

- SLP Chapters 3 & 4 have most of this material.
I hope to type this up when I have time.

Zach Swizek
Econ 8105

• Intro. to Dynamic programming

- We have planner problems of the following form:

$$\begin{aligned} \max_{\{c_t, k_{t+1}\}_{t=0}^{\infty}} \quad & \sum_{t=0}^{\infty} \beta^t \ln(c_t) \quad \text{[e.g. } f(k_t) = F(k_t) + (1-\delta)k_t \text{]} \\ \text{s.t.} \quad & c_t + k_{t+1} \leq f(k_t) \quad \forall t \\ & c_t, k_{t+1} \geq 0 \\ & k_0 \text{ given} \end{aligned}$$

Rewrite as:

$$\begin{aligned} \max_{\{k_{t+1}\}_{t=0}^{\infty}} \quad & \sum_{t=0}^{\infty} \beta^t \ln(f(k_t) - k_{t+1}) \\ \text{s.t.} \quad & k_{t+1} \in \{k_{t+1} \in K \mid 0 \leq k_{t+1} \leq f(k_t)\} \\ & \quad \quad \quad \uparrow K \in \mathbb{R}_+ \end{aligned}$$

Now to adapt to SLP notation, let:

$$F(k_t, k_{t+1}) = \ln(f(k_t) - k_{t+1}) \quad \text{"return function"}$$

$$\Gamma(k_t) = \{k_{t+1} \in K \mid 0 \leq k_{t+1} \leq f(k_t)\}$$

Then we have:

$$\begin{aligned} \text{(SP)} \quad & \max_{\{k_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} F(k_t, k_{t+1}) \\ \text{"sequence problem"} \quad & \text{s.t. } k_{t+1} \in \Gamma(k_t) \quad \forall t \\ & k_0 \text{ given} \end{aligned}$$

This is referred to as the "sequence problem" (SP) in SLP.

The solution to this problem: $\{\hat{k}_{t+1}\}_{t=0}^{\infty}$ ^{hat} is an infinite sequence of quantities. Because of their dynamic nature, these problems are difficult to solve. It's very difficult (or impossible) to solve this analytically by taking FOCs and using the optimality conditions. What about using a computer to optimize? Well, the computer would be solving for an infinite-dimensional vector, which isn't feasible either (to my knowledge).

We would rather have a simpler problem to solve; one that provides the same solution but for which you do not to solve an infinite dimensional optimization problem for. For example, from the SP above, we would rather have a problem of the form:

$$(FE) \quad V(k) = \max_{k'} \{ F(k, k') + \beta V(k') \} \quad \forall k \in K$$

^{Functional Equation}

$$\text{s.t. } k' \in \Gamma(k)$$

This is called the "functional equation" (FE) or the Bellman equation. It's called a FE b/c the function V is unknown. This is the "recursive" formulation of SP.

The beauty of the Bellman equation is that its solution is a function:

$$k^* = g(k), \quad \text{the "policy function"}$$

Once we solve for $g(k)$, it is easy to construct the sequence $\{k_{t+1}^*\}_{t=0}^{\infty}$: ^{star}

$$k_1^* = g(k_0); \quad k_2^* = g(k_1^*); \quad k_3^* = g(k_2^*); \quad \dots$$

and thus we get $\{k_{t+1}^*\}_{t=0}^{\infty}$.

So we have our solution the the SP: $\{\hat{k}_{t+1}\}_{t=0}^{\infty}$
and a sequence of solutions to the FE: $\{k_{t+1}^*\}_{t=0}^{\infty}$.

We want $\hat{k}_{t+1} = k_{t+1}^* \quad \forall t \geq 0$, given any initial k_0 .

When (i) SP and FE have the same solution(s) for all initial k_0 and (ii) they have the same value at the solution(s), we call this the "Principle of Optimality."

When do the SP and FE satisfy the principle of optimality? SLP chapter 4 discusses this.

For the specific planner's problem above, we will just assume it satisfies it for now.

This means that the solution to our original problem: $\{\hat{k}_{t+1}\}_{t=0}^{\infty}$

$$\begin{aligned} \max_{\{k_{t+1}\}_{t=0}^{\infty}} \quad & \sum_{t=0}^{\infty} \beta^t \ln(f(k_t) - k_{t+1}) \\ \text{s.t.} \quad & 0 \leq k_{t+1} \leq f(k_t) \quad \forall t \\ & k_0 \text{ given} \end{aligned}$$

is equivalent to the solutions given by the recursive form:
 $\{k_{t+1}^*\}_{t=0}^{\infty} = \{g(k_0), g(g(k_0)), g(g(g(k_0))), \dots\} = \{k_1^*, k_2^*, k_3^*, \dots\}$

$$\begin{aligned} V(k) = \max_{k'} \quad & \{ \ln(f(k) - k') + \beta V(k') \} \\ \text{s.t.} \quad & k' \in \Gamma(k) \end{aligned}$$

for any given initial capital stock k_0 .

