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.

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

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 binding[1]. Then the social planner's problem is simplified as:

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

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:

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

[1] An 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 $\lambda_t, t = 0, \cdots T; \mu_T$ 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:

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

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 $(c_0^*, \cdots, c_T^*, k_1^*, \cdots, k_{T+1}^*)$ 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:	$c_t + k_{t+1} = A k_t^{\alpha}, \forall 0 \leq t \leq T$
Initial Condition:	k_0 is given
Transversality Condition:	$k_{T+1} = 0$

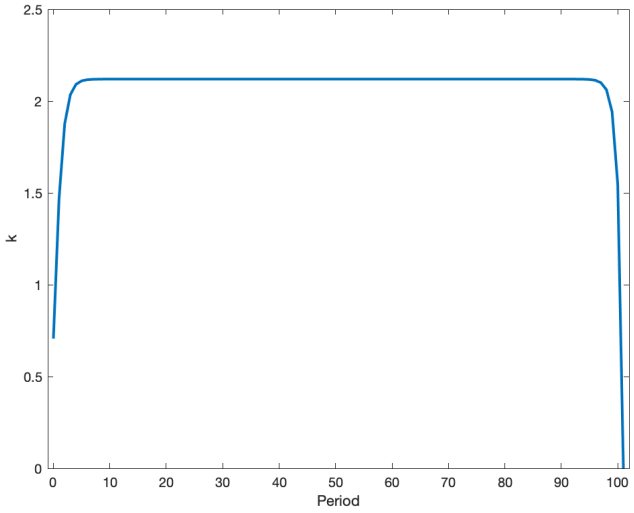
conditions are both necessary and sufficient for the optimizer of the social planner's problem and the transversality condtion can pin down the solution of the problem. To see this point, we substitute c_t of the resource constraint to the Euler condition. We have $T+1$ unknowns $k_1, \cdots, k_{T+1}$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

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

$$k_0, given; k_{T+1} = 0. \tag{3}$$

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 $k_1, \cdots, k_{T+1}$ is drafted in Figure 1.

