PPHA 44330: Energy and Environmental Economics, Fall 2020
Computational assignment, due before class Thursday, November 12

This assignment has three questions. The first two focus on a standard problem of socially optimal extraction of an exhaustible resource, with zero marginal costs. The third considers the value of an oil drilling option when the oil price is uncertain.

For all questions, please turn in (via email) all of your code, along with a pdf file that includes all of the required plots.

<u>Setup for questions 1 and 2</u>

The goal is for you to solve these problems on the computer for two different specifications for utility. Question 1 uses discretization to approximate the value function, while question 2 asks you to use interpolation between discrete states to obtain a better approximation.

The parameters of the problems are as follows. The initial resource stock is S_{tot} = 1000 units. Extraction costs are zero. Utility takes on one of two functional forms: $u(y) = 2y^{0.5}$ or $u(y) = 5y - 0.05y^2$ (linear demand). The discount rate r is 0.05, so the discount factor δ is 1 / 1.05. The time horizon is infinite.

1. <u>Solving the DDP by value function iteration; discrete state space</u>. For both utility functions, solve for the optimal extraction path on the computer, following the steps below:

   (a) Set up the state space (stock remaining) so that it runs from 0 to 1000 in 501 steps (so that the step size is 2). Let S denote the state space vector, and let N = 501 denote the number of discrete states.

   (b) Set up the action space (the amount to extract each period) so that it matches the state space. This way, after any action the remaining stock will exactly match a state. Denote the state space by A, and the number of possible actions by nA (=501).

   (c) Define utility given extraction. E.g., using Matlab implicit functions: u = @(x) 2*x^0.5;

   (d) Define a flow utility matrix U (N x nA) that gives the flow utility for every state and action. In the model, we don't want the planner to be able to extract more than the remaining stock, so let $U = -\infty$ for all cells in which the action implies more extraction than the remaining stock.

(e) Create another N x nA matrix that identifies the index of the state next period given that the initial state this period was row i and that the action taken was column j. If j calls for more extraction than the remaining stock, set the next state to have zero stock.

(f) Create a state transition matrix T that is N x ($N*nA$). Each N x N block should give the probability of transitioning from each row state to each column state, given a particular action. Because the model is deterministic and each action always results in next period's state hitting one of the discrete states exactly, this matrix should consist only of zeros and ones. In Matlab, you should try creating this matrix using Matlab's "sparse" command (or use the equivalent in whatever programming language you prefer). The matrix is small enough that you can probably get away with using a regular full matrix, but it's useful to learn how sparse matrices work.

(g) Initialize the value and control functions, and solve the model using value function iteration. At each iteration, you'll need to use the T matrix you calculated above to calculate an N x nA matrix (here labeled *Vnext*) that contains the expected value of next period for each current state and possible action (you can calculate *Vnext* by looping over actions). Once *Vnext* is calculated, the contraction mapping step should look something like (in Matlab): [V C] = max(U + delta * Vnext,[],2);. V will then hold the maximized value function and C will hold the optimal action (for the given iteration). Iterate until V converges to a tolerance of 1E-8 in the sup norm.

(h) Once you've solved the model, find the N by N optimal transition matrix T_{opt} that gives the state transition probabilities when the optimal action is taken in each state.

(i) Simulate the model for t = 80 periods, starting with the initial stock of 1000. Every period, you will use the optimal C and T_{opt} mappings to obtain each period's extraction and the stock remaining at the end of the period.

(j) For both utility functions, plot out the optimal extraction path and the price path against time. Notice that they're bumpy in spots rather than nice and smooth, even though we have 501 discrete states. We'll use interpolation to solve this problem in part 2.

2. Interpolating between the states.

(a) Set up the state space as you did in part 1. We'll define the action space differently, however, to more accurately model what happens when extraction rates are low. Set up the action space so that it still runs from 0 to 1000 over 501 steps, but now make the step size even over the square root of the action space (in other words, create 501 even steps over [0,1000^0.5], and then square each step to form the action space). This will bunch the actions closely together at the low end, which will help us capture the asymptotic decline of production in the model in which inverse demand goes to infinity as quantity goes to zero.

