Econ 626: Quantitative Methods II **Fall 2018**

Lecture 2: Dynamic Programming II

Lecturer: Prof. Daniel Levy *Scribes: Zhikun Lu*

Disclaimer: *Zhikun is fully responsible for the errors and typos appeared in the notes.*

Date: August 30, 2018

2.1 Infinite Horizon Model

(Growth Model – Lucas, Stokey and Prescott)

$$V(K_0) = \begin{cases} \max\limits_{\{C_0, K_1\}} [u(C_0) + \beta V(K_1)] \\ s.t. \quad C_0 + K_1 = f(K_0) \end{cases} \tag{2.1}$$

$$\Longrightarrow V(K_0) = \max_{\{K_1\}} [u(f(K_0) - K_1) + \beta V(K_1)] \tag{2.2}$$

FONC

$$u'(f(K_0) - K_1) = \beta V'(K_1) \tag{2.3}$$

By assumption

$$K_1 = g(K_0) \tag{2.4}$$

$$V(K_0) = u(f(K_0) - g(K_0)) + \beta V(g(K_0)) \tag{2.5}$$

Total differential:

$$\begin{aligned} V'(K_0) &= u'(f(K_0) - g(K_0))(f'(K_0) - g'(K_0)) + \beta V'(g(K_0))g'(K_0) \\ &= u'(f(K_0) - g(K_0))f'(K_0) - [u'(f(K_0) - g(K_0)) - \beta V'(g(K_0))]g'(K_0) \\ &= u'(f(K_0) - g(K_0))f'(K_0) \end{aligned} \tag{2.6}$$

Here, $[u'(f(K_0) - g(K_0)) - \beta V'(g(K_0))]g'(K_0) = 0$ by FONC (2.3), which follows from the envelope theorem.

Lead one period

$$V'(K_1) = u'(f(K_1) - K_2)f'(K_1) \tag{2.7}$$

$$\Longrightarrow u'(f(K_0) - K_1) = \beta u'(f(K_1) - K_2)f'(K_1) \tag{2.8}$$

which is a second order difference equation in K.

Assumptions

$$f(K_t) = K_t^{\alpha}, \quad 0 < \alpha < 1 \tag{2.9}$$

$$u(C_t) = \ln C_t \tag{2.10}$$

[1] Visit http://www.luzk.net/misc for updates.

Recall

$$V(K_t) = \begin{cases} \max_{\{C_t, K_{t+1}\}} [u(C_t) + \beta V(K_{t+1})] \\ s.t. \quad C_t + K_{t+1} = f(K_t) \end{cases} \tag{2.11}$$

Choice Variables: $\{C_t, K_{t+1}\}_{t=0}^{\infty}$

Set up a Lagrangian

$$\mathcal{L} = u(C_t) + \beta V(K_{t+1}) + \lambda_t[f(K_t) - C_t - K_{t+1}] \tag{2.12}$$

With our functional forms, it becomes

$$\mathcal{L} = \ln C_t + \beta V(K_{t+1}) + \lambda_t[K_t^\alpha - C_t - K_{t+1}] \tag{2.13}$$

FONC

$$[C_t] \quad \frac{1}{C_t} - \lambda_t = 0 \tag{2.14}$$

$$[K_{t+1}] \quad \beta V'(K_{t+1}) - \lambda_t = 0 \tag{2.15}$$

$$\implies \quad \frac{1}{C_t} = \beta V'(K_{t+1}) \tag{2.16}$$

Apply the envelope theorem again [`Benveniste-Scheinkman Theorem`]. Suppose we have a solution of all variables as a function of the state:

$$C_t = C_t(K_t) \tag{2.17}$$

$$K_{t+1} = K_{t+1}(K_t) \tag{2.18}$$

$$\lambda_t = \lambda_t(K_t) \tag{2.19}$$

then (2.13) $\implies$

$$V(K_t) = \mathcal{L}^* = \ln C_t(K_t) + \beta V(K_{t+1}(K_t)) + \lambda_t(K_t)[K_t^\alpha - C_t(K_t) - K_{t+1}(K_t)] \tag{2.20}$$

which is a function of K_t only.

