Lecture 6 Dynamic Programming
Consider a dynamic system that evolves over time according to:

\[ x_{t+1} = g(x_t, u_t) \]
Consider a dynamic system that evolves over time according to:

\[ x_{t+1} = g(x_t, u_t) \]

wherein \( x_t \) and \( u_t \) are, respectively, state and control variable(s). Canonical DP is to find a solution to the following optimization problem:
Consider a dynamic system that evolves over time according to:

\[ x_{t+1} = g(x_t, u_t) \]

wherein \( x_t \) and \( u_t \) are, respectively, state and control variable(s). Canonical DP is to find a solution to the following optimization problem:

\[
\text{Maximize } \sum_{t=0}^{T} \beta^t r(x_t, u_t) \text{ s.t. } \\
\text{ } x_0 \text{ given} \\
x_{t+1} = g(x_t, u_t) \\
\text{(StTr)}
\]
Introduction 2

The standard set of assumptions used in DP are:

1. $r(x_t, u_t)$ is a concave function
2. The set $\{(x_{t+1}, x_t) : x_{t+1} \leq g(x_t, u_t)\}$ is convex and compact
3. For most of the major properties of DP, it is not necessary to assume time invariance of $r$ or $g$. We can have instead: $r_t(x_t, u_t)$ and $x_{t+1} = g_t(x_t, u_t)$ [Notice also that we can always rewrite $g_t(x_t, u_t) = G(t, x_t, u_t)$]
4. $\beta < 1$ is not necessary but extremely powerful assumption. To begin with, if $\beta = 1$, clearly the maximand is unbounded and we need to patch up the DP by, for example, redefining the maximand as

$$\lim_{T \to \infty} \left[ \frac{1}{T} \sum_{t=0}^{T} r(x_t, u_t) \right]$$

5. $\{x_t\}$ is a vector of state variables whose dynamics follows the (StTr), and $\{u_t\}$ is a vector of control variables which enters in payoff $r(x_t, u_t)$ and/or transition equation $g_t(x_t, u_t)$. 

Numerical methods using MATLAB

April 2009 112 / 147
When $T < \infty$, the DP can be solved by backward induction: as of $t = T$, the problem above reduces to:
When $T < \infty$, the DP can be solved by backward induction: as of $t=T$, the problem above reduces to:

$$\text{Maximize}[r(x_T, u_T)]$$

with $x_T$ given
When $T < \infty$, the DP can be solved by backward induction: as of $t = T$, the problem above reduces to:

\[
\text{Maximize } [r(x_T, u_T)]
\]

\[
\text{with } x_T \text{ given}
\]

The solution is simply:
When $T < \infty$, the DP can be solved by backward induction: as of $t=T$, the problem above reduces to:

\[
\text{Maximize}[r(x_T, u_T)]
\]

\[
\text{with } x_T \text{ given}
\]

The solution is simply:

\[
u_T = \arg \max r(x_T, u_T)
\]
Introduction 3 Finite Horizon

- When $T < \infty$, the DP can be solved by backward induction: as of $t = T$, the problem above reduces to:

  $$\text{Maximize}[r(x_T, u_T)]$$

  with $x_T$ given

- The solution is simply:

  $$u_T = \arg \max r(x_T, u_T)$$

- which can be denoted as:
When $T < \infty$, the DP can be solved by backward induction: as of $t=T$, the problem above reduces to:

\[
\text{Maximize} [r(x_T, u_T)]
\]
\[
\text{with } x_T \text{ given}
\]

The solution is simply:

\[
u_T = \arg \max r(x_T, u_T)
\]

which can be denoted as:

\[
u_T = h_T(x_T)
\]
When $T < \infty$, the DP can be solved by backward induction: as of $t=T$, the problem above reduces to:

$$\text{Maximize}[r(x_T, u_T)]$$

with $x_T$ given

The solution is simply:

$$u_T = \text{arg max } r(x_T, u_T)$$

which can be denoted as:

$$u_T = h_T(x_T)$$

plugging this solution back to $r(x_T, u_T)$, we get:
When \( T < \infty \), the DP can be solved by backward induction: as of \( t = T \), the problem above reduces to:

\[
\text{Maximize}\left[r(x_T, u_T)\right] \\
\text{with } x_T \text{ given}
\]

The solution is simply:

\[
u_T = \arg \max r(x_T, u_T)
\]

which can be denoted as:

\[
u_T = h_T(x_T)
\]

plugging this solution back to \( r(x_T, u_T) \), we get:

\[
V_T(x_T) = r(x_T, h_T(x_T))
\]
Now as of $t = T - 1$, the problem is:
Now as of $t = T - 1$, the problem is:

$$\text{Maximize}\left[r(x_{T-1}, u_{T-1}) + \beta r(x_T, u_T)\right]$$

s.t.

$$x_T = g(x_{T-1}, u_{T-1})$$
Now as of $t = T - 1$, the problem is:

$$\text{Maximize} [r(x_{T-1}, u_{T-1}) + \beta r(x_T, u_T)]$$

s.t.

