

# The Growth Model: Discrete Time Dynamic Programming

Prof. Lutz Hendricks

Econ720

September 12, 2017

# The Issue

We solved the growth model in **sequence language**.

- ▶ the solution is a sequence of objects that satisfies a bunch of difference equations

An alternative: **recursive** formulation.

- ▶ dynamic programming

# Dynamic Programming: An Informal Introduction

The basic idea of DP is to transform a many period optimization problem into a static problem.

To do so, we summarize the entire future by a **value function**.

The value function

- ▶ tells us the maximum utility obtainable from tomorrow onwards for any value of the state variables.
- ▶ an indirect utility function

# Dynamic Programming: An Informal Introduction

Suppose we solve the planner's problem with starting date  $t^*$ :

$$\max \sum_{t=t^*}^{\infty} \beta^{t-t^*} u(c_t) \quad (1)$$

subject to  $k_{t+1} = f(k_t) - c_t \forall t$ .

Call the optimal solution  $c_t^*, k_t^*$  and the implied lifetime utility  $V$ .

## Value function

Claim: The only fact that we need to know to figure out  $V$  is  $k_{t^*}$

- ▶ other past choices do not show up in preferences or constraints
- ▶  $k_{t^*}$  is the **state variable** of the problem.

Therefore, we can define the **value function** (indirect utility function)

$$V(k_{t^*}) = \sum_{t=t^*}^{\infty} \beta^{t-t^*} u(c_{t^*}) \quad (2)$$

$$= \max \sum_{t=t^*}^{\infty} \beta^{t-t^*} u(f(k_t) - k_{t+1}) \quad (3)$$

# Time consistency

- ▶ What if we start the problem at  $t^* + 1$ ?
- ▶ Would the planner want to change his optimal choices of  $k_{t^*+2}, k_{t^*+3}$ , and so on?
- ▶ The answer is obviously “no,” ... although I won't prove this just yet.
- ▶ A problem with this property is known as **time consistent**:
  - ▶ Give the decision maker a choice to change his mind at a later date and he will choose the same actions again.
- ▶ Not all optimization problems have this property.
  - ▶ For example, changing the specification of discounting easily destroys time consistency (self-control problems arise).

# Stationarity

- ▶ Compare the value functions obtained from the problems starting at  $t^*$  and at  $t^* + 1$ .
- ▶ It is obvious that the function  $V(k_{t^*})$  does not depend on  $t^*$ .
- ▶ That is, solving the problem yields the same value function regardless of the starting date.
- ▶ Such a problem is called **stationary**.
- ▶ Not all optimization problems have this property.
  - ▶ For example, if the world ended at some finite date, then the problem at  $t^*+1$  looks different from the problem at  $t^*$ .

## Recursive structure

Now comes the key insight:

$$V(k_{t^*}) = u(c_{t^*}^*) + \beta \left[ \sum_{t=t^*+1}^{\infty} \beta^{t-(t^*+1)} u(c_{t^*}^*) \right] \quad (4)$$

$$= u(c_{t^*}^*) + \beta V(k_{t^*+1}^*) \quad (5)$$

$$= \max_{k_{t^*+1}} u(f(k_{t^*}^*) - k_{t^*+1}^*) + \beta V(k_{t^*+1}^*) \quad (6)$$

We have

- ▶ one term reflecting current period utility
- a second term summarizing everything that happens in the future, given optimal behavior, as a function of  $k_{t^*+1}^*$ .



## Recursive structure

Since this equation holds for any arbitrary start date, we may drop date subscripts.

This yields a **Bellman equation**:

$$V(k) = \max u(f(k) - k') + \beta V(k')$$

where the primes denote values in the next period.

Claim: **Solving the DP is equivalent to solving the original problem** (the Lagrangian).

- ▶ We will see conditions when this is true later.

## Recursive structure

The convenient part of this is: we have transformed a multiperiod optimization problem into a two period (almost static) one.

If we knew the value function, solving this problem would be trivial.

The bad news is that we have transformed an algebraic equation into a **functional equation**.

The solution of the problem is a value function  $V$  and an optimal policy function

$$c = \phi(k)$$

Note that  $c$  cannot depend on anything other than  $k$ , in particular not on  $k$ 's at other dates, because these don't appear in the Bellman equation.

## Solution

A solution to the planner's problem is now a pair of functions

$$[V(k), \phi(k)]$$

that solve the Bellman equation in the following sense.

