tilde{c}_t \\ \tilde{k}_t \end{bmatrix} \quad (24)$$

$$\begin{bmatrix} \tilde{c}_{t+1} \\ \tilde{k}_{t+1} \end{bmatrix} = \frac{1}{\beta u''f'} \begin{bmatrix} 1 & -\beta u'f'' \\ 0 & \beta u''f' \end{bmatrix} \begin{bmatrix} u'' & 0 \\ -1 & f' \end{bmatrix} \begin{bmatrix} \tilde{c}_t \\ \tilde{k}_t \end{bmatrix} \quad (25)$$

Or

$$\begin{bmatrix} \tilde{c}_{t+1} \\ \tilde{k}_{t+1} \end{bmatrix} = \begin{bmatrix} 1 + \frac{u'f''}{u''f'} & -\frac{u'f''}{u''} \\ -1 & f' \end{bmatrix} \begin{bmatrix} \tilde{c}_t \\ \tilde{k}_t \end{bmatrix} \quad (26)$$

where $\beta f'(\bar{k}) = 1$

Let

$$B = \begin{bmatrix} 1 + \frac{u'f''}{u''f'} & -\frac{u'f''}{u''} \\ -1 & f' \end{bmatrix}$$

and this matrix B can be diagonally-decomposed as

$$B = P\Lambda P^{-1} \quad (27)$$

where $\Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$, if the two eigenvalues λ_1, λ_2 of B are distinct, and P is the matrix each column of which is the corresponding eigenvector of each eigenvalue. Multiply $Q = P^{-1}$ to the left of $(\tilde{c}_t, \tilde{k}_t)'$, we have

$$P^{-1} \begin{bmatrix} \tilde{c}_t \\ \tilde{k}_t \end{bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \begin{bmatrix} \tilde{c}_t \\ \tilde{k}_t \end{bmatrix} = \begin{bmatrix} Q_{11}\tilde{c}_t + Q_{12}\tilde{k}_t \\ Q_{21}\tilde{c}_t + Q_{22}\tilde{k}_t \end{bmatrix}$$

Multiply P^{-1} to the left of both sides of the Equation (26).

$$P^{-1} \begin{bmatrix} \tilde{c}_{t+1} \\ \tilde{k}_{t+1} \end{bmatrix} = P^{-1}(P\Lambda P^{-1}) \begin{bmatrix} \tilde{c}_t \\ \tilde{k}_t \end{bmatrix} = \Lambda P^{-1} \begin{bmatrix} \tilde{c}_t \\ \tilde{k}_t \end{bmatrix} \quad (28)$$

Or

$$\begin{bmatrix} Q_{11}\tilde{c}_{t+1} + Q_{12}\tilde{k}_{t+1} \\ Q_{21}\tilde{c}_{t+1} + Q_{22}\tilde{k}_{t+1} \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} Q_{11}\tilde{c}_t + Q_{12}\tilde{k}_t \\ Q_{21}\tilde{c}_t + Q_{22}\tilde{k}_t \end{bmatrix} \quad (29)$$

Iterate the above equation and we have:

$$\begin{bmatrix} Q_{11}\tilde{c}_t + Q_{12}\tilde{k}_t \\ Q_{21}\tilde{c}_t + Q_{22}\tilde{k}_t \end{bmatrix} = \begin{bmatrix} \lambda_1^t & 0 \\ 0 & \lambda_2^t \end{bmatrix} \begin{bmatrix} Q_{11}\tilde{c}_0 + Q_{12}\tilde{k}_0 \\ Q_{21}\tilde{c}_0 + Q_{22}\tilde{k}_0 \end{bmatrix} = \begin{bmatrix} \lambda_1^t(Q_{11}\tilde{c}_0 + Q_{12}\tilde{k}_0) \\ \lambda_2^t(Q_{21}\tilde{c}_0 + Q_{22}\tilde{k}_0) \end{bmatrix} \quad (30)$$

Now we can see the stability of the system around the steady state. If we perturb a little from the steady state $(\bar{c}, \bar{k})'$, the location of $(c_t, k_t)'$ moves following (30). We say that this system is stable if $(\tilde{c}_t, \tilde{k}_t)$ converges to zero (thus $(c_t, k_t)'$ converges to the steady state $(\bar{c}, \bar{k})'$).

- $|\lambda_1| < 1, |\lambda_2| < 1$. Then the system is universally stable. Whatever value of the choice variable c_0 makes the system stable. In this case, there are multiple paths that $(c_t, k_t)'$ converges to the steady state $(\bar{c}, \bar{k})'$.
- $|\lambda_1| < 1, |\lambda_2| > 1$ ³. Then this system is divergent unless $Q_{21}\tilde{c}_0 + Q_{22}\tilde{k}_0 = 0$. This condition nails down the choice of $\tilde{c}_0 = -\frac{Q_{22}}{Q_{21}}\tilde{k}_0$ in the dynamic system. And this choice is the saddle path. Once \tilde{c}_0 is nailed down, we know that

$$\tilde{c}_t = -\frac{Q_{22}}{Q_{21}}\tilde{k}_t \quad (31)$$

$$\tilde{k}_t = \frac{\lambda_1^t(Q_{11}\tilde{c}_0 + Q_{12}\tilde{k}_0)}{-Q_{11}Q_{22}/Q_{21} + Q_{12}} \quad (32)$$

- $|\lambda_1| > 1, |\lambda_2| > 1$. This system is divergent no matter what choice of \tilde{c}_0 is. There is no path that $(c_t, k_t)'$ converges to the steady state $(\bar{c}, \bar{k})'$.

In our application of optimal growth model $\lambda_1 + \lambda_2 = 1 + \frac{(1-\alpha)\beta(1-\alpha)}{\alpha\beta} + \frac{1}{\beta}$, $\lambda_1\lambda_2 = \frac{1}{\beta}$, we can check that one eigenvalue is bigger than 1 and the other eigenvalue is less than one. So

³We assume $\lambda_1 < \lambda_2$.

there is one stable path which is the saddle path. We can use the same parameters as the finite-horizon case and see the local dynamics around the steady state. You may want to check the Matlab program to see the linearized version of this model. Play with it and see the above points.

```

1 %=====
2 % Chapter 2
3 % File: ch2_local_dynamics.m
4 % This program file is to solve check the local dynamics around the steady
5 % state of the optimal growth model
6 % Written: 2020.10.26
7 % Written by Bin Wang
8 %=====

9
10 clear all;
11 close all;
12 clc;

13
14 %% Parameters
15 A = 5;
16 alpha = 1/3;
17 beta = 0.99;
18 T = 100;
19 kbar = (A*alpha*beta)^(1/(1-alpha));
20 cbar = A*kbar^alpha-kbar;
21 fp = A*alpha*kbar^(alpha-1); %f'(kbar)
22 fpp = A*alpha*(alpha-1)*kbar^(alpha-2); %f''(kbar)
23 up = 1/cbar; %u'(cbar)
24 upp = -1/(cbar^2); %u''(cbar)
25 B1 = [beta*upp*fp beta*up*fpp; 0 1];
26 B2 = [upp 0; -1 fp];
27 B = B1\B2; %B matrix in the lecture notes

28
29 %diagonally-decompose the B matrix
30 [P Lambda]= eig(B);

31
32 %eigenvalue should be ordered asendingly
33 [Lambda_sort,index]=sort(diag(Lambda));
34 P_sort = P(:,index);
35 Lambda = diag(Lambda_sort);
36 P = P_sort;

37
38 %% You can check that one eigenvalue is less than one and one is bigger than
39 %one. So there is one stable path which is the saddle path.

```

```

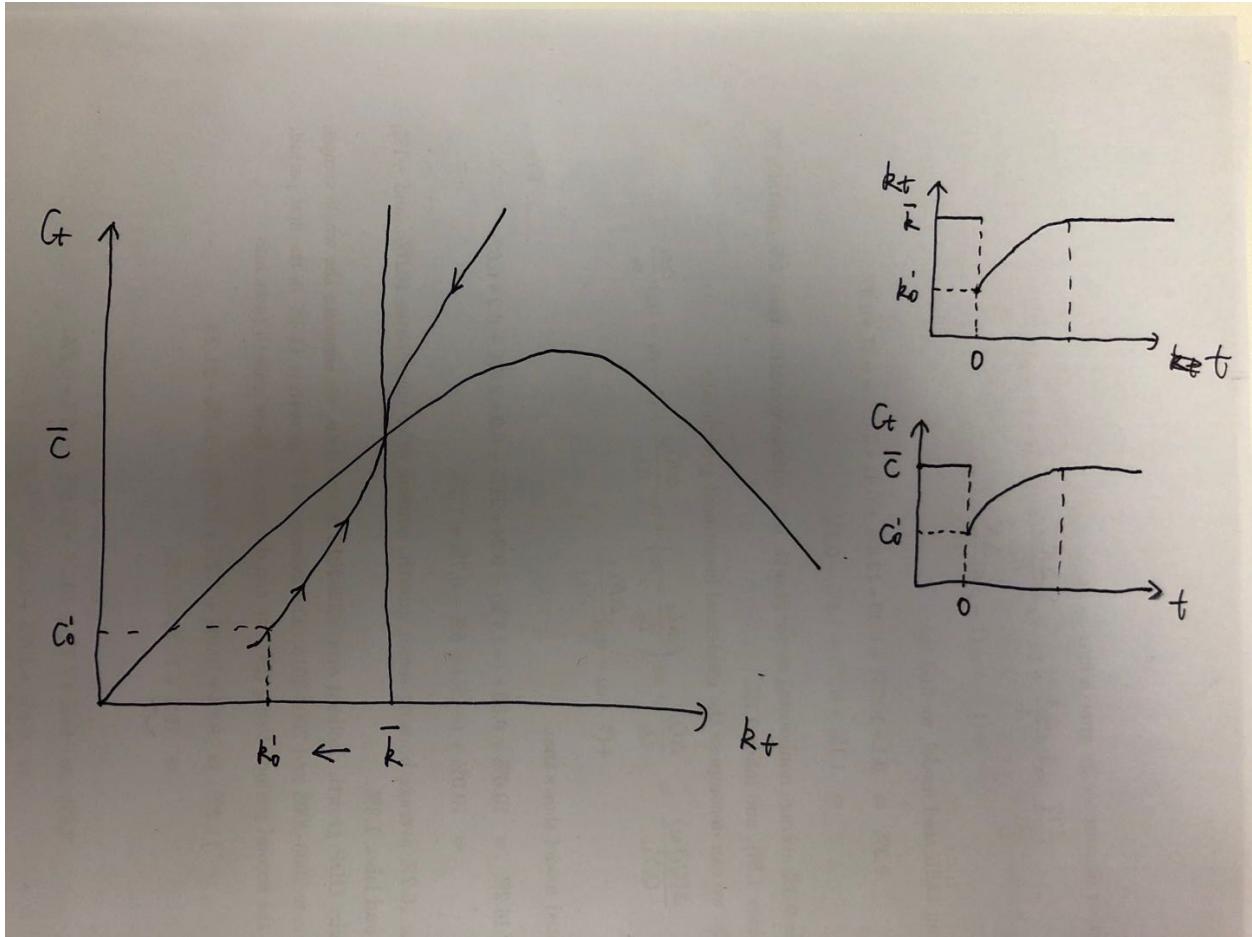
40 Q = inv(P);
41
42 %simulate the path
43 T=100;
44 k0 = 1/3*kbar;
45 k_tilde=zeros(T,1);
46 c_tilde=zeros(T,1);
47 k_tilde(1) = k0-kbar; %\tilde{k}_0$
48 c_tilde(1) = -Q(2,2)/Q(2,1)*k_tilde(1);
49 for t=2:T
50     k_tilde(t)=Lambda(1,1)^(t-1)*(Q(1,1)*c_tilde(1)+Q(2,2)*k_tilde(1))/(-Q(1
51         c_tilde(t)=-Q(2,2)/Q(2,1)*k_tilde(t);
52 end
53
54 figure;
55 plot(1:T,k_tilde,'LineWidth',2);
56 title('k_tilde');
57
58 figure;
59 plot(1:T,c_tilde,'LineWidth',2);
60 title('c_tilde');

```

Transition Dynamics In the previous chapter, we intensively use implicit function theorem to find the comparative statics of the static model. In the infinite-horizon model, the solution of the model is a infinite-sequence that maximize the sum of discounted utility. If there is an exogenous shock to the economy, the steady state of the economy will transit to another steady state. The two steady states are compared by the comparative static analysis. Additionally, we can see the process how the economy transit from the original steady state to the new steady state, which is also part of the solution.

For example, there is an exogenous shock to the capital that the typhoon destroys part of the existing capital \bar{k} . Then the state variable moves down. For example, capital becomes $k'_0 < \bar{k}$. Since the deep parameters do not change, the solution of the model doesn't change including the new steady state. Then the new c_t, k_t still stays on the original saddle path and converges to the original steady state, shown in Figure 9. This analysis is called **transition dynamics**.

Figure 9: Transition Dynamics



2.2 Dynamic Programming: General Form

A typical dynamic programming problem is to choose an infinite sequence of controls $\{u_t\}_{t=0}^{\infty}$ to maximize

$$\max_{\{u_t\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t r(x_t, u_t)$$

$$s.t. \quad x_{t+1} = g(x_t, u_t)$$

$$x_0 \in \mathbf{R}^n, \text{ given}$$

Dynamic programming seeks a time-invariant *policy function* h mapping the *state* x_t into the control u_t , such that the sequence $\{x_t\}_{t=0}^{\infty}$ generated by iterating the two functions.

$$u_t = h(x_t)$$

$$x_{t+1} = g(x_t, u_t)$$

Maximizing the objective bears the value function v :

$$v(x_0) = \max_{\{u_t\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t r(x_t, u_t)$$

$$s.t. \quad x_{t+1} = g(x_t, u_t)$$

x_0 , given

The SP can be solved by solving the following FE:

$$v(x) = \max_u r(x, u) + \beta v(x') \tag{33}$$

$$s.t. \quad x' = g(x, u) \tag{34}$$

The maximizer of the right hand side of the FE is a policy function $h(x)$ that satisfies:

$$v(x) = r[x, h(x)] + \beta v[g(x, h(x))]$$

We already show that in the optimal growth model the solution of the SP is the same as that of FE. In most of the problems we explore, Theorem 2 holds under fairly general conditions. Then we can just apply this theorem to find the solution of SP by solving the easier Bellman equation.

The first order condition for the FE (33) is:

$$r_u(x, u) + \beta v_x(g(x, u))g_u(x, u) = 0 \tag{35}$$

Notice that the first order condition holds at the optimal point $u = h(x)$. According to the Benveniste and Scheinkman envelope theorem, we can get the derivative of $v(x)$:

$$v_x(x) = \frac{\partial[r(x, u) + \beta v(g(x, u))]}{\partial x} = r_x(x, u) + \beta v_x(g(x, u))g_x(x, u) \tag{36}$$

For our problems, x' is not related to x so that $x' = g(x, u) = g(u)$ and then $g_x(x, u) = 0$. Then the second term disappears.

$$v_x(x) = \frac{\partial[r(x, u) + \beta v(g(x, u))]}{\partial x} = r_x(x, u) \quad (37)$$

Or you can directly differentiate $v(x)$ **without applying the envelope theorem**.

$$v(x) = r[x, h(x)] + \beta v[g(x, h(x))] \quad (38)$$

Then

$$\begin{aligned} v_x(x) &= r_x(x, h(x)) + r_u(x, h(x))h'(x) + \beta v_x(g(x, h(x)))[g_x(x, h(x)) + g_u(x, h(x))h'(x)] \\ &= r_x(x, h(x)) + \beta v_x(g(x, h(x)))g_x(x, h(x)) + h'(x)[r_u(x, h(x)) + \beta v_x(g(x, h(x)))g_u(x, h(x))] \\ &= r_x(x, h(x)) + \beta v_x(g(x, h(x)))g_x(x, h(x)) \\ &= r_x(x, h(x)) \end{aligned}$$

where the first equality comes from totally differentiating equation (38), the second comes from rearranging the terms, the third comes from applying the Euler equation, the fourth comes from $g_x(x, u) = 0$ ⁴.

Notice that the derivative of value function is also evaluated only at the optimal point $u = h(x)$. Substitute the derivative of value function (37) into the first order condition (35).

$$r_u(x, h(x)) + \beta r_x(g(x, h(x)), h(g(x, h(x))))g_u(x, h(x)) = 0 \quad (39)$$

This is the Euler equation of the dynamic programming. It states that at the optimal point the objective cannot be increased by changing the controls of any period. At the optimal point generated by $u = h(x)$, $x' = g(x, u)$, if we disturb u a bit, the decrease of present utility flow will be $r_u(x, h(x))$. The change of u brings about the change of the state of next period by $g_u(x, h(x))$, which brings about the change of future utility by $r_x(g(x, h(x)), h(g(x, h(x))))$. The multiplication of both, discounted to the present period, should be equal to the decrease of present utility flow $r_u(x, h(x))$ (or the sum of both should be zero).

⁴In the optimal growth model, we have $x_t = k_t$, $u_t = k_{t+1}$, $r(x_t, u_t) = \ln(Ak_t^\alpha - k_{t+1}) = \ln(Ax_t^\alpha - u_t)$, $g(x_t, u_t) = x_{t+1} = k_{t+1} = u_t = g(u_t)$. So that $g_x(x_t, u_t) = 0$.

The transversality condition is:

$$\lim_{t \rightarrow \infty} \beta^t v_x(x_t, u_t) x_t = 0 \quad (40)$$

The transversality condition states that either the value of the state, discounted to the present, should be zero or the state, discounted to the present, should be zero at the optimal plan.

You should be very clear that both Euler equation and transversality condition are sufficient and necessary. Absence of either one will not characterize the optimal solution of the dynamic programming problem. For the most of the problems we explore, Theorem 3 holds under fairly general conditions such as the utility function should be concave, twice differentiable and the constraint should be convex and compact. Students who are interested in details of these conditions should consult Chapter 5 of Acemoglu (2009).

2.3 Stochastic Control Problems

Consider the optimal growth model with uncertainty when we allow that technology A to change with time. We generally assume that the technology A_t is a Markov chain (\bar{A}, π_0, P) with

$$Prob(A_{t+1} = \bar{A}_j | A_t = \bar{A}_i) = P_{ij}.$$

The social planner is to maximize the discounted sum of expected utility given k_0 and A_0 .

$$\begin{aligned} & \max_{\{c_t, k_{t+1}\}_{t=0}^{\infty}} E_0 \sum_{t=0}^{\infty} \beta^t u(c_t) \\ \text{s.t. } & c_t + k_{t+1} \leq A_t k_t^{\alpha} \\ & c_t, k_{t+1} \geq 0 \\ & Prob(A_{t+1} = \bar{A}_j | A_t = \bar{A}_i) = P_{ij} \\ & k_0, A_0, \text{given} \end{aligned}$$

The notation E_0 is a conditional expectation which means $E(\cdot | A_0)$. Basically, the conditional expectation is a function of the information of the condition, the technology of Period 0,

which is exogenously given.