$$x_T = g(x_{T-1}, u_{T-1})$$

Using the result above, this is rewritten as
Now as of $t = T - 1$, the problem is:

\[
\text{Maximize} [r(x_{T-1}, u_{T-1}) + \beta r(x_T, u_T)]
\]

s.t.

\[
x_T = g(x_{T-1}, u_{T-1})
\]

Using the result above, this is rewritten as:

\[
\text{Maximize} [r(x_{T-1}, u_{T-1}) + \beta V_T(x_T)]
\]

s.t.

\[
x_T = g(x_{T-1}, u_{T-1})
\]
which is further rewritten as:

\[
\text{Maximize } \left[ r(x_T, u_T) + \beta V_T(g(x_T, u_T)) \right]
\]

with \( x_T \) given. This is again a simple maximization w.r.t. \( u_T \) only. We again express the solution in terms of the state variable:

\[
u_T = h_T(x_T)
\]

Plugging back this into the maximand above to get:

\[
V_T(x_T) = r(x_T, h_T(x_T)) + \beta V_T(g(x_T, h_T(x_T)))
\]

From \( t = T_2 \) on, we can repeat the same maximization, substitution processes back to the original problem.
which is further rewritten as:

\[
\text{Maximize}[r(x_{T-1}, u_{T-1}) + \beta V_T(g(x_{T-1}, u_{T-1}))]
\]

with \( x_{T-1} \) given
which is further rewritten as:

\[
Maximize[r(x_{T-1}, u_{T-1}) + \beta V_T(g(x_{T-1}, u_{T-1}))]
\]
with \(x_{T-1}\) given

This is again a simple maximization w.r.t. \(u_{T-1}\) only. We again express the solution in terms of the state variable:
which is further rewritten as:

\[ \text{Maximize} [r(x_{T-1}, u_{T-1}) + \beta V_T(g(x_{T-1}, u_{T-1}))] \]

with \( x_{T-1} \) given

This is again a simple maximization w.r.t. \( u_{T-1} \) only. We again express the solution in terms of the state variable:

\[ u_{T-1} = h_{T-1}(x_{T-1}) \]
which is further rewritten as:

$$\text{Maximize}[r(x_{T-1}, u_{T-1}) + \beta V_T(g(x_{T-1}, u_{T-1}))]$$

with $x_{T-1}$ given

This is again a simple maximization w.r.t. $u_{T-1}$ only. We again express the solution in terms of the state variable:

$$u_{T-1} = h_{T-1}(x_{T-1})$$

Plugging back this into the maximand above to get:
which is further rewritten as:

$$\text{Maximize}[r(x_{T-1}, u_{T-1}) + \beta V_T(g(x_{T-1}, u_{T-1}))]$$

with $x_{T-1}$ given

This is again a simple maximization w.r.t. $u_{T-1}$ only. We again express the solution in terms of the state variable:

$$u_{T-1} = h_{T-1}(x_{T-1})$$

Plugging back this into the maximand above to get:

$$V_{T-1}(x_{T-1}) = r(x_{T-1}, h_{T-1}(x_{T-1})) + \beta V_T(g(x_{T-1}, h_{T-1}(x_{T-1})))$$
which is further rewritten as:

\[
\text{Maximize}\left[r(x_{T-1}, u_{T-1}) + \beta V_T(g(x_{T-1}, u_{T-1}))\right]
\]

with \(x_{T-1}\) given

This is again a simple maximization w.r.t. \(u_{T-1}\) only. We again express the solution in terms of the state variable:

\[
u_{T-1} = h_{T-1}(x_{T-1})
\]

Plugging back this into the maximand above to get:

\[
V_{T-1}(x_{T-1}) = r(x_{T-1}, h_{T-1}(x_{T-1})) + \beta V_T(g(x_{T-1}, h_{T-1}(x_{T-1})))
\]

From \(t = T - 2\) on, we can repeat the same maximization, substitution processes back to the original problem.
The optimal solution is given by $T$ functions that determine the entire sequence of the control variable.
The optimal solution is given by $T$ functions that determine the entire sequence of the control variable.

$$u_t = h_t(x_t), \quad t = 1, 2, \ldots, T$$
The optimal solution is given by $T$ functions that determine the entire sequence of the control variable.

$$u_t = h_t(x_t), \ t = 1, 2, \ldots T$$

Although the backward induction gives us the solution, such a strategy has several drawbacks: first of all, the way that the control variable depends upon the state variable, i.e., $u_t = h_t(x_t)$, is clearly not invariant across time and the resulting solutions are often quite messy and becomes rapidly intractable.
The optimal solution is given by \( T \) functions that determine the entire sequence of the control variable.

\[
u_t = h_t(x_t), \ t = 1, 2, \ldots T
\]

Although the backward induction gives us the solution, such a strategy has several drawbacks: first of all, the way that the control variable depends upon the state variable, i.e., \( u_t = h_t(x_t) \), is clearly not invariant across time and the resulting solutions are often quite messy and becomes rapidly intractable.