1. Given  $V(k)$ , setting  $c = \phi(k)$  solves the max part of the Bellman equation.
2. Given that  $c = \phi(k)$ , the value function solves

$$V(k) = u(\phi(k)) + \beta V(f(k) - \phi(k))$$

## Solution: Intuition

Given  $V(k)$ , setting  $c = \phi(k)$  solves the max part of the Bellman equation.

This means:

Point by point, for each  $k$ :

$$\phi(k) = \arg \max_c u(c) + \beta V(f(k) - c) \quad (7)$$

$\phi(k)$  simply collects all the optimal  $c$ 's – one for each  $k$ .

## Solution: Intuition

$$V(k) = u(\phi(k)) + \beta V(f(k) - \phi(k))$$

Note that this uses the optimal policy function for  $c$ .

Think of the Bellman equation as a mapping in a function space:

$$V^{n+1} = T(V^n) = \max u(c) + \beta V^n(f(k) - c)$$

Given an input argument  $V^n$  the mapping produces an output arguments  $V^{n+1}$ .

The solution to the Bellman equation is the  $V$  that satisfies  $V = T(V)$ .

- ▶ a fixed point.

# The Planner's Problem with DP

The Planner's Bellman equation is

$$V(k) = \max u(c) + \beta V(f(k) - c)$$

with state  $k$  and control  $c$ .

The FOC for  $c$  is

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

Problem: we do not know  $V'$ .

## The Planner's Problem with DP

Differentiate the Bellman equation to obtain the envelope condition

(aka Benveniste-Scheinkman equation):

$$V'(k) = \beta V'(k')f'(k)$$

Now we can use the FOC to substitute out  $V'$  twice:

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

We obtain the same Euler equation as from the Lagrangian approach.

DP also tells us that the optimal  $c$  is a function only of  $k$ .

Therefore  $k'$  also depends only on  $k$ :

$$\begin{aligned}k' &= f(k) - \phi(k) \\ &= h(k)\end{aligned}$$

## Capital as control variable

There are other ways of setting up the Bellman equation.

With capital as the control:

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

FOC:

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

Envelope condition

$$V'(k) = u'(c)f'(k)$$

The general point: We cannot choose the state variables, but we can choose the control variables.



## Characterizing the Planner's Solution

It is here where DP has serious advantages over the Lagrangean: one can use results from **functional analysis** to establish properties of the value function and the policy function.

In our example it can be shown that the economy converges monotonically from any  $k_0$  to the steady state [Sargent (2009), p. 25, fn. 2]:

Note the difference relative to the OLG economy where much stronger assumptions are needed for this result.

# Nonstationary Dynamic Programming

What if time matters?

Case 1: Time matters because of an **aggregate state variable**.

- ▶ Example:  $f(k_t, A_t)$  where  $A_{t+1} = G(A_t)$ .
- ▶ Solution: Add  $A_t$  as a state variable to the value function.

Case 2: **Finite horizon** problems.

- ▶ Example: the household lives until date  $T$ .
- ▶ Solution: Add  $t$  as a state variable to the value function.

## Additional Constraints

Constraints are treated as in any optimization problem.

**Example:**

$\max \sum_{t=0}^{\infty} \beta^t u(c_t)$  subject to

▶  $k' = f(k) - c$

▶  $k' \geq 0$

Bellman equation:

$$V(k) = \max_{c, k'} u(c) + \beta V(k') + \lambda (f(k) - c - k') + \mu k' \quad (8)$$

First-order conditions: Kuhn Tucker for  $k'$ .

# Backward Induction

For the finite horizon problem: solve it backwards, starting with the last period.

**Example:**

$$\max \sum_{t=1}^T u(c_t) \quad (9)$$

subject to  $k_{t+1} = Rk_t - c_t$  and  $k_{T+1} \geq 0$ .

Bellman:  $V(k, t) = \max u(Rk - k') + \beta V(k', t + 1)$

Terminal value:  $V(k, T) = u(Rk)$

For  $T - 1$ :  $V(k, T - 1) = \max u(Rk - k') + \beta u(Rk')$

## Backward Induction

This is mainly useful for numerically solving the problem.  
Sometimes, one can solve finite horizon problems analytically  
(see Huggett et al. (2006) for an example).

Example: Non-separable Utility

## Example: Non-separable Utility

Consider the following growth economy, modified to include **habit persistence** in consumption.

