The Growth Model: Discrete Time Dynamic Programming

Prof. Lutz Hendricks

Econ720

September 14, 2016

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^*:

$$
\begin{aligned}
V(k_{t^*}) &= \max \sum_{t=t^*}^{\infty} \beta^{t-t^*} \ u(c_t) \\
& \quad + \sum\nolimits_{t=t^*}^{\infty} \lambda_t [f(k_t) - c_t - k_{t+1}]
\end{aligned}
$$

The result is an optimal sequence of choice variables (c_t, k_t) and a value function $V(k_{t*})$.

Given the initial condition k_{t^*} the maximum utility obtainable is $V(k_{t*})$.

Note that the value function is only a function of the initial capital stock.

Therefore, k_t is the **state variable** of the problem.

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: The right hand side of the Lagrangian can be broken into two terms:

$$
\begin{aligned}
V(k_{t*}) &= \max u(c_{t*}) + \lambda_{t*}[f(k_{t*}) - c_{t*} - k_{t*+1}] \\
&\quad + \beta V(k_{t*+1})
\end{aligned}
$$

- 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}.
- But since this equation holds for any arbitrary start date, we may drop date subscripts.

Recursive structure

This yields a **Bellman equation**:

$$\begin{aligned} V(k) &= \max u(c) + \lambda[f(k) - c - k'] \\ &\quad + \beta V(k') \end{aligned}$$

where the primes denote values in the next period.

Once we substitute the constraint into the second value function we have

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

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:

$$\phi(k) = \arg\max_c u(c) + \beta V(f(k) - c) \tag{1}$$

$\phi(k)$ 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 analyisis** 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:

$$V(k) = \max_{c,k'} u(c) + \beta V\left(k'\right) + \lambda\left(f(k) - c - k'\right) + \mu k' \tag{2}$$

First-order conditions: Kuhn Tucker for k'.

Backward Induction

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

Example:

$$\max \sum_{t=1}^{T} u(c_t) \tag{3}$$

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

$$k_{t+1} + c_t = f(k_t) + (1-\delta)k_t \tag{4}$$

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:

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

First order conditions:

$$\begin{aligned}&\beta^t u_1(t, t-1) + \beta^{t+1} u_2(t+1, t) = \lambda_t \\ &f'(k_t)\left(\beta^t u_1(t, t-1) + \beta^{t+1} u_2(t+1, t)\right) = \lambda_t (1-\delta) - \lambda_{t-1}\end{aligned} \tag{5}$$

Sequential Solution

Euler equation:

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

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:

$$U'(c_{t-1}) = U'(c_t)\left(f'(k_t) + 1 - \delta\right) \tag{6}$$

Interpretation

$$U'(c_{t-1}) = U'(c_t)\left(f'(k_t) + 1 - \delta\right) \tag{7}$$

- 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:

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

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

$$\begin{aligned} k' &= f(k) + (1-\delta)k - c \\ x &= f(k) - c \end{aligned}$$

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

$$\begin{aligned} V(k,z) &= \max_x u(f(k)-x,z) \\ &\quad +\beta V(x+(1-\delta)k, f(k)-x) \end{aligned}$$

FOC

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

Adding a state variable

The envelope conditions are

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

Now define

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

Then substitute out the V_z terms:

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

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:

$$\max \sum_{t=0}^{\infty} \beta^t \ \ln(c_t)$$

$$s.t. \ \ k_{t+1} \ = \ A \ k_t^{\theta} - c_t$$

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^{θ}, 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:

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

Envelope condition

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

First-order Conditions

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

$$
\begin{array}{rcl}
u'(f(k)-h(k)) & = & \beta V'(h(k)) \\
& \Rightarrow & \\
[f(k)-h(k)]^{-1} & = & \beta F/h(k) \\
h(k) & = & \beta F[f(k)-h(k)]
\end{array}
$$

The policy function is

$$
h(k) = \frac{\beta F}{1+\beta F} A k^{\theta} \tag{8}
$$

Envelope Condition

The envelope condition then implies

$$\begin{aligned} V'(k) &= u'(f(k)-h(k))f'(k) \\ &= \frac{A\theta k^{\theta-1}}{Ak^{\theta}-\beta F/(1+\beta F)Ak^{\theta}} \\ &= \frac{\theta k^{-1}}{1/(1+\beta F)} \\ &= \frac{\theta(1+\beta F)}{k} \end{aligned}$$

Solution

The guess implies $V'(k) = F/k$.

Substituting this into the previous equation yields an expression that can be solved for F:

$$\begin{aligned} F &= \theta(1+\beta F) \\ &= \theta/(1-\theta\beta) \end{aligned}$$

A bit of algebra shows that the policy function becomes

$$h(k) = \theta\beta A k^{\theta}$$

so that consumption is

$$f(k) - h(k) = (1-\theta\beta)Ak^{\theta}$$

Summary: Guess and Verify

- Use the guess for V in the FOC to get a policy function that depends on the unknown F: $k' = h(k; F)$.
- Use the guess for V in the envelope condition to get $V'(k; F)$ as a function of the unknown F.
- Get another expression for $V'(k; F)$ by differentiating the guess.
- Use the two expressions for V' to solve for F.

Summary: Guess and Verify

The claim is now that our guess satisfies the Bellman equation with this particular F.

We can verify this directly.

$$
\begin{aligned}
T(V) &= u(f(k)-h(k))+\beta\{E+F\ln(h(k))\} \\
&= \ln\left([1-\theta\beta]Ak^{\theta}\right)+\beta E+\beta\frac{\theta}{1-\theta\beta}\ln\left(\theta\beta Ak^{\theta}\right) \\
&= C_1+\left(\theta+\theta\frac{\theta\beta}{1-\theta\beta}\right)\ln(k) \\
&= C_1+F\ln(k)
\end{aligned}
$$

where C_1 is some constant (which could be used to find E).

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," .