(b) Figuring out the state transition matrix is now more challenging since most actions will lead to next period's state not hitting a discrete state exactly. What we want to do now is interpolate. Suppose next period's state, for a given entering state and action, is 502.5. Your code needs to find the indices for the nearest two states (502 and 504), and then assign probability weights to each of those states. In this example the weights would be 0.75 on 502 and 0.25 on 504 (simple linear interpolation). Thus, each row of each N by N block of the T matrix will usually have two non-zero entries that sum to one (a few rows will contain a single 1).

(c) Solve the model via value function iteration as before, finding the optimal actions and state transitions.

(d) Simulate the model for $t = 80$ periods, starting with the initial stock of 1000. Simulation is more challenging now because the initial state in each period will generally not be an exact discrete state. Thus, the optimal action each period cannot be read directly from the discrete policy function. We will solve this problem by linear interpolation. As in the example above, if (for example) the state entering some time period is 502.5, then the optimal action will be 0.75 times the action for state 502 and 0.25 times the action for state 504. (Note: you might find it helpful to write a sub-function to handle the interpolation.)

(e) For both utility functions, plot out the optimal extraction path and the price path against time. They should be much smoother now! (though still not perfectly smooth…though this can be resolved with tighter state and action spaces)

3. Drilling and real options. We now consider a different problem. You are the owner of a single well. If you drill the well, it will cost D = \$3 million, and the well will instantly produce X = 100,000 barrels of oil. Your problem is to decide when to drill the well, as a function of today's oil price.

In any period t, you observe the current oil price P_t. If you drill, you earn profits of $P_tX - D$. Given the setup of the problem, it is obvious that profits are positive iff the oil price is greater than \$30 / barrel.

Next period's oil price is equal to this period's price plus a random shock that is normally distributed with a mean of zero and a standard deviation of \$4. Thus, the expected price next period is today's price, but the realization of P_{t+1} is unknown at time t.

The discount rate r between periods is 0.05, so the discount factor δ is 1 / 1.05. The time horizon is infinite.

Your goal is to solve for the optimal "trigger price" P^* such that you should drill at time t if $P_t \geq P^*$, and solve for the value of the option as a function of P_t.

(a) Set up the state space as a vector of possible oil prices. Let prices vary from \$0 to \$80, with a step size of \$1 (so that the number of states $N = 81$). Define a vector that returns the firm's profits from drilling as a function of the price.

(b) Define a state transition matrix T that gives, for each row state i, the probability that the next period's price will be "in" the column state j. (This is the trickiest part of this question.) The matlab function normcdf will be useful here in obtaining the probability that next period's price will be above or below a given cutoff. For cutoffs, use the price halfway between states (so, for example, a next period's price of \$49.51 will be assigned to the \$50 price state, whereas a price of \$49.49 would be assigned to the \$49 price state). For the lowest price, the bottom cutoff is $-\infty$, and for the highest price use a top cutoff of $+\infty$.

To verify that you constructed T correctly, each row should sum to 1.

(Note that near the ends of the state space, the probabilities are such that the expected price next period is not the current price. This is an unavoidable issue in these kinds of problems, since it's not practical to define a state space that goes from $-\infty$ to $+\infty$. In practice, what researchers do is define a state space that extends well beyond the states of interest, and then test the robustness of the results to alternative state space definitions (I do this in my "Investment Under Uncertainty" paper). For instance, in this problem we're probably most interested in what happens between prices of \$30 and \$50 (the trigger price is probably in there somewhere), and in this price range the transition matrix is an accurate representation of the true price process. But we need lower and higher prices in the state vector, since we need to properly model future price expectations in the event that today's price is \$30 or \$50.)

(c) Use value function iteration to compute the value function and the control function. Report the "trigger price" at which drilling is optimal (you should find a trigger price of \$41). The fact that this trigger price exceeds \$30 reflects the option value of waiting!

(d) Create a graph showing the value function (i.e., put price on the x-axis, and the value of the drilling option on the y-axis). Intuitively explain its shape, and why it becomes a straight line for prices greater than \$41.