Fatih Guvenen University of Minnesota

Homework #2

Due back: Thursday, November 17, by **Noon**

Instructions: 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.

1. *This problem asks you to interpolate some utility functions using three different methods. There are built-in functions or libraries in all programming languages listed above that will do these interpolations automatically. While you can use them, you'll need to make sure that the algorithm isn't a black box, i.e., it allows you to set grid or knot points where you want, assign boundary conditions when relevant, etc. Please write in your report the name of the subroutine you used and the language. These are the three utility functions:

$$u(c) = \log(c)$$
$$u(c) = \sqrt{c}$$
$$u(c) = \frac{c^{1-\alpha}}{1-\alpha}, \quad \text{for } \alpha = 2, 5, 10.$$

   Suppose that you can evaluate these functions at as many points as necessary over the domain [0.10, 5]. So, how many points you choose to perform the interpolation is up to you, but you should aim to minimize the computational burden (i.e., as few function evaluations as possible). Assess the accuracy of each interpolation method by plotting the interpolated values on a finer grid than the one you used in your interpolation scheme (for example, take an equally spaced grids with increments of 0.01 (of c). Plot the function and what results from using each approximation.

   (a) *Conduct the experiment using (i) linear interpolation, (ii) cubic splines, (iii) Chebyshev polynomials. The first two interpolation schemes allow you to choose grid points any way you like. For (i) and (ii), try expanding grids with exponent $\theta = 1, 1.5, 3, 4$ so as to find the value to minimize computational cost for a given level of accuracy. Which interpolation scheme works the best, in terms of minimizing the computational burden? In cases (i) and (ii) how many points do you need to take so that your maximum error is less than 1% of the function value at each point? In case of (iii) what is the order of Cheb. Pols. you need for the desired accuracy?

(b) **The perils of extrapolation.* Extrapolation is evaluating the interpolated function at points outside of the grid it was originally constructed. So, for this part, calculate the value of each utility function at $c = 0.05$ and $c = 5.1, 5.5$, and 6, using the three different interpolation schemes. Is this feasible for all three methods above? Among the feasible ones, which one works "better" for extrapolation? (Define also what "better" should mean in this context.)