Figure 1: Equilibrium Capital




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

```
% Written by Bin Wang
%============================================================

clear all;
close all;
clc;

%% Parameters
A = 5;
alpha = 1/3;
beta = 0.99;
T = 100;
kbar = (A*alpha*beta)^(1/(1-alpha));
k0 = 1/3*kbar;
para = [A,alpha,beta,k0];
kt = zeros(T,1);
kTplus1 = 0;
kt = fsolve(@(k)(finiteRamsey(k,para)),k0*ones(T,1));

k = [k0;kt;kTplus1];

figure;
plot(0:1:(T+1),k,'LineWidth',2);
xlabel('Period');
ylabel('k');
xlim([-1 T+2]);

%% function
function F=finiteRamsey(k,para)
A=para(1);
alpha = para(2);
beta = para(3);
k0 = para(4);
kTplus1=0;
T = length(k);
F = zeros(T,1);
F(1) = A*alpha*beta*k(1)^(alpha-1)/(A*k(1)^alpha-k(2))
-1/(A*k0^alpha-k(1)); %k0 is given
for t=1:T-2
    F(t+1)= A*alpha*beta*k(t+1)^(alpha-1)/(A*k(t+1)^alpha
    -k(t+2))-1/(A*k(t)^alpha-k(t+1));
end
F(T)=A*alpha*beta*k(T)^(alpha-1)/(A*k(T)^alpha-kTplus1)
-1/(A*k(T-1)^alpha-k(T));
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**.

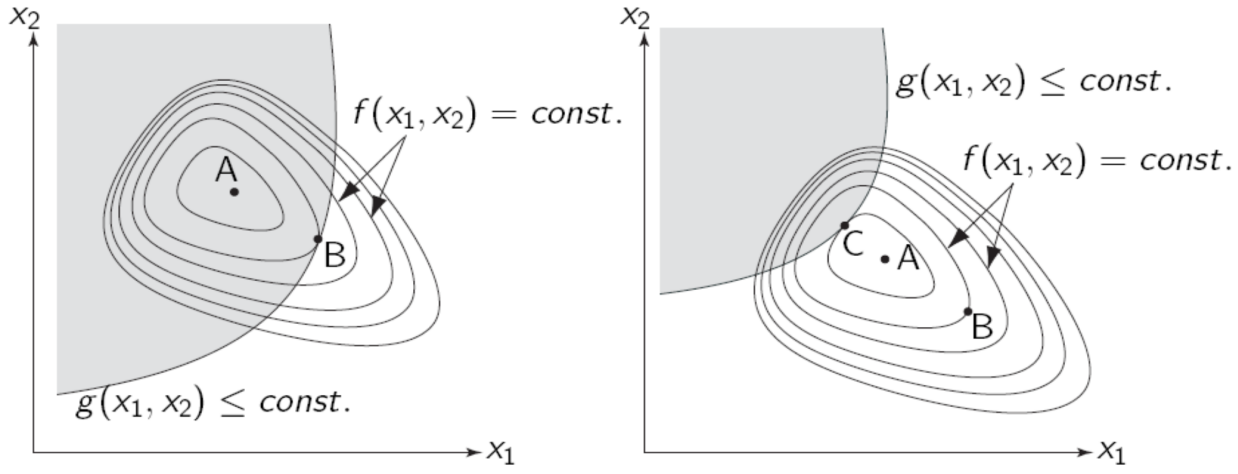
$$k = (A\alpha\beta)^{\frac{1}{1-\alpha}}$$

Kuhn-Tucker Theorem For the optimization problem

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

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

$$g(x_1^*, x_2^*) = g(x_1^A, x_2^A) < m \tag{4}$$

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.

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

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

$$g(x_1^A, x_2^A) > 0 \tag{7}$$

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

$$L(x_1, x_2, \lambda) = f(x_1, x_2) - \lambda(g(x_1, x_2) - m) \tag{8}$$

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

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

$$g(x_1^A, x_2^A) > 0, \lambda^A = 0 \tag{11}$$

- 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:

$$g(x_1^*, x_2^*) = g(x_1^C, x_2^C) = m \tag{12}$$

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 senario 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

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

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:

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

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

$$L(x_1, x_2, \lambda) = f(x_1, x_2) - \lambda(g(x_1, x_2, \lambda) \tag{13}$$

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

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**.

$$L(x_1, x_2, \lambda) = f(x_1, x_2) - \lambda(g(x_1, x_2, \lambda) \tag{14}$$

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

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-horzion:

$$\begin{aligned}
\textbf{SP: } v^*(k_0) = \max_{\{c_t\}_{t=0}^{\infty}, \{k_{t+1}\}_{t=0}^{\infty}} \quad & \sum_{t=0}^{\infty} \beta^t u(c_t) \\
s.t. \quad & c_t + k_{t+1} \leq f(k_t) \\
& c_t, k_{t+1} \geq 0 \\
& k_0, given
\end{aligned}$$

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

$$\begin{aligned}
\textbf{SP: } v^*(k_0) = \max_{\{k_{t+1}\}_{t=0}^{\infty}} \quad & \sum_{t=0}^{\infty} \beta^t u(f(k_t) - k_{t+1}) \\
s.t. \quad & k_{t+1} \geq 0 \\
& k_0, given
\end{aligned}$$

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.

$$\begin{aligned}
\textbf{FE: } v(k_t) &= \max_{k_{t+1}} \quad u(f(k_t) - k_{t+1}) + \beta v(k_{t+1}), \quad \forall k_t \\
s.t. \quad & 0 \leq k_{t+1} \leq f(k_t) \\
& k_t, given
\end{aligned}$$

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

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

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.

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

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.

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

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. □

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.

$$\begin{aligned}
&\textbf{FE: } v(k) = \max_{k'} \quad u(f(k) - k') + v(k'), \quad \forall k \\
&\quad s.t. \quad 0 \le k' \le f(k) \\
&\qquad\quad k, given
\end{aligned}$$

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 $v'(k')$.

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

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

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

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.

$$v'(k) = u'(f(k') - k'')f'(k') \tag{17}$$

Substitute it into the first order condition:

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

Or

$$u'(c) = \beta u'(c')f'(k') \tag{19}$$

This 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

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

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

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

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:

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

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

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

Let

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

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.