"sequence problem"

To provide some intuition as to why the SP can be written in the FE form, let's suppose $\{\hat{k}_{t+1}\}_{t=0}^{\infty}$ is the solution to the SP:

$$\begin{aligned} W(k_0) &= \max_{\{k_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t F(k_t, k_{t+1}) \\ &\quad \text{s.t. } k_{t+1} \in \Gamma(k_t) \quad \forall t, k_0 \text{ given} \\ &= \sum_{t=0}^{\infty} \beta^t F(\hat{k}_t, \hat{k}_{t+1}) \\ &= F(k_0, \hat{k}_1) + \sum_{t=1}^{\infty} \beta^t F(\hat{k}_t, \hat{k}_{t+1}) \\ &= F(k_0, \hat{k}_1) + \beta \sum_{t=0}^{\infty} \beta^t F(\hat{k}_{t+1}, \hat{k}_{t+2}) \\ &= F(k_0, \hat{k}_1) + \beta W(\hat{k}_1) \end{aligned}$$

So at the optimum, $W(k_0)$ (the planner's problem at time zero) is equal to the sum of:

- 1) The optimal $t=0$ period utility $F(k_0, \hat{k}_1) = u(f(k_0) - \hat{k}_1)$, and
- 2) The discounted value of solving the planner's problem at period 1, given initial capital $\hat{k}_1$: $\beta W(\hat{k}_1)$.

Which looks a lot like the maximum of FE given k_0 , where $g(k_0) = k_1^*$ (and since we assume principle of optimality holds, we know $\hat{k}_1 = k_1^*$)

$$\begin{aligned} V(k_0) &= \max_{k_1 \in \Gamma(k_0)} \{ F(k_0, k_1) + \beta V(k_1) \} \\ &= F(k_0, k_1^*) + \beta V(k_1^*) \end{aligned}$$

How do we solve for V in the FE?

Now, go back to our Bellman equation:

$$V(k) = \max_{k'} \{ u(f(k) - k') + \beta V(k') \}$$

s.t. $k' \in \Gamma(k)$

V is an unknown function. We want to find its true form or find an approximation that is very close to V .

The following steps describe how we solve for V (or close approx.):

Define the operator T :

$$T(w)(k) = \max_{k' \in \Gamma(k)} \{ u(f(k) - k') + \beta w(k') \}$$

Although "operator" may sound technical, don't worry. T is just a function whose input is a function and whose output is another function:

$$T: S \rightarrow S, \quad \text{where } S \text{ is a set of functions satisfying some properties}$$

For example, S could equal the set $C(K)$ of continuous + bounded functions, $f: K \rightarrow \mathbb{R}$, where $K \subseteq \mathbb{R}_+$ s.t. $k \in K$

Then we say $S \equiv C(K)$, and $T: C(K) \rightarrow C(K)$

Why do we need this operator T ?

We want to show that it is a contraction, which I define below. We eventually want to show that by applying the operator to a "guess" function many times, we will get a very good approximation of the unknown V function.

Definition: Contraction: Let (S, ρ) be a metric space and $T: S \rightarrow S$ be a function mapping S into S .
 T is a contraction mapping w/ modulus β if

$$\rho(T(v), T(w)) \leq \beta \rho(v, w)$$

$$\forall v, w \in S \text{ and } \beta \in (0, 1) \quad \blacksquare$$

What does $\rho(T(v), T(w)) < \beta \rho(v, w)$ mean?

It means that $T(v)$ and $T(w)$ are closer together than v and w .

Lets keep apply T :

$$\begin{aligned} \rho(T(T(v)), T(T(w))) &< \beta \rho(T(w), T(v)) \\ &< \beta (\beta \rho(v, w)) = \beta^2 \rho(v, w) \end{aligned}$$

If you keep applying T , say n times, you get:

$$\Rightarrow \rho(T^n(v), T^n(w)) < \beta^n \rho(v, w)$$

where we define $\underbrace{T(T(T \dots (T(v))))}_{\text{Applying } T \text{ } n \text{ times}} = T^n(v)$

And remember $\beta \in (0, 1) \Rightarrow \beta^n \rightarrow 0$ as $n \rightarrow \infty$;
 since $\rho(v, w)$ is finite by assumption, $\beta^n \rho(v, w) \rightarrow 0$ as $n \rightarrow \infty$.