This makes sense because in the finite horizon problem, the calendar time matters for the optimization since the remaining time horizon shrinks over time. Direct dependence of the solution on time is sometimes problematic also in empirical implementation.
Let us see how it works in a simple example.
Example

Let us see how it works in a simple example.

\[ V \equiv \text{Maximize} \sum_{t=0}^{T} \beta^t \log(c_t) \text{s.t.} \]

\[ k_0 \text{ given} \]

\[ k_{t+1} = A k_t^\alpha - c_t \]
Let us see how it works in a simple example.

\[ V \equiv \text{Maximize} \sum_{t=0}^{T} \beta^t \log(c_t) \text{s.t.} \]

\[ k_0 \text{ given} \]

\[ k_{t+1} = A k_t^\alpha - c_t \]

At \( t = T \), we trivially have
Example

- Let us see how it works in a simple example.

\[ V \equiv \text{Maximize } \sum_{t=0}^{T} \beta^t \log(c_t) \text{s.t.} \]

\[ k_0 \text{ given} \]

\[ k_{t+1} = A k_t^\alpha - c_t \]

- At \( t = T \), we trivially have

\[ c_T = A k_T^\alpha \]

\[ V_T = \log A + \alpha \log(k_T) \]
Let us see how it works in a simple example.

\[ V \equiv \text{Maximize } \sum_{t=0}^{T} \beta^t \log(c_t) \text{s.t. } \]
\[ k_0 \text{ given} \]
\[ k_{t+1} = Ak_t^\alpha - c_t \]

At \( t = T \), we trivially have

\[ c_T = Ak_T^\alpha \]
\[ V_T = \log A + \alpha \log(k_T) \]

At \( t = T - 1 \), we have
Example

- Let us see how it works in a simple example.

\[
 V \equiv \text{Maximize} \sum_{t=0}^{T} \beta^t \log(c_t) \text{s.t.} \\
 k_0 \text{ given} \\
 k_{t+1} = Ak_t^\alpha - c_t
\]

- At \( t = T \), we trivially have

\[
 c_T = Ak_T^\alpha \\
 V_T = \log A + \alpha \log(k_T)
\]

- At \( t = T - 1 \), we have

\[
 V_{T-1}(k_{T-1}) = \log(c_{T-1}) + \beta[\log A + \alpha \log(Ak_{T-1}^\alpha - c_{T-1})]
\]
Maximization yields:

\[ \max c^T \alpha \beta \]

Plugging back this into the maximand, we get:
Maximization yields:

\[
\frac{1}{c_{T-1}} - \frac{\alpha \beta}{A k_T^\alpha - c_{T-1}} = 0
\]

\[
c_{T-1} = \frac{A}{1 + \alpha \beta} k_T^\alpha
\]
Example 2

Maximization yields:

\[
\frac{1}{c_{T-1}} - \frac{\alpha \beta}{Ak_{T-1}^\alpha - c_{T-1}} = 0
\]

\[
c_{T-1} = \frac{A}{1 + \alpha \beta} k_{T-1}^\alpha
\]

Plugging back this into the maximand, we get:
Example 3

\[ V_{T-1}(k_{T-1}) = \log\left(\frac{A}{1 + \alpha\beta} k_{T-1}^{\alpha}\right) \]

\[ + \beta \left[ \log A + \alpha \log (A k_{T-1}^{\alpha} - \frac{A}{1 + \alpha\beta} k_{T-1}^{\alpha}) \right] \]

\[ = \log\left(\frac{A}{1 + \alpha\beta}\right) \]

\[ + \alpha\beta \log\left(\frac{A\alpha\beta}{1 + \alpha\beta}\right) + \alpha(1 + \alpha\beta) \log(k_{T-1}) \]
Example 4

- To get an idea how the dependence of control on time looks, try one more period:

\[
V(T^2(kT^2)) = \log(cT^2) + \beta \left[ \log(A_1 + \alpha \beta) + \alpha \beta \log(A_\alpha \beta + \alpha \beta) + \alpha (1 + \alpha \beta) \log(A_{kT^2}cT^2) \right]
\]

The focus is:

\[
cT^2 = \alpha \beta (1 + \alpha \beta) A_{\alpha T_1} cT^2 = 0
\]

Now we can guess (correctly!)
Example 4

To get an idea how the dependence of control on time looks, try one more period:

\[ V_{T-2}(k_{T-2}) = \log(c_{T-2}) \]
\[ + \beta[\log\left(\frac{A}{1 + \alpha\beta}\right) + \alpha\beta \log\left(\frac{A\alpha\beta}{1 + \alpha\beta}\right) + \alpha(1 + \alpha\beta) \log(A) \]
Example 4

To get an idea how the dependence of control on time looks, try one more period:

\[ V_{T-2}(k_{T-2}) = \log(c_{T-2}) \]

\[ + \beta [\log\left(\frac{A}{1 + \alpha \beta}\right) + \alpha \beta \log\left(\frac{A \alpha \beta}{1 + \alpha \beta}\right) + \alpha (1 + \alpha \beta) \log(Ak_{T-1})] \]

