Econ 260A: Dynamic Programming for Natural Resource Economics

Christopher Costello

October 3, 2018

1 Introduction

If your research is in natural resources, the environment, or macroeconomics, you are intimately familiar with inter-temporal allocation and the analytical challenges it poses. Classic examples include optimal growth, renewable resources, inventory, stock pollutants, etc. Often problems that are seemingly intractable analytically can be solved using dynamic programming. These notes have two objectives: First, I describe how to frame a discrete-time dynamic optimization problem as a dynamic programming problem, and show some basic analytical methods for analytically solving such problems. Second, because analytical solutions are often elusive, I introduce numerical approaches to solving dynamic programming problems.

2 Framing a problem as a Dynamic Programming Equation

Most discrete-time natural resource optimization problems can be intuitively written as a "Dynamic Programming Equation" or "Bellman Equation." Consider the following dynamic optimization problem:

$$\max_{\{u_t\}} \textstyle\sum_{t=0}^{T} \delta^t \pi(x_t, u_t) \tag{1}$$

$$\text{s.t. } x_{t+1} = G(x_t, u_t) \tag{2}$$

Here δ is the discount factor,[1] u_t is the control variable (or vector) in period t, x_t is the state variable (or vector), $\pi(x, u)$ is the current period payoff conditional

[1] If the discount rate is r, then the discount factor is $\delta \equiv \frac{1}{1+r}$.

on the current state and control, and the function $G(x, u)$ defines the equation of motion (Equation 2). Let's take a minute to think about the functions $\pi(x, u)$ and $G(x, u)$ for two classic problems:

- Renewable Resources
  - $\pi(x, u)$: Profit or utility of harvesting the resource. Depends on x because of a "stock effect". Depends on u because utility depends on how much is harvested.
  - $G(x, u)$: Growth of the resource depends on the current stock (so-called "density-dependence") and depends on how much is harvested.
- Pollution Problems
  - $\pi(x, u)$: Profit or utility may depend on the stock of pollution (carbon in atmosphere) and on the amount of pollution emitted this year.
  - $G(x, u)$: Growth in the pollution stock depends on the current pollution stock and on how much is added to the pollution stock.

The time horizon is finite T, but could also be infinite. The dynamic programming equation has a value function that is conditional on both time and the state. The DPE for this problem is:

$$\begin{aligned} V_t(x_t) &= \max_{u_t} [\pi(x_t, u_t) + \delta V_{t+1}(x_{t+1})] &(3) \\ &= \max_{u_t} [\pi(x_t, u_t) + \delta V_{t+1}(G(x_t, u_t))] &(4) \end{aligned}$$

Clearly, if the function $V_{t+1}(x)$ were known, this problem would be trivial to solve for the optimal $u_t^*(x_t)$, i.e. by simply maximization. Our problem is that we do not know the function $V_{t+1}(x)$.

- Note the interpretation of $V_t(x)$: It is the economic value, in period t, of having stock, x, conditional upon using that stock optimally in the future. This is closely related to the shadow value, which is the economic value of a marginal increment in stock, conditional upon optimal use (and is thus $\frac{dV_t(x)}{dx}$).
- Note also that, in general, the optimal control (u_t^*) will depend on the current state, x_t.
- Note that the state (x_t) and/or the control (u_t) could be vectors

2.1 Example: Renewable Resource Harvesting

The stock of a renewable resource at the beginning of period t is x_t. Harvest during period t is h_t. The period-t payoff from harvesting is h_t. The resource grows according to: $x_{t+1} = f(x_t - h_t)$. The initial stock is x_0.