The stochastic control problem still have a recursive structure. The FE is:

$$v(k, A) = \max_{c, k'} u(c) + \beta E_t[v(k', A')]$$

$$s.t. \quad c + k' = Ak^\alpha$$

$$\text{Prob}(A' = \bar{A}_j | A = \bar{A}_i) = P_{ij}$$

Where $E_t[v(k', A')]$ is just a conditional expectation conditional on realization of present technology A . Here in the stochastic case, we have two state variables k_t and A_t which are known in Period t . We also call A_t **exogenous state variable** since A_t is not influenced by choice variables in the problem. Substitute the definition of conditional expectation.

$$v(k, \bar{A}_i) = \max_{c, k'} u(c) + \beta \sum_{j=1}^n v(k', \bar{A}_j) P_{ij}$$

$$s.t. \quad c + k' = Ak^\alpha$$

where $\text{Prob}(A' = \bar{A}_j | A = \bar{A}_i) = P_{ij}$. Here in this stochastic problem, we need to keep track of the exogenously given stochastic technology, except for present capital. Then there are two state variables k and A in the stochastic case. We can still differentiate k' to the objective and get the first order condition.

$$-u'(Ak^\alpha - k') + \beta E[v_k(k', A')|A] = 0$$

From the envelope theorem

$$v_k(k, A) = u'(Ak^\alpha - k')A\alpha k^{\alpha-1}$$

Substitute the derivative of value function into the first order condition.

$$-u'(Ak^\alpha - k') + \beta E[u'(A'k'^\alpha - k'')A'\alpha k'^{\alpha-1}|A] = 0$$

This is the Euler equation in the stochastic problem and we can also write down the transver-

sality condition.

$$\lim_{t \rightarrow \infty} E_0 \beta^t v_k(k_t, A_t) k_t = 0$$

Then the solution of the stochastic optimal growth model is characterized by the four conditions: Every condition is essential for the solution. Absence of anyone will result a

Euler Condition:	$u'(c_t) = \beta E_t A_{t+1} \alpha k_{t+1}^{\alpha-1} u'(c_{t+1})$
Resource Constraint:	$c_t + k_{t+1} = A_t k_t^\alpha$
Initial Condition:	k_0, A_0 are given
Transversality Condition:	$\lim_{t \rightarrow \infty} \beta^t v_k(k_t, A_t) k_t = 0$

failure to find the solution.

2.4 Examples

2.4.1 Decentralized Growth Model

The optimal growth model is actually a centralized growth model which is a Pareto Optimum. Here we decentralize the model and show the competitive equilibrium of the neoclassical growth model.

The Representative Consumer The representative consumer is endowed with initial wealth a_0 in the beginning of Period 0. The consumer rents his wealth capital to the firm and works for the firm, and get capital rent $r_t a_t$, wage income w_t , and the profit redistributed by the firm π_t each period. The consumer decides c_t and a_{t+1} to maximize his lifetime sum of discounted utility. The wealth a_t depreciates fully in the end of the period as in the optimal growth model.

$$\begin{aligned} & \max_{c_t, a_{t+1}} \sum_{t=0}^{\infty} \beta^t u(c_t) \\ \text{s.t. } & c_t + a_{t+1} \leq r_t a_t + w_t + \pi_t \\ & c_t, a_{t+1} \geq 0 \\ & a_0, \quad \text{given} \end{aligned}$$

Write down the Bellman equation of this sequence problem:

$$v(a) = \max_{c,a'} u(c) + \beta v(a')$$

$$s.t. \quad c + a' = ra + w + \pi$$

Substitute c into the Bellman equation

$$v(a) = \max_{a'} u(ra + w + \pi - a') + \beta v(a')$$

The FOC is:

$$-u'(ra + w + \pi - a') + \beta v'(a') = 0$$

According to the envelope theorem:

$$v'(a) = u'(ra + w + \pi - a')r$$

Substituting the marginal value of wealth into the FOC gives the Euler equation.

$$-u'(ra + w + \pi - a') + \beta u'(r'a' + w' + \pi' - a'')r' = 0$$

The transversality condition is:

$$\lim_{t \rightarrow \infty} \beta^t v'_t(a_t) a_t = 0$$

Given r_t and w_t , the wealth initial condition, the budget constraint, the Euler equation, and the transversality condition characterize the solution of the representative consumer.

The Representative Firm The representative firm decides capital and labor demand given rental rate r_t and wage rate w_t .

$$\max_{k_t, n_t} \pi_t = Ak_t^\alpha n_t^{1-\alpha} - r_t k_t - w_t n_t$$

$$s.t. \quad k_t, n_t \geq 0$$

The FOCs are:

$$r_t = A\alpha k_t^{\alpha-1} n_t^{1-\alpha}$$

$$w_t = A(1 - \alpha)k_t^\alpha n_t^{-\alpha}$$

Market Clearing The goods market, capital market, and labor market are cleared:

$$c_t + a_{t+1} = Ak_t^\alpha n_t^{1-\alpha}$$

$$a_t = k_t$$

$$1 = n_t$$

Competitive Equilibrium The competitive equilibrium is an allocation $(\{c_t\}_{t=0}^\infty, \{l_t\}_{t=0}^\infty, \{a_t\}_{t=0}^\infty, \{\pi_t\}_{t=0}^\infty, \{n_t\}_{t=0}^\infty, \{k_{t+1}\}_{t=0}^\infty)$ and a price system $(\{r_t\}_{t=0}^\infty, \{w_t\}_{t=0}^\infty)$ such that:

- Given the price system, the allocation solves the representative' problem and the representative firm's problem.
- All markets are cleared.

We can check that the CE is also a PO by collapsing the equilibrium conditions of the CE. Notice that $r_t k_t + w_t n_t = Ak_t^\alpha$, $\pi_t = 0$, and $a_t = k_t, n_t = 1$. Then the Euler equation of the consumer is:

$$-u'(Ak^\alpha - k') + \beta u'(Ak'^\alpha - k'')A\alpha k'^{\alpha-1} = 0$$

This is the same as the Euler equation of the social planner's problem, and the transversality condition is also the same. All the equilibrium conditions can be collapsed as:

Euler Condition:	$u'(c_t) = \beta A\alpha k_{t+1}^{\alpha-1} u'(c_{t+1})$,
Resource Constraint:	$c_t + k_{t+1} = Ak_t^\alpha$,
Initial Condition:	k_0 is given
Transversality Condition:	$\lim_{t \rightarrow \infty} \beta^t v'(k_t) k_t = 0$

2.4.2 Optimal Growth Model with Exogenous Labor and Technology Growth

The optimal growth model presented above can be seen as a model with fixed size of population and fixed technology over time. In this example, we assume that population and technology both grow at a fixed rate of n and $(1 - \alpha)g$ respectively. Because the population grows in this economy, we should use representative household to represent this economy, as representative consumer can't illustrate the growth of this population. As population grows at a rate of n , the representative household also grows at this rate.

$$L_t = L_0(1 + n)^t$$

$$A_t = A_0(1 + (1 - \alpha)g)^t$$

The representative household's problem is:

$$\begin{aligned} & \max_{c_t, k_{t+1}} \sum_{t=0}^{\infty} \beta^t L_t u(c_t) \\ \text{s.t. } & C_t + K_{t+1} \leq A_t K_t^\alpha L_t^{1-\alpha} \\ & C_t, K_{t+1} \geq 0 \\ & K_0, L_0, A_0, \quad \text{given} \end{aligned}$$

where $C_t = L_t c_t$, $K_t = L_t k_t$ and $u(c_t) = \frac{c_t^{1-\theta}}{1-\theta}$. Capital letters stands for the aggregate variables and lower-case letters represent per capita variables. We can normalize all aggregate variables by

$$A_t^{\frac{1}{1-\alpha}} L_t = A_0^{\frac{1}{1-\alpha}} [(1 + (1 - \alpha)g)^{\frac{1}{1-\alpha}}]^t L_0 (1 + n)^t$$

Without harm, we can assume that $L_0 = A_0 = 1$ and then

$$A_t^{\frac{1}{1-\alpha}} L_t \approx (1 + g + n)^t$$

It states that $A_t^{\frac{1}{1-\alpha}} L_t$ grows at a rate of $g + n$. Let $\hat{k}_t = \frac{K_t}{A_t^{\frac{1}{1-\alpha}} L_t} = \frac{k_t}{A_t^{\frac{1}{1-\alpha}}}$ and $\hat{c}_t = \frac{C_t}{A_t^{\frac{1}{1-\alpha}} L_t} = \frac{c_t}{A_t^{\frac{1}{1-\alpha}}}$. The aggregate consumer's problem is:

$$\begin{aligned} & \max_{c_t, k_{t+1}} \sum_{t=0}^{\infty} (\beta(1+n)(1+(1-\theta)g))^t u(\hat{c}_t) \\ \text{s.t. } & \hat{c}_t + \hat{k}_{t+1}(1+g)(1+n) \leq \hat{k}_t^\alpha \\ & \hat{c}_t, \hat{k}_{t+1} \geq 0 \\ & \hat{k}_0, \quad \text{given} \end{aligned}$$

Let $\hat{\beta} = \beta(1+n)(1+(1-\theta)g) \approx \beta(1+n+(1-\theta)g)$ and this problem looks similar to the original optimal growth model now.

$$\begin{aligned} & \max_{c_t, k_{t+1}} \sum_{t=0}^{\infty} \hat{\beta}^t u(\hat{c}_t) \\ \text{s.t. } & \hat{c}_t + \hat{k}_{t+1}(1+g)(1+n) \leq \hat{k}_t^\alpha \\ & \hat{c}_t, \hat{k}_{t+1} \geq 0 \\ & \hat{k}_0, \quad \text{given} \end{aligned}$$

Write down the Bellman equation of this sequence problem:

$$\begin{aligned} v(\hat{k}) &= \max_{\hat{c}, \hat{k}'} u(\hat{c}) + \hat{\beta}v(\hat{k}') \\ \text{s.t. } & \hat{c} + \hat{k}'(1+g)(1+n) = \hat{k}^\alpha \end{aligned}$$

You can either substitute \hat{c} and consider \hat{k}' as the only control variable, or form the Lagrangian function. We apply the first here.

$$v(\hat{k}) = \max_{\hat{k}'} u(\hat{k}^\alpha - \hat{k}'(1+g)(1+n)) + \hat{\beta}v(\hat{k}')$$

The first order condition is:

$$-u'(\hat{k}^\alpha - \hat{k}'(1+g)(1+n))(1+g)(1+n) + \hat{\beta}v'(\hat{k}') = 0$$

According to the envelope theorem:

$$v'(\hat{k}) = u'(\hat{k}^\alpha - \hat{k}'(1+g)(1+n))\alpha\hat{k}^{\alpha-1}$$

Substituting the marginal value function into the first order condition gives the Euler equation:

$$-u'(\hat{k}^\alpha - \hat{k}'(1+g)(1+n))(1+g)(1+n) + \hat{\beta}u'(\hat{k}'^\alpha - \hat{k}''(1+g)(1+n))\alpha\hat{k}'^{\alpha-1} = 0$$

And the transversality condition is:

$$\lim_{t \rightarrow \infty} \hat{\beta}^t v'(\hat{k}_t) \hat{k}_t = 0$$

The optimal solution is characterized by the initial condition, resource constraint, Euler equation, and transversality condition.

Euler Condition:	$u'(\hat{c}_t)(1+g)(1+n) = \hat{\beta}\alpha\hat{k}_{t+1}^{\alpha-1}u'(\hat{c}_{t+1}),$
Resource Constraint:	$\hat{c}_t + \hat{k}_{t+1}(1+g)(1+n) = \hat{k}_t^\alpha,$
Initial Condition:	\hat{k}_0 is given
Transversality Condition:	$\lim_{t \rightarrow \infty} \hat{\beta}^t v'(\hat{k}_t) \hat{k}_t = 0$

This \hat{k} converges to the steady state as in our original optimal growth model. Then the aggregate variables grow at the rate of $g + n$, and the per capita variables grow at the exogenous rate of g . It means that per capita consumption, capital, and output all grow at an exogenous rate determined by technology, which is not explained in the model. This is known as the Ramsey model. We will return to the growth theory later on.

One Note: Uzawa's Theorem You may wonder why we propose normalizing all aggregate variables by $A_t^{\frac{1}{1-\alpha}}L_t$. This is related to a well-established theorem about the form of production function if we want to model a balanced growth path of the economy.

Theorem 4. *The balanced growth in the long run $g_Y = g_K = g_C$ is only possible if all technological progress is labor-augmenting or Harrod-neutral in the aggregate production function. That is,*

$$\tilde{F}(A_t, K_t, L_t) = F(K_t, A_t L_t)$$

We will not prove this theorem here but you should know what Uzawa's Theorem is. And we can see that the Cobb-Douglas aggregate production function is labor-augmenting.

$$A_t K_t^\alpha L_t^{1-\alpha} = (A_t^{\frac{1}{1-\alpha}})^{1-\alpha} K_t^\alpha L_t^{1-\alpha} = K_t^\alpha (A_t^{\frac{1}{1-\alpha}} L_t)^{1-\alpha}.$$

2.4.3 Habit Formation

Consumption data show that consumption is more persistent than what the optimal growth model implies. Some economists come up with the theory of "Catch up with the Jones" theory of consumption. It states that consumer inclines to consume the quantity as what other consumers do.

$$u(c_t, \bar{c}_{t-1}) = \frac{(c_t - \bar{c}_{t-1})^{1-\theta}}{1-\theta}$$

This utility function means the larger the difference between current consumption and previous average consumption is, the smaller the marginal utility of consumption is. Basically, this utility function penalizes behavior perturbing away from previous consumption, which makes the consumption series more persistent. Other setups are the same for the representative consumer.

The social planner solves the following problem:

$$\max_{c_t, k_{t+1}} \sum_{t=0}^{\infty} \beta^t u(c_t, \bar{c}_{t-1})$$

$$s.t. \quad c_t + k_{t+1} \leq A k_t^\alpha$$

$$c_t, k_{t+1} \geq 0$$

$$k_0, \quad given$$

Here the social planner has two state variables k_t and \bar{c}_{t-1} . The control variables are still c_t and k_{t+1} . The Bellman equation is:

$$v_t(k_t, \bar{c}_{t-1}) = \max_{c_t, k_{t+1}} u(c_t, \bar{c}_{t-1}) + \beta v_{t+1}(k_{t+1}, \bar{c}_t)$$

$$s.t. \quad c_t + k_{t+1} = A k_t^\alpha$$

Substitute k_{t+1} :

$$v_t(k_t, \bar{c}_{t-1}) = \max_{k_{t+1}} u(Ak_t^\alpha - k_{t+1}, \bar{c}_{t-1}) + \beta v_{t+1}(k_{t+1}, \bar{c}_t)$$

The first order condition is:

$$-\frac{\partial u(Ak_t^\alpha - k_{t+1}, \bar{c}_{t-1})}{\partial c_t} + \beta \frac{\partial v_{t+1}(k_{t+1}, \bar{c}_t)}{\partial k_{t+1}} = 0$$

According to the envelope theorem:

$$\frac{\partial v_t(k_t, \bar{c}_{t-1})}{k_t} = \frac{\partial u(Ak_t^\alpha - k_{t+1}, \bar{c}_{t-1})}{\partial c_t} A\alpha k_t^{\alpha-1}$$

Substitute this marginal value to the first order condition:

$$-\frac{\partial u(Ak_t^\alpha - k_{t+1}, \bar{c}_{t-1})}{\partial c_t} + \beta \frac{\partial u(Ak_{t+1}^\alpha - k_{t+2}, \bar{c}_t)}{\partial c_{t+1}} A\alpha k_{t+1}^{\alpha-1} = 0$$

In the equilibrium, the average consumption \bar{c}_t equals to the representative's consumption.

The Euler equation is:

$$-\frac{\partial u(Ak_t^\alpha - k_{t+1}, c_{t-1})}{\partial c_t} + \beta \frac{\partial u(Ak_{t+1}^\alpha - k_{t+2}, c_t)}{\partial c_{t+1}} A\alpha k_{t+1}^{\alpha-1} = 0$$

The transversality condition is:

$$\lim_{t \rightarrow \infty} \beta^t \frac{\partial v_t(k_t, \bar{c}_{t-1})}{k_t} k_t = 0$$

The initial condition, resource condition, the Euler equation, and the transversality condition characterize the optimal solution.

2.5 Lagrangian Multiplier

In the above sections, we show how we can solve the dynamic problems. We know that both the Euler equation and transversality conditions are both necessary and sufficient for our solution. Though technically beautiful, it's burdensome to formulate the functional equation to get the first order conditions and transversality condition every time. Is there any shortcut to get the conditions?

Once you know that transversality condition is essential, you know that the stable path

that satisfy the first order conditions which converges to the steady state is the solution of the dynamic problem. Then one easy way to solve the problem is to find the FOCs first. And then any stable path (transversality) that satisfy all these FOCs is the solution. Normally, we can obtain easily the FOCs by formulate the infinite-horizon Lagrangian function. Still use the optimal growth model as the example:

$$L(\{c_t, k_{t+1}\}_{t=0}^{\infty}) = \sum_{t=0}^{\infty} \beta^t [u(c_t) + \lambda_t (Ak_t^{\alpha} - c_t - k_{t+1})]$$

The FOCs are:

$$L_{c_t} = u'(c_t) - \lambda_t = 0, t = 0, 1, 2, \dots$$

$$L_{k_{t+1}} = -\lambda_t + \beta \lambda_{t+1} A \alpha k_{t+1}^{\alpha-1} = 0, t = 0, 1, 2, \dots$$

$$L_{\lambda_t} = Ak_t^{\alpha} - c_t - k_{t+1} = 0, t = 0, 1, 2, \dots$$

Then we can get the FOCs of this optimal growth model:

$$u'(c_t) = \beta u'(c_{t+1}) A \alpha k_{t+1}^{\alpha-1}, t = 0, 1, 2, \dots$$

$$Ak_t^{\alpha} - c_t - k_{t+1} = 0, t = 0, 1, 2, \dots$$

Then we can solve this problem by finding a **stable** path that satisfy the FOCs, which also satisfies the transversality condition.

3 Practical Dynamic Programming: Global Solution

We can apply dynamic programming to get the characterization conditions for the solution. However, for most of macroeconomic problems, we cannot obtain closed-form solutions. Practically, we count on computer programs to solve the dynamic programming problems numerically. Basically, we can divide the numerical solution method to two categories: (1) global solution; (2) local solution. The global solution illustrates the value function along the domain of the state variable while local solution only explores the dynamics in the neighbourhood of the steady state of the state variable. Here in this section, we explores the global solution method.