The foc is:
Example 4

- To get an idea how the dependence of control on time looks, try one more period:

\[ V_{T-2}(k_{T-2}) = \log(c_{T-2}) \]

\[ + \beta \log\left(\frac{A}{1 + \alpha \beta}\right) + \alpha \beta \log\left(\frac{A\alpha \beta}{1 + \alpha \beta}\right) + \alpha (1 + \alpha \beta) \log(\ldots) \]

- The foc is:

\[
\frac{1}{c_{T-2}} - \frac{\alpha \beta (1 + \alpha \beta)}{Ak_{T-1}^{\alpha} - c_{T-2}} = 0
\]

\[ c_{T-2} = \frac{A}{1 + \alpha \beta (1 + \alpha \beta)} k_{T-1}^{\alpha} \]
Example 4

To get an idea how the dependence of control on time looks, try one more period:

\[ V_{T-2}(k_{T-2}) = \log(c_{T-2}) \]
\[ + \beta \log\left( \frac{A}{1 + \alpha \beta} \right) + \alpha \beta \log\left( \frac{A\alpha \beta}{1 + \alpha \beta} \right) + \alpha(1 + \alpha \beta) \log(c_{T-2}) \]

The foc is:

\[ \frac{1}{c_{T-2}} - \frac{\alpha \beta(1 + \alpha \beta)}{Ak_{T-1}^\alpha - c_{T-2}} = 0 \]
\[ c_{T-2} = \frac{A}{1 + \alpha \beta(1 + \alpha \beta)} k_{T-1}^\alpha \]

Now we can guess (correctly!)
Example 5

\[ c_{T-j} = \frac{A}{1 + \alpha \beta + (\alpha \beta)^2 + \ldots + (\alpha \beta)^j} k_{T-j}^\alpha \]
\[ c_{T-j} = \frac{A}{1 + \alpha \beta + (\alpha \beta)^2 + \ldots + (\alpha \beta)^j} k^\alpha_{T-j} \]

And, more importantly, we can guess (again correctly!)
Example 5

\[ c_{T-j} = \frac{A}{1 + \alpha \beta + (\alpha \beta)^2 + \ldots + (\alpha \beta)^j} k^\alpha_{T-j} \]

And, more importantly, we can guess (again correctly!)

\[ \lim_{T,j \to \infty} c_{T-j} = \frac{A}{1 + \frac{1}{1 - \alpha \beta}} k^\alpha_{T-j} \]
In general, though, this example is misleading in the sense that the recursive structure in the finite horizon DP may not always generate simply general solution formulae as we have in the example. On the other hand, the recursive structure is extremely convenient in programming using MATLAB or any other computer language.

As the example above suggests, under a certain set of conditions, the infinite horizon DP has a particularly desirable property that the optimal control, $u_t$, is a unique and time invariant function of $k_t$, the state variable:

$$u_t = h(x_t)$$

This makes sense, because, after all, in an infinite horizon DP, the calendar time should not affect the optimal policy because, tomorrow will look just like today or yesterday as far as the initial condition, i.e., the state variable, is the same.

If the dynamics of the state variable also has time invariant property, i.e., $g_t(x_t, u_t) = g(x_t, u_t)$, then the entire structure of the DP is time invariant. This suggests that the maximand
Infinite Horizon DP

- In general, though, this example is misleading in the sense that the recursive structure in the finite horizon DP may not always generate simply general solution formulae as we have in the example. On the other hand, the recursive structure is extremely convenient in programming using MATLAB or any other computer language.

- As the example above suggests, under a certain set of conditions, the infinite horizon DP has a particularly desirable property that the optimal control, \( u_t \), is a unique and time invariant function of \( k_t \), the state variable: \( u_t = h(x_t) \)
In general, though, this example is misleading in the sense that the recursive structure in the finite horizon DP may not always generate simply general solution formulae as we have in the example. On the other hand, the recursive structure is extremely convenient in programming using MATLAB or any other computer language.

As the example above suggests, under a certain set of conditions, the infinite horizon DP has a particularly desirable property that the optimal control, $u_t$, is a unique and time invariant function of $k_t$, the state variable: $u_t = h(x_t)$

This makes sense, because, after all, in an infinite horizon DP, the calendar time should not affect the optimal policy because, tomorrow will look just like today or yesterday as far as the initial condition, i.e., the state variable, is the same.
Infinite Horizon DP

- In general, though, this example is misleading in the sense that the recursive structure in the finite horizon DP may not always generate simply general solution formulae as we have in the example. On the other hand, the recursive structure is extremely convenient in programming using MATLAB or any other computer language.

- As the example above suggests, under a certain set of conditions, the infinite horizon DP has a particularly desirable property that the optimal control, $u_t$, is a unique and time invariant function of $k_t$, the state variable: $u_t = h(x_t)$

- This makes sense, because, after all, in an infinite horizon DP, the calendar time should not affect the optimal policy because, tomorrow will look just like today or yesterday as far as the initial condition, i.e., the state variable, is the same.