1. What is the state variable for this problem? (The period-t state is x_t.)
2. What is the control variable for this problem? (The period-t control is h_t.)
3. What is the DPE?

$$\begin{aligned} V_t(x_t) &= \max_{h_t} \left[h_t + \delta V_{t+1}\left(x_{t+1}\right)\right] &(5) \\ &= \max_{h_t} \left[h_t + \delta V_{t+1}\left(f(x_t - h_t)\right)\right] &(6) \end{aligned}$$

4. What is the function $V_t(x_t)$? (It is the value of entering period-t with stock x_t and optimally using that stock in the future. But we do not immediately know the functional form of $V(x)$; it must be derived.)
5. If we knew the functional form of $V_{t+1}(x_{t+1})$, what would be the optimal choice of h_t? (If we knew $V_{t+1}(x_{t+1})$, we could simply take the first order condition, which would yield:

$$1 = \delta \frac{dV_{t+1}}{dx_{t+1}} \frac{dx_{t+1}}{d(x_t - h_t)} \tag{7}$$

Which implies that in period-t we should harvest to the point where the value of the last unit of harvest (which is 1) equals the discounted value of the additional growth we would have received in the future from that last unit. This is a kind of inter-temporal arbitrage or equi-marginal principle.

3 Analytical Solutions: Backward Induction

While writing down the DPE and taking the first order condition allows us to interpret the marginal conditions for optimality, we still do not have an explicit solution for how much to harvest. This is because we do not know the form of $V(x)$. One way to obtain the explicit solution to a dynamic programming problem is to use backward induction. While this approach will always work in principle, it only occasionally works in practice. That's because the form of $V(x)$ often gets analytically unwieldy as one moves backward in time. Fortunately, for this problem, we can get analytical traction. Here's how to do backward-induction:

Suppose this is a finite horizon where T is the final date, after which there is no value, so $V_{T+1}(x_{T+1}) = 0$ (you should think about how this would work if, instead, you had a salvage value at time $T+1$). The period-T DPE is:

$$\begin{aligned} V_T(x_T) &= \max_{h_T} \left[h_T + \delta V_{T+1}\left(x_{T+1}\right)\right] &(8) \\ &= \max_{h_T} \left[h_T + 0\right] &(9) \end{aligned}$$

which is straightforward to solve: $h_T^* = x_T$ (this is due to the constraint $h_t \leq x_t$, without which the optimal policy would be $h_T = \infty$). Plugging $h_T^* = x_T$ into the DPE gives us the period-T value function:

$$V_T(x_T) = x_T \tag{10}$$

This is extremely convenient, because as we step back to period $T-1$ we know the analytical form of the value function one period hence. The period $T-1$ DPE is:

$$\begin{aligned} V_{T-1}(x_{T-1}) &= \max_{h_{T-1}} \left[h_{T-1} + \delta V_T\left(x_T\right)\right] &(11)\\ &= \max_{h_{T-1}} \left[h_{T-1} + \delta x_T\right] &(12)\\ &= \max_{h_{T-1}} \left[h_{T-1} + \delta f\left(x_{T-1} - h_{T-1}\right)\right] &(13)\end{aligned}$$

Taking the first order condition gives:

$$f'(x_{T-1} - h_{T-1}) = \frac{1}{\delta} = 1 + r \tag{14}$$

Which implies that in period $T-1$ we should harvest down to the point where the growth rate equals the discount rate. Let's call this optimal level of "residual stock" y_{T-1}^* (where $y_{T-1}^* \equiv x_{T-1} - h_{T-1}^*$). Importantly, note that y_{T-1}^* is independent of x_{T-1} (think of it as a number that is state-independent). Then, we can re-write the DPE as follows:

$$V_{T-1}(x_{T-1}) = x_{T-1} - y_{T-1}^* + \delta f\left(y_{T-1}^*\right) \tag{15}$$