1. Guess and verify. Most dynamic problems in macroeconomics do not have an explicit solution. But our example of the optimal growth model with full appreciation is the only example that has a closed-form solution. You can guess the solution is $v(k) = E + F \ln k$ and this solution should satisfy the four characterizing solutions. This should be easy for you and I leave it as an exercise. You should find that optimal policy function is $k' = \beta\alpha Ak^\alpha$.

Basically, there is only one example that can be solved by this method. So it is of little use for our purpose.

2. Value function iteration. This method is directly implied by dynamic programming. According to contraction mapping theorem, starting from arbitrary initial value function $v_0(x)$, the value function $v_j(x)$ converges to the solution $v(x)$ by iterating the following FE:

$$v_{j+1}(x) = \max_u r(x, u) + \beta v_j(x')$$

$$\text{s.t. } x' = g(x, u)$$

3. Howard's improvement algorithm. The Howard's improvement algorithm is also called policy function iteration. It consists the following steps:

- (a) Starting from an arbitrary policy function $u = h_0(x)$, and according to the func-

tion compute the associated value function:

$$v_0(x) = \sum_{t=0}^{\infty} \beta^t r[x_t, h_0(x_t)]$$

$$s.t. \quad x_{t+1} = g[x_t, h(x_t)]$$

(b) From value function v_j , generate a new policy $u = h_{j+1}(x)$ that solves the FE:

$$\max_u \quad r(x, u) + \beta v_j(x')$$

$$s.t. \quad x' = g(x, u)$$

(c) From the newly-generated policy, compute the associated value function:

$$v_{j+1}(x) = \sum_{t=0}^{\infty} \beta^t r[x_t, h_{j+1}(x_t)]$$

$$s.t. \quad x_{t+1} = g[x_t, h(x_t)]$$

Iterate step (b) and (c) until the policy function converges to the solution $h(x)$.

- The policy function iteration method is usually faster to get the solution than the value function iteration. Numerically the choice of initial function is important for the speed even though theoretically you can always get the solution.
- It is hard to write program code to directly compute the value function according to the policy function as Step (c). We usually use the FE to compute the value function $v_{j+1}(x)$ according to policy function $h_{j+1}(x)$.

$$v_{j+1}(x_t) = r(x_t, h_{j+1}(x_t)) + \beta v_{j+1}(x_{t+1})$$

$$s.t. \quad x_{t+1} = g(x_t, h_{j+1}(x_t))$$

Suppose the domain of the state x_t is discretized as (x^1, \dots, x^N) . The key is to construct a transition matrix P similar to the transition matrix of Markov chain where

$$P_{mn} = \begin{cases} 1 & if \quad g(x_t, h_{j+1}(x_t)) = x_{t+1}, x_t = x^m, x_{t+1} = x^n \\ 0 & otherwise \end{cases}$$

Then the FE can be transformed as:

$$v_{j+1}(x_t) = r(x_t, h_{j+1}(x_t)) + \beta Pv_{j+1}(x_t)$$

The value function can be calculated as:

$$v_{j+1}(x_t) = (I - \beta P)^{-1}r(x_t, h_{j+1}(x_t))$$

Next, we will use concrete examples to show you how to apply the two iteration methods.

3.1 Value Function Iteration

We find the solution of this problem:

$$v(k) = \max_{k'} \ln(Ak^\alpha - k') + \beta v(k')$$

The value function iteration is to iterate the following contraction mapping until v_j converges.

$$v_{j+1}(k) = \max_{k'} \ln(Ak^\alpha - k') + \beta v_j(k')$$

We use the same parameters as the finite case of the optimal growth model. First, we should know that the optimal growth model features that the equilibrium capital converges to the steady state \bar{k} . So we can discretize the capital k around the steady state.

$$\bar{k} = (A\alpha\beta)^{\frac{1}{1-\alpha}}$$

In my program, I choose to discretize the domain from one fifth of the steady state to five times of this value, $k = [k^1, k^2, \dots, k^N]$ with $k^1 = \frac{1}{5}\bar{k}$ and $k^N = 5\bar{k}$.

```

1 A = 5;
2 alpha = 1/3;
3 beta = 0.99;
4 kbar = (A*alpha*beta)^(1/(1-alpha));
5 k=1/5*kbar:0.05:5*kbar;      %discretize the domain of k

```

Then we should start with an initial value function for the algorithm. Basically, this is a function and so we should assign a number to each value of capital in the domain.

```

1 N = length(k);      %number of points in the domain of value function

```

```

2 | v0 = zeros(N,1);      %initial value function v0=0 or consider v0 as v_{j}

```

Next, we can start with the iteration. Basiclly, for the value function $v_j(k)$, we should find a maximizer k' for the objective $\ln(Ak^\alpha - k') + \beta v_j(k')$ and the maximum value will be $v_{j+1}(k)$. This should be true for each value in the domain of k . The easiest way to do this is to build a matrix $vk{k}$. The dimension of this matrix is $N \times N$ where N is the dimension of k . Each entry of this matrix $vk{k}$ records the value of the objective $\ln(Ak^\alpha - k') + \beta v_j(k')$ when $k = k^i$ and $k' = k^j$. The maximization process is to find a maximum value along each row of this matrix.

$$vk{k} = \begin{bmatrix} \ln(Ak^{1\alpha} - k^1) + \beta v_j(k^1) & \cdots & \ln(Ak^{1\alpha} - k^N) + \beta v_j(k^N) \\ \vdots & & \vdots \\ \ln(Ak^{N\alpha} - k^1) + \beta v_j(k^1) & \cdots & \ln(Ak^{N\alpha} - k^N) + \beta v_j(k^N) \end{bmatrix}$$

where $vk{k}_{ij} = \ln(Ak^{i\alpha} - k^j) + \beta v_j(k^j)$

```

1 | vkk = zeros(N,N);      %Matrix vkk with vkk_{ij} as the value of
2 | %the objctive if k=ki and k'= kj
3 | %each value of vkk_{ij} is given by
4 | %ln(A*k_{i})^alpha-k_{j})+beta*v0_{j}
5 | for i=1:N
6 |   for j=1:N
7 |     if A*k(i)^alpha-k(j)<=1e-5
8 |       vkk(i,j) = -1e5;          %if consumption is negative,
9 |       %make the value a value that can't be chosen
10 |     else
11 |       vkk(i,j)= log(A*k(i)^alpha-k(j))+beta*v0(j);
12 |     end
13 |   end

```

For each iteration, we should find v_{j+1} given v_j . After the iteration, assgin the newly-found value function v_{j+1} ($v1$ in the program) to v_j ($v0$ in the program). If the difference between v_{j+1} and v_j is very small (less than 10^{-5} in the program), we can think that we have found the solution.

```

1 | %choose k' for each k
2 | [v1,d1]= max(vkk,[],2);           %choose the maximum value along the row of
3 | diff = max(abs(v1-v0));           %the difference between v_{j} and v_{j+1}
4 | v0 = v1;                          %give the new function value to v_{j}

```

```

5   d0 = d1;
6   it = it+1;

```

The whole program is as follows:

```

1 %=====
2 % Chapter 2
3 % File: ch2_solve_k_infinite_vfi.m
4 % This program file is to solve the finite optimal growth model
5 % Written: 2020.09.19
6 % Written by Bin Wang
7 %=====

8
9 clear all;
10 close all;
11 clc;

12
13 %% Parameters
14 A = 5;
15 alpha = 1/3;
16 beta = 0.99;
17 kbar = (A*alpha*beta)^(1/(1-alpha));
18 k=1/5*kbar:0.02:5*kbar;           %discretize the domain of k
19 N = length(k);                  %number of points in the domain of val
20 v0 = zeros(N,1);                %initial value function v0=0 or consider
21 tol = 1e-5;                     %tolerance of the convergence
22 maxI = 1e5;                     %maximum iteration
23 diff = 2;                       %value function difference of each ite

24
25 it = 0;                         %count of iteration

26
27 diary('vfi.log')               %records all outputs
28 while diff>tol & it<=maxI
29     if mod(it,100)==0
30         disp(['Iteration          ' 'diff']);
31         disp([it diff]);
32     end
33     vkk = zeros(N,N);            %Matrix vkk with vkk_{ij} as the value
34     %each value of vkk_{ij} is given by ln(A*k_{i})^alpha-k_{j})+beta*v0_{j}
35     for i=1:N
36         for j=1:N
37             if A*k(i)^alpha-k(j)<=1e-5
38                 vkk(i,j) = -1e5;      %if consumption is negative, make the
39             else
40                 vkk(i,j)= log(A*k(i)^alpha-k(j))+beta*v0(j);

```

```

41         end
42     end
43 end
44 %=====
45
46 %choose k' for each k
47 [v1,d1]= max(vkk,[],2); %chose the maximum value along the row
48 diff = max(abs(v1-v0)); %the difference between v_{j'} and v_{j}
49 v0 = v1; %give the new function value to v_{j'}
50 d0 = d1;
51 it = it+1;
52 end
53
54 %%
55 if diff<tol
56     disp(['Value function iteration succeeds']);
57     disp(['Iteration           ' 'diff']);
58     disp([it diff]);
59 end
60 figure;
61 plot(k,v0,'LineWidth',2);
62 xlabel('k');
63 ylabel('v');
64 fig = gcf;
65 fig.PaperUnits = 'inches';
66 fig.PaperPosition = [0 0 8 4];
67 saveas(fig,'ch2_solve_k_infinite_vfi_value_function.png');
68 figure;
69
70 plot(k,k(d0),'LineWidth',2);
71 xlabel('k');
72 ylabel("k'");
73 ylim([0 12]);
74 fig = gcf;
75 fig.PaperUnits = 'inches';
76 fig.PaperPosition = [0 0 8 8];
77 saveas(fig,'ch2_solve_k_infinite_vfi_policy_function.png');

78
79
80 figure;
81 plot(k,A*k.^alpha-k(d0),'LineWidth',2);hold on;
82 plot(kbar*ones(length(0:0.05:12),1),0:0.05:12,'r','LineWidth',2);
83 plot(0:0.05:12,A*(0:0.05:12).^alpha-(0:0.05:12),'r','LineWidth',2);
84 xlabel('k');
85 ylabel("c");

```

```

86 ylim([0 12]);
87 xlim([0 12]);
88 fig = gcf;
89 fig.PaperUnits = 'inches';
90 fig.PaperPosition = [0 0 8 8];
91 saveas(fig,'ch2_solve_k_infinite_vfi_dynamic_diagram.png');
92
93 diary off;

```

We can record the difference of subsequent value functions and see that difference becomes smaller and smaller. This is what contraction mapping theorem tells us.

```

1 Iteration      diff
2      0          2
3
4 Iteration      diff
5    100.0000    0.5383
6
7 Iteration      diff
8    200.0000    0.1970
9
10 Iteration     diff
11   300.0000    0.0721
12
13 Iteration     diff
14   400.0000    0.0264
15
16 Iteration     diff
17   500.0000    0.0097
18
19 Iteration     diff
20   600.0000    0.0035
21
22 Iteration     diff
23   700.0000    0.0013
24
25 Iteration     diff
26   800.0000    0.0005
27
28 Iteration     diff
29   900.0000    0.0002
30
31 Iteration     diff
32   1.0e+03 *

```

```

33      1.0000      0.0000
34
35
36 Iteration      diff
37      1.0e+03 *
38
39      1.1000      0.0000
40
41 Value function iteration succeeds
42 Iteration      diff
43      1.0e+03 *
44
45      1.1840      0.0000

```

The solution of this optimal growth model is a value function $v(k)$ shown in Figure 10 or a policy function $k' = h(k)$ shown in Figure 11. We can also draw the dynamic diagram we analyzed above and the saddle path in Figure 12.

Figure 10: Value Function

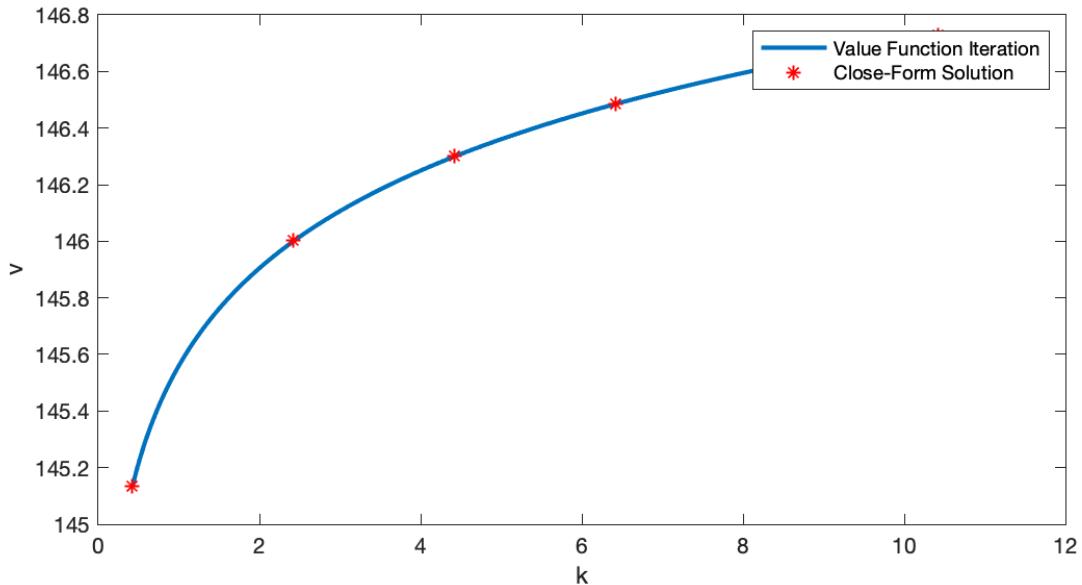
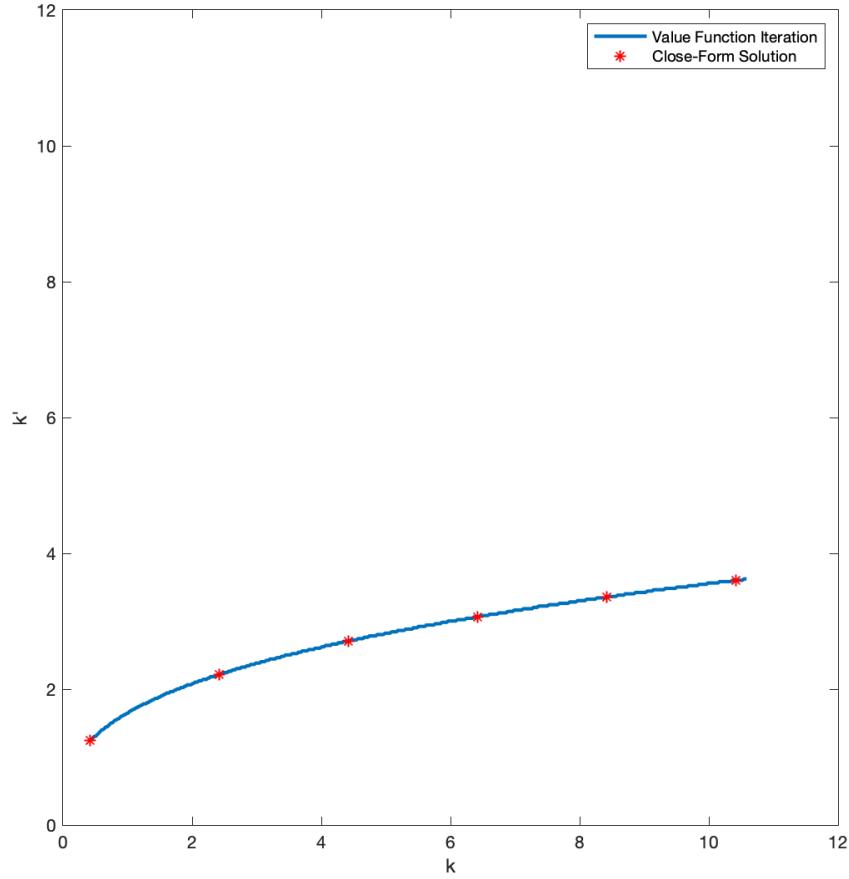


Figure 11: Policy Function



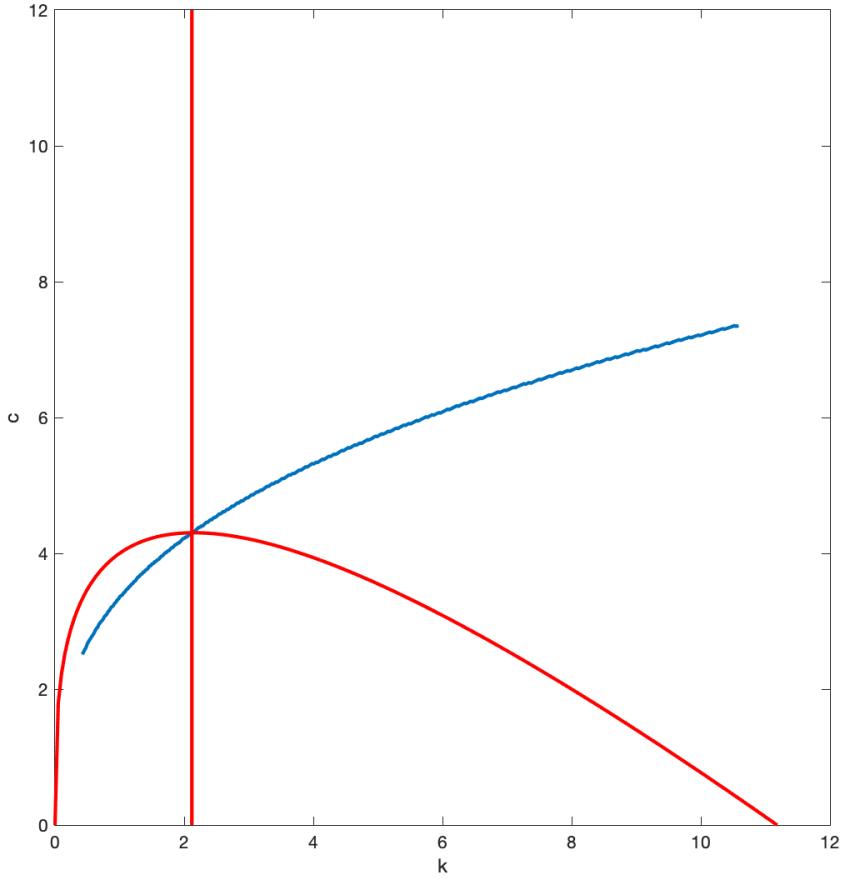
Stochastic Optimal Growth Model There are two state variables in the stochastic optimal growth model k and A .

$$v(k, A) = \max_{k'} \ln(Ak^\alpha - k') + \beta E[v(k', A')|A]$$

The parameters are calibrated the same as above deterministic case. We choose to discretize the domain from one fifth of the steady state to five times of this value, $k = [k^1, k^2, \dots, k^N]$ with $k^1 = \frac{1}{5}\bar{k}$ and $k^N = 5\bar{k}$. For the stochastic A_t is assumed to be a Markov chain $\{(A^1, A^2), P, \pi_0\}$ where

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}$$

Figure 12: Saddle Path



$P_{ij} = \text{Prob}(A_{t+1} = A^j | A_t = A^i)$ and $P_{i1} + P_{i2} = 1, i = 1, 2$. Now we need to find a solution of the problem, which is a value function v of two arguments k, A . As k has a dimension of N and A has a dimension of 2, the number of points of the domain is $2N$. This is called **curse of dimension**, which states that computation power grows exponentially with number of states.