The social planner solves

$$\max \sum_{t=0}^{\infty} \beta^t u(c_t, c_{t-1})$$

subject to the feasibility constraints

$$k_{t+1} + c_t = f(k_t) + (1 - \delta)k_t \quad (10)$$

$f$  satisfies Inada conditions.

Compute and interpret the first-order necessary conditions for the planner's problem.

## Sequential Solution

This problem does not fit the DP approach without some modification.

We first solve it using a Lagrangian:

$$\Gamma = \sum_{t=1}^{\infty} \beta^t u(f(k_t) - x_t, f(k_{t-1}) - x_{t-1}) \\ + \sum_{t=1}^{\infty} \lambda_t (x_t + (1 - \delta)k_t - k_{t+1})$$

First order conditions:

$$\beta^t u_1(t, t-1) + \beta^{t+1} u_2(t+1, t) = \lambda_t \tag{11}$$
$$f'(k_t) (\beta^t u_1(t, t-1) + \beta^{t+1} u_2(t+1, t)) = \lambda_t (1 - \delta) - \lambda_{t-1}$$



## Sequential Solution

Euler equation:

$$\lambda_{t-1} = \lambda_t [1 - \delta + f'(k_t)]$$

Define the total marginal utility of consumption as

$$U'(c_{t-1}) = \beta^{t-1} u_1(t-1, t-2) + \beta^t u_2(t, t-1)$$

The Euler Equation then becomes:

$$U'(c_{t-1}) = U'(c_t) (f'(k_t) + 1 - \delta) \quad (12)$$

## Interpretation

$$U'(c_{t-1}) = U'(c_t) (f'(k_t) + 1 - \delta) \quad (13)$$

- ▶ Give up one unit of  $c_{t-1}$ . This costs  $U'(c_{t-1})$ .
- ▶ We can increase  $x_{t-1}$  by 1 and raise  $k_t$  by 1.
- ▶ We eat the results next period at marginal utility  $U'(c_t)$ .
- ▶ We can eat
  - ▶ the additional output  $f'(k_t)$ ;
  - ▶ the undepreciated capital  $1 - \delta$ ;

# Sequential Solution

A **solution** of the hh problem is:

Sequences  $\{x_t, k_t\}$  that satisfy

1. the EE
2. the flow budget constraint.
3. The boundary conditions  $k_1$  given and a TVC:

$$\lim_{t \rightarrow \infty} U'(c_t)k_t = 0$$

## DP Solution

For DP to work, it must be possible to write the problem as

$$V(s) = \max u(s, c) + \beta V(s')$$

where  $s$  is the state and  $c$  is the control.

The current problem does not fit that pattern:

$$V(k) = \max u(c, c_{-1}) + \beta V(k')$$

subject to the law of motion

$$k' = f(k) + (1 - \delta)k - c$$

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

Nonseparable utility is the problem.

## Adding a State Variable

The solution is to define an additional state variable

$$z = c_{-1}$$

Then the Bellman equation is

$$V(k, z) = \max_x u(f(k) - x, z) \\ + \beta V(x + (1 - \delta)k, f(k) - x)$$

FOC

$$u_1(c, z) = \beta V_k(k', z') - \beta V_z(k', z')$$

## Adding a state variable

The envelope conditions are

$$\begin{aligned}V_z &= u_2(c, z) \\V_k &= u_1(c, z)f'(k) + \beta V_k(\cdot)(1 - \delta) \\&\quad + \beta V_z(\cdot)f'(k)\end{aligned}$$

Now define

$$U'(c) = u_1(c, z) + \beta u_2(c', z')$$

Then substitute out the  $V_z$  terms:

$$\begin{aligned}U'(c) &= \beta V_k(\cdot) \\V_k &= U'(c)f'(k) + (1 - \delta)\beta V_k(\cdot)\end{aligned}$$

Substitute out the  $V_k$  terms and we get the same EE as with the Lagrangean.

Guess and Verify

## Guess and Verify

- ▶ In very special cases it is possible to solve for the value function in closed form.
- ▶ A common case is
  - ▶ log utility,  $u(c) = \ln(c)$ , and
  - ▶ Cobb-Douglas technology with full depreciation:  $f(k) = Ak^\theta$ .
- ▶ Then we can use the “guess and verify” method.



## Guess and Verify

The general approach is:

1. Guess a functional form for  $V$ . Stick this into the right-hand-side of the Bellman equation.
2. Solve the max problem given the guess for  $V$ . The result is on the left hand side a new value function,  $V^1$ .
3. If  $V = V^1$  the guess was correct.