Since y_{T-1}^* is independent of x_{T-1}, we can define: $A \equiv -y_{T-1}^* + \delta f\left(y_{T-1}^*\right)$, and the value function simply becomes:

$$V_{T-1}(x_{T-1}) = x_{T-1} + A \tag{16}$$

for a constant, A (note that the period-$T-1$ value function in Equation 16 looks a lot like the period-T value function in Equation 10). Stepping back one more period, the DPE is:

$$\begin{aligned} V_{T-2}(x_{T-2}) &= \max_{h_{T-2}} \left[h_{T-2} + \delta V_{T-1}\left(x_{T-1}\right)\right] &(17)\\ &= \max_{h_{T-2}} \left[h_{T-2} + \delta\left(x_{T-1} + A\right)\right] &(18)\\ &= \max_{h_{T-2}} \left[h_{T-2} + \delta\left[f\left(x_{T-2} - h_{T-2}\right) + A\right]\right] &(19)\end{aligned}$$

Taking the first order condition gives:

$$1 = \delta f'(x_{T-2} - h_{T-2}) \tag{20}$$

which implies that we should harvest down to $y^*_{T-2} \equiv x_{T-2} - h^*_{T-2}$ and that this value is independent of the current state x_{T-2}. It also implies that $y^*_{T-2} = y^*_{T-1}$. The period $T-2$ value function is:

$$V_{T-2}(x_{T-2}) = x_{T-2} + B \tag{21}$$

Where the constant B is given by:

$$B \equiv \delta \left[f(y^*_{T-2}) + A \right] - y^*_{T-2} \tag{22}$$

In this manner, one can continue to work backwards. For this simple problem, the pattern has become clear. We can *conjecture* the form of the period-t value function as follows:

$$V_t(x_t) = x_t + \Theta_t \tag{23}$$

for a time-specific constant Θ_t that is independent of x_t. To verify this conjecture, we can plug it into the DPE and see if it is internally consistent. Under the conjectured value function, we have:

$$\begin{aligned} V_t(x_t) &= \max_{h_t} \left[h_t + \delta V_{t+1}(x_{t+1}) \right] && (24) \\ &= \max_{h_t} \left[h_t + \delta \left(f(x_t - h_t) + \Theta_{t+1} \right) \right] && (25) \end{aligned}$$

which yields a first order condition:

$$1 = \delta f'(x_t - h_t). \tag{26}$$

This first order condition implies that under the conjectured value function (in $t+1$), the optimal policy in period t is to harvest down to a specific level that is independent of x_t. This implies that the period-t value function is $V_t(x_t) = x_t + \Theta_t$, which proves the conjecture.

Note that we could also have explicitly calculated the value function itself in any period t. The take-home message is that backward induction can often be used to analytically derive the optimal policy function. For this problem, the optimal policy (in all periods $t < T$) is to harvest the stock down to the level at which $1 = \delta f'(x_t - h_t)$, and in period T the optimal policy is to harvest the entire stock.

4 Numerical Solutions: Value Function Iteration

While simple dynamic problems may be solved analytically, many realistic problems are too cumbersome to solve analytically. In such cases, one approach is to resort to numerical optimization techniques. Here we introduce the simplest method for numerically solving dynamic programming problems, called "Value Function Iteration" or VFI. Note that it can be used for deterministic or stochastic problems and that states and/or controls could be vectors. If you have a continuous state or control space, you must modify VFI slightly.

The intuition behind VFI is as follows: Start at the end of time (or if the problem is infinite-horizon, start a long distance into the future). Assume there is no future. For many possible values of the period-T state, numerically optimize and thus solve for the value function in period T. Step back a period, invoke the period-T value function, and solve for the period $T-1$ value function, etc. Repeat this process until you reach the present period. Often, the main interest of the researcher is the *policy function*, not the *value function*. The policy function often converges to a single function within 10 periods. The value function will only converge at the rate of interest (so you might have to step back for 20-30 periods, depending on the discount rate).

The recipe is as follows:

1. Discretize the state and control space

2. "Guess" or insert (if known) the correspondence: $V_{T+1}(x_{T+1})$. If you assume no future, this would just be set equal to 0.