$$\begin{aligned}
v(k) = \max_{c,k'} \quad & u(c)+\beta v(k'), \\
s.t. \quad & c+k' = f(k) \\
& k, given
\end{aligned}$$

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:

$$\begin{aligned} L_c &= u'(c) - \lambda = 0 \\ L_{k'} &= \beta v'(k')\lambda = 0 \\ L_\lambda &= f(k) - c - k' = 0 \end{aligned}$$

Combine these conditions:

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

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:

$$\begin{aligned} u'(c) &= \beta u'(c') f(k') \\ c &= f(k) - k' \end{aligned}$$

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

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

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:	$c_t + k_{t+1} = A k_t^{\alpha}$
Initial Condition:	k_0 is given
Transversality Condition:	$\lim_{t\to\infty} \beta^t u'(c_t) f'(k_t) k_t = 0$

only difference lies on the transversality condition.

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 (c, k) space, we can analyze how each point (c_t, k_t) in the space evolve by calculating $(c_{t+1} - c_t, k_{t+1} - k_t)$. If both are positive, the point (c_t, k_t) moves northeastward. If both are negative, the point (c_t, k_t) moves southwestward. Likewise, you can analyze the other two scenarios.

Before calculating the difference of (c, k), 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:

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

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.

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

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

$$c_{t+1} - c_t = \frac{u'}{u''}\left[1 - \frac{1}{\beta A \alpha k_t^{\alpha-1}}\right]$$

When $k_t > \bar{k} = (A\alpha\beta)^{\frac{1}{1-\alpha}}$, $c_{t+1} - c_t < 0$. When $k_t < \bar{k} = (A\alpha\beta)^{\frac{1}{1-\alpha}}$, $c_{t+1} - c_t > 0$. And when $k_t = \bar{k} = (A\alpha\beta)^{\frac{1}{1-\alpha}}$, $c_{t+1} - c_t = 0$ and thus c_t stays. This can be shown in Figure . To the left, c_t moves up. To the right of $\bar{k}$, c_t moves down.

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

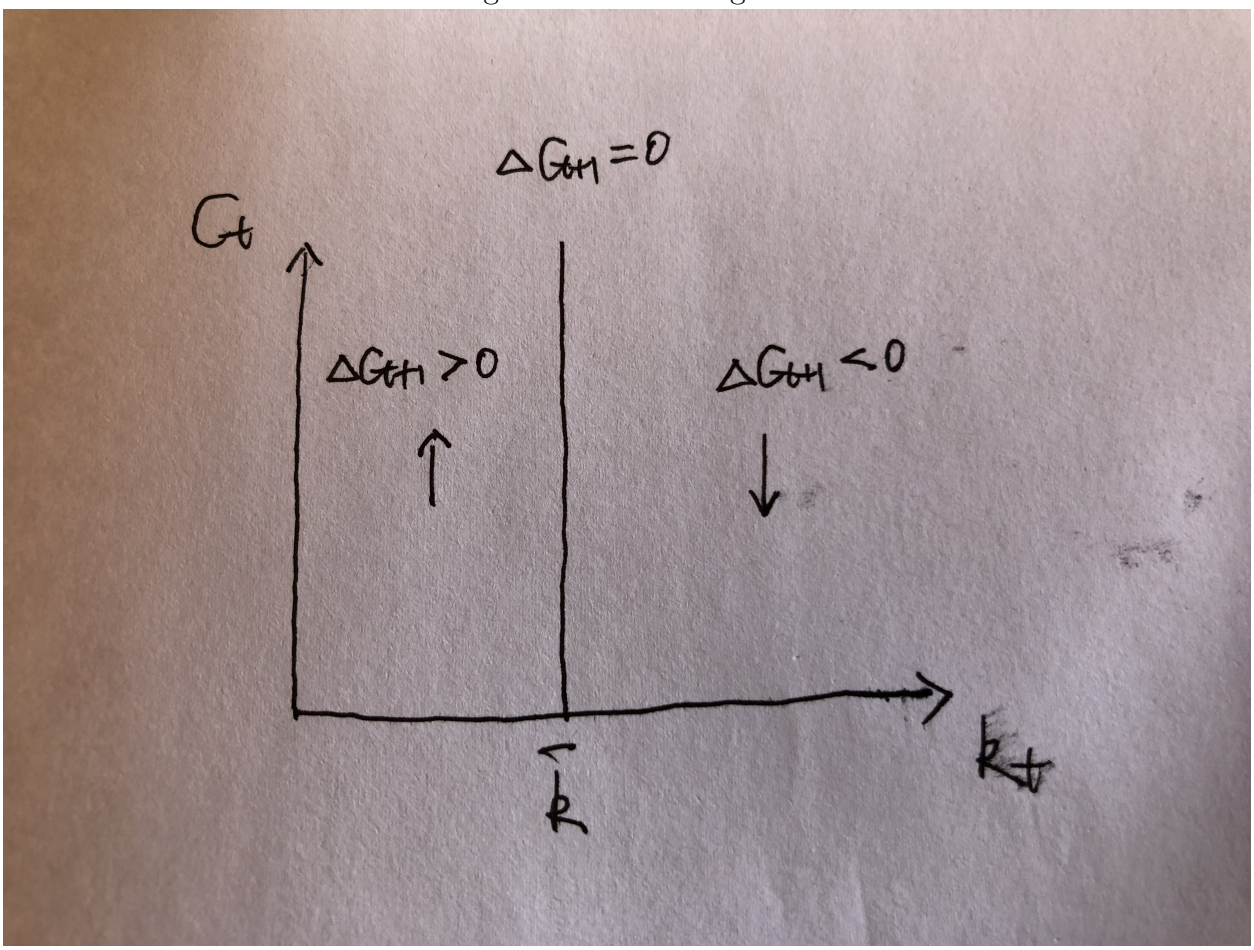
$$c_t + k_{t+1} = Ak_t^{\alpha}$$

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

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

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 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

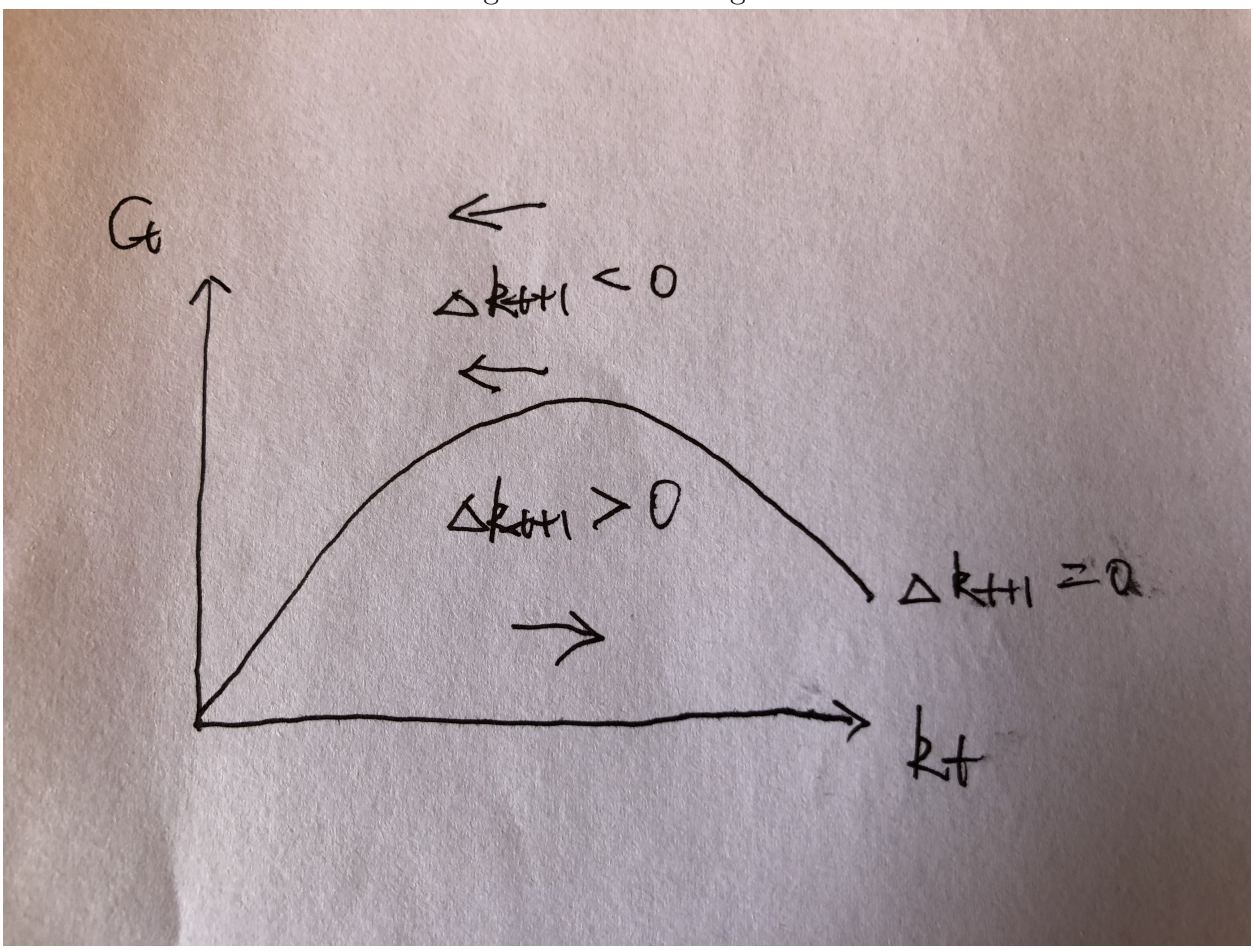
Figure 3: Phase Diagram



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 equationand 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 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

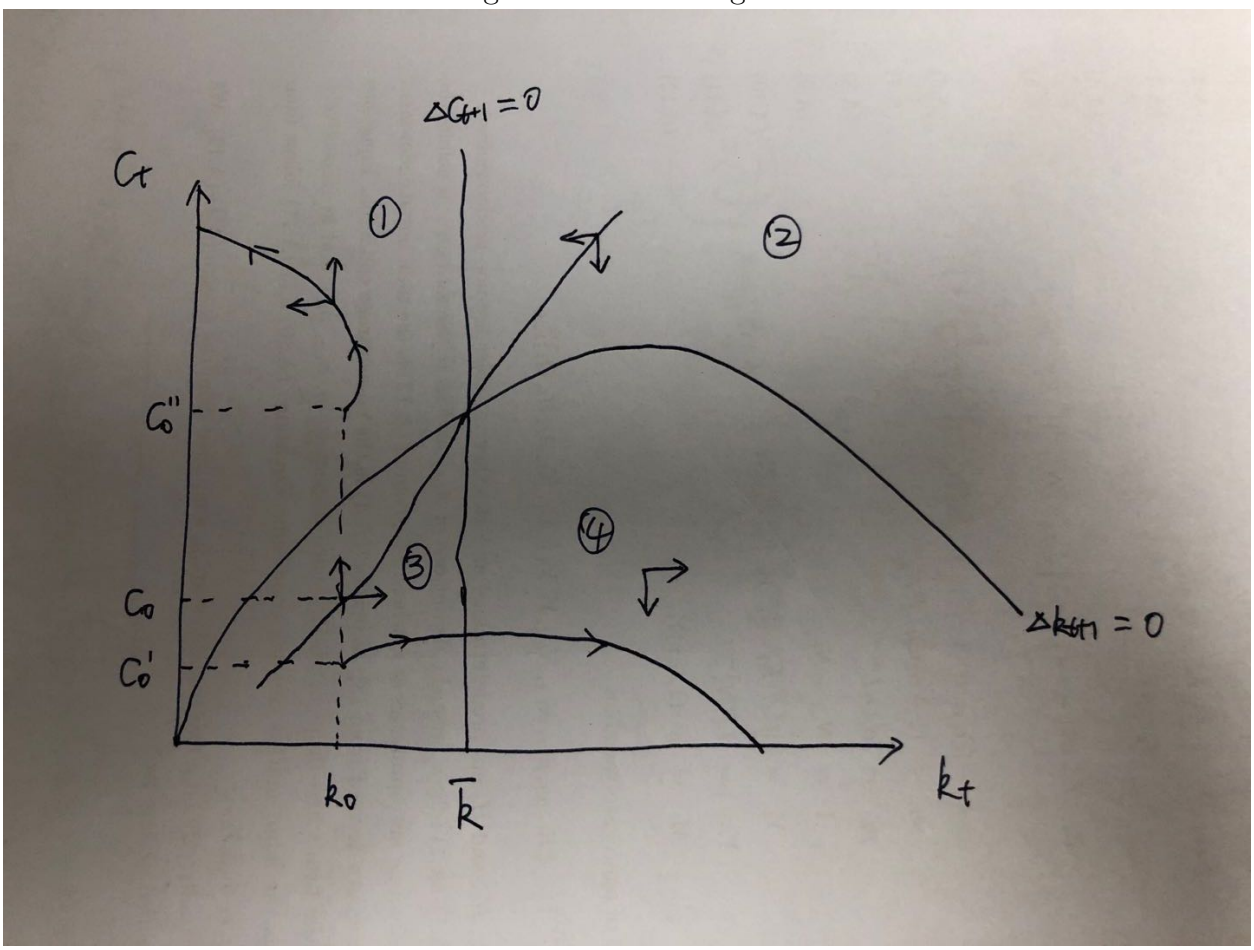
Figure 4: Phase Diagram



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 $c_0 = (1 - \alpha\beta) A k_0^\alpha$. We can check numerically whether other path that $c_0 > (1 - \alpha\beta) A k_0^\alpha$ or $c_0 < (1 - \alpha\beta) A k_0^\alpha$ 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.

[2] This result will be shown in the next subsection.

Figure 5: Phase Diagram



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 neibourhood of the steady state. We call it **local dynamics** around the steady state. Consider the first order conditions:

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

Figure 6: Saddle Path and Transversality Condtion

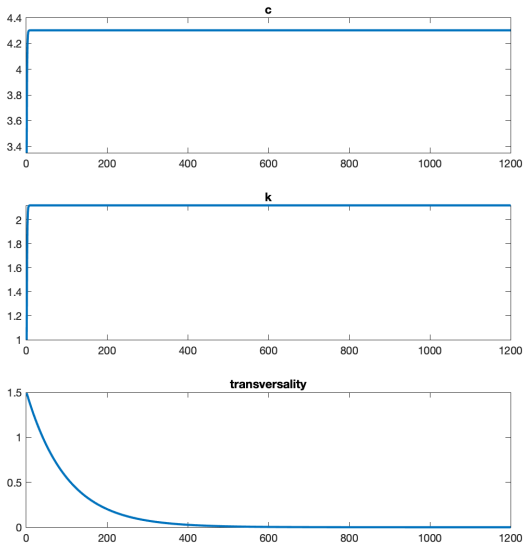
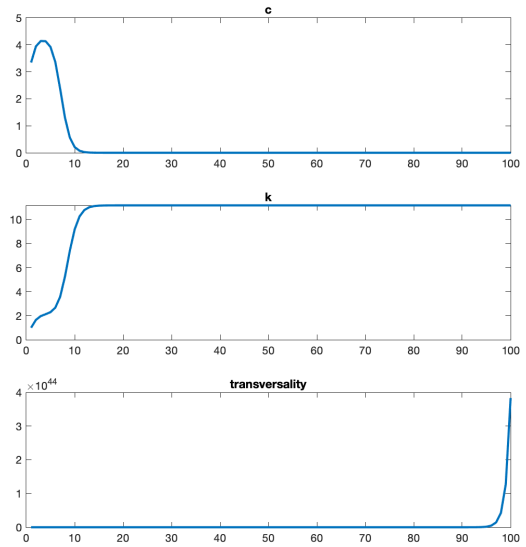


Figure 7: Saddle Path and Transversality Condtion

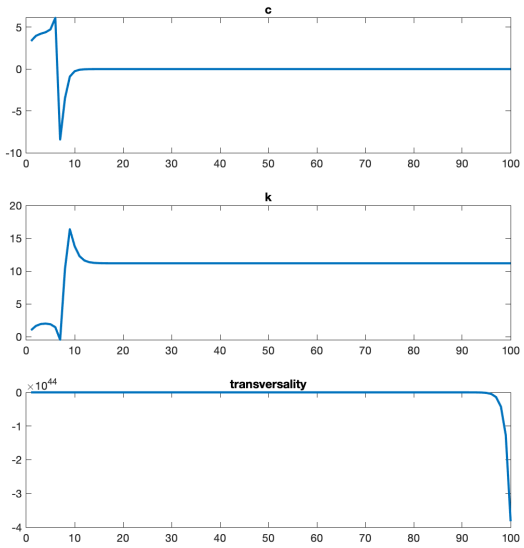


We can apply the Taylor series theorem to get the first-order Taylor expansion around the steady state to the LHS and RHS of both equations. Let $\tilde{c}_t = c_t - \bar{c}$ and $\tilde{k}_t = k_t - \bar{k}$.

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

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

Figure 8: Saddle Path and Transversality Condtion



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:

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

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

Or

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

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

$$B = P\Lambda P^{-1} \tag{27}$$

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).

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

Or

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

Iterate the above equation and we have:

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

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

$$\tilde{c}_t = -\frac{Q_{22}}{Q_{21}}\tilde{k}_t \tag{31}$$

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

- $|\lambda_1| > 1, |\lambda_2| > 1$. 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 $\lambda_1 + \lambda_2 = 1 + \frac{(1-\alpha\beta)(1-\beta)}{\alpha\beta} + \frac{1}{\beta}, \lambda_1\lambda_2 = \frac{1}{\beta}$, we can check that one eigenvalue is bigger than 1 and the other eigenvalue is less than one. So 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.

