Discrete Time Dynamic Programming, Continued

IV. Constraints at Infinity

We could have set up our problem with two sorts of additional terms. The objective function could have been expanded to the form

$$E\left\{\sum_{t=0}^{\infty} b^t U(C_t, S_t)\right\} + \lim_{t\to\infty} b^t E\big[W(S_t)\big], \tag{23}$$

and the constraints could have been expanded to include the requirement that

$$\lim_{t\to\infty} b^t G(S_t) \le 0 \ . \tag{24}$$

Note that in (24) there is no expectation operator, so we are requiring that the inequality hold with certainty.

With these additions, the problem retains its recursive structure and the Bellman equation still functions in the same way as a necessary and (with some side conditions) sufficient condition for an optimum. You might go through the arguments yourself step by step to be sure that this is true.

V. Exogenous States

In many economic models the assumption off i.i.d. shocks e, (F) in the first part of the notes, is a source of some difficulty. The objective function is often interpretable as a utility function or profit function and the constraints (2) on the evolution of the state are often interpretable as budget constraints or production relations. But in such objects the serial dependence properties of "exogenous disturbances" is not naturally taken to be restricted. "Technology" in a production function, for example, is often taken to drift upward in a serially correlated way, and "income" in a consumer's optimization problem is naturally taken to be serially correlated. Such situations can be accomodated by including in the equations for evolution of the state (2) a description of the serial dependence in the exogenous disturbances. In this case the exogenous components of the problem might be labeled Z, and their stochastic evolution described by an equation of the form

$$Z_{t+1} = g(Z_t, h_{t+1}) \tag{25}$$

in which h_t satisfies both (E) and (F). The state vector in the problem is then taken to include both endogenous states, which we might label K, and exogenous states Z. The full state vector $S = [K'\ Z']'$. The equations of evolution given in (2) can then be expanded to the specialized form

$$\begin{aligned} K_{t+1} &= f(C_t, K_t, Z_t, e_{t+1}) \\ Z_{t+1} &= g(Z_t, h_{t+1}) \end{aligned} \ . \tag{26}$$

Here h is playing the role of a component of e in the more general specification (2).

We point out this special case here not only because it shows how serial dependence in stochastic components of the model can be accommodated within a dynamic programming setup, but also because it often helps simplify finding and interpreting solutions to recognize a structure like that in(25)-(26) when it is present.

VI.First Order Conditions

The usual techniques of calculus, in particular the Kuhn-Tucker conditions for an optimum, can be applied to the Bellman equation (3) when *U* and *V* are differentiable and have appropriate concavity properties and when the constraints defined in (C) also have differentiable forms. Slightly abusing notation, we will use V' to stand for $D_S V = \frac{\partial V}{\partial S}$ in what follows. Assume that for each value *S* of the state vector and each *C* in $\Gamma(S)$,

$$E\left[V'(S)\cdot D_C f(C,S,\varepsilon)\right],\ D_C U(C,S) \tag{27}$$

are well-defined.

Assume also that the set $\Gamma(S)$ can be characterized by the inequality (or vector of inequalities)

$$H(C,S)\leq 0\ . \tag{28}$$

The usual Kuhn-Tucker theorem then asserts that a necessary condition for (3) to hold (assuming that the lub in (3) is attained) is that

$$\frac{\partial U(C,S)}{\partial C}+\beta E\left\{V'(f(C,S,\varepsilon))\cdot\frac{\partial f}{\partial C}\right\}=\lambda(S)\frac{\partial H(C,S)}{\partial C}\ , \tag{29}$$

with $\lambda(S)>0$ for those elements of λ corresponding to which the inequality in (28) is an equality, and $\lambda(S)=0$ for those where the inequality is strict. Recognizing that optimal *C* is a function $C^*(S)$ of *S*, we can differentiate left and right-hand sides of (3) with respect to *S* to obtain

$$V'(S)=\frac{\partial U}{\partial S}+\beta E\left\{V'(S)\cdot\frac{\partial f}{\partial S}\right\}+\frac{\partial}{\partial C}\left\{U+\beta E\left[V(f(C^*(S),S,\varepsilon))\right]\right\}\cdot\frac{\partial C^*}{\partial S}\ . \tag{30}$$

In writing (30) we assume that C^* is differentiable. Notice that the term in brackets following $\partial/\partial C$ is exactly the left-hand side of (29). The fact that this term is zero in the case with no constraints on *C* is a special case of a theorem called the envelope theorem, and (30) with that term substituted out is sometimes called the "envelope condition" in dynamic programming jargon.

Because (30) involves the unknown function C^*, it is not directly very useful. Using (29) we can convert(30) to the form

$$V'(S)=\frac{\partial U}{\partial S}+\beta E\left\{V'(f(C,S,\varepsilon))\frac{\partial f}{\partial S}\right\}+\lambda(S)\frac{\partial H(C,S)}{\partial C}\cdot\frac{\partial C^*(S)}{\partial S}\ . \tag{31}$$

Then we observe that, if $\lambda(S)>0$, so the constraint that *H*=0 is binding,

$$\frac{\partial H(C^*(S),S)}{\partial S} = D_1 H(C^*(S),S) \cdot \frac{\partial C^*(S)}{\partial S} + D_2 H(C^*(S),S) = 0 \ . \tag{32}$$

Using (32) in (31) then gives us the usual form of the envelope condition

$$V' = \frac{\partial U}{\partial S} + \beta E\left[V'\big(f(C,S,\varepsilon)\big) \cdot \frac{\partial f}{\partial S} \right] - \lambda(S) \cdot \frac{\partial H(C,S)}{\partial S} \ . \tag{33}$$

Equations (29) and (33) together are sometimes called the Euler equations for the problem. Note that one way to remember them is to form the "Hamiltonian-like" expression

$$V(S) - U(C,S) - \beta E\big[V(f(C,S,\varepsilon))\big] + \lambda H(C,S) \tag{34}$$

Equations (29) and (33) are then the partial derivatives of (34) with respect to *C* and *S*, respectively.

When (28) holds with equality, (28), (29) and (33) are a system with as many equations as the sum of the dimensions of the vectors *C*, *S* and λ. If it were not for the expectation operators in the system, we could solve it for $V'(S)$, *C* and λ for any given S_0. Because of the expectation operator, we have to treat the system as a functional equation. One way to use it computationally, for example, would be to postulate a functional form for V' and use the postulated form in computing the expectations in (29) and (33). Then the system can ordinarily be solved jointly for *C*, λ and $V'(S)$ for any given *S*, producing (if we solve it for many values of *S*) a new candidate guess for V'. Iterating this process we might hope to arrive at a V' function that satisfies the equations, and in the process at a C^* function that defines optimal decisions. Methods like this are, or can be made to be, numerically more efficient than value function iteration when the problem is smooth enough to allow their application.