Macro Theory B

Dynamic Programming

Ofer Setty*
The Eitan Berglas School of Economics - Tel Aviv University

March 12, 2015

1 Dynamic optimization with finite horizon

The economy has a social planner whose horizon is finite, T.

The planner's problem is:

$$Max_{\{c_t,k_{t+1}\}_{t=0}^{T}} U(c_0, c_1...c_T) = \sum_{t=0}^{T} \beta^t u(c_t), \quad 0 < \beta < 1,$$

$$\text{subject to} \qquad c_t + k_{t+1} \leq f(k_t) + (1-\delta)k_t,$$

$$k_0 \text{ given},$$

$$c_t \geq 0, \quad t = 0, 1, ..., T,$$

$$k_{t+1} \geq 0, \quad t = 0, 1, ..., T.$$

We assume:

1. Time-separability of preferences - so, for example, there are no habits or durable goods.

*This set of notes is based on those of Zvi Hercowitz in Semester Bet 2010.

2. $u_1(c) > 0, u_{11} < 0, \lim_{c\to\infty} u_1(c) = 0$, $\lim_{c\to 0} u_1(c) = \infty$. The latter assumption implies that c will never be zero. Hence the non-negativity constraint for consumption will never bind and can be omitted.

3. $f_1(k) > 0, f_{11} < 0, \lim_{k\to\infty} f_1(k) = 0$, $\lim_{k\to 0} f_1(k) = \infty$. The latter assumption implies that $k_t > 0$, $t = 1, ...T$, and hence the restriction for $t = 1, ..T-1$ can be omitted. However, we still need the restriction $k_{T+1} \geq 0$, otherwise, the planner will let k_{T+1} go to $-\infty$. We expect k_{T+1} to be 0.

4. There are no financial assets, only physical investment. This corresponds to a planning problem in a closed economy.

The Lagrangian function is:

$$L = \sum_{t=0}^{T} \beta^t \{u(c_t) + \lambda_t [f(k_t) + (1-\delta)k_t - c_t - k_{t+1}]\} + \mu_T \beta^T k_{T+1},$$

See below for the interpretation of μ_T.

The first-order conditions are:

$$\frac{\partial L}{\partial c_t} = \beta^t [u_1(c_t) - \lambda_t] = 0, \qquad t = 0, 1, ..., T, \tag{1}$$

$$\frac{\partial L}{\partial k_{t+1}} = -\beta^t \lambda_t + \beta^{t+1} \lambda_{t+1} [f_1(k_{t+1}) + 1 - \delta] = 0, \quad t = 0, 1..., T-1, \tag{2}$$

$$\frac{\partial L}{\partial k_{T+1}} = \beta^T (-\lambda_T + \mu_T) = 0. \tag{3}$$

Optimality conditions formulated as Kuhn-Tucker conditions. These are a generalization of the Lagrange multipliers method for nonlinear restrictions which may not bind.

The Kuhn-Tucker conditions are the conditions above plus:

$$\lambda_t \left[f(k_t) + (1-\delta)k_t - c_t - k_{t+1} \right] = 0, \quad t = 0, 1, ..., T,$$
$$\lambda_t \geq 0, \quad t = 0, 1, ..., T,$$
$$\mu_T \beta^T k_{T+1} = 0,$$
$$\mu_T \geq 0.$$

Discussion of the solution: Given that

$$u_1(c_t) > 0, \quad t = 0, 1, ..., T,$$

$$\lambda_t > 0, \quad t = 0, 1, ..., T$$

from (1). This implies two things: first, the resource constraints bind at all times, and second, from (3), $\mu_T > 0$, which in turn implies that $k_{T+1} = 0$. Using (1) and (2), and replacing (1) into (2), the dynamic behavior is the solution to the system

$$\begin{aligned} 0 &= -u_1(f(k_t) + (1-\delta)k_t - k_{t+1}) \\ &\quad + \beta u_1(f(k_{t+1}) + (1-\delta)k_{t+1} - k_{t+2}) \left[f_1(k_{t+1}) + 1 - \delta \right], \\ &\text{for } t = 0, 1, ..., T-1, \\ k_0 &= \text{given}, \\ 0 &= \beta^T \mu_T k_{T+1} \end{aligned}$$

The first is called the "Euler Equation," the second is the initial condition, and the third is the "transversality" condition, or final condition.