Now the value function is also a function of stochastic technology A_t . For simplicity, we assume A_t takes two values from (A^1, A^2) . The optimal policy of the FE is also dependent

on the state of technology $k' = h(k, A)$. We can write down the FE of each technology state:

$$v(k, A^1) = \max_{k'} \ln(A^1 k^\alpha - k') + \beta[P_{11}v(k', A^1) + P_{12}v(k', A^2)]$$

$$v(k, A^2) = \max_{k'} \ln(A^2 k^\alpha - k') + \beta[P_{21}v(k', A^1) + P_{22}v(k', A^2)]$$

To apply the value function iteration, we can start with arbitrary value functions $(v_0(k, A^1), v_0(k, A^2))$.

Given $(v_j(k, A^1), v_j(k, A^2))$, calculate $(v_{j+1}(k, A^1), v_{j+1}(k, A^2))$ using following FE:

$$v_{j+1}(k, A^1) = \max_{k'} \ln(A^1 k^\alpha - k') + \beta[P_{11}v_j(k', A^1) + P_{12}v_j(k', A^2)]$$

$$v_{j+1}(k, A^2) = \max_{k'} \ln(A^2 k^\alpha - k') + \beta[P_{21}v_j(k', A^1) + P_{22}v_j(k', A^2)]$$

In the program, we should make some changes to the dimensions of some variables. We parameterized A to be a two dimensional vector $A = (4, 5)$. v and k are both $N \times M$ matrix where $M = 2$. The vk matrix which records the value of the objective function is now three dimensional $N \times N \times M$, where vk_{ijm} is assigned the value of the objective of $k = k^i, k' = k^j, A = A^m$. The transition matrix is parameterized as:

$$P = \begin{bmatrix} 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

```

1 A = [4 5];
2 M = length(A);
3
4 vkk = zeros(N,N,M); %Matrix vkk with vkk_{ijm} as the value of the objective
5 %each value of vkk_{ijm} is given by
6 %ln(A(m)*k_{i}^alpha-k_{j})+beta*P_{m,1}*v0_{j,1}+beta*P_{m,2}*v0_{j,2}
7 for m=1:M
8     for i=1:N
9         for j=1:N
10            if A(m)*k(i)^alpha-k(j)<=1e-5
11                vkk(i,j,m) = -1e5; %if consumption is negative, make it -ve
12            else
13                vkk(i,j,m)= log(A(m)*k(i)^alpha-k(j)) ...
14                +beta*P(m,1)*v0(j,1)+ beta*P(m,2)*v0(j,2);
15            end
16        end
17    end
18 end

```

The solution of the stochastic optimal growth model is given by the value function $v(k, A)$. As technology A is a two-dimensional vector, we can graph the solution by $v(k, A^1)$ and $v(k, A^2)$. Similarly, we can also show the policy function $k' = h(v, A)$ by $h(v, A^1)$ and $h(v, A^2)$.

We can see from Figure 13 that $v(k, A^1)$ is higher than $v(k)$ of deterministic problem with $A = 4$ while $v(k, A^2)$ is lower than $v(k)$ of deterministic problem with $A = 5$. This is intuitive in the sense that there is a probability of getting higher technology in the future when the present technology is low. Similarly, there is a probability of getting lower technology in the future when the present technology is high. And $v(k, A^1)$ is lower than $v(k, A^2)$. It states that if the economy starts with a low technology, the sum of discounted utility will be lower.

For the stochastic optimal growth model, we can get a closed-form solution for the policy function $k_t = \alpha\beta Ak_t^\alpha$. We can check that the numerical policy functions are quite accurately replicating the theoretical closed-form solutions, as shown in Figure 14.

Figure 13: Value Function

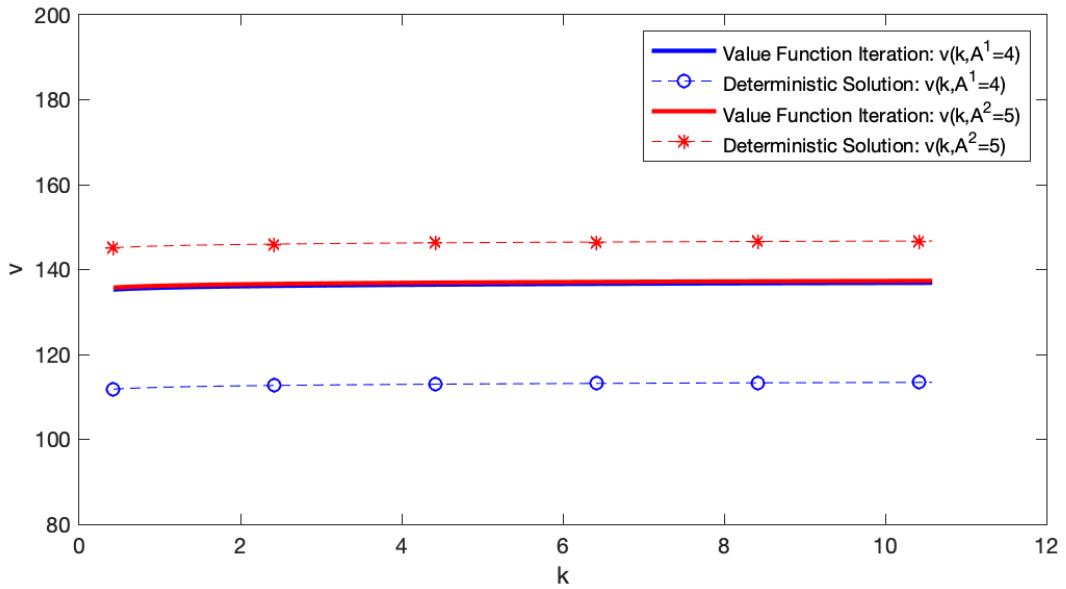
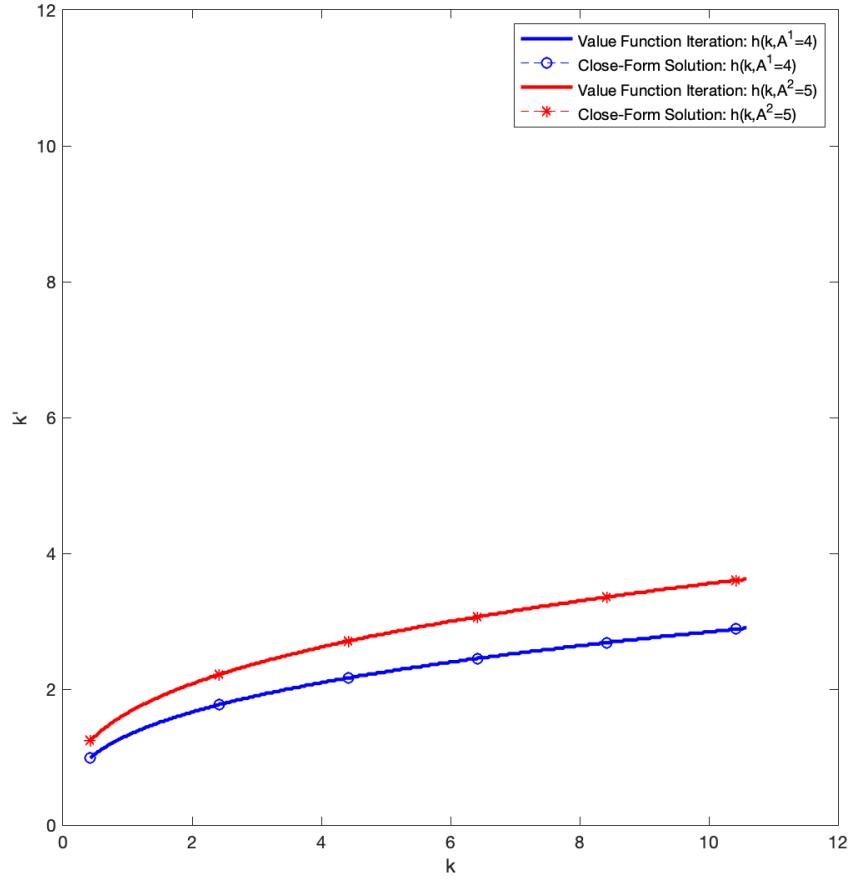


Figure 14: Policy Function



3.2 Policy Function Iteration

We still use the optimal growth model as the example to show the algorithm.

$$v(k) = \max_{k'} \ln(Ak^\alpha - k') + \beta v(k')$$

The policy function iteration is to iterate the policy until v_j converges.

$$v_{j+1}(k) = \max_{k'} \ln(Ak^\alpha - k') + \beta v_j(k')$$

In my program, I choose to discretize the domain from one fifth of the steady state to five times of this value, $k = [k^1, k^2, \dots, k^N]$ with $k^1 = \frac{1}{5}\bar{k}$ and $k^N = 5\bar{k}$.

- Starting from an arbitrary policy function $k' = h_0(k)$ (we choose $k' = \frac{1}{5}Ak^\alpha$ in the program), and construct a transition matrix P where

$$P_{mn} = \begin{cases} 1 & \text{if } k' = h_0(k), k = k^m, k' = k^n \\ 0 & \text{otherwise} \end{cases}$$

Then the FE can be transformed as:

$$v_0(k) = \ln(Ak^\alpha - h_0(k)) + \beta Pv_0(k)$$

The value function can be calculated as:

$$v_0(k) = (I - \beta P)^{-1} \ln(Ak^\alpha - h_0(k))$$

- From value function v_j , generate a new policy $u = h_{j+1}(x)$ that solves the FE:

$$\max_{k'} \ln(Ak^\alpha - k') + \beta v_j(k')$$

- From the newly-generated policy, compute the associated value function as Step (1): construct a transition matrix P where

$$P_{mn} = \begin{cases} 1 & \text{if } k' = h_{j+1}(k), k = k^m, k' = k^n \\ 0 & \text{otherwise} \end{cases}$$

Then the FE can be transformed as:

$$v_{j+1}(k) = \ln(Ak^\alpha - h_{j+1}(k)) + \beta Pv_{j+1}(k)$$

The value function can be calculated as:

$$v_{j+1}(k) = (I - \beta P)^{-1} \ln(Ak^\alpha - h_{j+1}(k))$$

Iterate step (b) and (c) until the policy function converges to the solution $h(x)$. The key difference from the value function iteration is that we need to recalculate the value function according to the newly-found policy in each iteration. In the program, we should construct the transition matrix P in each iteration including the initial step:

```
1 %compute the value function v0 according to the policy d0=====
```

```

2 Pkk = zeros(N,N); %construct the transition matrix
3 for i=1:N
4     for j=1:N
5         if j==d0(i)
6             Pkk(i,j)=1; %Pkk(i,j)=1 if k'=h(k), k=k^i, k'=k^j
7         end
8     end
9 end
10 v0=(eye(N)-beta*Pkk)\(log(A*k.^alpha-k(d0)))';
11 %=====

```

For beginners, it is perhaps a little bit harder to understand this algorithm. Let's just use a very simple example to understand this. Assume that the domain of the state has only three points (k^1, k^2, k^3) . The value of the capital of each period should be chosen from the three points. So the value function has three points in its domain $v(k^1), v(k^2), v(k^3)$. The value function is a solution of the FE:

$$v(k) = \max_{k'} \ln(Ak^\alpha - k') + \beta v(k')$$

For each $k \in \{k^1, k^2, k^3\}$, we should find a new capital $k' \in \{k^1, k^2, k^3\}$. For example,

$$v(k^1) = \max \{ \ln(Ak^{1\alpha} - k^1) + \beta v(k^1), \ln(Ak^{1\alpha} - k^2) + \beta v(k^2), \ln(Ak^{1\alpha} - k^3) + \beta v(k^3) \}$$

Similarly,

$$v(k^2) = \max \{ \ln(Ak^{2\alpha} - k^1) + \beta v(k^1), \ln(Ak^{2\alpha} - k^2) + \beta v(k^2), \ln(Ak^{2\alpha} - k^3) + \beta v(k^3) \}$$

$$v(k^3) = \max \{ \ln(Ak^{3\alpha} - k^1) + \beta v(k^1), \ln(Ak^{3\alpha} - k^2) + \beta v(k^2), \ln(Ak^{3\alpha} - k^3) + \beta v(k^3) \}$$

Suppose we start with the initial policy $h_0(k^1) = k^2, h_0(k^2) = k^1, h_0(k^3) = k^2$. Then the FE is:

$$v_0(k^1) = \ln(Ak^{1\alpha} - k^2) + \beta v_0(k^2)$$

$$v_0(k^2) = \ln(Ak^{2\alpha} - k^1) + \beta v_0(k^1)$$

$$v_0(k^3) = \ln(Ak^{3\alpha} - k^2) + \beta v_0(k^2)$$

It amounts to:

$$v_0(k^1) = \ln(Ak^{1\alpha} - k^2) + \beta(0, 1, 0) * (v_0((k^1), v_0(k^2), v_0(k^3))'$$

$$v_0(k^2) = \ln(Ak^{2\alpha} - k^1) + \beta(1, 0, 0) * (v_0((k^1), v_0(k^2), v_0(k^3))'$$

$$v_0(k^3) = \ln(Ak^{3\alpha} - k^2) + \beta(0, 1, 0) * (v_0((k^1), v_0(k^2), v_0(k^3))'$$

Or

$$\begin{bmatrix} v_0(k^1) \\ v_0(k^2) \\ v_0(k^3) \end{bmatrix} = \begin{bmatrix} \ln(Ak^{1\alpha} - h_0(k^1)) \\ \ln(Ak^{2\alpha} - h_0(k^2)) \\ \ln(Ak^{3\alpha} - h_0(k^3)) \end{bmatrix} + \beta \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v_0(k^1) \\ v_0(k^2) \\ v_0(k^3) \end{bmatrix}$$

This is just

$$v_0(k) = \ln(Ak^\alpha - h_0(k)) + \beta Pv_0(k)$$

where

$$P = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Each entry of 1 means that the maximizer is this one according to the policy. Then the value function v_0 associated with policy function $h_0(k)$ is:

$$v_0(k) = (I - \beta P)^{-1} \ln(Ak^\alpha - h_0(k))$$

In each iteration, we compare the new policy $h_{j+1}(k)$ with the old policy $h_j(k)$ until they converge to the same value.

```

1 diff = max(abs(d1-d0)); %the difference between h_{j} and h_{j+1}
2 d0 = d1;
3 v0 = v1;
```

We can see that the policy function iteration is much faster than the value function iteration.

The former needs only 7 iterations while the latter need more than 1000 iterations.

1	Iteration	diff
---	-----------	------

```

2      0      2
3
4 Iteration      diff
5      1      35
6
7 Iteration      diff
8      2      7
9
10 Iteration      diff
11      3      6
12
13 Iteration      diff
14      4      5
15
16 Iteration      diff
17      5      3
18
19 Iteration      diff
20      6      1
21
22 Policy function iteration succeeds
23 Iteration      diff
24      7      0

```

Stochastic Optimal Growth Model We can also apply the policy function iteration to the stochastic optimal growth model to get the solution. Remember that the key step of policy function iteration is to calculate the value function associated with the updated policy function. In our parameterized optimal growth model, the stochastic technology is only two-dimensional.

$$v(k, A) = \max_{k'} \ln(Ak^\alpha - k') + \beta E[v(k', A')|A]$$

We can write down the value function to each possible present technology.

$$v(k, A^1) = \max_{k'} \ln(A^1 k^\alpha - k') + \beta P_{11} v(k', A^1) + \beta P_{12} v(k', A^2)$$

$$v(k, A^2) = \max_{k'} \ln(A^2 k^\alpha - k') + \beta P_{21} v(k', A^1) + \beta P_{22} v(k', A^2)$$

If the optimal policy is $k' = h_j(k)$, the two FE are:

$$v_j(k, A^1) = \ln(A^1 k^\alpha - h_j(k)) + \beta P_{11} v(h_j(k), A^1) + \beta P_{12} v(h_j(k), A^2)$$

$$v_j(k, A^2) = \ln(A^2 k^\alpha - h_j(k)) + \beta P_{21} v(h_j(k), A^1) + \beta P_{22} v(h_j(k), A^2)$$

As the deterministic case, we can define a $N \times N$ transition matrix Pkk associated with the policy $h_j(k)$.

$$Pkk_{abm} = \begin{cases} 1 & \text{if } k' = h_j(k, A^m), k = k^a, k' = k^b \\ 0 & \text{otherwise} \end{cases}$$

The two FE are now:

$$v_j(k, A^1) = \ln(A^1 k^\alpha - h_j(k)) + \beta P_{11} Pkk_{..1} * v(k, A^1) + \beta P_{12} Pkk_{..1} * v(k, A^2)$$

$$v_j(k, A^2) = \ln(A^2 k^\alpha - h_j(k)) + \beta P_{21} Pkk_{..2} * v(k, A^1) + \beta P_{22} Pkk_{..2} * v(k, A^2)$$

Stack the two vectors $(v_j(k, A^1)', v_j(k, A^2)')$, then

$$\begin{bmatrix} v_j(k, A^1) \\ v_j(k, A^2) \end{bmatrix} = \begin{bmatrix} \ln(A^1 k^\alpha - h_j(k)) \\ \ln(A^2 k^\alpha - h_j(k)) \end{bmatrix} + \begin{bmatrix} \beta P_{11} Pkk_{..1} & \beta P_{12} Pkk_{..1} \\ \beta P_{21} Pkk_{..2} & \beta P_{22} Pkk_{..2} \end{bmatrix} \begin{bmatrix} v_j(k, A^1) \\ v_j(k, A^2) \end{bmatrix}$$

Then we can calculate the value functions associated with policy function $v_j(k, A)$.

$$\begin{bmatrix} v_j(k, A^1) \\ v_j(k, A^2) \end{bmatrix} = \left[I_{2N} - \beta \begin{pmatrix} P_{11} Pkk_{..1} & P_{12} Pkk_{..1} \\ P_{21} Pkk_{..2} & P_{22} Pkk_{..2} \end{pmatrix} \right]^{-1} \begin{bmatrix} \ln(A^1 k^\alpha - h_j(k)) \\ \ln(A^2 k^\alpha - h_j(k)) \end{bmatrix} \quad (41)$$

The key difference lies in the construction of the transition matrix Pkk and calculate the associated value function $v(k, A)$. We should vectorize the Bellman equation and calculate the vectorized value function and then reshape it to the matrix again in the program.

```

1 %compute the value function v0 according to the policy d0=====
2 Pkk = zeros(N,N,M); %construct the transition matrix
3 for m=1:M
4     for i=1:N
5         for j=1:N
6             if j==d0(i,m)
7                 Pkk(i,j,m)=1; %Pkk(i,j,m)=1 if k'=h(k,A^m), k=k^i, k'=k^j, A=
8             end
9         end

```

```

10      end
11  end
12 objValue = [log(A(1)*k.^alpha-k(d0(:,1))), log(A(2)*k.^alpha-k(d0(:,2)))]';
13 PPkk = zeros(2*N,2*N);
14 PPkk(1:N,1:N)= P(1,1)*squeeze(Pkk(:,:,1));
15 PPkk(1:N,N+1:2*N)= P(1,2)*squeeze(Pkk(:,:,1));
16 PPkk(N+1:2*N,1:N)= P(2,1)*squeeze(Pkk(:,:,2));
17 PPkk(N+1:2*N,N+1:2*N)= P(2,2)*squeeze(Pkk(:,:,2));
18 v0Vector = (eye(M*N) - beta*PPkk)\objValue;
19 v0 = reshape(v0Vector,N,2);
20 %=====

```

You can check from the program that the value functions and policy functions are the same as those obtained from the value function method. Still, we can get the results in 7 iterations, which is much faster than the value function iteration.