- If the dynamics of the state variable also has time invariant property, i.e., $g_t(x_t, u_t) = g(x_t, u_t)$, then the entire structure of the DP is time invariant. This suggests that the maximand
\[
\max_{\{u_s\}_{s=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t r(u_t, x_t)
\]

s.t.

\[
x_{t+1} = g(x_t, u_t)
\]
\[
\max_{\{u_s\}_{s=0}^\infty} \sum_{t=0}^\infty \beta^t r(u_t, x_t)
\]

s.t.

\[x_{t+1} = g(x_t, u_t)\]

can be written as a function of the initial condition, \(x_0\):
Infinite Horizon DP 2

\[ \max_{\{u_s\}_{s=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t r(u_t, x_t) \]

s.t.

\[ x_{t+1} = g(x_t, u_t) \]

can be written as a function of the initial condition, \( x_0 \):

\[ V(x_0) = \max_{\{u_s\}_{s=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t r(u_t, x_t) \]
Now we know: [Bellman Principle]
Now we know: [Bellman Principle]

\[ V(x_0) = \max_{u_s} \sum_{t=0}^{\infty} \beta^t r(u_t, x_t) = r(u_0, x_0) + \beta \max_{u_s} \sum_{t'=0}^{\infty} \beta^{t'} r(u_{t'+1}, x_{t'+1}) = r(u_0, x_0) + \beta V(x_1) \]

\[ x_1 = g(x_0, u_0) \]
Now we know: [Bellman Principle]

\[
V(x_0) = \max_{\{u_s\}_{s=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t r(u_t, x_t) = r(u_0, x_0) + \beta \max_{\{u_s\}_{s=0}^{\infty}} \sum_{t'=0}^{\infty} \beta^{t'} r(u_{t'+1}, x_{t'+1})
\]

\[
= r(u_0, x_0) + \beta V(x_1)
\]

\[
x_1 = g(x_0, u_0)
\]

This property holds for any calendar time. Hence we have
Now we know: [Bellman Principle]

\[ V(x_0) = \max_{\{u_s\}_{s=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t r(u_t, x_t) = r(u_0, x_0) + \beta \max_{\{u_s\}_{s=0}^{\infty}} \sum_{t'=0}^{\infty} \beta^{t'} r(u_{t'+1}, x_{t'+1}) \]

\[ = r(u_0, x_0) + \beta V(x_1) \]

\[ x_1 = g(x_0, u_0) \]

This property holds for any calendar time. Hence we have

\[ V(x) = \max_u [r(x, u) + \beta V(x')] \]

\[ x' = g(x, u) \]
Now we know: [Bellman Principle]

$$V(x_0) = \max \sum_{t=0}^{\infty} \beta^t r(u_t, x_t) = r(u_0, x_0) + \beta \max \sum_{t'=0}^{\infty} \beta^{t'} r(u_{t'+1}, x_{t'+1})$$

$$= r(u_0, x_0) + \beta V(x_1)$$

$$x_1 = g(x_0, u_0)$$

This property holds for any calendar time. Hence we have

$$V(x) = \max_u [r(x, u) + \beta V(x')]$$

$$x' = g(x, u)$$

Inserting the constraint, we get [Bellman Equation]:

$$V(x) = \max_u [r(x, u) + \beta V(x')]$$

$$x' = g(x, u)$$
Bellman Equation

Now we know: [Bellman Principle]

\[ V(x_0) = \max_{\{u_s\}_{s=0}^\infty} \sum_{t=0}^{\infty} \beta^t r(u_t, x_t) = r(u_0, x_0) + \beta \max_{\{u_s\}_{s=0}^\infty} \sum_{t'=0}^{\infty} \beta^{t'} r(u_{t'+1}, x_{t'+1}) \]

\[ = r(u_0, x_0) + \beta V(x_1) \]

\[ x_1 = g(x_0, u_0) \]

This property holds for any calendar time. Hence we have

\[ V(x) = \max_u [r(x, u) + \beta V(g(x, u))] \]

Inserting the constraint, we get [Bellman Equation]:

\[ V(x) = \max_u [r(x, u) + \beta V(g(x, u))] \]
By now it is clear that the optimal policy for control variable is represented also by a time invariant policy function, $h(x)$. 

\begin{equation}
V(x) = r(x, h(x)) + \beta V(g(x, h(x)))
\end{equation} (* is a Functional equation)

Obviously, $V(x)$ is unknown and there exists no miracle formulae to find one. Notice the property above does not depend in any way upon whether or not $h(x)$ is obtained by a conventional first order condition(s). It does require, however, that $h(x)$ is the solution for the single period maximization problem, given $V(x)$ and $g(x, u)$. 

(Institute)
By now it is clear that the optimal policy for control variable is represented also by a time invariant policy function, $h(x)$.

Hence we have:

$$V(x) = r(x, h(x)) + \beta V(g(x, h(x))) \quad (*)$$
By now it is clear that the optimal policy for control variable is represented also by a time invariant policy function, \( h(x) \).

Hence we have:

\[
V(x) \equiv r(x, h(x)) + \beta V(g(x, h(x))) \quad (*)
\]
By now it is clear that the optimal policy for control variable is represented also by a time invariant policy function, $h(x)$.

