Macro III

Homework 2- Approximate the Value Function

In this homework you will again consider the cake-eating problem from homework 1. The size of the cake, preferences, time periods and the grid on the state space is the same as before. The one thing that is different is that in the DP problem the control variable is NOT restricted to lie on a prespecified grid. Thus, you will need to carry out a one dimensional maximization to solve the problem on the right-hand-side of Bellman's equation.

1. Graph the optimal value function for this problem in period 1 and the computed value function. Do this also for the computed optimal decision rule in period 1 and the actual optimal decision rule. Do both of these for $N = 101$ grid points - see homework 1 for how to pick grid points.

2. Change the number of grid points to $N = 201$ and present the same two graphs from 1.

3. Include a copy of your computer code with your homework.

SUGGESTIONS:

1. Create an array $V(NX, NJ)$ for the value function, where NX is the number of grid points on the state and $NJ = J + 1$ is the number of periods. Initialize the value function to zero in period $J + 1$.

2. Your algorithm to solve this homework should be two DO LOOPS with a maximization operation sandwiched in between.

3. To evaluate the right-hand-side of Bellman's equation, you will have to interpolate the term for the value function next period. Do this by linear interpolation. [A quadratic interpolation may plausibly lead to a better approximation as the object to be interpolated is concave, but a linear interpolation has the advantage of simplicity.] There are two steps. First, find the two gridpoints points that lie on each side of the point to be interpolated. This can be done in some computer languages with a single command or by a subroutine in others- this is essentially "searching an ordered table" as described in Numerical Recipes. Second, hit the value function at the two interpolating gridpoints with the correct weights. You may want to put some of the maximization code or the interpolation code in a subroutine to improve the elegance of your code.

4. To carry out the maximization, you can use a built-in one dimensional maximization routine in some computer languages. Of course, you can write your own but this is NOT recommended. Judd (1998, CH 4) gives a discussion of some standard approaches to this problem. These methods make use of different assumptions- derivatives, concavity, ... This is a deep topic.