4 Practical Dynamic Programming: Local Methods

For some applications in macroeconomics, we are interested in the economic system around the neighbourhood of the steady state. Or the solution is not that highly-nonlinear so that the local dynamics is exact enough for our questions. The local methods are also called perturbation methods, since these approaches explore the dynamic system of the economy when it is perturbed from the steady state. In this section, we first use the optimal growth model as an example to illustrate this method, linear approximation, log-linear approximation, and higher-order approximation. Then we use general notation to study this method.

4.1 Perturbation: Optimal Growth Model

The social planner's problem is:

$$\begin{aligned} \max_{\{c_t\}_{t=0}^{\infty}, \{k_{t+1}\}_{t=1}^{\infty}} \quad & \sum_{t=0}^{\infty} \beta^t u(c_t) \\ \text{subject to: } \quad & c_t + k_{t+1} \leq A k_t^{\alpha}, \forall t \\ & c_t, k_{t+1} \geq 0, \forall t, \\ & k_0, \quad \text{given.} \end{aligned}$$

By dynamic programming, we know that the solution is characterized by;

Euler Condition:	$u'(c_t) = \beta A \alpha k_{t+1}^{\alpha-1} u'(c_{t+1})$
Resource Constraint:	$c_t + k_{t+1} = A k_t^\alpha$
Initial Condition:	k_0 is given
Transversality Condition:	$\lim_{t \rightarrow \infty} \beta^t v_k(k_t) k_t = 0$

Basic Idea of Perturbation The basic idea of perturbation method is to use the Euler condition and resource constraint to find the local dynamic system around the steady state that can be dealt with easily such as linear approximation, log-linear approximation or higher-order approximation. But not all dynamic path is the solution because the solution should also satisfy the transversality condition. And we know that any path that doesn't satisfy the transversality condition will not converge to the steady state. Thus only the stable path that converges to the steady state is the solution of the model. You may refresh your memory about the local dynamics of the deterministic optimal growth above.

In our model, we can first find the steady state that we start with.

$$\bar{k} = (A\alpha\beta)^{\frac{1}{1-\alpha}}, \bar{c} = A\bar{k}^\alpha - \bar{k}.$$

Around the steady state, we can approximate the dynamic system so that the system is easier to be dealt with. As we already linearized the dynamic system above, here let's log-linearize the system by Taylor series expansion. And let $\hat{x} = \ln x - \ln \bar{x}, x = k, c$. This \hat{x} means percentage change from the steady state. We proceed by rewriting all the equations by using $x_t = e^{\ln x_t}, x = c, k$.

$$\begin{aligned} u'(e^{\ln c_t}) &= \beta A \alpha (e^{\ln k_{t+1}})^{\alpha-1} u'(e^{\ln c_{t+1}}) \\ e^{\ln c_t} + e^{\ln k_{t+1}} &= A(e^{\ln k_t})^\alpha \end{aligned}$$

Then approximate these two equations around $\ln \bar{c}$ and $\ln \bar{k}$

$$\begin{aligned} u'(e^{\ln \bar{c}}) + u''(e^{\ln \bar{c}}) e^{\ln \bar{c}} \hat{c}_t &= \beta \alpha A \left((e^{\ln \bar{k}})^{\alpha-1} + (\alpha-1)(e^{\ln \bar{k}})^{\alpha-2} (e^{\ln \bar{k}}) \hat{k}_{t+1} \right) (u'(e^{\ln \bar{c}}) + u''(e^{\ln \bar{c}}) e^{\ln \bar{c}} \hat{c}_{t+1}) \\ e^{\ln \bar{c}} + e^{\ln \bar{c}} \hat{c}_t + e^{\ln \bar{k}} + e^{\ln \bar{k}} \hat{k}_{t+1} &= A((e^{\ln \bar{k}})^\alpha + \alpha(e^{\ln \bar{k}})^{\alpha-1} e^{\ln \bar{k}} \hat{k}_t) \end{aligned}$$

Simplify them by $e^{\ln \bar{x}} = \bar{x}, x = c, k$

$$\begin{aligned} u'(\bar{c}) + u''(\bar{c})\bar{c}\hat{c}_t &= \beta\alpha A \left((\bar{k}^{\alpha-1} + (\alpha-1)\bar{k}^{\alpha-2}\bar{k}\hat{k}_{t+1}) (u'(\bar{c}) + u''(\bar{c})\bar{c}\hat{c}_{t+1}) \right) \\ \bar{c} + \bar{c}\hat{c}_t + \bar{k} + \bar{k}\hat{k}_{t+1} &= A(\bar{k}^\alpha + \alpha\bar{k}^{\alpha-1}\bar{k}\hat{k}_t) \end{aligned}$$

Simplify further:

$$u''(\bar{c})\bar{c}\hat{c}_t = \beta\alpha A u'(\bar{c})(\alpha-1)\bar{k}^{\alpha-2}\bar{k}\hat{k}_{t+1} + \beta\alpha A \bar{k}^{\alpha-1} u''(\bar{c})\bar{c}\hat{c}_{t+1} \quad (42)$$

$$\bar{c}\hat{c}_t + \bar{k}\hat{k}_{t+1} = A\alpha\bar{k}^{\alpha-1}\bar{k}\hat{k}_t \quad (43)$$

Stack these two equations into matrix:

$$\begin{bmatrix} A\alpha\beta\bar{k}^{\alpha-1}u''\bar{c} & A\alpha\beta(\alpha-1)\bar{k}^{\alpha-1}u' \\ 0 & \bar{k} \end{bmatrix} \begin{bmatrix} \hat{c}_{t+1} \\ \hat{k}_{t+1} \end{bmatrix} = \begin{bmatrix} u''\bar{c} & 0 \\ -\bar{c} & A\alpha\bar{k}^\alpha \end{bmatrix} \begin{bmatrix} \hat{c}_t \\ \hat{k}_t \end{bmatrix}$$

Then

$$\begin{aligned} \begin{bmatrix} \hat{c}_{t+1} \\ \hat{k}_{t+1} \end{bmatrix} &= \begin{bmatrix} A\alpha\beta\bar{k}^{\alpha-1}u''\bar{c} & A\alpha\beta(\alpha-1)\bar{k}^{\alpha-1}u' \\ 0 & \bar{k} \end{bmatrix}^{-1} \begin{bmatrix} u''\bar{c} & 0 \\ -\bar{c} & A\alpha\bar{k}^\alpha \end{bmatrix} \begin{bmatrix} \hat{c}_t \\ \hat{k}_t \end{bmatrix} \\ &= -\frac{\bar{c}}{\bar{k}} \begin{bmatrix} \bar{k} & -A\alpha\beta(\alpha-1)\bar{k}^{\alpha-1}u' \\ 0 & A\alpha\beta\bar{k}^{\alpha-1}u''\bar{c} \end{bmatrix} \begin{bmatrix} u''\bar{c} & 0 \\ -\bar{c} & A\alpha\bar{k}^\alpha \end{bmatrix} \begin{bmatrix} \hat{c}_t \\ \hat{k}_t \end{bmatrix} \\ &= \begin{bmatrix} 1 + (1-\alpha)(\frac{1}{\alpha\beta} - 1) & (\alpha-1)\frac{1}{\beta} \\ -(\frac{1}{\alpha\beta} - 1) & \frac{1}{\beta} \end{bmatrix} \begin{bmatrix} \hat{c}_t \\ \hat{k}_t \end{bmatrix} \end{aligned}$$

Let $B = \begin{bmatrix} 1 + (1-\alpha)(\frac{1}{\alpha\beta} - 1) & (\alpha-1)\frac{1}{\beta} \\ -(\frac{1}{\alpha\beta} - 1) & \frac{1}{\beta} \end{bmatrix}$ and this B matrix can be diagonally-decomposed as:

$$B = P\Lambda P^{-1} \quad (44)$$

We can find P and Λ accordingly:

$$\begin{aligned} |B - \lambda I| &= \begin{vmatrix} 1 + (1 - \alpha)(\frac{1}{\alpha\beta} - 1) - \lambda & (\alpha - 1)\frac{1}{\beta} \\ -(\frac{1}{\alpha\beta} - 1) & \frac{1}{\beta} - \lambda \end{vmatrix} \\ &= \lambda^2 - (1 + \frac{1}{\beta} + \frac{1}{\alpha\beta} - \frac{1}{\beta} - 1 + \alpha)\lambda + \frac{1}{\beta} + \frac{1}{\beta}(1 - \alpha)(\frac{1}{\alpha\beta} - 1) + (\alpha - 1)(\frac{1}{\alpha\beta} - 1) \\ &= \lambda^2 - (\frac{1}{\alpha\beta} + \alpha)\lambda + \frac{1}{\beta} = 0 \end{aligned}$$

Then we know that there are distinct eigenvalues: one is less than one and the other is bigger than one.

$$\lambda_1 = \alpha < 1, \quad \lambda_2 = \frac{1}{\alpha\beta} > 1 \quad (45)$$

We can also get the corresponding eigenvectors P_1, P_2 for λ_1, λ_2 .

$$\begin{bmatrix} 1 + (1 - \alpha)(\frac{1}{\alpha\beta} - 1) - \alpha & (\alpha - 1)\frac{1}{\beta} \\ -(\frac{1}{\alpha\beta} - 1) & \frac{1}{\beta} - \alpha \end{bmatrix} P_1 = \begin{bmatrix} \frac{1-\alpha}{\alpha\beta} & \frac{\alpha-1}{\beta} \\ \frac{\alpha\beta-1}{\alpha\beta} & \frac{1-\alpha\beta}{\beta} \end{bmatrix} P_1 = 0$$

One eigenvector for eigenvalue $\lambda_1 = \alpha$ is $P_1 = (\alpha, 1)'$. Similarly,

$$\begin{bmatrix} 1 + (1 - \alpha)(\frac{1}{\alpha\beta} - 1) - \frac{1}{\alpha\beta} & (\alpha - 1)\frac{1}{\beta} \\ -(\frac{1}{\alpha\beta} - 1) & \frac{1}{\beta} - \frac{1}{\alpha\beta} \end{bmatrix} P_2 = \begin{bmatrix} \frac{\alpha\beta-1}{\beta} & \frac{\alpha-1}{\beta} \\ \frac{\alpha\beta-1}{\alpha\beta} & \frac{\alpha-1}{\alpha\beta} \end{bmatrix} P_2 = 0$$

One eigenvector for eigenvalue $\lambda_2 = \frac{1}{\alpha\beta}$ is $P_2 = (1 - \alpha, \alpha\beta - 1)'$. So the matrix B can be diagonally-decomposed as:

$$B = \begin{bmatrix} 1 + (1 - \alpha)(\frac{1}{\alpha\beta} - 1) & (\alpha - 1)\frac{1}{\beta} \\ -(\frac{1}{\alpha\beta} - 1) & \frac{1}{\beta} \end{bmatrix} = \begin{bmatrix} \alpha & 1 - \alpha \\ 1 & \alpha\beta - 1 \end{bmatrix} \begin{bmatrix} \alpha & 0 \\ 0 & \frac{1}{\alpha\beta} \end{bmatrix} \begin{bmatrix} \alpha & 1 - \alpha \\ 1 & \alpha\beta - 1 \end{bmatrix}^{-1} \quad (46)$$

Then we can know that $(\hat{c}_t, \hat{k}_t)'$ can be simulated by:

$$\begin{aligned} \begin{bmatrix} \hat{c}_t \\ \hat{k}_t \end{bmatrix} &= P\Lambda^t P^{-1} \begin{bmatrix} \hat{c}_0 \\ \hat{k}_0 \end{bmatrix} \\ &= \begin{bmatrix} \alpha & 1 - \alpha \\ 1 & \alpha\beta - 1 \end{bmatrix} \begin{bmatrix} \alpha^t & 0 \\ 0 & (\frac{1}{\alpha\beta})^t \end{bmatrix} \begin{bmatrix} \alpha & 1 - \alpha \\ 1 & \alpha\beta - 1 \end{bmatrix}^{-1} \begin{bmatrix} \hat{c}_0 \\ \hat{k}_0 \end{bmatrix} \end{aligned}$$

Move matrix P to the left hand side:

$$\begin{bmatrix} \alpha & 1-\alpha \\ 1 & \alpha\beta-1 \end{bmatrix}^{-1} \begin{bmatrix} \hat{c}_t \\ \hat{k}_t \end{bmatrix} = \begin{bmatrix} \alpha^t & 0 \\ 0 & (\frac{1}{\alpha\beta})^t \end{bmatrix} \begin{bmatrix} \alpha & 1-\alpha \\ 1 & \alpha\beta-1 \end{bmatrix}^{-1} \begin{bmatrix} \hat{c}_0 \\ \hat{k}_0 \end{bmatrix}$$

Let

$$Q = P^{-1} = \begin{bmatrix} \alpha & 1-\alpha \\ 1 & \alpha\beta-1 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{\alpha\beta}{\alpha^2\beta-1} & \frac{\alpha-1}{\alpha^2\beta-1} \\ \frac{-1}{\alpha^2\beta-1} & \frac{\alpha}{\alpha^2\beta-1} \end{bmatrix}$$

Then the economic system is:

$$\begin{bmatrix} \frac{\alpha\beta}{\alpha^2\beta-1}\hat{c}_t + \frac{\alpha-1}{\alpha^2\beta-1}\hat{k}_t \\ \frac{-1}{\alpha^2\beta-1}\hat{c}_t + \frac{\alpha}{\alpha^2\beta-1}\hat{k}_t \end{bmatrix} = \begin{bmatrix} \alpha^t & 0 \\ 0 & (\frac{1}{\alpha\beta})^t \end{bmatrix} \begin{bmatrix} \frac{\alpha\beta}{\alpha^2\beta-1}\hat{c}_0 + \frac{\alpha-1}{\alpha^2\beta-1}\hat{k}_0 \\ \frac{-1}{\alpha^2\beta-1}\hat{c}_0 + \frac{\alpha}{\alpha^2\beta-1}\hat{k}_0 \end{bmatrix}$$

$$= \begin{bmatrix} \alpha^t \left(\frac{\alpha\beta}{\alpha^2\beta-1}\hat{c}_0 + \frac{\alpha-1}{\alpha^2\beta-1}\hat{k}_0 \right) \\ (\frac{1}{\alpha\beta})^t \left(\frac{-1}{\alpha^2\beta-1}\hat{c}_0 + \frac{\alpha}{\alpha^2\beta-1}\hat{k}_0 \right) \end{bmatrix}$$

Clearly when t goes to infinity, α^t goes to zero and $(\frac{1}{\alpha\beta})^t$ goes to infinity as $\alpha < 1, \frac{1}{\alpha\beta} > 1$.

To get a stable path, the only way is to choose \hat{c}_0 so that

$$\frac{-1}{\alpha^2\beta-1}\hat{c}_0 + \frac{\alpha}{\alpha^2\beta-1}\hat{k}_0 = 0$$

Then the saddle path requires

$$\hat{c}_0 = \alpha\hat{k}_0$$

And

$$\frac{-1}{\alpha^2\beta-1}\hat{c}_t + \frac{\alpha}{\alpha^2\beta-1}\hat{k}_t = 0$$

$$\hat{c}_t = \alpha\hat{k}_t$$

Substitute it into the first row:

$$\hat{c}_t = \alpha\alpha^t\hat{k}_0$$

$$\hat{k}_t = \alpha^t\hat{k}_0$$

where $\hat{k}_0 = \ln k_0 - \ln \bar{k} = \ln k_0 - \ln(A\alpha\beta)^{\frac{1}{1-\alpha}}$. And then

$$c_t = e^{\hat{c}_t + \ln \bar{c}}$$

$$k_t = e^{\hat{k}_t + \ln \bar{k}}$$

This is the basic way to solve the dynamic system by diagonally-decomposing the matrix. In most of the DSGE models, we have stochastic shocks which make the solution method more difficult. Here we introduce two widely-used methods, Blanchard-Kahn's method and Sims' method, to show how we can solve the DSGE models. Still we use the stochastic optimal growth model as the example to show the two methods.