Thus, applying T $n \rightarrow \infty$ times implies

$$\rho(T^\infty(v), T^\infty(w)) \rightarrow 0.$$

I.e. the distance between $T^n(v)$ and $T^n(w)$ goes to zero as n goes to ∞ .

Now that we understand contractions, we can state the Contraction mapping theorem (CMT)

Definition: CMT: If (S, ρ) is a complete metric space and $T: S \rightarrow S$ is a contraction with modulus β , then:

Every Cauchy sequence in S converges to element in S

i.e. unique

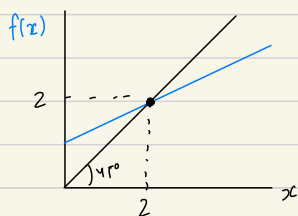
i) T has exactly one fixed point v in S

ii) For any initial $v_0 \in S$, $\rho(T^n(v_0), v) \leq \beta^n \rho(v_0, v)$

A couple of quick points on this definition:

↳ i) Fixed point: $T(v) = v$

Q: What is fixed point ($x^* = f(x^*)$) of $f(x) = 1 + \frac{1}{2}x$?



We can find the fixed point(s) of a function in 1 dimension by checking where it crosses the 45° line

A: $f(x^*) = x^*$ when $x^* = 2$

↳ ii) Why didn't we write (ii) as:

$$\rho(T^n(v_0), T^n(v)) \leq \beta^n \rho(v_0, v) \quad ?$$

Since v is fixed point, $T(v) = v$, then $T^n(v) = v$.
E.g. if $n=2$: $T^2(v) = T(\underbrace{T(v)}_v) = T(v) = v$

The CMT tells us that if (S, ρ) is a complete metric space and T is a contraction w/ $\beta \in (0, 1)$, then for any initial function v_0 , applying the operator T to v_0 many times will cause the distance between $T^n(v_0)$ and v to go to zero:

$$\rho(T^n(v_0), v) \rightarrow 0 \text{ as } n \rightarrow \infty \text{ where } T(v) = v.$$

And remember our operator T :

$$T(w)(k) = \max_{k' \in \Gamma(k)} \{ u(f(k) - k') + \beta w(k') \}$$

We want the fixed point $T(V) = V$ so we can write the Bellman equation:

$$T(V)(k) = V(k) = \max_{k' \in \Gamma(k)} \{ u(f(k) - k') + \beta V(k') \}$$

Oftentimes we cannot solve for V exactly, but if T is a contraction, we can apply T repeatedly (N times) so that $\rho(T^N(v_0), V)$ is very small for any v_0 . Thus we can use $T^N(v_0)$ as a good approximation of V .

How do we know if T is a contraction?

We can check Blackwell's sufficient conditions for a contraction.

Before we define Blackwell's sufficient conditions, let's define our metric (how we measure distance), the sup norm:

$$\rho(V, W) \equiv \sup_{k \in K} |V(k) - W(k)|$$

The sup norm tells us the maximum distance between two functions for any given input value.

• Def. Blackwell's sufficient conditions for a contraction:

Let $K \subseteq \mathbb{R}_+$, and let $C(K)$ be a space of bounded functions $v: K \rightarrow \mathbb{R}$, with the sup norm.

Let $T: C(K) \rightarrow C(K)$ be an operator satisfying:

i) Monotonicity: If $v, w \in C(K)$ satisfy $v(k) \leq w(k) \quad \forall k \in K$, then: $(Tv)(k) \leq (Tw)(k) \quad \forall k \in K$

ii) Discounting: $\exists$ some $\beta \in (0, 1)$ such that $[T(v+a)](k) \leq (Tv)(k) + \beta a \quad \forall v \in C(K), a \geq 0, k \in K$

Then T is a contraction with modulus β . ■

[Note: $(v+a)(k) = v(k) + a$]

Other than the boundedness condition*, it is straightforward to show our operator above satisfies Blackwell's conditions and thus is a contraction. We will do this later in the course.

Therefore, we can apply the CMT. Solving for $V = T(V)$ (or more correctly, a very good approximation of V) is fairly simple to implement on a computer.

Professor Kehue even did it in Excel!

This process is called value function iteration (VFI).

On the next page, I talk about VFI.