```
%============================================================
% Chapter 2
% File: ch2_local_dynamics.m
% This program file is to solve check the local dynamics around the steady
% state of the optimal growth model
% Written: 2020.10.26
% Written by Bin Wang
%============================================================

clear all;
```

[3] We assume $\lambda_1 < \lambda_2$.

```
close all;
clc;

%% Parameters
A = 5;
alpha = 1/3;
beta = 0.99;
T = 100;
kbar = (A*alpha*beta)^(1/(1-alpha));
cbar = A*kbar^alpha-kbar;
fp = A*alpha*kbar^(alpha-1);                    %f'(kbar)
fpp = A*alpha*(alpha-1)*kbar^(alpha-2);         %f''(kbar)
up = 1/cbar;                                    %u'(cbar)
upp = -1/(cbar^2);                              %u''(cbar)
B1 = [beta*upp*fp beta*up*fpp; 0 1];
B2 = [upp 0; -1 fp];
B = B1\B2;                                      %B matrix in the lecture notes

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

%eigenvalue should be ordered asendingly
[Lambda_sort,index]=sort(diag(Lambda));
P_sort = P(:,index);
Lambda = diag(Lambda_sort);
P = P_sort;

%% You can check that one eigenvalue is less than one and one is bigger than
%one. So there is one stable path which is the saddle path.
Q = inv(P);

%simulate the path
T=100;
k0 = 1/3*kbar;
k_tilde=zeros(T,1);
c_tilde=zeros(T,1);
k_tilde(1) = k0-kbar;  %\tilde{k}_0$
c_tilde(1) = -Q(2,2)/Q(2,1)*k_tilde(1);
for t=2:T
   k_tilde(t)=Lambda(1,1)^(t-1)*(Q(1,1)*c_tilde(1)+Q(2,2)*k_tilde(1))/(-Q(1
   c_tilde(t)=-Q(2,2)/Q(2,1)*k_tilde(t);
end

figure;
plot(1:T,k_tilde,'LineWidth',2);
```

```
title('k tilde');

figure;
plot(1:T,c_tilde,'LineWidth',2);
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**.

2.2 Dynamic Programming: General Form

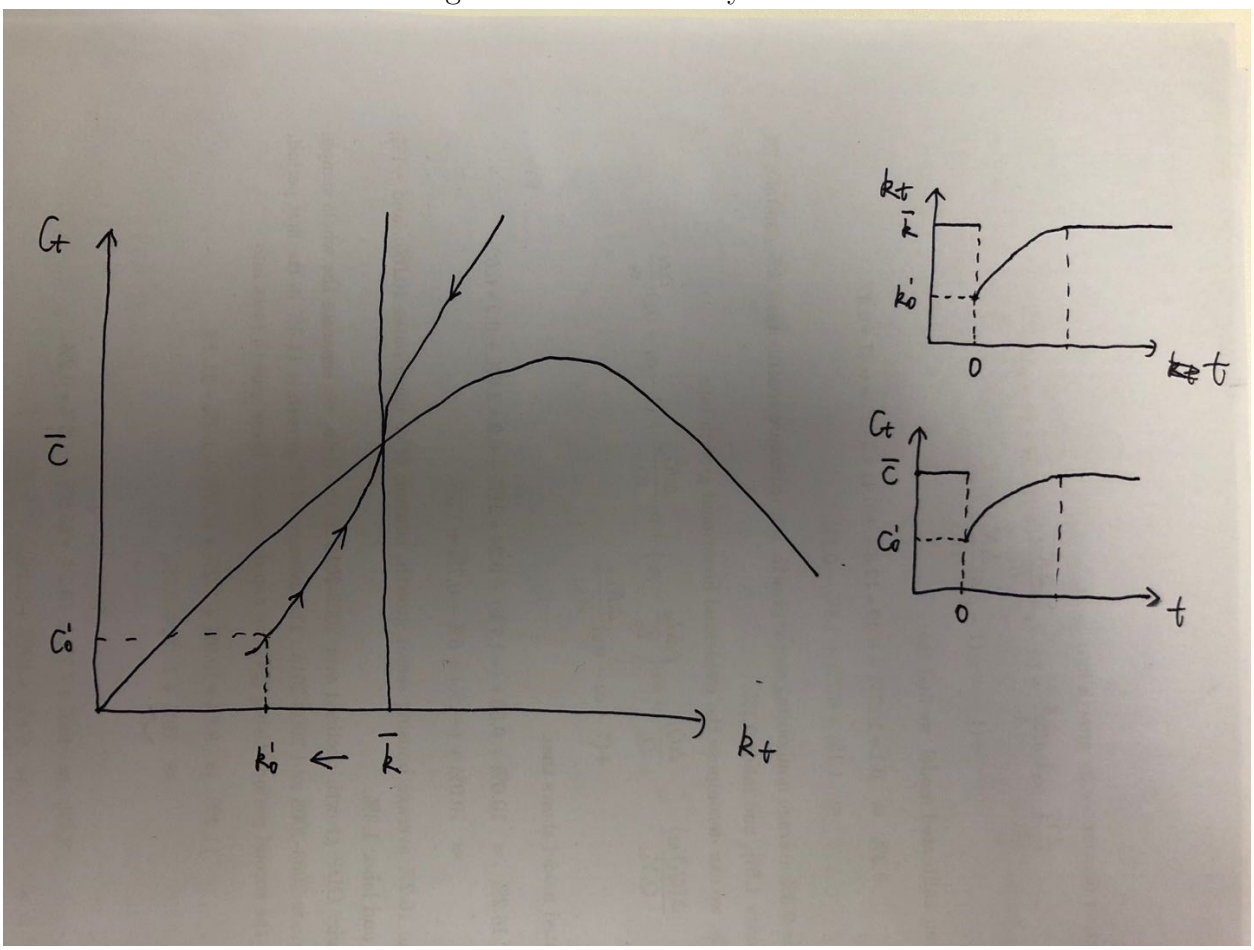
A typical dynamic programming problem is to choose an infinite sequence of controls $\{u_t\}_{t=0}^{\infty}$ to maximize

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

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

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

Figure 9: Transition Dynamics



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 $v(x)$:

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

Notice that the derivative of value function is also evaluated only at the optimal point $u = h(x)$. Substitute the derivative of value function (36) into the first order condition (35).

$$r_u(x, h(x)) + \beta r_x(g(x, h(x)), h(g(x, h(x)))) g_u(x, h(x)) = 0 \tag{37}$$

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).