The social planner solves the following problem:

$$\begin{aligned} & \max_{\{c_t\}_{t=0}^{\infty}, \{k_{t+1}\}_{t=1}^{\infty}} E_0 \sum_{t=0}^{\infty} \beta^t u(c_t) \\ & c_t + k_{t+1} \leq A_t k_t^{\alpha}, \forall t \\ & \ln A_{t+1} = (1 - \rho)\bar{A} + \rho \ln A_t + \epsilon_{t+1}, \quad \epsilon_{t+1} \sim N(0, \sigma^2) \\ & k_0, A_0, \quad \text{given}. \end{aligned}$$

Now the stochastic productivity A_t follows an AR(1) process. This is a continuous version for the discrete Markov chain. You can see that the AR(1) stochastic process A_t satisfies the Markov property that $\text{Prob}(A_{t+1}|A_t, A_{t-1}, \dots) = \text{Prob}(A_{t+1}|A_t)$. Since productivity is stochastic, c_1, c_2, \dots are all stochastic and then the sum of discounted utility is stochastic. Then the social planner's objective is to maximize his lifetime expected utility which is conditional on information of Period 0 productivity A_0 . To solve the model, we first obtain the solution characterization conditions by dynamic programming:

$$v(k, A) = \max_{c, k'} u(c) + \beta E[v(k', A')|A] + \lambda(Ak^{\alpha} - c - k')$$

The first order conditions w.r.t. c, k' are:

$$u'(c) - \lambda = 0$$

$$\beta E[v_k(k', A')|A] - \lambda = 0$$

From the envelope theorem:

$$v_k(k, A) = \lambda A \alpha k^{\alpha-1} = u'(c) A \alpha k^{\alpha-1}$$

Substitute it into the first order conditions and the Euler equation is:

$$u'(c) = E[\beta u'(c') A' \alpha k'^{\alpha-1} | A]$$

The solution of the model is characterized by: Along with the technology law of motion:

Euler Condition:	$u'(c_t) = E_t \beta A_{t+1} \alpha k_{t+1}^{\alpha-1} u'(c_{t+1}),$
Resource Constraint:	$c_t + k_{t+1} = A_t k_t^\alpha,$
Initial Condition:	$k_0, A_0,$ given
Transversality Condition:	$\lim_{t \rightarrow \infty} \beta^t v_k(k_t, A_t) k_t = 0$

$$\ln A_t = (1 - \rho) \ln \bar{A} + \rho \ln A_{t-1} + \epsilon_t.$$

As above, we log-linearize the Euler equation and resource constraint by Taylor series expansion around the logarithm of the steady state.

$$\begin{aligned} u'(e^{\ln c_t}) &= E_t \beta e^{\ln A_{t+1}} \alpha (e^{\ln k_{t+1}})^{\alpha-1} u'(e^{\ln c_{t+1}}) \\ e^{\ln c_t} + e^{\ln k_{t+1}} &= e^{\ln A_t} (e^{\ln k_t})^\alpha \end{aligned}$$

Let $\hat{x} = \ln x - \ln \bar{x}, x = A, c, k$ and get the Taylor series around $\ln \bar{A}, \ln \bar{c},$ and $\ln \bar{k}$:

$$\begin{aligned} u'(e^{\ln \bar{c}}) + u''(e^{\ln \bar{c}}) e^{\ln \bar{c}} \hat{c}_t &= E_t \beta \alpha (e^{\ln \bar{A}} + e^{\ln \bar{A}} \hat{A}_{t+1}) \left((e^{\ln \bar{k}})^{\alpha-1} + (\alpha-1)(e^{\ln \bar{k}})^{\alpha-2} (e^{\ln \bar{k}}) \hat{k}_{t+1} \right) * \\ &\quad (u'(e^{\ln \bar{c}}) + u''(e^{\ln \bar{c}}) e^{\ln \bar{c}} \hat{c}_{t+1}) \\ e^{\ln \bar{c}} + e^{\ln \bar{c}} \hat{c}_t + e^{\ln \bar{k}} + e^{\ln \bar{k}} \hat{k}_{t+1} &= (e^{\ln \bar{A}} + e^{\ln \bar{A}} \hat{A}_t) ((e^{\ln \bar{k}})^\alpha + \alpha (e^{\ln \bar{k}})^{\alpha-1} e^{\ln \bar{k}} \hat{k}_t) \end{aligned}$$

Simplify them by $e^{\ln \bar{x}} = \bar{x}$, $x = A, c, k$

$$\begin{aligned} u'(\bar{c}) + u''(\bar{c})\bar{c}\hat{c}_t &= \beta\alpha(\bar{A} + \bar{A}\hat{A}_{t+1}) \left((\bar{k}^{\alpha-1} + (\alpha-1)\bar{k}^{\alpha-2}\bar{k}\hat{k}_{t+1}) \right) (u'(\bar{c}) + u''(\bar{c})\bar{c}\hat{c}_{t+1}) \\ \bar{c} + \bar{c}\hat{c}_t + \bar{k} + \bar{k}\hat{k}_{t+1} &= (\bar{A} + \bar{A}\hat{A}_t)(\bar{k}^\alpha + \alpha\bar{k}^{\alpha-1}\bar{k}\hat{k}_t) \end{aligned}$$

Simplify further:

$$\begin{aligned} u''(\bar{c})\bar{c}\hat{c}_t &= \beta\alpha\bar{A}\bar{k}^{\alpha-1}u'(\bar{c})E_t\hat{A}_{t+1} + \beta\alpha\bar{A}u'(\bar{c})(\alpha-1)\bar{k}^{\alpha-2}\bar{k}\hat{k}_{t+1} + \beta\alpha\bar{A}\bar{k}^{\alpha-1}u''(\bar{c})\bar{c}E_t\hat{c}_{t+1} \\ (47) \end{aligned}$$

$$\bar{c}\hat{c}_t + \bar{k}\hat{k}_{t+1} = \bar{A}\bar{k}^\alpha\hat{A}_t + \bar{A}\alpha\bar{k}^{\alpha-1}\bar{k}\hat{k}_t \quad (48)$$

We have $\beta\bar{A}\alpha\bar{k}^{\alpha-1} = 1$, $\bar{k} = (\bar{A}\alpha\beta)^{\frac{1}{\alpha\beta}}$, $\frac{\bar{c}}{\bar{k}} = \bar{A}\bar{k}^{\alpha-1} - 1 = \frac{1}{\alpha\beta} - 1$.

$$-\hat{c}_t = \rho\hat{A}_t + (\alpha-1)\hat{k}_{t+1} - E_t\hat{c}_{t+1} \quad (49)$$

$$(\frac{1}{\alpha\beta} - 1)\hat{c}_t + \hat{k}_{t+1} = \frac{1}{\alpha\beta}\hat{A}_t + \frac{1}{\beta}\hat{k}_t \quad (50)$$

And the law of motion of technology is linear:

$$\hat{A}_{t+1} = \rho\hat{A}_t + \epsilon_{t+1} \quad (51)$$

You should notice that on the right hand side of the log-linearized Euler equation there are $E_t A_{t+1}$ and $E_t \hat{c}_{t+1}$ terms but there is no expectation symbol before k_{t+1} . It is because the capital at the beginning of next period k_{t+1} is actually determined in Period t . You can see this point from the resource constraint $c_t + k_{t+1} = A_t k_t$. As A_t and k_t are known in this period, the choice of c_t is not a contingent plan and is deterministic. Thus the capital of next period k_{t+1} should be also deterministic. We call this kind of variable **predetermined** or **beginning-of-period variable**.

Some papers or books render this not convenient as we should differentiate which variables are predetermined. They advocate **end-of-period variable** so that any time t variables are deterministic and $t+1$ variables are stochastic. For example, in the stochastic optimal growth model, we can change the time convention for the capital k_t by moving the time a period backward. The capital at the beginning of Period t is k_{t-1} instead of k_t . The end

of period capital will be k_t instead of k_{t+1} . Likewise, the given capital at the beginning of Period 0 is k_{-1} and the resource constraint will be $c_t + k_t = A_t k_{t-1}^\alpha$. We will come back to this point later on. Right now, we still follow the beginning of period convention.

4.2 Blanchard and Kahn's Method

The earliest method to solve the linearized DSGE model was developed by [Blanchard and Kahn \[1980\]](#). Basically, it requires that the linearized system is written as:

$$\begin{bmatrix} \hat{k}_{t+1} \\ E_t \hat{c}_{t+1} \end{bmatrix} = B \begin{bmatrix} \hat{k}_t \\ \hat{c}_t \end{bmatrix} + E \hat{A}_t \quad (52)$$

The predetermined variables should appear in the first several rows of the linearized system. Basically, in our application, the system is:

$$\begin{bmatrix} \alpha\beta \bar{A} u' (\alpha - 1) \bar{k}^{\alpha-1} & \alpha\beta \bar{A} \bar{k}^{\alpha-1} u'' \bar{c} \\ \bar{k} & 0 \end{bmatrix} \begin{bmatrix} \hat{k}_{t+1} \\ E_t \hat{c}_{t+1} \end{bmatrix} = \begin{bmatrix} 0 & u'' \bar{c} \\ \bar{A} \alpha \bar{k}^\alpha & -\bar{c} \end{bmatrix} \begin{bmatrix} \hat{k}_t \\ \hat{c}_t \end{bmatrix} \quad (53)$$

$$+ \begin{bmatrix} -\alpha\beta \bar{A} \bar{k}^{\alpha-1} u' \rho \\ \bar{A} \bar{k}^\alpha \end{bmatrix} \hat{A}_t \quad (54)$$

$$\begin{bmatrix} \alpha - 1 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \hat{k}_{t+1} \\ E_t \hat{c}_{t+1} \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ \frac{1}{\beta} & -(\frac{1}{\alpha\beta} - 1) \end{bmatrix} \begin{bmatrix} \hat{k}_t \\ \hat{c}_t \end{bmatrix} + \begin{bmatrix} -\rho \\ \frac{1}{\alpha\beta} \end{bmatrix} \hat{A}_t \quad (55)$$

$$\begin{bmatrix} \hat{k}_{t+1} \\ E_t \hat{c}_{t+1} \end{bmatrix} = \begin{bmatrix} \frac{1}{\beta} & -(\frac{1}{\alpha\beta} - 1) \\ -\frac{1}{\beta}(1-\alpha) & 1 + (1-\alpha)(\frac{1}{\alpha\beta} - 1) \end{bmatrix} \begin{bmatrix} \hat{k}_t \\ \hat{c}_t \end{bmatrix} + \begin{bmatrix} \frac{1}{\alpha\beta} \\ \rho - (1-\alpha)\frac{1}{\alpha\beta} \end{bmatrix} \hat{A}_t \quad (56)$$

[Blanchard and Kahn \[1980\]](#) start with this equation and decompose the B matrix according to a Jordan decomposition.

$$B = \Lambda^{-1} J \Lambda$$

where the diagonal elements of J , consisting of the eigenvalues of B , are ordered in increasing absolute value in moving from left to right. J can be written as:

$$J = \begin{bmatrix} J_1 & 0 \\ 0 & J_2 \end{bmatrix}$$

In our application, the B matrix can be diagonalized and so the Jordan decomposition is the same as diagonalization. You can check that there are two distinct eigenvalues $\alpha, \frac{1}{\alpha\beta}$ for the matrix B . And the corresponding eigenvectors for these two eigenvalues are $(1, \alpha)'$ and $(1 - \alpha\beta, \alpha - 1)'$ respectively. Then in the Jordan decomposition, the J and Λ matrix are:

$$J = \begin{bmatrix} \alpha & 0 \\ 0 & \frac{1}{\alpha\beta} \end{bmatrix}$$

$$\Lambda = \begin{bmatrix} 1 & 1 - \alpha\beta \\ \alpha & \alpha - 1 \end{bmatrix}^{-1} = \frac{1}{\alpha^2\beta - 1} \begin{bmatrix} \alpha - 1 & \alpha\beta - 1 \\ -\alpha & 1 \end{bmatrix} = \begin{bmatrix} \frac{\alpha - 1}{\alpha^2\beta - 1} & \frac{\alpha\beta - 1}{\alpha^2\beta - 1} \\ \frac{-\alpha}{\alpha^2\beta - 1} & \frac{1}{\alpha^2\beta - 1} \end{bmatrix}$$

Substitute this Jordan decomposition to our baseline equation.

$$\begin{bmatrix} \hat{k}_{t+1} \\ E_t \hat{c}_{t+1} \end{bmatrix} = \begin{bmatrix} \frac{\alpha - 1}{\alpha^2\beta - 1} & \frac{\alpha\beta - 1}{\alpha^2\beta - 1} \\ \frac{-\alpha}{\alpha^2\beta - 1} & \frac{1}{\alpha^2\beta - 1} \end{bmatrix}^{-1} \begin{bmatrix} \alpha & 0 \\ 0 & \frac{1}{\alpha\beta} \end{bmatrix} \begin{bmatrix} \frac{\alpha - 1}{\alpha^2\beta - 1} & \frac{\alpha\beta - 1}{\alpha^2\beta - 1} \\ \frac{-\alpha}{\alpha^2\beta - 1} & \frac{1}{\alpha^2\beta - 1} \end{bmatrix} \begin{bmatrix} \hat{k}_t \\ \hat{c}_t \end{bmatrix} + \begin{bmatrix} E_1 \\ E_2 \end{bmatrix} \hat{A}_t \quad (57)$$

where $E_1 = \frac{1}{\alpha\beta}, E_2 = \rho - (1 - \alpha)\frac{1}{\alpha\beta}$. Premultiply $\begin{bmatrix} \frac{\alpha - 1}{\alpha^2\beta - 1} & \frac{\alpha\beta - 1}{\alpha^2\beta - 1} \\ \frac{-\alpha}{\alpha^2\beta - 1} & \frac{1}{\alpha^2\beta - 1} \end{bmatrix}$ to the LHS and RHS of the above equation.

$$\begin{bmatrix} \frac{\alpha - 1}{\alpha^2\beta - 1} & \frac{\alpha\beta - 1}{\alpha^2\beta - 1} \\ \frac{-\alpha}{\alpha^2\beta - 1} & \frac{1}{\alpha^2\beta - 1} \end{bmatrix} \begin{bmatrix} \hat{k}_{t+1} \\ E_t \hat{c}_{t+1} \end{bmatrix} = \begin{bmatrix} \alpha & 0 \\ 0 & \frac{1}{\alpha\beta} \end{bmatrix} \begin{bmatrix} \frac{\alpha - 1}{\alpha^2\beta - 1} & \frac{\alpha\beta - 1}{\alpha^2\beta - 1} \\ \frac{-\alpha}{\alpha^2\beta - 1} & \frac{1}{\alpha^2\beta - 1} \end{bmatrix} \begin{bmatrix} \hat{k}_t \\ \hat{c}_t \end{bmatrix} \quad (58)$$

$$+ \begin{bmatrix} \frac{\alpha - 1}{\alpha^2\beta - 1} & \frac{\alpha\beta - 1}{\alpha^2\beta - 1} \\ \frac{-\alpha}{\alpha^2\beta - 1} & \frac{1}{\alpha^2\beta - 1} \end{bmatrix} \begin{bmatrix} E_1 \\ E_2 \end{bmatrix} \hat{A}_t \quad (59)$$

Let

$$\begin{bmatrix} \tilde{x}_{1t} \\ \tilde{x}_{2t} \end{bmatrix} = \begin{bmatrix} \frac{\alpha-1}{\alpha^2\beta-1} & \frac{\alpha\beta-1}{\alpha^2\beta-1} \\ \frac{-\alpha}{\alpha^2\beta-1} & \frac{1}{\alpha^2\beta-1} \end{bmatrix} \begin{bmatrix} \hat{k}_t \\ \hat{c}_t \end{bmatrix} = \frac{1}{\alpha^2\beta-1} \begin{bmatrix} (\alpha-1)\hat{k}_t + (\alpha\beta-1)\hat{c}_t \\ -\alpha\hat{k}_t + \hat{c}_t \end{bmatrix}$$

Then

$$\begin{bmatrix} E_t \tilde{x}_{1t+1} \\ E_t \tilde{x}_{2t+1} \end{bmatrix} = \begin{bmatrix} \alpha & 0 \\ 0 & \frac{1}{\alpha\beta} \end{bmatrix} \begin{bmatrix} \tilde{x}_{1t} \\ \tilde{x}_{2t} \end{bmatrix} + \begin{bmatrix} \frac{\alpha-1}{\alpha^2\beta-1} & \frac{\alpha\beta-1}{\alpha^2\beta-1} \\ \frac{-\alpha}{\alpha^2\beta-1} & \frac{1}{\alpha^2\beta-1} \end{bmatrix} \begin{bmatrix} E_1 \\ E_2 \end{bmatrix} \hat{A}_t \quad (60)$$

The transformation effectively "decouples" the system into stable portion and the non-stable portion because $0 < \alpha < 1, \frac{1}{\alpha\beta} > 1$. For the stable portion, we know that \tilde{x}_{1t} converges because $0 < \alpha < 1$. For the unstable portion, we can't solve it by letting $\tilde{x}_{20} = 0$ because $E_0 \tilde{x}_{21} = D_2 \hat{A}_0$ and this technology term will diverge as t goes to infinity. We solve this by performing a forward iteration. As $\frac{1}{\alpha\beta} > 1$, we know that $0 < \frac{1}{\alpha\beta} < 1$. For the unstable portion:

$$E_t \tilde{x}_{2t+1} = \frac{1}{\alpha\beta} \tilde{x}_{2t} + D_2 \hat{A}_t$$

where $D_2 = \frac{1}{\alpha^2\beta-1}[-\frac{1}{\beta} + (\rho + \frac{\alpha-1}{\alpha\beta})]$. Then

$$\tilde{x}_{2t} = \alpha\beta E_t \tilde{x}_{2t+1} - \alpha\beta D_2 \hat{A}_t$$

Substitute $\tilde{x}_{2t+1} = \alpha\beta E_{t+1} \tilde{x}_{2t+2} - \alpha\beta D_2 \hat{A}_{t+1}$ into the above equation and use the law of iterated expectations $E_t [E_{t+1} \tilde{x}_{2t+2}] = E_t \tilde{x}_{2t+2}$.

$$\tilde{x}_{2t} = (\alpha\beta)^2 E_t \tilde{x}_{2t+2} + (\alpha\beta)(-\alpha\beta D_2 E_t \hat{A}_{t+1}) + (-\alpha\beta D_2 \hat{A}_t)$$

Forward this equation continually and we can get:

$$\tilde{x}_{2t} = -\alpha\beta D_2 \hat{A}_t - (\alpha\beta)^2 D_2 E_t \hat{A}_{t+1} - (\alpha\beta)^3 D_2 E_t \hat{A}_{t+2} + \dots \quad (61)$$

$$= - \sum_{j=0}^{\infty} (\alpha\beta)^{j+1} D_2 E_t \hat{A}_{t+j} \quad (62)$$

We know that $E_t \hat{A}_{t+1} = \rho \hat{A}_t$ and so $E_t \hat{A}_{t+j} = \rho^j \hat{A}_t$.

$$\tilde{x}_{2t} = - \sum_{j=0}^{\infty} (\alpha\beta)^{j+1} D_2 \rho^j \hat{A}_t = - \frac{D_2 \alpha \beta}{1 - \alpha \beta \rho} \hat{A}_t \quad (63)$$

$$\frac{1}{\alpha^2 \beta - 1} [-\alpha \hat{k}_t + \hat{c}_t] = - \frac{D_2 \alpha \beta}{1 - \alpha \beta \rho} \hat{A}_t \quad (64)$$

Then, we have the solution:

$$\hat{c}_t = \alpha \hat{k}_t - \frac{D_2 \alpha \beta (\alpha^2 \beta - 1)}{1 - \alpha \beta \rho} \hat{A}_t \quad (65)$$

$$= \alpha \hat{k}_t - \frac{\frac{1}{\alpha^2 \beta - 1} [-\frac{1}{\beta} + (\rho + \frac{\alpha-1}{\alpha \beta})] \alpha \beta (\alpha^2 \beta - 1)}{1 - \alpha \beta \rho} \hat{A}_t \quad (66)$$

$$= \alpha \hat{k}_t + \hat{A}_t \quad (67)$$

Once we know the choice variable of \hat{c}_t , we can substitute c_t to the dynamic system to get \hat{k}_{t+1} as Equation (68). Equation (65) and (68) are the approximate policy functions of the optimal growth model. Once we know the state of the dynamic system \hat{k}_t, \hat{A}_t , we know that the social planner should choose \hat{c}_t and \hat{k}_{t+1} optimally to maximize her life time discounted utility.

$$\hat{k}_{t+1} = \frac{1}{\beta} \hat{k}_t - \left(\frac{1}{\alpha \beta} - 1 \right) \hat{c}_t + \frac{1}{\alpha \beta} \hat{A}_t \quad (68)$$

$$= \frac{1}{\beta} \hat{k}_t - \left(\frac{1}{\alpha \beta} - 1 \right) (\alpha \hat{k}_t + \hat{A}_t) + \frac{1}{\alpha \beta} \hat{A}_t \quad (69)$$

$$= \alpha \hat{k}_t + \hat{A}_t \quad (70)$$

We know that the optimal growth model has a closed form solution that the policy functions are

$$k_{t+1} = \alpha \beta A_t k_t^\alpha \quad (71)$$

$$c_t = (1 - \alpha \beta) A_t k_t^\alpha \quad (72)$$

If we log-linearize these two policy functions around steady states, we can find that $\hat{k}_t = \hat{c}_t = \alpha \hat{k}_t + \hat{A}_t$, which are the local solutions found by the Blanchard and Kahn's method. We

know that:

$$\hat{A}_t = \rho \hat{A}_{t-1} + \epsilon_t$$

Substitute the law of motion of technology into the policy functions and we can get:

$$\hat{c}_t = \alpha \hat{k}_t + rho \hat{A}_{t-1} \epsilon_t \quad (73)$$

$$\hat{k}_{t+1} = \alpha \hat{k}_t + \rho \hat{A}_{t-1} + \epsilon_t \quad (74)$$

Let $s_t = (\hat{c}_t, \hat{k}_{t+1}, \hat{A}_t)'$, then the two policy functions and the law of motion of technology can be expressed as:

$$s_t = \Phi_1 s_{t-1} + \Phi_\epsilon \epsilon_t \quad (75)$$

where

$$\Phi_1 = \begin{bmatrix} 0 & \alpha & \rho \\ 0 & \alpha & \rho \\ 0 & 0 & \rho \end{bmatrix}$$

$$\Phi_\epsilon = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

We can see that the solution expressed in terms of s_t is an AR(1) process. With information s_{t-1} in the beginning of period t and new information ϵ_t during the period, the social planner decides s_t according to her preference and technology under the resource constraint.