3. Step back 1 period (to period T), and calculate:

$$V_T(x_T) = \max_{u_T} \left[\pi(x_T, u_T) + \delta V_{T+1}(G(x_T, u_T))\right] \tag{27}$$

   for every possible value, x_T. This gives the *policy function*, $u_T^*(x_T)$ and the value function $V_T(x_T)$.

4. Step back 1 period (to period $T-1$), obtain the *policy function*, $u_{T-1}^*(x_{T-1})$ and the value function $V_{T-1}(x_{T-1})$. Repeat until you get to time 0.

This procedure will give a set of $T+1$ value functions and a set of $T+1$ policy functions. For infinite horizon problems, we are typically interested in the properties of the policy function.

5 A Renewable Resource Example

A renewable resource stock x_t is harvested to maximize profit. Harvest in period t is h_t and the current period profit from harvest is: $\Pi(h) = \alpha h - (\beta h^2)/2$, which corresponds to a downward sloping linear demand curve. The discount factor is δ and the equation of motion (i.e. the growth of the resource stock) is:

$$x_{t+1} = (x_t - h_t) + r(x_t - h_t)(1 - (x_t - h_t)/K) \tag{28}$$

for scalar parameters r and K. The dynamic programming equation is:

$$V_t(x_t) = \max_{h_t} \Pi(h_t) + \delta V_{t+1}((x_t - h_t) + r(x_t - h_t)(1 - (x_t - h_t)/K)) \tag{29}$$

Again, the problem is that the value function is not known.

The pseudo-code for solving this problem is as follows:

1. Set all parameters (including T).
2. Discretize the state space into N equally spaced values on $[0, K]$, you can make this grid finer by increasing N.
3. Set the $T+1$ value function to zero (so V_{T+1} is an $Nx1$ vector of zeros, each one corresponding to a different level of the stock).
4. Loop backwards over time, going from T to 1.
   (a) Loop over the N possible stock values
      i. For a given stock size x, define a vector of possible harvest values (ranging from 0 to x)
      ii. For each harvest, calculate the next period's stock and the corresponding value (from the "known" value function).
      iii. The value of stock level x and harvest level h this period is: $\Pi(h) + \delta V_{t+1}(x_{t+1})$.
      iv. Pick the harvest level that maximizes that expression
   (b) End the loop over stock
5. End the loop over time.