* When we have $u(\cdot) = \log(\cdot)$, we usually just ignore the boundedness condition. To be clear, we will always bound $\log(\cdot)$ from above, and just ignore that it is not bounded below. But to truly satisfy Blackwell's, we need $u(\cdot)$ to be bounded.

Value Function Iteration

(Adapted from Amol Amol's notes)

Suppose we have verified that our operator T :

$$T(w)(k) = \max_{k' \in \Gamma(k)} \{ \ln(f(k, k')) + \beta w(k') \}$$

satisfies Blackwell's sufficient conditions^{*} and thus is a contraction with $\beta \in (0, 1)$. Then by the CMT,
(i) T has a unique fixed point $T(V) = V$ and
(ii) For any initial function v_0 , $T^n(v_0)$ converges to V as n goes to infinity (i.e. $\{T(v_0), T^2(v_0), \dots\} \rightarrow V$).

So let's use the CMT; we can create the sequence $T(v_0), T^2(v_0), T^3(v_0), \dots$ which we know converges to the fixed point V . This is the process of value function iteration:

1) Make an initial guess; a function $v_0(k)$.

↳ We guess $v_0(k) = 0$. Remember, we can pick any initial v_0 as long as it is in $C(K)$. Since $v_0(k) = 0$ is continuous and bounded we are free to choose it.

[Continued next page]

^{*} Again, other than boundedness from below, ignore this for now.

2) Solve $T(v_0)(k) = v_1(k)$.

Plug in $v_0(k') = 0$:

$$\hookrightarrow T(v_0)(k) = v_1(k) = \max_{k'} \left\{ \ln(Ak^\alpha - k') + \beta \underbrace{v_0(k')}_{=0} \right\}$$

s.t. $0 \leq k' \leq Ak^\alpha$

KKT FOCs: $\frac{1}{Ak^\alpha - k'} = \gamma$; $\gamma k' = 0$; $\gamma \geq 0$; $k' \geq 0$

$$\Rightarrow \frac{k'}{Ak^\alpha - k'} = \gamma k' = 0$$

$\Rightarrow k'^* = g_1(k) = 0$ is optimal policy.

Plug $g_1(k) = k'^* = 0$ into $T(v_0)(k) = v_1(k)$:

$$\begin{aligned} v_1(k) &= \ln(Ak^\alpha - \underbrace{g_1(k)}_{=0}) \\ &= \ln(Ak^\alpha) \end{aligned}$$

We have solved $v_1(k) = \ln(Ak^\alpha)$.

3) Solve $T^2(v_0)(k) = T(v_1)(k) = v_2(k)$

Plug this in from above

$$\hookrightarrow T(v_1)(k) = v_2(k) = \max_{k'} \left\{ \ln(Ak^\alpha - k') + \beta \underbrace{v_1(k')}_{\text{from above}} \right\}$$

s.t. $0 \leq k' \leq Ak^\alpha$

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

s.t. $0 \leq k' \leq Ak^\alpha$

FOC: $\frac{1}{Ak^\alpha - k'} = \frac{\beta A \alpha k'^{\alpha-1}}{Ak'^\alpha}$

Which you can solve for the optimal policy:

$$\Rightarrow k^* = g_2(k) = \frac{\alpha\beta}{1+\alpha\beta} A k^\alpha$$

Plug this into $T(v_1)(k) = v_2(k)$:

$$\begin{aligned} v_2(k) &= \ln(Ak^\alpha - g_2(k)) + \beta \ln(Ag_2(k)^\alpha) \\ &= \ln\left(Ak^\alpha - \frac{\alpha\beta}{1+\alpha\beta} Ak^\alpha\right) + \beta \ln\left(A\left(\frac{\alpha\beta}{1+\alpha\beta} k^\alpha\right)^\alpha\right) \end{aligned}$$

Thus we have solved $v_2(k)$.

4) Solve $T(v_2)(k) = v_3(k)$

$\vdots$

n) Solve $T(v_{n-2})(k) = v_{n-1}(k)$

$n+1$) Solve $T(v_{n-1})(k) = v_n(k)$

$\vdots$

And so on.

Using VFI, we construct the sequence of functions:

$$\begin{array}{ccccccc} v_0, & T(v_0), & T(T(v_0)), & T(T(T(v_0))), & \dots & T^n(v_0), & \dots \\ v_0, & v_1, & v_2, & v_3, & \dots, & v_n, & \dots \end{array}$$

which we know will converge to fixed point $V = T(V)$.

We can implement VFI on the computer, and although we do not iterate an infinite amount of times, we can get a very close approximation of V using $T^n(v_0)$, with N high enough.