The transversality condition is:

$$\lim_{t\to\infty} \beta^t v_x(x_t, u_t) x_t = 0 \tag{38}$$

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 $(\bar{A}, \pi_0, P)$ 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.

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

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:

$$\begin{aligned} & v(k,A) = \max_{c,k'} u(c) + \beta E_t[v(k',A')] \\ s.t. \quad & c + k' = Ak^{\alpha} \\ & Prob(A' = \bar{A}_j | A = \bar{A}_i) = P_{ij} \end{aligned}$$

Where $E_t[v(k',A')]$ is just a conditional expectation conditional on realization of present technology A. Substitute the definition of conditional expectation.

$$\begin{aligned} & v(k,\bar{A}_i) = \max_{c,k'} u(c) + \beta \sum_{j=1}^{n} v(k',\bar{A}_j) P_{ij} \\ s.t. \quad & c + k' = Ak^{\alpha} \end{aligned}$$

where $Prob(A' = \bar{A}_j | A = \bar{A}_i) = P_{ij}$. 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 substitete 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 transversality condition.

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

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

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.

$$\max_{c_t, a_{t+1}} \sum_{t=0}^{\infty} \beta^t u(c_t)$$

$$s.t. \quad c_t + a_{t+1} \leq r_t a_t + w_t + \pi_t$$

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

$$a_0, \quad given$$

Write down the Bellman equation of this sequence problem:

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

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

Substitute c into the Bellman equation

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

The FOC is:

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

According to the envelope theorem:

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

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

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

The transversality condition is:

$$\lim_{t\to\infty 0} \beta^t v_t'(a_t)a_t = 0$$

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:

$$c_t = Ak_t^{\alpha} n_t^{1-\alpha}$$
$$a_t = k_t$$
$$1 = n_t$$

Competitive Equilibrium The competitive equilibrium is an allocation $(c_t, a_t, \pi_t, n_t, k_t)$ and a price system r_t, w_t 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:

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

This is the same as the Euler equation of the social planner's problem, and the transversality condition is also the same.

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:

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

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:

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

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.

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

Write down the Bellman equation of this sequence problem:

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

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.

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

The first order condition is:

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

According to the envelope theorem:

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

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

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

And the transversality condition is:

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

The optimal solution is characterized by the initial condition, resource constraint, Euler equation, and transversality condition. 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 $g_Y = g_K = g_L$ 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-Douglous 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:

$$\begin{aligned} &\max_{c_t,k_{t+1}} \sum_{t=0}^{\infty} \beta^t u(c_t, \bar{c}_{t-1}) \\ s.t. \quad & c_t + k_{t+1} \leq A k_t^{\alpha} \\ & c_t, k_{t+1} \geq 0 \\ & k_0, \quad given \end{aligned}$$

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:

$$\begin{aligned} & v_t(k_t, \bar{c}_{t-1}) = \max_{c_t,k_{t+1}} u(c_t, \bar{c}_{t-1}) + \beta v_{t+1}(k_{t+1}, \bar{c}_t) \\ s.t. \quad & c_t + k_{t+1} = A k_t^{\alpha} \end{aligned}$$

Substitute k_{t+1}:

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

The first order condition is:

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

According to the envelope theorem:

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

Substitute this marginal value to the first order condition:

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

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:

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

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

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 neibourhood 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 $k' = \beta\alpha A k^{\alpha}$.

   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:

$$\begin{aligned} v_{j+1}(x) &= \max_{u} r(x,u) + \beta v_j(x') \\ s.t. \quad & x' = g(x,u) \end{aligned}$$

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 $(x^1, \cdots, x^N)$. 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:

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

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:

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

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

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

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, $k = [k^1, k^2, \cdots, k^N]$ with $k^1 = \frac{1}{5}\bar{k}$ and $k^N = 5\bar{k}$.

```
A = 5;
alpha = 1/3;
beta = 0.99;
kbar = (A*alpha*beta)^(1/(1-alpha));
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.

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

```
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 vkk. The dimension of this matrix is $N \times N$ where N is the dimension of k. Each entry of this matrix vkk 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.

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

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

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

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

```
  d0 = d1;
  it = it+1;
```

The whole program is as follows:

```
%=========================================================
% Chapter 2
% File: ch2_solve_k_infinite_vfi.m
% This program file is to solve the finite optimal growth model
% Written: 2020.09.19
% Written by Bin Wang
%=========================================================

clear all;
close all;
clc;

%% Parameters
A = 5;
alpha = 1/3;
beta = 0.99;
kbar = (A*alpha*beta)^(1/(1-alpha));
k=1/5*kbar:0.02:5*kbar;                 %discretize the domain of k
N = length(k);                          %number of points in the domain of va
v0 = zeros(N,1);                        %initial value function v0=0 or consi
tol = 1e-5;                             %tolerance of the convergence
maxI = 1e5;                             %maximum iteration
diff = 2;                               %value function difference of each it

it = 0;                                 %count of iteration

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

```
            end
        end
    end
    %=========================================================================

    %choose k' for each k
    [v1,d1]= max(vkk,[],2);           %chose the maximum value along the row
    diff = max(abs(v1-v0));           %the difference between v_{j} and v_{
    v0 = v1;                          %give the new function value to v_{j}
    d0 = d1;
    it = it+1;
end

%%
if diff<tol
    disp(['Value function iteration succeds']);
    disp(['Iteration    '      'diff']);
    disp([it diff]);
end
figure;
plot(k,v0,'LineWidth',2);
xlabel('k');
ylabel('v');
fig = gcf;
fig.PaperUnits = 'inches';
fig.PaperPosition = [0 0 8 4];
saveas(fig,'ch2_solve_k_infinite_vfi_value_function.png');
figure;

plot(k,k(d0),'LineWidth',2);
xlabel('k');
ylabel("k'");
ylim([0 12]);
fig = gcf;
fig.PaperUnits = 'inches';
fig.PaperPosition = [0 0 8 8];
saveas(fig,'ch2_solve_k_infinite_vfi_policy_function.png');


figure;
plot(k,A*k.^alpha-k(d0),'LineWidth',2);hold on;
plot(kbar*ones(length(0:0.05:12),1),0:0.05:12,'r','LineWidth',2);
plot(0:0.05:12,A*(0:0.05:12).^alpha-(0:0.05:12),'r','LineWidth',2);
xlabel('k');
ylabel("c");
```

```
ylim([0 12]);
xlim([0 12]);
fig = gcf;
fig.PaperUnits = 'inches';
fig.PaperPosition = [0 0 8 8];
saveas(fig,'ch2_solve_k_infinite_vfi_dynamic_diagram.png');