Interpretation of μ_T: In general the constraint would be:

$$k_{T+1} - \chi \geq 0, \quad \chi \geq 0.$$

Hence, μ_T represents the marginal utility of *lowering* χ. I.e., how much we lose if we are required to leave some capital behind.

The system is a second-order difference equation in k. These are $T + 2$ equations with $T + 2$ unknowns: $k_0, k_1, ...k_{T+1}$. The two equations for the initial and the terminal conditions directly solve for 2 of the unknowns, remaining the T Euler equations with T unknowns.

Interpretation of the Euler equation:

$$u_1(c_t) = \beta u_1(c_{t+1}) \left[f_1(k_{t+1}) + 1 - \delta \right].$$

Marginal utility cost of investment—in terms of forgone utility from consumption—equals discounted value of the physical return next period translated in utility terms.

2 Dynamic optimization with infinite horizon

The main reason for considering the infinite horizon is intergenerational altruism: parents care about their children, and know that they will care about their children, and so on.

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

$$\text{subject to} \qquad c_t + k_{t+1} \leq f(k_t) + (1 - \delta) k_t,$$

$$k_0 \text{ given},$$

$$\lim_{T \to \infty} \beta^T u_1(c_T) k_T = 0.$$

The last equation is the "transversality condition"; a counterpart to the final condition in the finite-horizon case. In that case the final condition was $\beta^T u_1(c_T) k_{T+1} = 0$.

The solution to the problem is the decision rule

$$k_{t+1} = g(k_t),$$

An optimal decision rule is a mapping of a state to decisions that maximizes some function. I.e., the optimal choice for any possible value of the state.

We need to compute $g(\cdot)$ which solves the optimization problem.

The Euler equation, as seen previously, is

$$-u_1(f(k) + (1-\delta)k - k') + \beta u_1(f(k') + (1-\delta)k' - k'')(f_1(k') + (1-\delta)) = 0.$$

A prime indicates one period ahead, and a double prime two periods ahead.

As mentioned earlier, this is a second order non-linear difference equation in k. Because of the infinite number of periods, the solution method used previously, solution of a set of equations with an equal number of unknowns, is not applicable now. Here, we solve the problem using a Dynamic Programming procedure.

2.1 The Bellman equation

Solution of the problem (decision rule):

$$k' = g(k).$$

Infinite-horizon Dynamic Programming logic:

First, because of the infinite horizon, the planning horizon is constant over time, and hence the decision rule is also constant over time. In particular, the *decision rule* this period is the same as the *decision rule* next period.

Second. The feature above turns the dynamic problem—where the current decision depends on the future decision—into a functional equation in one functional unknown.

Define the "value" function $V(\cdot)$ as the solution to our problem given the initial condition:

$$V(k_0) = Max \sum_{t=0}^{\infty} \beta^t u(f(k_t) + (1-\delta)k_t - k_{t+1}).$$

The Bellman (Richard E.Bellman (1920–1984)) *functional* equation is

$$V(k) = Max_{k'} \left[u(f(k) + (1-\delta)k - k') + \beta V(k') \right]. \tag{4}$$

$V(k)$ is the solution to this equation. The first-order condition is:

$$-u_1(f(k) + (1-\delta)k - k') + \beta V_1(k') = 0. \tag{5}$$

This equation is implicitly $k' = g(k)$. If we knew the function V we could solve for g. The only problem is that we'll know V only when the problem is solved.

Note that $V(k_0)$ is the value function and that $V(k) = Max_{k'} \left[u(f(k) + (1-\delta)k - k') + \beta V(k') \right]$ is the Bellman equation. Do not mix the two.

2.2 The Envelope theorem and first order conditions

Note that (5) should be identical to the original Euler equation. This can be shown as follows:

Given (4), (the envelope theorem), $V_1(k)$ becomes

$$V_1(k) = u_1(c) \left[f_1(k) + (1-\delta) \right]$$

Iterating forward and substituting into (5) yields:

$$-u_1(c) + \beta u_1(c') \left[f_1 (k') + (1 - \delta) \right] = 0,$$

which is the Euler equation in the original form.

2.3 Solution methods

2.3.1 Value function iteration

If we have no educated guess for the functional form of the value function then we start with any value. Under some regularity conditions a contraction mapping is guaranteed and we will end up with the optimal value.