6 R Code

Here is the actual R code for solving this problem:

```
# This code solves the dynamic programming code for Econ 260A

rm(list=ls(all=TRUE))
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
```

```
width=2;
library(ggplot2)
library(dplyr)
library(stringr)
library(cowplot)

DynamicObjective260A=function(H,X,a,b,delta,r,K,q,Xgrid,V)
{
  #To be used in optimization...this thing is minimized by choosing H
  Pi = a*H - b*(H^2)/2
  Xprime = (X-H) + r*(X-H)*(1-(X-H)/K)
  Vnext = spline(Xgrid,V,xout=Xprime,method="natural")
  negout = -(Pi + delta*Vnext$y)
  return(negout)
}

a=10
b=.2
delta=1/1.1
r=.8
K=100
q=.5
X0=K/5

T=30
XL=50

Xgrid = seq(.1,K,length.out=XL) #Grid over state space
Vmat = matrix(0,nrow=length(Xgrid),ncol=T+1)
Hstar=matrix(NA,nrow=length(Xgrid),ncol=T) #Initialize the control vector
Vnew=matrix(NA,nrow=length(Xgrid),ncol=1) #Initialize the "new" value function

for (t in seq(T,1,-1))
{
  print(t)
  for (i in seq(1,length(Xgrid),1))
  {
    guess=0
    X = Xgrid[i]
    #This finds optimal policy function
```

```
    Thing = optim(par=guess,fn=DynamicObjective260A,gr=NULL,lower=0,upper=X,X=X,a=a
    Htmp = Thing$par
    Valtmp = Thing$value
    Hstar[i,t] = Htmp #the optimal Harvest for each stock
    Vnew[i,1] = -Valtmp
   }
  Vmat[,t]=Vnew
}

xg=data.frame(Xgrid)
hdf=data.frame(Hstar)
vdf=data.frame(Vmat[,1:T])
II=data.frame(index=1:XL)
hdf2 = bind_cols(II,xg,hdf)
vdf2 = bind_cols(II,xg,vdf)
hdf_new = gather(data=hdf2,key="Xtime",value="harvest_opt",num_range('X',1:T)) %>%
  mutate(T_end=T+1-as.numeric(str_replace(Xtime, 'X', ''))) %>%
  select(-Xtime)
vdf_new = gather(data=vdf2,key="Xtime",value="value_function",num_range('X',1:T)) %
  mutate(T_end=T+1-as.numeric(str_replace(Xtime, 'X', ''))) %>%
  select(-Xtime)
DF = left_join(hdf_new,vdf_new,by=c("index", "Xgrid", "T_end"))
DF_converge = DF %>%
  filter(T_end==T)

P1 = ggplot(data=DF) +
  geom_line(aes(x=Xgrid,y=harvest_opt,color=T_end,group=T_end),size=1) +
  xlab("Stock, X") +
  ylab("Opt. Harvest") +
  scale_color_continuous(name="Time to End")

P2 = ggplot(data=DF) +
  geom_line(aes(x=Xgrid,y=value_function,color=T_end,group=T_end),size=1) +
  xlab("Stock, X") +
  ylab("Value Fn.") +
  scale_color_continuous(name="Time to End")

P3 = plot_grid(P1,P2,nrow=2)

ggsave(filename="../Converge.pdf",plot=P3)
```

```
# Now do Forward Sweep
XX=vector() #Initialize vectors
HH=vector()
Pi=vector()
Pipv=vector()
XX[1] = X0 #Starting stock for forward sweep

#Forward Simulation
simtime = seq(1,T,length.out=T)
for (tt in 1:T)
{
HHtmp= spline(Xgrid,Hstar[,tt],xout=XX[tt],method="natural") #Interpolate to find h
HH[tt] = HHtmp$y #Use interpolated harvest in period tt
XX[tt+1] = (XX[tt]-HH[tt]) + r*(XX[tt]-HH[tt])*(1-(XX[tt]-HH[tt])/K)
Pi[tt] = a*HH[tt] - b*(HH[tt]^2)/2
Pipv[tt] = (delta^tt)*Pi[tt]
}

DFsim = data.frame(time=simtime,X=XX[1:T],H=HH[1:T],Profit=Pi[1:T],PVProfit=Pipv[1

S1 = ggplot(data=DFsim) +
  geom_line(aes(x=time,y=H),color="blue",size=2) +
  xlab("Harvest")

S2 = ggplot(data=DFsim) +
  geom_line(aes(x=time,y=X),color="blue",size=2) +
  xlab("Stock")


S3 = ggplot(data=DFsim) +
  geom_line(aes(x=time,y=Profit),color="blue",size=2)


S4 = ggplot(data=DFsim) +
  geom_line(aes(x=time,y=PVProfit),color="blue",size=2)
```

```
P4 = plot_grid(S1,S2,S3,S4,ncol=2,nrow=2)
P4
```

This produces 2 graphs. The first shows the evolution of the policy function and value function over time, see Figure 1. The bottom panel is the value function for each of the 30 periods. The highest one is the value function at time $t = 1$. The top panel is the policy function. The lowest one corresponds to period $t = 1$. Note that because the policy function converged (it took 6 or 7 periods), this would also be the policy function for an infinite horizon problem. The optimal policy is to harvest an amount that is increasing in the current stock. No harvest occurs below a stock of about 27. At a stock of about 50, you should harvest about 15. At a stock of 100, you should harvest about 35.

The second figure shows key variables over time in a forward simulation. Note what happens at the end of the time horizon - the stock is drawn down. See Figure 2.