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.

```
Iteration    diff
     0     2

Iteration    diff
  100.0000    0.5383

Iteration    diff
  200.0000    0.1970

Iteration    diff
  300.0000    0.0721

Iteration    diff
  400.0000    0.0264

Iteration    diff
  500.0000    0.0097

Iteration    diff
  600.0000    0.0035

Iteration    diff
  700.0000    0.0013

Iteration    diff
  800.0000    0.0005

Iteration    diff
  900.0000    0.0002

Iteration    diff
   1.0e+03 *
```

```
33
34      1.0000      0.0000
35
36  Iteration     diff
37     1.0e+03 *
38
39      1.1000      0.0000
40
41  Value function iteration succeds
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

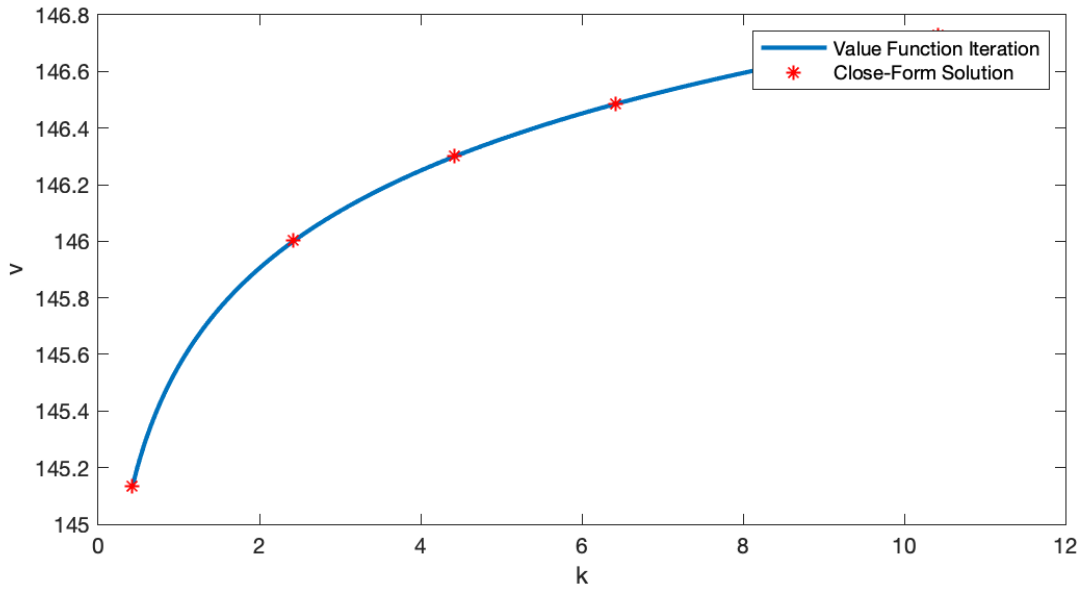
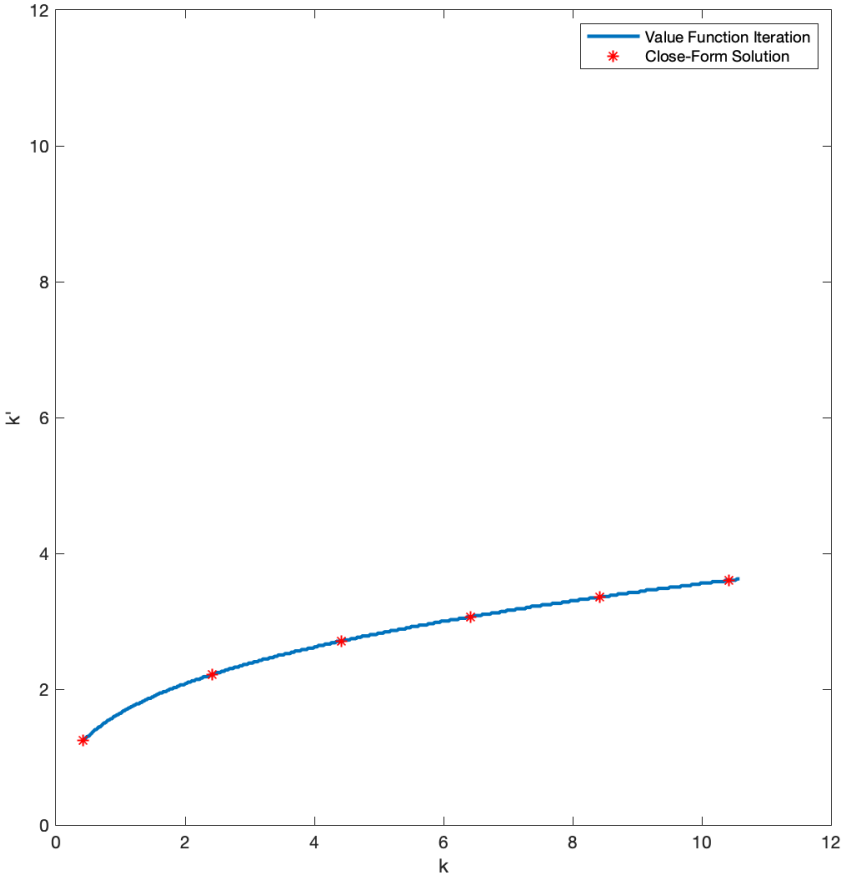


Figure 11: Policy Function



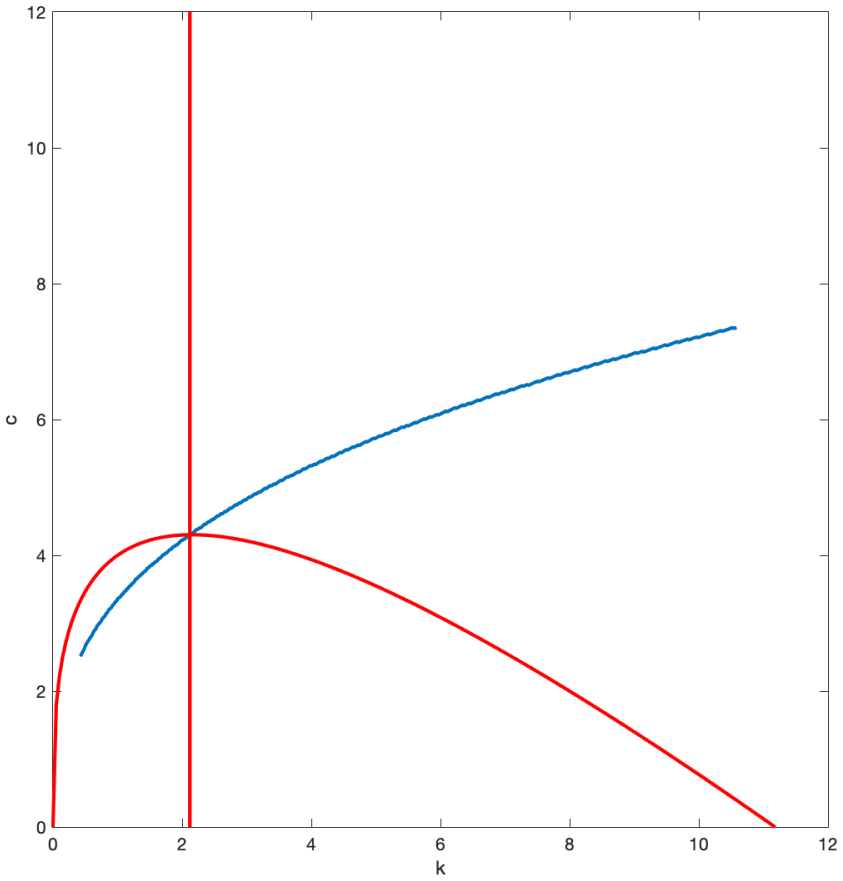
Stochastic Optimal Growth Model There are two state variables in the stochastic optimal growth model k and A.

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

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, $k = [k^1, k^2, \cdots, k^N]$ 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



$P_{ij} = Prob(A_{t+1} = A^j | A_t = A^i)$ 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 vkk matrix which records the value of the objective function is now three dimensional $N \times N \times M$, where vkk_{ijm} is assigned the value of the objective of $k = k^i, k' = k^j, A = A^m$. The transition matrix is parameterized as:

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