Differentiate w.r.t. K_t

$$V'(K_t) = \frac{1}{C_t(K_t)} C_t'(K_t) + \beta V'(K_{t+1}) K_{t+1}'(K_t) + \lambda_t'(K_t)[K_t^\alpha - C_t(K_t) - K_{t+1}(K_t)] + \lambda_t(K_t)[\alpha K_t^{\alpha-1} - C_t'(K_t) - K_{t+1}'(K_t)] \tag{2.21}$$

$$= \frac{1}{C_t(K_t)} C_t'(K_t) + \beta V'(K_{t+1}) K_{t+1}'(K_t) + \lambda_t(K_t)[\alpha K_t^{\alpha-1} - C_t'(K_t) + K_{t+1}'(K_t)] \quad (\text{as } K_t^\alpha - C_t - K_{t+1} = 0) \tag{2.22}$$

$$= C_t'(K_t)\left[\frac{1}{C_t(K_t)} - \lambda_t(K_t)\right] + K_{t+1}'(K_t)[\beta V'(K_{t+1}(K_t)) - \lambda_t(K_t)] + \lambda_t(K_t)\alpha K_t^{\alpha-1} \tag{2.23}$$

$$= \lambda_t(K_t)\alpha K_t^{\alpha-1} \quad \text{(since the first two terms are 0 by FONC)} \tag{2.24}$$

$$= \frac{1}{C_t(K_t)} \alpha K_t^{\alpha-1} \tag{2.25}$$

Lead one period forward

$$V'(K_{t+1}) = \frac{1}{C_{t+1}(K_{t+1})} \alpha K_{t+1}^{\alpha-1} \tag{2.26}$$

Note: we could obtain the same result directly from (2.12) with envelope theorem. If we take the value function of (2.12)

$$V(K_t) = \max\{u(C_t) + \beta V(K_{t+1}) + \lambda_t[f(K_t) - C_t - K_{t+1}]\} \tag{2.27}$$

and ignore the dependence of C_t and K_{t+1}, because we are at a maximum point, then by the envelope theorem:

$$V'(K_t) = \lambda_t f'(K_t) \overset{\text{lead one-period}}{\Longrightarrow} V'(K_{t+1}) = \lambda_{t+1} f'(K_{t+1}), \tag{2.28}$$

which is the same as (2.26) because of (2.14).

Next, (2.26) and (2.16) lead to

$$\frac{1}{C_t} = \beta \frac{1}{C_{t+1}} \alpha K_{t+1}^{\alpha-1} \tag{2.29}$$

which is a first order difference equation in C. Recall the budget constraint

$$C_t + K_{t+1} = K_t^{\alpha}, \tag{2.30}$$

which is a first order difference equation in K.

Note that (2.29) and (2.30) are connected. Together, they form a system of two first order difference equations.

$$\begin{cases} \dfrac{1}{C_t} = \beta \frac{1}{C_{t+1}} \alpha K_{t+1}^{\alpha-1} \\ C_t + K_{t+1} = K_t^{\alpha} \end{cases}$$

2.2 Solution Methods

Three method for solving dynamic programming problems:

1. Policy function iteration
2. Value function iteration
3. Guess "intelligently" (not easy)

2.2.1 Policy Function Iteration

From (2.29), we have

$$\frac{K_{t+1}}{C_t} = \frac{\alpha\beta}{C_{t+1}} K_{t+1}^{\alpha} \tag{2.31}$$

(2.30) $\Longrightarrow$

$$1 + \frac{K_{t+1}}{C_t} = \frac{K_t^{\alpha}}{C_t} \tag{2.32}$$

Together $\Longrightarrow$

$$\alpha\beta \frac{K_{t+1}^{\alpha}}{C_{t+1}} = \frac{K_t^{\alpha}}{C_t} - 1 \tag{2.33}$$

which is a first order difference equation in $\dfrac{K^{\alpha}}{C}$.

Solve the difference equation by sucessive substitution forward:

$$\frac{K_t^\alpha}{C_t} = 1 + \alpha\beta \frac{K_{t+1}^\alpha}{C_{t+1}} \tag{2.34}$$

$$\frac{K_{t+1}^\alpha}{C_{t+1}} = 1 + \alpha\beta \frac{K_{t+2}^\alpha}{C_{t+2}} \tag{2.35}$$

$$\frac{K_t^\alpha}{C_t} = 1 + \alpha\beta(1 + \alpha\beta \frac{K_{t+2}^\alpha}{C_{t+2}}) \tag{2.36}$$

$$\frac{K_t^\alpha}{C_t} = 1 + \alpha\beta + (\alpha\beta)^2 \frac{K_{t+2}^\alpha}{C_{t+2}} \tag{2.37}$$

$$\frac{K_{t+2}^\alpha}{C_{t+2}} = 1 + \alpha\beta \frac{K_{t+3}^\alpha}{C_{t+3}} \tag{2.38}$$

$$\frac{K_t^\alpha}{C_t} = 1 + \alpha\beta + (\alpha\beta)^2(1 + \alpha\beta \frac{K_{t+3}^\alpha}{C_{t+3}}) \tag{2.39}$$

$$\frac{K_t^\alpha}{C_t} = 1 + \alpha\beta + (\alpha\beta)^2 + (\alpha\beta)^3 \frac{K_{t+3}^\alpha}{C_{t+3}} \tag{2.40}$$