Figure 1: Value Function and Policy Function

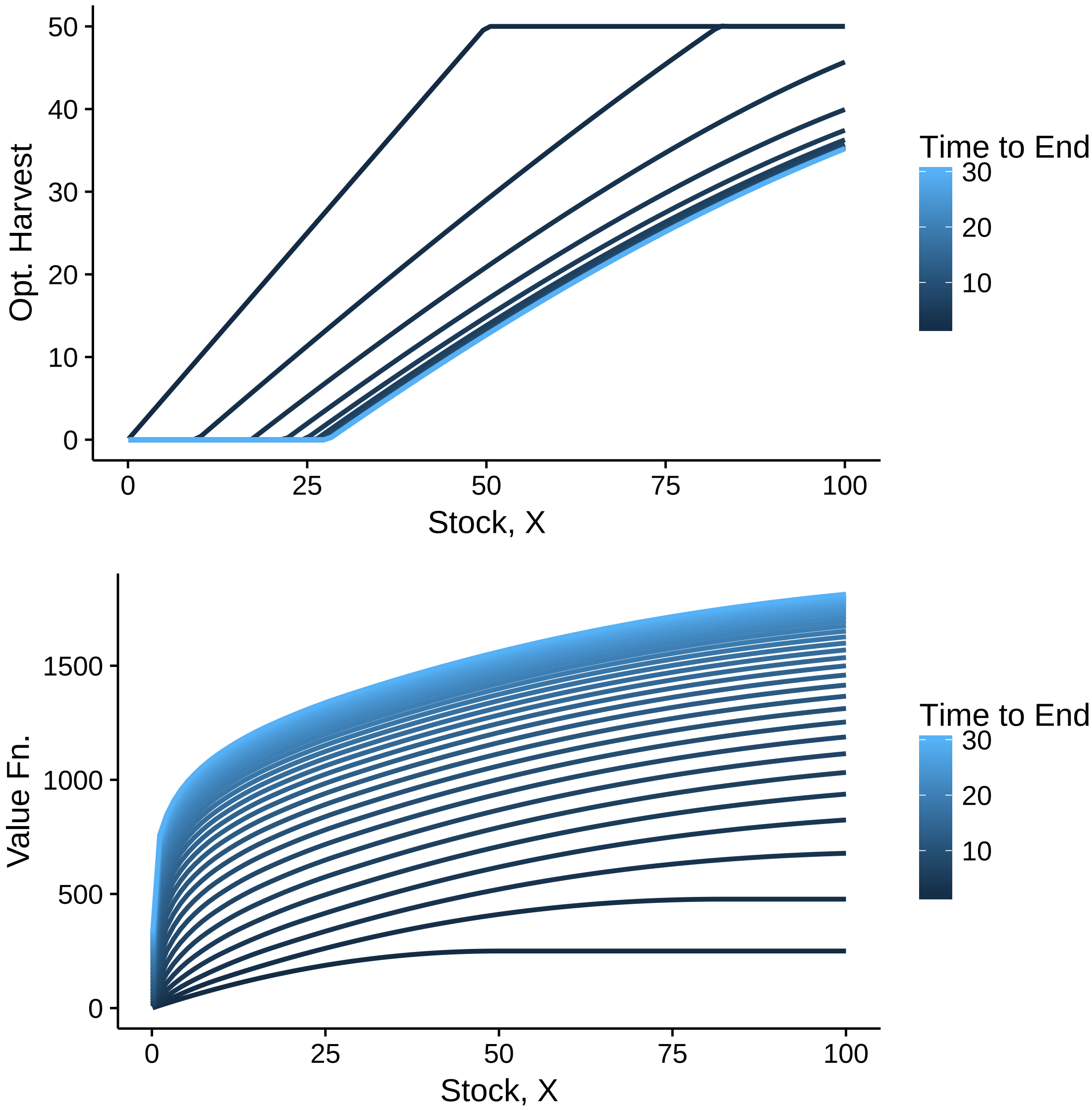

Figure 2: Key Variables over Time