Basically, the solution of DSGE models is just an AR(1) process. The parameters Φ_1, Φ_ϵ are functions of deep parameters of the DSGE model such as α, β, ρ . After solving the model, we can use the solution to judge whether the proposed model can explain economic phenomenon, such as **calibration**.

Calibration Calibration is a econometric methodology in economics to test whether the proposed model explains the data. Here below is a typical process how calibration works.

1. Propose a model to explain data;
2. Use a proportion of data to calibrate values of parameters;
3. Check whether the calibrated model can match moments of other proportion of the data.

For example, in the optimal growth model, we need to calibrate three parameters α, β, ρ . Our purpose is to check whether the optimal growth model can explain the overall macroeconomic data. We collect the macroeconomic data. Use some proportion of the data to get the values of these three parameters. The proportion of capital income in total income is the value of α , which is normally $\frac{1}{3}$. β is discount factor, which is normally obtained from real interest rate $\frac{1}{1+r}$. ρ is the persistence parameter of Solow residual, approximately 0.9. That's what we mean "use some proportion of the data", because we do not use all the data we can get.

Then we can use the calibrated model to simulate all the endogenous variables such as c_t, k_t . We can check the means, standard deviations of all these variables. Compare these moments with their data counterpart. If they look similar, we can conclude that the optimal growth model explains the macroeconomy successfully. Calibration is a popular method since 1980s and nowadays it is still a popular way to test macroeconomic models.

Uncertainty Equivalence You may notice that the solution is an AR(1) process which is related to deep parameters α, β, ρ but is not related to the standard deviation σ of the technology shock ϵ_t . This is called uncertainty equivalence in macroeconomics. No matter what magnitude of the uncertainty, the law of motion of the state variables solved from the DSGE model is the same. The solution would be the same under low uncertainty and high uncertainty. This is a characteristic under first order perturbation.

General Form of Blanchard and Kahn’s Method The first order conditions of DSGE models can be written as:

$$\begin{bmatrix} x_{1t+1} \\ E_t x_{2t+1} \end{bmatrix} = B \begin{bmatrix} x_{1t} \\ x_{2t} \end{bmatrix} + Ef_t \quad (76)$$

where x_{1t} is a $n_1 \times 1$ vector of endogenous predetermined variables (endogenous state variables), and x_{2t} is a $n_2 \times 1$ vector of endogenous nonpredetermined (choice) variables, f_t is a $m \times 1$ vector of exogenous state variables. Our aim is to find the solution of the dynamic system by expressing the nonpredetermined variables x_{2t} as a policy function of predetermined variables x_{1t} and exogenous state variables f_t .

The B matrix can be decomposed as $B = \Lambda^{-1} J \Lambda$ where the diagonal elements of Jordan matrix J , consisting of the eigenvalues of B , are ordered in increasing absolute value. We can write J as:

$$J = \begin{bmatrix} J_1 & 0 \\ 0 & J_2 \end{bmatrix} \quad (77)$$

where the eigenvalues in J_1 lie on or within the unit circle, and those in J_2 lie outside of the unit circle. J_1 is said to be stable and J_2 is said to be unstable or explosive. The matrices Λ and E are partitioned conformably as

$$\Lambda = \begin{bmatrix} \Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22} \end{bmatrix}, \quad E = \begin{bmatrix} E_1 \\ E_2 \end{bmatrix} \quad (78)$$

We assume that J_1 is $n'_1 \times n'_1$ and J_2 is $n'_2 \times n'_2$. Then $n'_1 + n'_2 = n_1 + n_2$. And Λ_{11} is $n'_1 \times n'_1$, Λ_{12} is $n'_1 \times n'_2$, Λ_{21} is $n'_2 \times n'_1$, and Λ_{22} is $n'_2 \times n'_2$. E_1 is $n'_1 \times 1$ and E_2 is $n'_2 \times 1$.

Blanchard and Khan Condition For this decomposition, we have three scenarios:

1. $n'_2 = n_2$. If the number of explosive eigenvalues is equal to the number of nonpredetermined variables, the system is said to be saddle-path stable. We also mentioned this in the analysis of local dynamics of the optimal growth model. We call this **Blanchard and Kahn’s Condition**.

2. $n'_2 < n_2$. If the number of explosive eigenvalues is less than the number of nonpredetermined variables, the system is said to have infinite solutions. It states that there are more variables to choose to extinguish the explosiveness of the system, and so there are multiple choices.
3. $n'_2 > n_2$. If the number of explosive eigenvalues is more than the number of nonpredetermined variables, the system is said to have no solution. It states that there are less variables to choose to extinguish the explosiveness of the system, and so there is no choice. Now matter how we choose, the system is explosive.

You can use the optimal growth model to understand the above results. There are two eigenvalues for this system λ_1, λ_2 and there is one nonpredetermined variable or choice variable c_t . If one eigenvalue is explosive, then this system is saddle-path stable. One choice of c_t is just enough to extinguish the explosiveness. If no eigenvalues are explosive $\lambda_1, \lambda_2 < 1$, then $n'_2 < n_2$ and we can choose multiple path of c_t since there is no explosiveness in the system. If two eigenvalues are both explosive $\lambda_1, \lambda_2 > 1$, then $n'_2 > n_2$ and there is no solution, because we can only choose one variable but there are two sources of explosiveness.

We can proceed by decomposing B .

$$\begin{bmatrix} x_{1t+1} \\ E_t x_{2t+1} \end{bmatrix} = \Lambda^{-1} J \Lambda \begin{bmatrix} x_{1t} \\ x_{2t} \end{bmatrix} + E f_t \quad (79)$$

Premultiply the system by Λ .

$$\Lambda \begin{bmatrix} x_{1t+1} \\ E_t x_{2t+1} \end{bmatrix} = J \Lambda \begin{bmatrix} x_{1t} \\ x_{2t} \end{bmatrix} + \Lambda E f_t \quad (80)$$

$$\begin{bmatrix} \hat{x}_{1t+1} \\ E_t \hat{x}_{2t+1} \end{bmatrix} = \begin{bmatrix} J_1 & 0 \\ 0 & J_2 \end{bmatrix} \begin{bmatrix} \hat{x}_{1t} \\ \hat{x}_{2t} \end{bmatrix} + \begin{bmatrix} D_1 \\ D_2 \end{bmatrix} f_t \quad (81)$$

where

$$\begin{bmatrix} \hat{x}_{1t} \\ \hat{x}_{2t} \end{bmatrix} = \Lambda \begin{bmatrix} x_{1t} \\ x_{2t} \end{bmatrix} = \begin{bmatrix} \Lambda_{11}x_{1t} + \Lambda_{12}x_{2t} \\ \Lambda_{21}x_{1t} + \Lambda_{22}x_{2t} \end{bmatrix}$$

$$\begin{bmatrix} D_1 \\ D_2 \end{bmatrix} = \Lambda \begin{bmatrix} E_1 \\ E_2 \end{bmatrix}$$

For the decomposed system (81), the upper portion is stable but the lower portion is not stable. For the lower portion,

$$E_t \hat{x}_{2t+1} = J_2 \hat{x}_{2t} + D_2 f_{2t}$$

We can solve this unstable portion by forward iteration.

$$\begin{aligned} \hat{x}_{2t} &= J_2^{-1} E_t \hat{x}_{2t+1} - J_2^{-1} D_2 f_{2t} \\ &= J_2^{-2} E_t \hat{x}_{2t+2} - J_2^{-2} D_2 E_t f_{2t+1} - J_2^{-1} D_2 f_{2t} \\ &= \dots \\ &= - \sum_{t=0}^{\infty} J_2^{-(i+1)} D_2 E_t f_{2t+i} \end{aligned}$$

If f_{2t} is an AR(1) process as technology, then $f_{2t} = \Xi f_{2t-1} + e_t$ where $e_t \sim N(0, V)$. Then $E_t f_{2t+i} = \Xi^i f_{2t}$. Then,

$$\hat{x}_{2t} = - \sum_{t=0}^{\infty} J_2^{-(i+1)} D_2 \Xi^i f_{2t}$$

Or

$$x_{2t} = -\Lambda_{22}^{-1} \Lambda_{21} x_{1t} - \Lambda_{22}^{-1} \sum_{t=0}^{\infty} J_2^{-(i+1)} D_2 \Xi^i f_{2t} \quad (82)$$

In the optimal growth model, there is only one shock $\Xi = \rho$. Then

$$\sum_{t=0}^{\infty} J_2^{-(i+1)} D_2 \Xi^i = \sum_{t=0}^{\infty} J_2^{-(i+1)} D_2 \rho^i = J_2^{-1} D_2 (1 - J_2^{-1} \rho)^{-1}$$

Then the unstable portion is:

$$x_{2t} = -\Lambda_{22}^{-1}\Lambda_{21}x_{1t} - \Lambda_{22}^{-1}J_2^{-1}D_2(1 - J_2^{-1}\rho)^{-1}f_{2t} \quad (83)$$

The stable portion is:

$$x_{1t+1} = B_{11}x_{1t} + B_{22}x_{2t} + E_1f_t \quad (84)$$

Let $s_t = (x_{1t+1}, x_{2t}, f_t)$, then the law of motion of s_t can be expressed as:

$$s_t = \Phi_1 s_{t-1} + \Phi_\epsilon \epsilon_t \quad (85)$$

4.3 Sims' Method

Sims [2002] proposes a new solution method to the linearized DSGE model. First he proposes to express all the variables in the "end-of-period" convention instead of "beginning-of-period" convention. The variable decided in Period t should be expressed as a variable in Period t . For example, in the optimal growth model, the present k_{t+1} is actually decided in Period t instead of Period $t+1$. Then we should express k_t as the capital in the end of Period t and the capital in the beginning of Period $t+1$. Then the first order conditions of the optimal growth model is:

$$u'(c_t) = \beta E_t u'(c_{t+1}) A_{t+1} k_t^{\alpha-1}$$

$$c_t + k_t = A_t k_{t-1}^\alpha$$

where k_{-1} is given. The system can be log-linearized as:

$$u''(\bar{c}) \bar{c} \hat{c}_t = \beta \alpha \bar{A} \bar{k}^{\alpha-1} u'(\bar{c}) E_t \hat{A}_{t+1} + \beta \alpha \bar{A} u'(\bar{c}) (\alpha - 1) \bar{k}^{\alpha-2} \bar{k} \hat{k}_t + \beta \alpha \bar{A} \bar{k}^{\alpha-1} u''(\bar{c}) \bar{c} E_t \hat{c}_{t+1} \quad (86)$$

$$\bar{c} \hat{c}_t + \bar{k} \hat{k}_t = \bar{A} \bar{k}^\alpha \hat{A}_t + \bar{A} \alpha \bar{k}^{\alpha-1} \bar{k} \hat{k}_{t-1} \quad (87)$$

Then we do not need to differentiate predetermined variables from nonpredetermined variables. All $t+1$ variables are stochastic and all t variables are known. Then express the

rational expectation forecast error η_t as the difference between the actual realization of a random variable x_t and its one-period ahead rational expectation:

$$\eta_t = x_t - E_{t-1}x_t \quad (88)$$

In the optimal growth model, let $s_t = (\hat{c}_t, \hat{k}_t, \hat{A}_t)'$. We can express s_t in the canonical form in Sims [2002]:

$$\Gamma_0 s_t = \Gamma_1 s_{t-1} + \Psi \epsilon_t + \Pi \eta_t$$

Let $\eta_{ct} = c_t - E_{t-1}c_t$, and $\eta_t = \eta_{ct}$, and move the time period of the Euler equation by one period backward.

$$\begin{aligned} \beta\alpha\bar{A}\bar{k}^{\alpha-1}u''(\bar{c})\bar{c}\hat{c}_t &= u''(\bar{c})\bar{c}\hat{c}_{t-1} - \beta\alpha\bar{A}\bar{k}^{\alpha-1}u'(\bar{c})\rho\hat{A}_{t-1} - \beta\alpha\bar{A}u'(\bar{c})(\alpha-1)\bar{k}^{\alpha-2}\bar{k}\hat{k}_{t-1} \\ &+ \beta\alpha\bar{A}\bar{k}^{\alpha-1}u''(\bar{c})\bar{c}\eta_{ct} \\ \bar{c}\hat{c}_t + \bar{k}\hat{k}_t - \bar{A}\bar{k}^\alpha\hat{A}_t &= \bar{A}\alpha\bar{k}^{\alpha-1}\bar{k}\hat{k}_{t-1} \\ \hat{A}_t &= \rho\hat{A}_{t-1} + \epsilon_t \end{aligned}$$

Then

$$\begin{aligned} \Gamma_0 &= \begin{bmatrix} \beta\alpha\bar{A}\bar{k}^{\alpha-1}u''(\bar{c})\bar{c} & 0 & 0 \\ \bar{c} & \bar{k} & -\bar{A}\bar{k}^\alpha \\ 0 & 0 & 1 \end{bmatrix} \\ \Gamma_1 &= \begin{bmatrix} u''\bar{c} & -\beta\alpha\bar{A}u'(\bar{c})(\alpha-1)\bar{k}^{\alpha-2}\bar{k} & -\beta\alpha\bar{A}\bar{k}^{\alpha-1}u'(\bar{c})\rho \\ 0 & \bar{A}\alpha\bar{k}^\alpha & 0 \\ 0 & 0 & \rho \end{bmatrix} \\ \Psi &= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{aligned}$$

$$\Pi = \begin{bmatrix} \beta\alpha\bar{A}\bar{k}^{\alpha-1}u''(\bar{c})\bar{c} \\ 0 \\ 0 \end{bmatrix}$$

Let's continue with the canonical form without referring to the optimal growth model case to save the difficulty in calculating the inverse of matrix.

$$\Gamma_0 s_t = \Gamma_1 s_{t-1} + \Psi \epsilon_t + \Pi \eta_t$$

$$s_t = \Gamma_0^{-1} \Gamma_1 s_{t-1} + \Gamma_0^{-1} \Psi \epsilon_t + \Gamma_0^{-1} \Pi \eta_t$$

If $\Gamma_0^{-1} \Gamma_1$ has full rank, it can be diagonally-decomposed as:

$$\Gamma_0^{-1} \Gamma_1 = P \Lambda P^{-1} \quad (89)$$

where Λ is a diagonal matrix with the eigenvalues ordered by ascending absolute value on the principal diagonal. P is the matrix comprised of eigenvectors in each column. Premultiply P^{-1} to the system:

$$P^{-1} s_t = \Lambda P^{-1} s_{t-1} + P^{-1} \Gamma_0^{-1} \Psi \epsilon_t + P^{-1} \Gamma_0^{-1} \Pi \eta_t$$

As in Blanchard and Kahn's method, Sims also partition the system into stable portion and explosive portion according to the absolute value of the eigenvectors. Let

$$\Lambda = \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix} \quad (90)$$

where Λ_1 is in the unit circle and Λ_2 is outside the unit circle. Suppose Λ_1 is $n_1 \times n_1$ and Λ_2 is $n_2 \times n_2$. $n = n_1 + n_2$ are the number of all endogenous variables in the system. Let

$$\tilde{s}_t = P^{-1} s_t = \begin{bmatrix} \tilde{s}_{1t} \\ \tilde{s}_{2t} \end{bmatrix} \quad (91)$$

where s_{1t} is $n_1 \times 1$ and s_{2t} is $n_2 \times 1$. Similarly, we can decompose the terms related to ϵ_t and

η_t conformably.

$$\begin{bmatrix} \tilde{s}_{1t} \\ \tilde{s}_{2t} \end{bmatrix} = \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix} \begin{bmatrix} \tilde{s}_{1t-1} \\ \tilde{s}_{2t-1} \end{bmatrix} + P^{-1}\Gamma_0^{-1}\Psi\epsilon_t + P^{-1}\Gamma_0^{-1}\Pi\eta_t$$

$$\begin{bmatrix} \tilde{s}_{1t} \\ \tilde{s}_{2t} \end{bmatrix} = \begin{bmatrix} \Lambda_1\tilde{s}_{1t-1} \\ \Lambda_2\tilde{s}_{2t-1} \end{bmatrix} + \begin{bmatrix} Q^S \\ Q^U \end{bmatrix} \Gamma_0^{-1}\Psi\epsilon_t + \begin{bmatrix} Q^S \\ Q^U \end{bmatrix} \Gamma_0^{-1}\Pi\eta_t$$

where $Q = P^{-1} \begin{bmatrix} Q^S \\ Q^U \end{bmatrix}$.