```
A = [4 5];
M = length(A);

vkk = zeros(N,N,M);  %Matrix vkk with vkk_{ijm} as the value of the objctive
    %each value of vkk_{ijm} is given by
    %ln(A(m)*k_{i}^alpha-k_{j})+beta*P_{m,1)*v0_{j,1}+beta*P_{m,2}*v0_{j,2}
    for m=1:M
        for i=1:N
            for j=1:N
                if A(m)*k(i)^alpha-k(j)<=1e-5
                    vkk(i,j,m) = -1e5;     %if consumption is negative, make
                else
                    vkk(i,j,m)= log(A(m)*k(i)^alpha-k(j)) ...
                    +beta*P(m,1)*v0(j,1)+ beta*P(m,2)*v0(j,2);
                end
            end
        end
    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 $k_t = \alpha\beta A k_t^{\alpha}$. 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

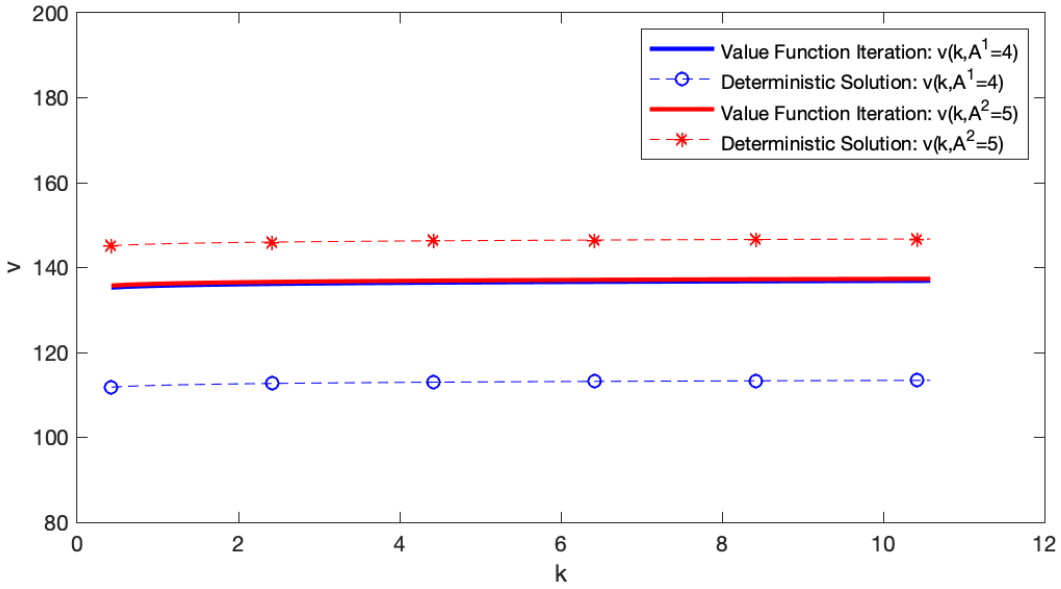
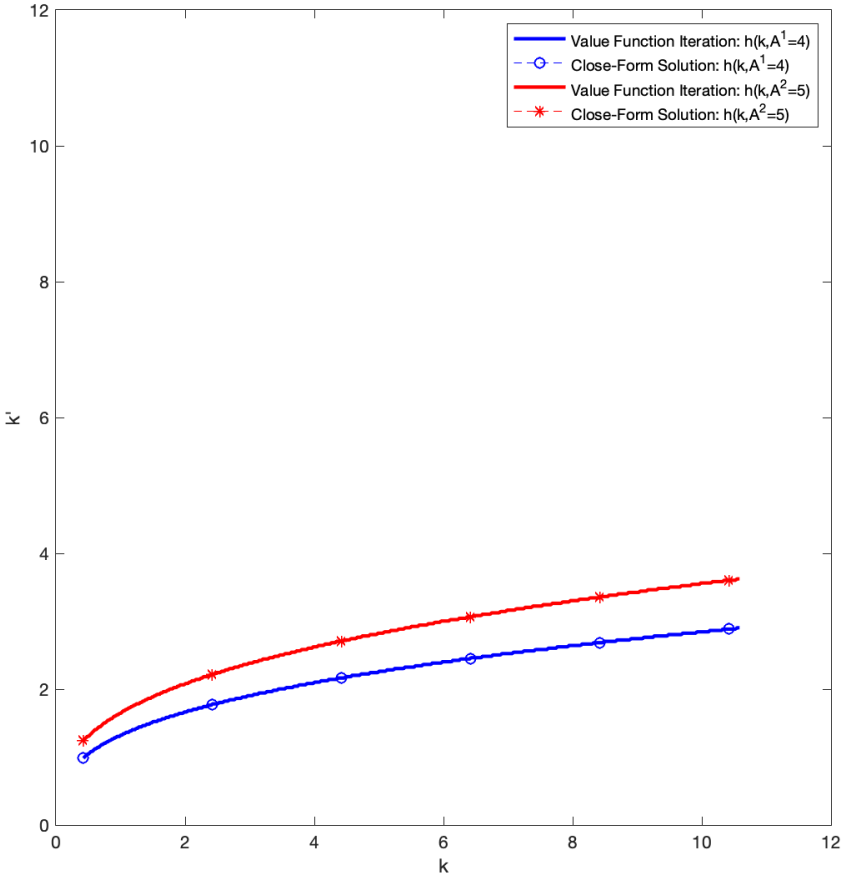


Figure 14: Policy Function



3.2 Policy Function Iteration

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

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

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

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

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

1. 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

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

Then the FE can be transformed as:

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

The value function can be calculated as:

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

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

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

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

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

Then the FE can be transformed as:

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

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:

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

```
Pkk = zeros(N,N);        %construct the transition matrix
for i=1:N
    for j=1:N
        if j==d0(i)
            Pkk(i,j)=1;     %Pkk(i,j)=1 if k'=h(k), k=k^i,k'=k^j
        end
    end
end
v0=(eye(N)-beta*Pkk)\(log(A*k.^alpha-k(d0)))';
%=========================================================================
```

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:

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

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,

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

Similarly,

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

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

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

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.

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

```
Iteration     diff
```

```
     0      2

Iteration     diff
     1     35

Iteration     diff
     2      7

Iteration     diff
     3      6

Iteration     diff
     4      5

Iteration     diff
     5      3

Iteration     diff
     6      1

Policy function iteration succeeds
Iteration     diff
     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.

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

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

$$v(k, A^1) = \max_{k'} \quad \ln(A^1 k^{\alpha} - k') + \beta P_{11} v(k', A^1) + \beta P_{12} v(k', A^2)$$
$$v(k, A^2) = \max_{k'} \quad \ln(A^2 k^{\alpha} - k') + \beta P_{21} v(k', A^1) + \beta P_{22} v(k', A^2)$$

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 $h_j(k)$.

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

The two FE are now:

$$v_j(k, A^1) = \ln(A^1 k^\alpha - h_j(k)) + \beta P_{11} Pkk_{..1} * v(k, A^1) + \beta P_{12} Pkk_{..1} * v(h, A^2)$$
$$v_j(k, A^2) = \ln(A^2 k^\alpha - h_j(k)) + \beta P_{21} Pkk_{..2} * v(k, A^1) + \beta P_{22} Pkk_{..2} * v(k, A^2)$$

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 $v_j(k, A)$.

$$\begin{bmatrix} v_j(k, A^1) \\ v_j(k, A^2) \end{bmatrix} = \left[I_{2N} - \beta \begin{pmatrix} P_{11} Pkk_{..1} & P_{12} Pkk_{..1} \\ P_{21} Pkk_{..2} & P_{22} Pkk_{..2} \end{pmatrix} \right]^{-1} \begin{bmatrix} \ln(A^1 k^\alpha - h_j(k)) \\ \ln(A^2 k^\alpha - h_j(k)) \end{bmatrix} \tag{39}$$

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.