Practically, to solve the functional equation we conjecture a form for the function $V(k)$, say $V^1(k)$, compute $V_1^1(k')$, substitute into (5)

$$-u_1 \left(f(k) + (1 - \delta)k - k' \right) + \beta V_1^1(k') = 0$$

This is one equation in k'. The solution:

$$k' = g^1(k).$$

Substitute this into the Bellman equation to get the resulting value function:

$$V^2(k) = u(f(k) + (1 - \delta)k - g^1(k)) + \beta V^1(g^1(k)).$$

If $V^1 = V^2$, done. Otherwise, numerical methods can be used to approximate V by successive iterations until convergence. Note that convergence of V involves convergence of g.

This example with $\delta = 1,\ f(k) = k^a$ and $u(c) = \ln c$ can be actually solved by hand.

$$V(k) = Max_{k'}\left[\ln(k^a - k') + \beta V(k')\right]. \tag{6}$$

$Guess : V^1(k) = 0$. In this case choose optimally $k' = 0$. Then:

$$V(k) = \ln(k^a) = \alpha \ln(k) \tag{7}$$

Therefore: $V^2(k) = \alpha \ln(k)$.

$$V(k) = Max_{k'}\left[\ln(k^a - k') + \beta\alpha \ln(k')\right]. \tag{8}$$

The FOC is:

$$\begin{aligned}
\frac{1}{k^a - k'} &= \frac{\beta\alpha}{k'} \\
k' &= \beta\alpha k^a - \beta\alpha k^{'} \\
k' &= \frac{\beta\alpha}{1+\beta\alpha} k^a \\
V^3(k) &= \ln(k^a - k') + \beta\alpha \ln(k') \\
&= \ln(k^a - \frac{\beta\alpha}{1+\beta\alpha} k^a) + \beta\alpha \ln\left(\frac{\beta\alpha}{1+\beta\alpha} k^a\right) \\
&= \ln(k^a) + \ln\left(1 - \frac{\beta\alpha}{1+\beta\alpha}\right) + \beta\alpha \ln(k^a) + \beta\alpha \ln\left(\frac{\beta\alpha}{1+\beta\alpha}\right) \\
&= \alpha(1+\alpha\beta)\ln(k) + \ln\left(\frac{1}{1+\beta\alpha}\right) + \beta\alpha \ln\left(\frac{\beta\alpha}{1+\beta\alpha}\right) \qquad (9)
\end{aligned}$$

Continue till:

$$V(k) = \frac{\alpha}{1-\alpha\beta}\ln(k) + \frac{1}{1-\beta}\frac{\alpha\beta}{1-\alpha\beta}\ln(\alpha\beta) + \frac{1}{1-\beta}\ln(1-\alpha\beta) \tag{10}$$

2.3.2 Guess and verify

Since we already know the solution we will guess the functional form and find the parameters:

$$
\begin{aligned}
V(k) &= E + F\ln(k) \\
V(k) &= Max_{k'}\left[\ln(k^a - k') + \beta\left(E + F\ln(k')\right)\right]
\end{aligned}
$$

From the FOC:

$$
k' = \frac{\beta F k^a}{1 + \beta F}
$$

Substitute into the Bellman equation to get:

$$
\begin{aligned}
F &= \frac{\alpha}{1 - \beta\alpha} \\
E &= \frac{\ln(1 - \beta\alpha) + \frac{\alpha\beta}{1-\beta\alpha}\ln(\alpha\beta)}{1 - \beta}
\end{aligned}
$$

2.3.3 Policy function iteration (Euler equation solution)

The Euler equation can be written as

$$
u_1\left(f(k) + (1-\delta)k - k'\right) = \beta u_1\left(f(k') + (1-\delta)k' - g(k')\right)\left[f_1(k') + (1-\delta)\right],
$$

which also an implicitly $k' = g(k)$. It also can be expressed as *functional* equation in $g(.)$, similarly as the Bellman equation is in $V(.)$:

$$
u_1\left(f(k) + (1-\delta)k - g(k)\right) = \beta u_1\left(f(k') + (1-\delta)k' - g(k')\right)\left[f_1(k') + (1-\delta)\right].
$$

Practically, we conjecture the solution $g^1(k)$:

$$u_1(f(k) + (1-\delta)k - k') = \beta u_1\left(f(k') + (1-\delta)k' - g^1(k')\right)[f_1(k') + (1-\delta)]$$

This is one equation in k'. The solution is:

$$k' = g^2(k)$$

If $g^1 = g^2$, done. Otherwise, numerical methods can be used to approximate g by successive iterations until convergence.