Appendix C : Useful for understanding SLP notation.

State vs. Control Variables

Consider planner's problem with $f(k_t) = A k_t^\alpha$ (A and α given and fixed)

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

$$\begin{aligned} \text{s.t.} \quad & c_t + k_{t+1} \leq A k_t^\alpha \quad \forall t \\ & c_t, k_{t+1} \geq 0 \quad \forall t \\ & k_0 \text{ given} \end{aligned}$$

In each period t , the planner must pick c_t and k_{t+1} . These are the control variables. That is, in period t , the planner "controls" these variables.

What other variables are important to the planner at time τ ? Look at period τ feasibility:

$$\underbrace{c_\tau}_{\text{control}} + \underbrace{k_{\tau+1}}_{\text{control}} \leq A \underbrace{k_\tau}_{?}$$

The planner doesn't "control" k_τ at period τ ; it is already determined*. Additionally, it gives very valuable information to the planner. It tells the planner how much c_τ and $k_{\tau+1}$ can be chosen; it determines the set of feasible choices at τ . We call k_τ a "state" variable; it describes the current "state" of the planning problem.

* The planner chose k_τ in period $\tau-1$.

Control variables: Chosen by planner at τ

State variables: i) Are predetermined at τ ; in a sense, they are "given" at time τ
ii) Provide information to the decisionmaking process of the planner at period τ .

Question: Consider the planner's problem at period τ from above. Suppose we add $k_{\tau-1}$ as a state variable.

Why is this unnecessary?

Check the two conditions on state variables from above:

i) Are $k_{\tau-1}$ and k_{τ} predetermined at time τ ? Yes. This is not an issue.

ii) Do they jointly provide information to the decisionmaking process?

No! We do not need to know $k_{\tau-1}$ at τ to choose c_{τ} and $k_{\tau+1}$; we only need to know k_{τ} .

Distinguishing between state and control variables is important in dynamic programming.

Consider the Bellman equation for the problem above:

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

$$\text{s.t. } c + k' \leq f(k)$$

$$c, k' \geq 0$$

We can immediately tell that k is a state variable; the input (what is "given") for V is k ($V: \mathbb{R} \rightarrow \mathbb{R}$).

In general, the Bellman equation will be a function of the state variable(s).

Thus, c and k' are control variables; the planner chooses ("controls") c and k' to maximize utility. This is directly related to the SLP notation.

The SLP notation for the functional equation (FE) is:

$$V(x) = \sup_{y \in F(x)} \{ F(x, y) + \beta V(y) \} \quad \forall x \in X$$

Note that x represents the state variable(s). y represents current period control variable(s) that are next period's state variable(s).

For our planner's problem above we have:

$$\begin{aligned} F(x, y) &= u(f(k) - k') \\ x &= k \\ y &= k' \end{aligned}$$

Note that this notation requires that we find a way to "hide" the other control variables (e.g. c). We know that c is also a control variable to the planner, and since it is not a next-period state variable, we want to find a way to write the problem such that the choosing of c is "hidden." We set:

$$\begin{aligned} \underbrace{v(k_t, k_{t+1})}_{\text{in PS3}} &= F(k, k') = \max_c u(c) \\ &\quad \text{s.t. } c + k' \leq f(k), \quad c \geq 0 \\ &= u(f(k) - k') \end{aligned}$$

By setting $F(k, k')$ this way, we guarantee that we are choosing the optimal value of c for each pair (k, k') , but without it being an explicit choice variable. This allows us to focus on the variables that make the problem dynamic and is generalizable.

Thus we can write the Bellman solely in terms of today's state variable (k) and tomorrow's state variable k' (k' is also today's control variable):

$$V(k) = \max_{\substack{\text{today's} \\ \text{state}}} \left\{ F(k, k') + \beta V(k') \right\}$$

today's control s.t. $k' \in \Gamma(k) = \{k' \in \mathbb{R}_+ \mid 0 \leq k' \leq f(k)\}$

tomorrow's state

To be clear, this is exactly the same problem as the FE from earlier;

$$V(k) = \max_{c, k'} \left\{ u(c) + \beta V(k') \right\}$$

s.t. $c + k' \leq f(k)$
 $c, k' \geq 0$

it is just written in a slightly different form.

Also note that $\Gamma(k)$ is a function of current state variable k ; this makes sense as the planner needs to observe k to know which values of k' are feasible. As I said earlier, state variables provide important information to the decisionmaking process! Thus, the set of feasible k' is a function of state variable k .