```
%compute the value function v0 according to the policy d0=========
Pkk = zeros(N,N,M); %construct the transition matrix
for m=1:M
    for i=1:N
        for j=1:N
            if j==d0(i,m)
                Pkk(i,j,m)=1; %Pkk(i,j,m)=1 if k'=h(k,A^m), k=k^i,k'=k^j,A=
            end
        end
```

```
    end
end
objValue = [log(A(1)*k.^alpha-k(d0(:,1))),log(A(2)*k.^alpha-k(d0(:,2)))]';
PPkk = zeros(2*N,2*N);
PPkk(1:N,1:N)= P(1,1)*squeeze(Pkk(:,:,1));
PPkk(1:N,N+1:2*N)= P(1,2)*squeeze(Pkk(:,:,1));
PPkk(N+1:2*N,1:N)= P(2,1)*squeeze(Pkk(:,:,2));
PPkk(N+1:2*N,N+1:2*N)= P(2,2)*squeeze(Pkk(:,:,2));
v0Vector = (eye(M*N) - beta*PPkk)\objValue;
v0 = reshape(v0Vector,N,2);
%=========================================================================
```

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.

3.3 Projection Method

The essence of projection method is to use a collection of functions with known function form to approximate the value function or policy function with unknown function form, and thus find the solution of the dynamic problem.

The projection method usually finds solutions from the equilibrium conditions, namely the first order conditions of the dynamic programming problem. In the optimal growth model, the first order condition after substituting the resource constraint is:

$$F(c(k)) = \frac{1}{c(k)} - \beta E_t \frac{1}{c(Ak^\alpha - c(k))} \alpha A (Ak^\alpha - c(k))^{\alpha-1} = 0$$

The state variable is k and the solution is given by a policy function $c(k)$ and $k'(k) = Ak^\alpha - c(k)$. Projection method tries to approximate $c(k)$ by $r+1$ functions with known function form $(p_r^0(k), p_r^1(k), \cdots, p_r^r(k))'$ where

$$p_r(k) = \sum_{i=0}^{r} \chi_i p_r^i(k) = p_r(k)\chi$$

where $\chi = (\chi_0, \chi_1, \cdots, \chi_r)'$. And we call $F(p_r(k))$ the **residual function** in the projection method.

This comes from the Weierstrass theorem which states that any continuous function $c(\cdot)$ can be approximated uniformly by a sequence of r polynomials $p_r(k)$. One candidate collection of functions is linear combination of monomials in k.

$$p_r(k) = \sum_{i=0}^{r} \chi_i k^i$$

Numerically the monomials are not good enough since this candidate has a collinear problem as k^i and k^{i+1} are rather close to each other. Basically, the collection of orthogonal polynomials is used extensively. The definition of an orthogonal polynomial is based on the inner product between two functions f_1 and f_2, given the weighting functions w:

$$\langle f_1, f_2 \rangle = \int_{\underline{k}}^{\bar{k}} f_1(k) f_2(k) w(k) dk$$

The collection of polynomials $T_r(k)$ is defined to be mutually orthogonal with respect to weighting function $w(k)$ iff $\langle T_i(k), T_j(k) \rangle = 0$ for $i \neq j$.

Chebyshev Polynomials One such collection is the Chebyshev collection, which is defined over $s \in [-1, 1]$ and the weighting function $w(s) = (1 - s^2)^{-\frac{1}{2}}$. Given the r-th order Chebyshev polynomials for constructing the policy function $p_r(k)$.

$$p_r(k(s)) = \sum_{i=0}^{r} \chi_i T_r^i(s), \quad s = 2\frac{k - \underline{k}}{\bar{k} - \underline{k}} - 1$$

where

$$T_r^i(s) = \cos(i \arccos(s))$$

The Chebyshev polynomials are also given the the recursive form:

$$T_r^1(s) = 1, \quad T_r^2(s) = s$$
$$T_r^{i+1}(s) = 2kT_r^i(s) - T_r^{i-1}(s), i = 1, \cdots, r-1$$

Notice that the domain of Chebyshev functions locates in $[-1, 1]$. We should transform the domain of k to s by $s = 2\frac{k-\underline{k}}{\bar{k}-\underline{k}} - 1$.

Implementation The implementation of the projection method boils down to choose χ to as close to as:

$$F(p_r(k)) = 0$$

In the case of optimal growth model, the projection method is to choose χ such that

$$F(p_r(k)) = \frac{1}{p_r(k)} - \beta E_t \frac{1}{p_r(Ak^\alpha - p_r(k))} \alpha A (Ak^\alpha - p_r(k))^{\alpha-1} = 0$$

But how do we judge whether $F(p_r(k))$ is as close as zero? Alternative choices of the criterion that $F(p_r(k))$ is as close as zero stands out as another dimension that the projection methods are differentiated except different choice of polynomial functions. Returning to the use of inner products, we seek to choose χ to minimize the following inner products:

$$\langle F(p_r(k)), f_i(k) \rangle = \int_{\underline{k}}^{\bar{k}} F(p_r(k)) f_i(k) w(k) dk$$

Alternative specifications of $f_i(k)$ and $w(k)$ determines different varieties of projection methods.

Variety	$f(k)$	w(k)
Least Squares	$f_i(k) = F(p_r(k))$	$w(k) = 1$
Galerkin	$f_i(k) = p_r^i(k)$	$w(k) = 1$
Collocation	$f_i(k) = 1$	$w(k) = \begin{cases} 0 & if \quad s \neq k_i \\ 1 & if \quad k = k_i \end{cases}$

Then the algorithm of projection method to find the policy function solution $c(k)$ of the optimal growth problem is:

1. Choose a finite state space $[\underline{k}, \bar{k}]$ and a collection of functions $p_r^i(k), i = 0, 1, \cdots, r$ and let:

$$p_r(k) = \sum_0^r \chi_i p_r^i(k)$$

2. Define the residual function:

$$F(p_r(k)) = \frac{1}{p_r(k)} - \beta E_t \frac{1}{p_r(Ak^\alpha - p_r(k))} \alpha A (Ak^\alpha - p_r(k))^{\alpha - 1}$$

3. Choose a projection function $f_i(k)$ and a weighting function $w(k)$ and compute the inner product. We choose χ to minimize the inner product.

$$\min_{\chi} \quad \langle F(p_r(k)), f(k) \rangle = \int_{\underline{k}}^{\bar{k}} F(p_r(k)) f_i(k) w(k) dk, i = 0, \cdots, n$$

Choices of state grid, polynomial functions, and the objective function to minimize the distance give different variety of projection method.

Equalized Grid, Chebyshev Polynomial, Least Squares Solution We discretize the state space by equal increment in

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:

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

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:	$c_t + k_{t+1} = Ak_t^{\alpha}$
Initial Condition:	k_0 is given
Transversality Condition:	$\lim_{t\to\infty} \beta^t v_k(k_t) k_t = 0$

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. Let's linearize the system by Taylor series expansion. And let $\tilde{x} = x - \bar{x}, x = k, c.$

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

Simplify:

$$u''(\bar{c})\tilde{c}_t = \beta\alpha A u'(\bar{c})(\alpha-1)\bar{k}^{\alpha-2}\tilde{k}_{t+1} + \beta\alpha A\bar{k}^{\alpha-1}u''(\bar{c})\tilde{c}_{t+1} \tag{40}$$
$$\tilde{c}_t + \tilde{k}_{t+1} = A\alpha\bar{k}^{\alpha-1}\tilde{k}_t \tag{41}$$

Notice that all terms with order higher than one are omitted.