## Guess and Verify: Example

Consider the growth model with log utility and Cobb-Douglas production / full depreciation.

The planner solves:

$$\begin{aligned} \max \quad & \sum_{t=0}^{\infty} \beta^t \ln(c_t) \\ \text{s.t.} \quad & k_{t+1} = A k_t^\theta - c_t \end{aligned}$$

## Guess

Guess

$$V(k) = E + F \ln(k)$$

This is inspired by the hope that  $V$  should inherit the form of  $u$ .

Having capital stock  $k$  amounts to having output  $Ak^\theta$ , which would suggest

$$\begin{aligned} V(k) &\cong \ln(Ak^\theta) \\ &= \ln(A) + \theta \ln(k) \end{aligned}$$

Note that the guess for  $V$  contains some unknown constants  $(E, F)$  which we determine as we go along.

## First-order Conditions

FOC:

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

or

$$1/c = \beta F/k' \tag{14}$$

Envelope condition

$$V'(k) = u'(c)f'(k)$$

or

$$F/k = f'(k)/c \tag{15}$$

## Policy Function

We can use the FOC to obtain the policy function in terms of the unknown parameters.

$$F_c = k' / \beta = f'(k)k \quad (16)$$

Note that

$$f'(k)k = \theta f(k) \quad (17)$$

Here, we are lucky and the  $F$  drops out

$$k' = h(k) = \beta \theta f(k) \quad (18)$$

$$c = (1 - \beta \theta) f(k) \quad (19)$$

Result (as expected): the saving rate is constant.

## Recover $E$ and $F$

Now we need to recover  $E$  and  $F$  (and make sure they are indeed constants)

Substitute everything we know into the Bellman equation:

$$E + F \ln(k) = \ln((1 - \beta\theta)f(k)) + \beta \{E + F \ln(\beta\theta f(k))\} \quad (20)$$

Note that  $\ln(f(k)) = \ln(A) + \theta \ln(k)$ .

Collect all the terms that involve  $k$  to solve for  $F$

$$F \ln(k) = \theta \ln(k) + \beta F \theta \ln(k) \quad (21)$$

implies

$$F = \frac{\theta}{1 - \theta\beta} \quad (22)$$

Collect all the constant terms to solve for  $E$

$$E = \ln(1 - \theta\beta) + \ln(A) + \beta E + \beta F \ln(\theta\beta) + \beta F A \quad (23)$$

## Summary: Guess and Verify

1. Guess a value function (including unknown parameters).
2. Write first-order and Envelope conditions using the guess.
3. Solve for policy function.
4. Substitute policy function into Bellman equation to recover unknown parameters (and check the guess).

# Applications

Examples where guess + verify is used:

Huggett et al. (2006), Huggett et al. (2011), Manuelli and Seshadri (2014)

(all models of human capital accumulation)



# DP vs Lagrangian

What does DP buy us compared with a Lagrangian?

- ▶ With **uncertainty**, DP tends to be more convenient than a Lagrangian.
- ▶ Results from functional analysis can often be used to find **properties** of the optimal policy function such as monotonicity, continuity, and existence.
- ▶ DP can have **computational** advantages. There are methods for numerically approximating policy functions.

# Reading

- ▶ Acemoglu (2009), ch. 6. Also ch. 5 for background material we will discuss in detail later on.
- ▶ Ljungqvist and Sargent (2004), ch. 3 (Dynamic Programming), ch. 7 (Recursive CE).
- ▶ Stokey et al. (1989), ch. 1 is a nice introduction.

## References I

- Acemoglu, D. (2009): *Introduction to modern economic growth*, MIT Press.
- Huggett, M., G. Ventura, and A. Yaron (2006): “Human Capital and Earnings Distribution Dynamics,” *Journal of Monetary Economics*, 53, 265–290.
- (2011): “Sources of Lifetime Inequality,” *American Economic Review*, 101, 2923–54.
- Ljungqvist, L. and T. J. Sargent (2004): *Recursive macroeconomic theory*, 2nd ed.
- Manuelli, R. E. and A. Seshadri (2014): “Human Capital and the Wealth of Nations,” *The American Economic Review*, 104, 2736–2762.
- Sargent, T. J. (2009): *Dynamic macroeconomic theory*, Harvard University Press.
- Stokey, N., R. Lucas, and E. C. Prescott (1989): “Recursive Methods in Economic Dynamics,” .