Hence we have:

$$V(x) \equiv r(x, h(x)) + \beta V(g(x, h(x))) \quad (*)$$

In a nutshell, infinite horizon DP is reduced to find a pair of functions, $V(x), h(x)$ which jointly satisfy the above equation. [$(*)$ is a Functional equation].
By now it is clear that the optimal policy for control variable is represented also by a time invariant policy function, $h(x)$.

Hence we have:

$$V(x) = r(x, h(x)) + \beta V(g(x, h(x)))$$  \hfill (*)

In a nutshell, infinite horizon DP is reduced to find a pair of functions, $V(x), h(x)$ which jointly satisfy the above equation. \hfill ([(*]) is a Functional equation).

Obviously, $V(x)$ is unknown and there exists no miracle formulae to find one.
By now it is clear that the optimal policy for control variable is represented also by a time invariant policy function, \( h(x) \).

Hence we have:

\[
V(x) \equiv r(x, h(x)) + \beta V(g(x, h(x)))
\] (*)

In a nutshell, infinite horizon DP is reduced to find a pair of functions, \( V(x), h(x) \) which jointly satisfy the above equation. [(*) is a Functional equation].

Obviously, \( V(x) \) is unknown and there exists no miracle formulae to find one.

Notice the property above does not depend in any way upon whether or not \( h(x) \) is obtained by a conventional first order condition(s). It does require, however, that \( h(x) \) is the solution for the single period maximization problem, given \( V(x) \) and \( g(x, u) \).
Unfortunately, except for some very special cases, neither $h(x)$ or $V(x)$ has closed form solution.
Unfortunately, except for some very special cases, neither $h(x)$ or $V(x)$ has closed form solution.

This should not be surprising, knowing that Maximum Principle does not have such for the identical maximization problem.
Unfortunately, except for some very special cases, neither \( h(x) \) or \( V(x) \) has closed form solution.

This should not be surprising, knowing that Maximum Principle does not have such for the identical maximization problem.

BUT, the great advantage of DP is that it lends itself easily to numerical approximations of the solution.
Unfortunately, except for some very special cases, neither $h(x)$ or $V(x)$ has closed form solution.

This should not be surprising, knowing that Maximum Principle does not have such for the identical maximization problem.

BUT, the great advantage of DP is that it lends itself easily to numerical approximations of the solution.

In particular, starting from any candidate value function, say, $v_0(x)$, you can get the sequence of solution:
1. Obtain $h_0(x)$ using $v_0(x)$: to do that, simply choose $h_0(x)$ to maximize

$$h_0(x) = \arg\max \left[ r(x, h_0(x)) + \beta V_0(g(x, h_0(x))) \right]$$

2. Use $h_0(x)$ and compute:

$$v_1(x) = r(x, h_0(x)) + \beta V_0(g(x, h_0(x)))$$

3. Obtain $h_1(x)$ using $v_1(x)$

4. Repeat the process to obtain $v_2(x), v_3(x), ..$
Bellman Equation 5

- Under certain set of conditions (which are relatively mild), the sequence \( \{v_j(x)\} \) converges to the true maximand, \( V(x) \).
  
  [Contraction Mapping].
Under certain set of conditions (which are relatively mild), the sequence \( \{v_j(x)\} \) converges to the true maximand, \( V(x) \). [Contraction Mapping].

I.e., consider a properly defined space in which a candidate function \( v(x) \) resides. Start with a candidate \( v_0(x) \).
Under certain set of conditions (which are relatively mild), the sequence \( \{ v_j(x) \} \) converges to the true maximand, \( V(x) \). [Contraction Mapping].

I.e., consider a properly defined space in which a candidate function \( v(x) \) resides. Start with a candidate \( v_0(x) \).

Then, the mapping defined by the procedure above, say, \( T v_0(x) \) generates \( v_1(x) \). The mapping is a contraction if the 'distance',

\[
\| T v_0(x) - v_0(x) \| 
\]
Under certain set of conditions (which are relatively mild), the sequence \( \{v_j(x)\} \) converges to the true maximand, \( V(x) \).

**[Contraction Mapping].**

I.e., consider a properly defined space in which a candidate function \( v(x) \) resides. Start with a candidate \( v_0(x) \).

Then, the mapping defined by the procedure above, say, \( Tv_0(x) \) generates \( v_1(x) \). The mapping is a contraction if the 'distance',

\[
\| Tv_0(x) - v_0(x) \|
\]

shrinks, so that it eventually converges to a unique fixed point where

\[
Tv(x) = v(x)
\]
Under certain set of conditions (which are relatively mild), the sequence \( \{v_j(x)\} \) converges to the true maximand, \( V(x) \).

[Contraction Mapping].

I.e., consider a properly defined space in which a candidate function \( v(x) \) resides. Start with a candidate \( v_0(x) \).

