Fatih Guvenen University of Minnesota

Econ 8185: Homework Assignment #1

Due back: Thursday, November 3rd, by **1:00PM**

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/Julia/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 either MatLab/Julia/Python, Fortran, or C/C++ to do the homework, but Fortran, Julia, or C is strongly preferred and recommended.

The first two questions will help you learn some basic but important issues about numerical algorithms. Before attempting these questions, carefully study Chapter 1 of the *Numerical Recipes* book, which covers many other essential topics for computation.

1. ***[15 points] Rounding error:** Using the well-known formula for the roots of a quadratic equation ($x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$) compute the larger root of the following quadratic equation: $ax^2 + bx + c = 0$ for the following values: $a = 1$, $b = 100000$, and $c = 10^n, n = -1, -2, ..., -8$. Now use the following alternative method. First calculate $q = \frac{1}{2}\left(-b + \text{sign}(b)\sqrt{b^2 - 4ac}\right)$, and then obtain the two roots: $x_1 = q/a$ and $x_2 = c/q$. Notice that the two calculations should give you precisely the same answer if it were not for the rounding error of the computer arithmetic. Also note that in some real life computational problems you will need precision up to the 10th or 12th significant digit (for example, to clear the asset market). How close are the answers for different values of n?

2. **[15 points] Truncation error/Unstable Algorithms:** Consider the "Golden Mean" which is the number given by:
$$\phi = \frac{\sqrt{5} - 1}{2} \approx 0.61803398.$$
   (a) First, verify analytically that the powers of this number satisfy the following recursion (Hint: trivial!)
   $$\phi^{n+1} = \phi^{n-1} - \phi^n$$
   This recursion suggests a seemingly efficient way to compute higher powers of ϕ since it only requires subtraction which is much faster than multiplication (required to exponentiate ϕ directly).

   (b) But how well does the clever algorithm described in (a) works in practice? To see this, compute ϕ^n for $n = 2, 3, ..., 20$ using two separate methods. First, write a simple program using the recursion above and the fact that $\phi^0 = 1$ and $\phi^1 = 0.61803398$. Second, simply raise ϕ to the appropriate power. Compare the values obtained using the two methods. How do they compare to each other as n increases? This (somewhat contrived) exercise illustrates the dangers of rounding and truncation errors that combine together to create a dirty mess.

3. ***[70 points] VFI.** Write a computer program to solve the neoclassical growth model using value function iteration on a discrete grid. 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 $U(c) = \log(c)$ and assume that the savings choice, k', is restricted to lie on the same discrete capital grid you construct. Center your grid at the steady-state capital stock $\overline{k}$, as defined by $f'(\overline{k}) = \beta^{-1}$. Start with a small number (say, 11) of equally-spaced grid points, and then increase this number to, say, 1001. Take the zero function as your initial guess for the value function (despite the fact that it is a very poor choice from an efficiency stand point, as we discussed in class. We will learn to pick better ones soon).

   (a) [20 points] Using basic VFI with no acceleration methods, obtain numerical results both for the case of full depreciation ($\delta = 1$) and for the case of less-than-full depreciation ($\delta = 0.10$). For ($\delta = 1$), compare your numerical findings to the analytical (closed-form) solutions for the value function and the decision rule.

   (b) [20 points] Now repeat part (a) for $\delta = 0.10$ but now applying modified policy iteration using $m = 5, 10, 25, 100$, 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.

   (c) [30 points] Repeat part (b) using MacQueen-Porteus bounds acceleration method. Apply the bounds in (i) every iteration, and (ii) every 5 iterations. Compare the convergence time in (i) and (ii) to each other and to basic VFI and MPI. Prepare a table so as to make comparisons easier.