Continue substituting forward infinitely many times, we get

$$\frac{K_t^\alpha}{C_t} = 1 + \alpha\beta + (\alpha\beta)^2 + (\alpha\beta)^3 + \cdots + \lim_{s\to\infty} (\alpha\beta)^s \frac{K_{t+s}^\alpha}{C_{t+s}}. \tag{2.41}$$

If $\lim_{s\to\infty}(\alpha\beta)^s \frac{K_{t+s}^\alpha}{C_{t+s}} = 0$, then

$$\frac{K_t^\alpha}{C_t} = \sum_{s=0}^{\infty} (\alpha\beta)^s = \frac{1}{1-\alpha\beta}, \tag{2.42}$$

which leads to our policy function

$$C_t^* = (1-\alpha\beta)K_t^\alpha. \tag{2.43}$$

Comment

(2.34) $\Longrightarrow$ after N times substitution

$$\frac{K_t^\alpha}{C_t} = 1 + \alpha\beta + (\alpha\beta)^2 + \cdots + (\alpha\beta)^N + (\alpha\beta)^{N+1} \frac{K_{t+N+1}^\alpha}{C_{t+N+1}}. \tag{2.44}$$

Assumption

$$\lim_{N\to\infty} (\alpha\beta)^{N+1} \frac{K_{t+N+1}^\alpha}{C_{t+N+1}} = 0. \tag{2.45}$$

As $N \to \infty$, $(\alpha\beta)^{N+1} \to 0$. Therefore by assuming the above limit, we are imposing a limit on how fast $\frac{K_{t+N+1}^\alpha}{C_{t+N+1}}$ grows in the future. Specifically, we require that $\frac{K_{t+N+1}^\alpha}{C_{t+N+1}}$ will not grow as $N \to \infty$ at a rate that exceeds the rate in which $(\alpha\beta)^{N+1}$ shrinks.

(2.43) is the `consumption function`, where $1-\alpha\beta$ is the MPC.

Plug into $C_t + K_{t+1} = K_t^\alpha$:

$$(1-\alpha\beta)K_t^\alpha + K_{t+1} = K_t^\alpha \tag{2.46}$$

$$K_{t+1}^* = \alpha\beta K_t^\alpha \tag{2.47}$$

$$\Longrightarrow \frac{K_{t+1}^*}{K_t^\alpha} = \alpha\beta$$

confirming our assertion that the steady state saving rate is $\alpha\beta$. Thus, (2.43) abd (2.47) are the optimal policy functions.

2.2.2 Guess a "solution"

Sometimes, based on our experience, we may be able to tell something about the properties of the policy functions.

For example

Suppose that we can guess that

$$\frac{K_{t+1}}{K_t^\alpha} = \text{constant} \equiv \Gamma, \tag{2.48}$$

but we don't know its value. Then

$$K_{t+1} = \Gamma K_t^\alpha \tag{2.49}$$

$$C_t = (1-\Gamma)K_t^\alpha \tag{2.50}$$

$$\frac{K_{t+1}}{C_t} = \frac{\Gamma}{1-\Gamma} \tag{2.51}$$

From (2.31),

$$\frac{\Gamma}{1-\Gamma} = \frac{K_{t+1}}{C_t} = \frac{\alpha\beta}{C_{t+1}}K_{t+1}^\alpha \Longrightarrow C_{t+1} = \frac{1-\Gamma}{\Gamma}\alpha\beta K_{t+1}^\alpha \tag{2.52}$$

(2.50) $\Longrightarrow$

$$C_{t+1} = (1-\Gamma)K_{t+1}^\alpha \tag{2.53}$$

Comparing (2.52) and (2.53), we conclude that

$$\frac{1-\Gamma}{\Gamma}\alpha\beta = 1-\Gamma \tag{2.54}$$

$$\Gamma = \alpha\beta \tag{2.55}$$

2.2.3 Value function iteration

Value function changes from iteration to iteration, i.e., every period, until convergence.

Typically, we start with some initial functional form, often as simple as $V(\cdot) = 0$, and iterate, until convergence.