Then, the mapping defined by the procedure above, say, \( T v_0(x) \) generates \( v_1(x) \). The mapping is a contraction if the 'distance',

\[
\| T v_0(x) - v_0(x) \|
\]

shrinks, so that it eventually converges to a unique fixed point where

\[
T v(x) = v(x)
\]

Because of this property, we can use the brute force (of computer) to solve just about any well defined DP with any degree of accuracy, assuming we have sufficient (computer) time.
If we can somehow exclude corner solutions, we obtain full differentiation of the Bellman Equation to get

\[
\frac{\partial V_0(x)}{\partial x} = \frac{\partial r(x, h(x))}{\partial x} + \beta \frac{\partial g(x, h(x))}{\partial x} V_0(g(x, h(x)))
\]

In particular, if

\[
g(x, h(x)) = g(h(x))
\]

[transition equation does not depend directly upon x], we get Benveniste-Scheinkman formula

\[
V_0(x) = \frac{\partial r(x, h(x))}{\partial x}
\]

Using this, we get [Euler Equation]

\[
\frac{\partial r(x_t, u_t)}{\partial u_t} + \frac{\partial g(u_t)}{\partial u_t} \frac{\partial r(x_{t+1}, u_{t+1})}{\partial x_t} + 1 = 0
\]
If we can somehow exclude corner solutions, we obtain full differentiation of the Bellman Equation to get:

$$V'(x) = \frac{\partial r(x, h(x))}{\partial x} + \beta \frac{\partial g(x, h(x))}{\partial x} V'(g(x, h(x)))$$
Euler Equation

If we can somehow exclude corner solutions, we obtain full differentiation of the Bellman Equation to get

\[ V'(x) = \frac{\partial r(x, h(x))}{\partial x} + \beta \frac{\partial g(x, h(x))}{\partial x} V'(g(x, h(x))) \]

In particular, if \( g(x, h(x)) = g(h(x)) \) [transition equation does not depend directly upon \( x \)], we get Benveniste-Scheinkman formula
If we can somehow exclude corner solutions, we obtain full differentiation of the Bellman Equation to get

\[ V'(x) = \frac{\partial r(x, h(x))}{\partial x} + \beta \frac{\partial g(x, h(x))}{\partial x} V'(g(x, h(x))) \]

In particular, if \( g(x, h(x)) = g(h(x)) \) [transition equation does not depend directly upon \( x \)], we get Benveniste-Scheinkman formula

\[ V'(x) = \frac{\partial r(x, h(x))}{\partial x} \]
If we can somehow exclude corner solutions, we obtain full differentiation of the Bellman Equation to get

\[ V'(x) = \frac{\partial r(x, h(x))}{\partial x} + \beta \frac{\partial g(x, h(x))}{\partial x} V'(g(x, h(x))) \]

In particular, if \( g(x, h(x)) = g(h(x)) \) [transition equation does not depend directly upon \( x \)], we get Benveniste-Scheinkman formula

\[ V'(x) = \frac{\partial r(x, h(x))}{\partial x} \]

Using this, we get [Euler Equation]
Euler Equation

- If we can somehow exclude corner solutions, we obtain full differentiation of the Bellman Equation to get

\[ V'(x) = \frac{\partial r(x, h(x))}{\partial x} + \beta \frac{\partial g(x, h(x))}{\partial x} V'(g(x, h(x))) \]

- In particular, if \( g(x, h(x)) = g(h(x)) \) [transition equation does not depend directly upon \( x \)], we get Benveniste-Scheinkman formula

\[ V'(x) = \frac{\partial r(x, h(x))}{\partial x} \]

- Using this, we get [Euler Equation]

\[ \frac{\partial r(x_t, u_t)}{\partial u_t} + \frac{\partial g(u_t)}{\partial u_t} \frac{\partial r(x_{t+1}, u_{t+1})}{\partial x_{t+1}} = 0 \]
Consider, for example, a simple optimal consumption problem:
Consider, for example, a simple optimal consumption problem:

\[
\max_{\{u_s\}_{s=0}^\infty} \sum_{t=0}^\infty \beta^t u(c_t)
\]

\[
k_{t+1} = f(k_t) - c_t
\]
Consider, for example, a simple optimal consumption problem:

\[
\max_{\{u_s\}_{s=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u(c_t)
\]

\[
k_{t+1} = f(k_t) - c_t
\]

The Bellman equation for this problem is
Consider, for example, a simple optimal consumption problem:

\[
\max_{\{u_s\}_{s=0}^\infty} \sum_{t=0}^\infty \beta^t u(c_t)
\]

\[k_{t+1} = f(k_t) - c_t\]

The Bellman equation for this problem is

\[
V(k_t) = u(c_t) + \beta V(k_{t+1})
\]

\[= u[f(k_t) - k_{t+1}] + \beta V(k_{t+1})\]
Consider, for example, a simple optimal consumption problem:

\[
\max_{\{u_s\}_{s=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u(c_t)
\]

\[k_{t+1} = f(k_t) - c_t\]

The Bellman equation for this problem is

\[V(k_t) = u(c_t) + \beta V(k_{t+1})\]

\[= u[f(k_t) - k_{t+1}] + \beta V(k_{t+1})\]

We have
Consider, for example, a simple optimal consumption problem:

\[
\max \left\{ u_s \right\}_{s=0}^{\infty} \sum_{t=0}^{\infty} \beta^t u(c_t)
\]

\[
k_{t+1} = f(k_t) - c_t
\]

