Fatih Guvenen University of Minnesota

Homework #3

Due back: Tuesday, November 22, by 5pm

You are allowed to work in groups of up to 2 students and submit together. Please make sure to specify the members of the group clearly. Please make sure to submit a clear report in the form of an executive summary, that describes your answer to each question and that is self contained, along with any piece of code (Fortran/Matlab) that generates your results. Please make every effort to produce a complete package so that I can understand what you did, how you did it, and what you found. Please submit a single zip archive containing all your files.

Questions indicated by a "star" are required for everybody, including those who audit the class. You can use MatLab, Python, Julia, Fortran, or C/C++ to do the homework.

For each question, please discuss your answers in your report. (Please do not merely provide some numbers and a code).

1. *Use one-sided finite differences to compute an approximation to the first derivative of $g(p) = 0.5p^{-0.5} + 0.5p^{-0.2}$ at $p = 1.5$. Let the increment ϵ in the finite differences range across all the values in the set $\{10^{-1}, 10^{-2}, ..., 10^{-10}\}$. For which value of ϵ the approximate first derivative the most accurate?

2. *Repeat the third problem using two-sided finite differences to approximate the first derivative.

3. Use the bisection, secant, and Newton's methods to compute an estimate of p_0, where $g(p_0) = 0.75$ (and g is defined in the first problem). For each method, report how many iterations are required to compute an estimate $\hat{p}$ satisfying $|f(\hat{p}) - f(p_0)| < 10^{-6}$.

4. Repeat the last problem using Brent's method as described in Chapter 9.3 of Numerical Recipes in Fortran.

5. *This question asks you to solve the same dynamic programming problem you studied in HW1 but this time making tomorrow's capital a continuous choice not restricted to a grid and assuming that A is not fixed but follows a pe rsistent stochastic process. Therefore you will need to use an interpolation method and an optimization routine. Here is the same problem as before, but now restated to feature a continuous choice:

   Let the production function take the form $f(k) = Ak^{\alpha} + (1 - \delta)k$, where $A > 0$, $0 < \alpha < 1$, and $0 \leq \delta \leq 1$. Let the utility function be $U(c) = c^{1-\alpha}/(1 - \alpha)$, $\alpha = 2$, and assume that the savings choice, k', can take *continuous values* above a certain positive minimum $k' \geqslant \underline{k} \geq 0$. The productivity parameter A follows a 15-state Markov process with a persistence of 0.98. Pick the minimum and maximum of your grid judiciously so as to cover the entire ergodic set *and* make sure to choose the maximum capital grid point such that you do not have to extrapolate *too much* at the highest capital grid (recall we discussed in class what too much

extrapolation means). Start with a small number (say, 31) of grid points and use an expanding grid that works the best for your problem. Then increase this number in increments of 20 or 30, up to you, to say, 500+, and solve the problems below again until the algorithm converges. Use a convergence criterion defined on the decision rule as follows:

$$\max_i \left(\frac{k_i^{n+1} - k_i^n}{1 + |k_i^{n+1} - k_i^n|} \right) < 10^{-6}, \tag{1}$$

where k^n is the n indexes iteration number and i indexes grid number. [Hint: Solving the problem at the highest capital grid today, k_N, requires extra care because you need to allow the agent to choose $k' > k_N$ tomorrow. In other words, you cannot restrict her choice set to lie inside the capital grid. This is why the maximization problem needs to extrapolate beyond k_N in tomorrow's grid to consider such k' choices]

(a) Solve the described problem using the basic VFI algorithm as well as by applying the modified policy iteration using $m = 5, 20, 100$, and 500 Howard steps. Compare both the solution (value function and decision rules) and the time it takes to obtain a solution to the basic VFI (which you also need to redo for the continuous case). Was 31 grid points for k sufficient to get an accurate solution? If not, what was the minimum number of grid points you needed for the VFI algorithm to converge. Plot your value functions and decision rules and apply spline interpolation to see if there are any wiggles that might cause problems.

(b) Solve the described problem using the basic VFI algorithm *again*, but now using the same setup as in HW1 (You don't need to do the Howard iteration part). Specifically, assume that the choice of k' lies on a discrete grid tomorrow. Now you are free to take an equally-spaced or expanding grid for k (you can guess which one makes more sense). You won't need to do interpolation, and you can choose whichever optimization routine you want. The goal is to get the same visual accuracy on the value function and the same tolerance for the capital decision for equation 1 as in part (a) but using the discrete grid. How many grid points for k did you need to pick? Compare the timing to that of the basic VFI above. How do they compare?