Start with inital guess $V_0(K_{T+1}) = 0$

$$V_1(K_T) = \begin{cases} \max\limits_{\{C_T, K_{T+1}\}} [u(C_T) + \beta V_0(K_{T+1})] \\ s.t. \quad C_T + K_{T+1} = K_T^\alpha \end{cases} \tag{2.56}$$

$$\Longrightarrow \begin{cases} K_{T+1} = 0 \\ C_T = K_T^\alpha \end{cases}, \tag{2.57}$$

Hence, $V_1(K_T) = \ln(K_T^\alpha)$. Let's continue:

$$V_2(K_{T-1}) = \begin{cases} \max\limits_{\{C_{T-1}, K_T\}} [u(C_{T-1}) + \beta V_1(K_T)] \\ s.t. \quad C_{T-1} + K_T = K_{T-1}^\alpha \end{cases} \tag{2.58}$$

$$\Longrightarrow V_2(K_{T-1}) = \begin{cases} \max\limits_{\{C_{T-1}, K_T\}} \ln(C_{T-1}) + \beta\ln(K_T^\alpha) \\ s.t. \quad C_{T-1} + K_T = K_{T-1}^\alpha \end{cases} \tag{2.59}$$

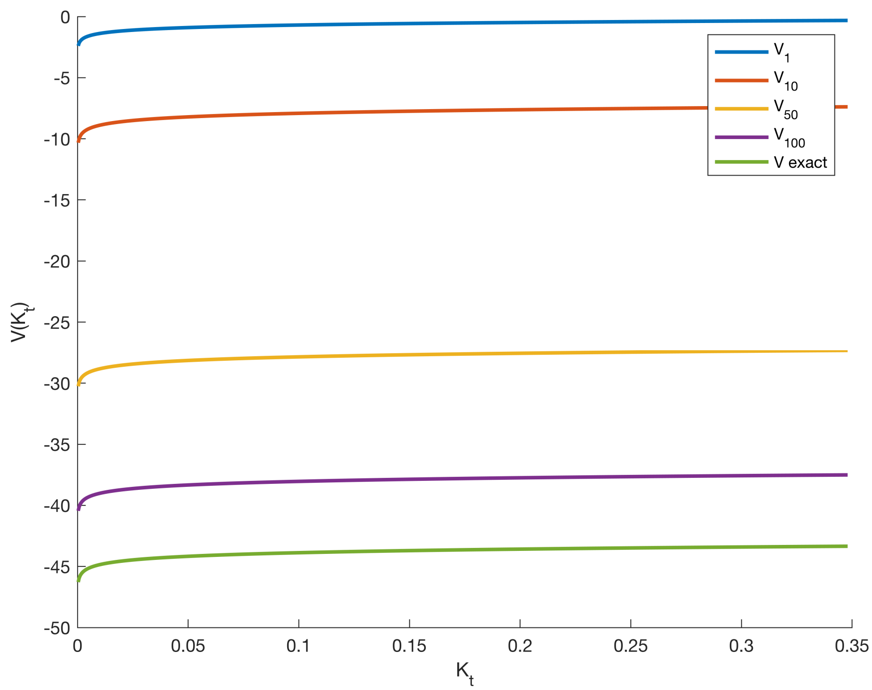

Figure 2.1: An Numerical Example with $\alpha = 0.3, \beta = 0.98$

Code to reproduce figure (2.1)

```
clear all;clc;
% luzhikun
beta = 0.98; alpha = 0.3; delta = 1;

f_ss = @(k)alpha*k^(alpha-1)+1-delta-1/beta
k_ss = fsolve(f_ss,1)
k_initial = 0.5*k_ss

num_state = 1000;
phi = 2*k_ss/num_state;
k_state = phi:phi:(2*k_ss);

[K_x, K_y] = meshgrid(k_state,k_state);

v = zeros(1,num_state);
epsilon = 10^(-20)

num_iter = 100

xx = [1, 10, 50, 100]
figure(1)
hold on
```

```
for ii = 1:num_iter
  v_primitive = v;
  c =  max( K_x.^alpha + (1 - delta)*K_x - K_y, epsilon);
  v_matrix =  log( c ) + beta*v'*ones(1,num_state);
  [v_improved, k_choice_vector] = max(v_matrix);
  v = v_improved;
  %error_ = max(v_improved - v_primitive)
  if (ii == xx(1))|(ii == xx(2))|(ii == xx(3))|(ii == xx(4))
    plot(k_state, v, 'linewidth', 2)
  end
end

% exact solution
v_exact = v;
a = 1/(1-beta)*(log(1-alpha*beta)+alpha*beta/(1-alpha*beta)*log(alpha*beta));
b = alpha/(1-alpha*beta);
v_exact = a+b*log(k_state);
plot(k_state, v_exact, 'linewidth', 2)

hold off

%title('Value function iteration');
xlabel('K_t'); ylabel('V(K_t)');
legend('V_1', 'V_{10}', 'V_{50}', 'V_{100}','V exact')
```

References