We do not need to worry about the stable portion since $\lim_{t \rightarrow 0} \Lambda_1^t = 0$. But for the explosive portion.

$$\tilde{s}_{2t} = \Lambda_2\tilde{s}_{2t-1} + Q^U\Gamma_0^{-1}\Psi\epsilon_t + Q^U\Gamma_0^{-1}\Pi\eta_t$$

Because $\lim_{t \rightarrow 0} \Lambda_1^t = \infty$, the only way that makes this system stable is to let $\tilde{s}_{2t} = 0$ and $Q^U\Gamma_0^{-1}\Psi\epsilon_t + Q^U\Gamma_0^{-1}\Pi\eta_t = 0$. This is similar to the approach we used in the analysis of deterministic version. We need to choose the forecast error η_t such that $Q^U\Gamma_0^{-1}\Psi\epsilon_t + Q^U\Gamma_0^{-1}\Pi\eta_t = 0$.

$$Q^U\Gamma_0^{-1}\Pi\eta_t = -Q^U\Gamma_0^{-1}\Psi\epsilon_t$$

This equation system can have unique solution, multiple solution or no solution. As there are n_2 explosive eigenvalues and thus n_2 equations for n_η unknowns η .

1. $n_\eta = n_2$. Then there is one unique solution. We have only one choice of forecast error $\eta_t = -(Q^U\Gamma_0^{-1}\Pi)^{-1}Q^U\Gamma_0^{-1}\Psi\epsilon_t$ to make the system stable.
2. $n_\eta > n_2$. We have more forecast errors than explosive eigenvalues. Then we have multiple choices for the forecast error to make the system stable.
3. $n_\eta < n_2$. Then the number of forecast errors is less than the number of explosive eigenvalues. Then whatever choice of the forecast errors, the choice is not enough to extinguish the explosiveness of the system.

This condition is the same as that of Blanchard and Kahn's method. We still call it **Blanchard and Kahn Condition**.

As in the saddle path, we have $n_\eta = n_2$ and then there is unique solution. The choice of forecast error is:

$$\eta_t = \Xi \epsilon_t, \quad \Xi = -(Q^U \Gamma_0^{-1} \Pi)^{-1} Q^U \Gamma_0^{-1} \Psi$$

Substitute it into the stable portion of the system.

$$\begin{aligned}\tilde{s}_{1t} &= \Lambda_1 \tilde{s}_{1t-1} + Q^S \Gamma_0^{-1} \Psi \epsilon_t + Q^S \Gamma_0^{-1} \Pi \eta_t \\ &= \Lambda_1 \tilde{s}_{1t-1} + Q^S \Gamma_0^{-1} \Psi \epsilon_t + Q^S \Gamma_0^{-1} \Pi \Xi \epsilon_t \\ &= \Lambda_1 \tilde{s}_{1t-1} + (Q^S \Gamma_0^{-1} \Psi + Q^S \Gamma_0^{-1} \Pi \Xi) \epsilon_t\end{aligned}$$

The whole system is:

$$\begin{bmatrix} \tilde{s}_{1t} \\ \tilde{s}_{2t} \end{bmatrix} = \begin{bmatrix} \Lambda_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{s}_{1t-1} \\ \tilde{s}_{2t-1} \end{bmatrix} + \begin{bmatrix} Q^S \Gamma_0^{-1} \Psi + Q^S \Gamma_0^{-1} \Pi \Xi \\ 0 \end{bmatrix} \epsilon_t$$

Then we can go back to the original equation system by premultiplying P . Since $P \tilde{s}_t = s_t$.

$$s_t = P \begin{bmatrix} \Lambda_1 & 0 \\ 0 & 0 \end{bmatrix} P^{-1} s_{t-1} + P \begin{bmatrix} Q^S \Gamma_0^{-1} \Psi + Q^S \Gamma_0^{-1} \Pi \Xi \\ 0 \end{bmatrix} \epsilon_t$$

Or

$$s_t = \Phi s_{t-1} + \Phi \epsilon_t$$

This is the general solution for the log-linearized DSGE model.

Basically, in most of our applications in macroeconomics, $\Gamma_0^{-1} \Gamma_1$ is not of full rank so we can not use the eigenvalue decomposition to divide the stable and explosive portions of the system. Sims [2002] applies an alternative QZ decomposition, a general approach to decompose matrix, to obtain the stable and explosive portions of the system. Sims writes the procedure '*gensys.m*' to solve the linearized DSGE model and you can use this procedure to save the coding effort. All you need to do is just input the four matrix $\Gamma_0, \Gamma_1, \Psi, \Pi$ and

the procedure will give you the resulting policy function.

How about the scenario that there are multiple solutions? Basically, what you need to do is to solve the equation $Q^U \Gamma_0^{-1} \Pi \eta_t = -Q^U \Gamma_0^{-1} \Psi \epsilon_t$ to get infinite many solutions by following similar steps. In macroeconomics, we call this scenario **indeterminacy** or **sunspot**. Interested students can read the paper by Lubik and Schorfheide [2003] for more details.

4.4 Dynare

Dynare is a procedure built on top of MatLab. It is the procedure widely used in macroeconomics to solve and estimate the DSGE models. It is easy and quite simple. You don't even need to log-linearize your first order conditions to get the solution. You can download it from the following website: <http://www.dynare.org/download>.

To run Dynare, we should create mod files, which are *.mod* files. Simply type "Dynare filename" to implement the procedure. After installing dynare, remember to "set the path" to let Matlab know where to find these Dynare procedures.

Select the "Set Path" entry in the "File" menu, then click on "Add Folder...", and select the Matlab subdirectory of 'your Dynare installation. Note that you should not use "Add with Subfolders...". Apply the settings by clicking on "Save". Note that MATLAB will remember this setting next time you run it.

For more information, visit https://www.dynare.org/resources/quick_start/#configuring-matlab

Time Convention Notice that Dynare follows Sims' time convention that all predetermined variables should appear as a "t-1" variable instead of a beginning-of-period "t" variables. For example, in our optimal growth model, the capital in the beginning of Period t should be denoted as k_{t-1} . And all "t+1" variable means expectation of "t+1" variable conditional on time t information by default. Then if a equation has time $t + 1$ variable but do not have expectation, we should move the time period backward by one period. For

example, the motion of technology is:

$$\ln A_{t+1} = (1 - \rho) \ln \bar{A} + \rho \ln A_t + \epsilon_{t+1} \quad (92)$$

Here all time $t + 1$ variables A_{t+1}, ϵ_{t+1} do not actually have expectation, then we should move the time period backward by one as:

$$\ln A_t = (1 - \rho) \ln \bar{A} + \rho \ln A_{t-1} + \epsilon_t \quad (93)$$

Write the Dynare Program First, you have to save the program as a *.mode* program as *.m* program of Matlab. The dynare program has several essential sections. The first one is the declaration of the endogenous variables including endogenous state variables, exogenous state variables, and control variables. Remember that the number of endogenous variables should be equal to the number of first order conditions. In our example, we have three endogenous variables c_t, k_t, A_t .

```
1 var c k A;
```

Then declare the shocks of your model.

```
1 varexo epsilon;
```

The third is to declare the parameters of your model. In our example, we have $\beta, \alpha, \rho, \sigma, \bar{A}$ five parameters⁵.

```
1 parameters alpha beta rho sigma Abar;
```

Then we should calibrate all the parameters.

```
1 alpha = 1/3;
2 beta = 0.99;
3 rho = 0.9;
4 sigma = 0.01;
5 Abar = 5;
```

Next, we should declare the first order conditions of the DSGE model. We should start with "model" and end with "end" in the mod file in this section. Here we have several ways to

⁵ \bar{A} is the steady state of technology, which is given in the example.

write out the model. The first approach is to write down the original equations but express all the endogenous variables as logarithm. As Dynare linearizes your model around the steady state, we can write the FOCs by expressing all the variables in terms of the logarithms.

$$u'(e^{\ln c_t}) = \beta E_t u'(e^{\ln c_{t+1}}) e^{\ln A_{t+1}} \alpha (e^{\ln k_t})^{\alpha-1} \quad (94)$$

$$e^{\ln c_t} + e^{\ln k_t} = e^{\ln A_t} (e^{\ln k_t})^\alpha \quad (95)$$

$$\ln(e^{\ln A_t}) = (1 - \rho)\bar{A} + \rho \ln(e^{\ln A_{t-1}}) + \epsilon_t \quad (96)$$

```

1 model;
2 1/exp(c) = 1/exp(c(+1))*exp(A(+1))*alpha*(exp(k))^(alpha-1);
3 exp(c) + exp(k)=exp(A)*(exp(k(-1)))^alpha;
4 log(exp(A)) = (1-rho)*log(Abar) + rho*log(exp(A(-1))) + epsilon;
5 end;
```

Notice that all time $t + 1$ variables x with conditional expectation appear as $x(+1)$ in the mod file. All the time $t - 1$ variables appears as $x(-1)$. We do not add any postfix for time t variables.

Then, we should declare a set of approximate values for the steady state of the endogenous variables. Dynare will find the steady states for you.

```

1 %declare a set of approximate values for the steady state
2 initval;
3 k = log(2);
4 c = log(2);
5 A = log(5);
6 end;
7
8 steady; %find the steady state
```

At last, call solution method by dynare.

```

1 stoch_simul(order=1,nograph); %first order perturbation
```

The mod file is finished and the whole mod file is as follows.

```

1 %=====
2 % Chapter 2
3 % File: ramsey.mod
4 % This program file is to solve the infinite optimal growth model with
```

```

5 % Sims' method by dynare
6 % Written: 2020.12.07
7 % Written by Bin Wang
8 %=====
9
10
11 % Preamble
12 var c k A;
13
14 varexo epsilon;
15
16 parameters alpha beta rho sigma Abar;
17
18 % parameter values
19 alpha = 1/3;
20 beta = 0.99;
21 rho = 0.9;
22 sigma = 0.01;
23 Abar = 5;
24
25 model;
26 1/exp(c) = 1/exp(c(+1))*exp(A(+1))*alpha*(exp(k))^(alpha-1);
27 exp(c) + exp(k)=exp(A)*(exp(k(-1)))^alpha;
28 log(exp(A)) = (1-rho)*log(Abar) + rho*log(exp(A(-1))) + epsilon;
29 end;
30
31 initval;
32 k = log(2);
33 c = log(2);
34 A = log(5);
35 end;
36
37 %shocks;
38 %var epsilon = sigma^2;
39 %end;
40
41 steady;
42
43 stoch_simul(order=1);

```

Save the mod file in the working path you choose. Then type "dynare ramsey.mod" in the command window⁶. The output of Dynare first appears in the command window of Matlab.

⁶You should first add the path of Dynare. Or you should tell Matlab where to find the procedure of

```

1 Using 64-bit preprocessor
2 Starting Dynare (version 4.6.3).
3 Calling Dynare with arguments: none
4 Starting preprocessing of the model file ...
5 Found 3 equation(s).
6 Evaluating expressions...done
7 Computing static model derivatives (order 1).
8 Computing dynamic model derivatives (order 1).
9 Processing outputs ...
10 done
11 Preprocessing completed.
12
13
14 STEADY-STATE RESULTS:
15
16 c 1.45939
17 k 0.766238
18 A 1.60944
19
20 MODEL SUMMARY
21
22 Number of variables: 3
23 Number of stochastic shocks: 1
24 Number of state variables: 2
25 Number of jumpers: 2
26 Number of static variables: 0
27
28
29 MATRIX OF COVARIANCE OF EXOGENOUS SHOCKS
30 Variables epsilon
31 epsilon 0.000000
32
33 POLICY AND TRANSITION FUNCTIONS
34
35 A
36 Constant 1.459386 0.766238 1.609438
37 k(-1) 0.333333 0.333333
38 0
39 A(-1) 0.900000 0.900000 0.900000
40 epsilon 1.000000 1.000000 1.000000

```

Dynare everytime to run mod file by "addpath ...".

```

41 THEORETICAL MOMENTS
42 VARIABLE      MEAN   STD. DEV.    VARIANCE
43 c             1.4594   0.0000   0.0000
44 k             0.7662   0.0000   0.0000
45 A             1.6094   0.0000   0.0000
46
47 All endogenous are constant or non stationary, not displaying correlations
48
49 Total computing time : 0h00m00s

```

Check the "policy and transition functions" part of the output. You can read that as:

$$c = 1.459386 + 0.333333k(-1) + 0.9A(-1) + \epsilon \quad (97)$$

$$k = 0.766238 + 0.333333k(-1) + 0.9A(-1) + \epsilon \quad (98)$$

$$A = 1.609438 + 0.9A(-1) + \epsilon \quad (99)$$

As we express all the variables in logarithms, this policy functions are actually:

$$\ln c_t = 1.459386 + 0.333333 \ln k_{t-1} + 0.9 \ln A_{t-1} + \epsilon_t \quad (100)$$

$$\ln k_t = 0.766238 + 0.333333 \ln k_{t-1} + 0.9 \ln A_{t-1} + \epsilon_t \quad (101)$$

$$\ln A_t = 1.609438 + 0.9 \ln A_{t-1} + \epsilon_t \quad (102)$$

Remember that we solve this model by Blanchard and Kahn's method, the policies are:

$$\hat{c}_t = \alpha \hat{k}_{t-1} + \rho \hat{A}_{t-1} + \epsilon_t \quad (103)$$

$$\hat{k}_t = \alpha \hat{k}_{t-1} + \rho \hat{A}_{t-1} + \epsilon_t \quad (104)$$

$$\hat{A}_t = \rho \hat{A}_{t-1} + \epsilon_t \quad (105)$$

Notice that $\hat{x}_t = \ln x_t - \ln \bar{x}$, $x = c, k, A$. We can see that they are equivalent.

How can we get the solution exactly the same as what we did in the Blanchard and Kahn's method or Sims' method. You have to type the log-linearized FOCs in the model section. Before that, you have to solve the steady states by yourself and add these steady states as parameters declared in the parameter section.

```

1 %=====
2 % Chapter 2

```

```

3 % File: ramsey1.mod
4 % This program file is to solve the infinite optimal growth model with
5 % Sims' method by dynare
6 % log-linearized FOCs
7 % In this program x is the percentage change from steady state x=c,k,A
8 % Then the steady states of x should be all 0.
9 % Written: 2020.12.07
10 % Written by Bin Wang
11 %=====
12
13 %addpath /Applications/Dynare/4.6.3/matlab;
14
15 % Preamble
16 var c k A;
17
18 varexo epsilon;
19
20 parameters alpha beta rho sigma Abar kbar cbar up upp;
21
22 % parameter values
23 alpha = 1/3;
24 beta = 0.99;
25 rho = 0.9;
26 sigma = 0.01;
27 Abar = 5;
28
29 %steady states
30 kbar = (Abar*alpha*beta)^(1/(1-alpha));
31 cbar = Abar*kbar^alpha - kbar;
32 up = 1/cbar; %u'
33 upp = -1/cbar^2; %u ''
34
35 model;
36 upp*cbar*c = beta*alpha*Abar*kbar^(alpha-1)*up*A(+1)+beta*alpha*Abar*up*(alpha-1);
37 cbar*c + kbar*k = Abar*kbar^alpha*A + Abar*alpha*kbar^(alpha-1)*k(-1);
38 A = rho*A(-1) + epsilon;
39 end;
40
41 initval;
42 k = 0;
43 c = 0;
44 A = 0;
45 end;
46
47 %shocks;

```

```

48 %var epsilon = sigma^2;
49 %end;
50
51 steady;
52
53 stoch_simul(order=1);

```

You can see that we add $\bar{c}, \bar{k}, u', u''$ as parameters in this setup. Right now c, k, A in the mod file actually means percentage change from their steady states $\hat{c}_t, \hat{k}_t, \hat{A}_t$. The equations in the model section are log-linearized FOCs of the optimal growth model.

$$u''(\bar{c})\bar{c}\hat{c}_t = \beta\alpha\bar{A}\bar{k}^{\alpha-1}u'(\bar{c})E_t\hat{A}_{t+1} + \beta\alpha\bar{A}u'(\bar{c})(\alpha - 1)\bar{k}^{\alpha-2}\bar{k}\hat{k}_t + \beta\alpha\bar{A}\bar{k}^{\alpha-1}u''(\bar{c})\bar{c}E_t\hat{c}_{t+1} \quad (106)$$

$$\bar{c}\hat{c}_t + \bar{k}\hat{k}_t = \bar{A}\bar{k}^\alpha\hat{A}_t + \bar{A}\alpha\bar{k}^{\alpha-1}\bar{k}\hat{k}_{t-1} \quad (107)$$

$$\hat{A}_t = \rho\hat{A}_{t-1} + \epsilon_t \quad (108)$$

Notice that we didn't move the Euler equation backward by one period as Sims' method. That's because Dynare requires all time $t + 1$ variables are actually conditional expectation and time t variables have no expectation. If we move the time backward, $E_{t-1}\hat{c}_t$ will be considered as \hat{c}_t without expectation. The output of this mod file is:

```

1 Using 64-bit preprocessor
2 Starting Dynare (version 4.6.3).
3 Calling Dynare with arguments: none
4 Starting preprocessing of the model file ...
5 Found 3 equation(s).
6 Evaluating expressions...done
7 Computing static model derivatives (order 1).
8 Computing dynamic model derivatives (order 1).
9 Processing outputs ...
10 done
11 Preprocessing completed.
12
13
14 STEADY-STATE RESULTS:
15
16 c 0
17 k 0
18 A 0

```

```

19
20 MODEL SUMMARY
21
22 Number of variables: 3
23 Number of stochastic shocks: 1
24 Number of state variables: 2
25 Number of jumpers: 2
26 Number of static variables: 0
27
28
29 MATRIX OF COVARIANCE OF EXOGENOUS SHOCKS
30 Variables epsilon
31 epsilon 0.000000
32
33 POLICY AND TRANSITION FUNCTIONS
34
35 A
36 k(-1) 0.333333 0.333333
37 0
38
39
40 THEORETICAL MOMENTS
41 VARIABLE MEAN STD. DEV. VARIANCE
42 c 0.0000 0.0000 0.0000
43 k 0.0000 0.0000 0.0000
44 A 0.0000 0.0000 0.0000
45
46 All endogenous are constant or non stationary, not displaying correlations
47
48 Total computing time : 0h00m00s

```

You can see that steady states of all three variables are zero as expected. The policy functions are the same as what we get in Blanchard and Kahn's method.

References

- Olivier Jean Blanchard and Charles M Kahn. The Solution of Linear Difference Models under Rational Expectations. *Econometrica*, 48(5):1305–1311, July 1980. URL <https://ideas.repec.org/a/ecm/emetrp/v48y1980i5p1305-11.html>.
- Thomas A. Lubik and Frank Schorfheide. Computing sunspot equilibria in linear rational expectations models. *Journal of Economic dynamics and control*, 28(2):273–285, 2003. URL <http://www.sciencedirect.com/science/article/pii/S0165188902001537>.
- Christopher A Sims. Solving Linear Rational Expectations Models. *Computational Economics*, 20(1-2):1–20, October 2002. URL <https://ideas.repec.org/a/kap/compec/v20y2002i1-2p1-20.html>.