The Bellman equation for this problem is

\[
V(k_t) = u(c_t) + \beta V(k_{t+1})
\]

\[
= u[f(k_t) - k_{t+1}] + \beta V(k_{t+1})
\]

We have

\[
V(k) = \max_{k'} [u(f(k) - k') + \beta V(k')]
\]
Euler Equation 3

- Benveniste-Scheinkman formula is

\[ V_0(k) = u_0(f(k)k_0) f_0(k_0) \]

The focus for \( k_0 \) is

\[ u_0(f(k)k_0) + \beta V_0(k_0) = 0 \]

Thus we get:

\[ u_0(f(k)k_0) = \beta u_0(f(k_0)k_0) f_0(k_0) \]
Euler Equation 3

- Benveniste-Scheinkman formula is

\[ V'(k) = u'(f(k) - k')f'(k) \]
Benveniste-Scheinkman formula is

\[ V'(k) = u'(f(k) - k')f'(k) \]

The foc for \( k' \) is
Benveniste-Scheinkman formula is

\[ V'(k) = u'(f(k) - k')f'(k) \]

The foc for \( k' \) is

\[-u'(f(k) - k') + \beta V'(k') = 0\]
Benveniste-Scheinkman formula is

\[ V'(k) = u'(f(k) - k')f'(k) \]

The foc for \( k' \) is

\[ -u'(f(k) - k') + \beta V'(k') = 0 \]

Thus we get:
Euler Equation 3

- Benveniste-Scheinkman formula is

\[ V'(k) = u'(f(k) - k')f'(k) \]

- The foc for \( k' \) is

\[ -u'(f(k) - k') + \beta V'(k') = 0 \]

- Thus we get:

\[ u'(f(k) - k') = \beta u'(f(k') - k')f'(k') \]
Euler Equation 4

Or,
Euler Equation 4

Or,

\[ u'(c_t) = \beta u'(c_{t+1}) f'(k_{t+1}) \]

\[ 1 = \frac{\beta u'(c_{t+1})}{u'(c_t)} f'(k_{t+1}) \]
Euler Equation 4

- Or,

\[ u'(c_t) = \beta u'(c_{t+1}) f'(k_{t+1}) \]

\[ 1 = \frac{\beta u'(c_{t+1})}{u'(c_t)} f'(k_{t+1}) \]

- MRS between today and tomorrow consumption is equal to the MPK, or natural rate of interest rate plus one.
Euler Equation 4

- Or,

\[
\begin{align*}
  u'(c_t) &= \beta u'(c_{t+1}) f'(k_{t+1}) \\
  1 &= \frac{\beta u'(c_{t+1})}{u'(c_t)} f'(k_{t+1})
\end{align*}
\]

- MRS between today and tomorrow consumption is equal to the MPK, or natural rate of interest rate plus one.

- Needless to say, this condition can be obtained also using Maximum Principle.
The example is now extended to cover uncertainty. The most common and easiest one to handle is additive uncertainty on the state transition function:

\[ g(x, u) = b + \epsilon \]

wherein \( \epsilon \) is a mean zero, finite variance iid. Denote by \( f \) the distribution of \( \epsilon \). The Bellman equation now becomes

\[ V(W, \epsilon) = u(c) + \beta E[V(W_{c+}, \epsilon)] \]

where in the expectation is over \( f \). But then, there is not much we can do on this,... indicating that analytical power of DP is rather limited unless we resort to the numerical solution or simulation.
The example is now extended to cover uncertainty. The most common and easiest one to handle is additive uncertainty on the state transition function:

\[ g = \hat{g}(x, u) + \epsilon \]
The example is now extended to cover uncertainty. The most common and easiest one to handle is additive uncertainty on the state transition function:

\[ g = \hat{g}(x, u) + \epsilon \]

wherein \( \epsilon \) is a mean zero finite variance iid. Denote by \( f \) the distribution of \( \epsilon \).
The example is now extended to cover uncertainty. The most common and easiest one to handle is additive uncertainty on the state transition function:

\[ g = \hat{g}(x, u) + \epsilon \]

wherein \( \epsilon \) is a mean zero finite variance iid. Denote by \( f \) the distribution of \( \epsilon \).

The Bellman equation now becomes

\[ V(W, \epsilon) = u(c) + \beta E [V(W - c, \epsilon) | \epsilon] \]
Uncertainty

- The example is now extended to cover uncertainty. The most common and easiest one to handle is additive uncertainty on the state transition function:

\[ g = \hat{g}(x, u) + \epsilon \]

- wherein \( \epsilon \) is a mean zero finite variance iid. Denote by \( f \) the distribution of \( \epsilon \).

- The Bellman equation now becomes

\[ V(W, \epsilon) = u(c) + \beta E \left[ V(W - c, \epsilon) | \epsilon \right] \]

- where in the expectation is over \( f \). But then, there is not much we can do on this,... indicating that analytical power of DP is rather limited unless we resort to the numerical solution or simulation.