

**Assignment 1**

Due: October 3rd by 2pm.

1. (a) Obtain time-series (as long as possible) of quarterly Canadian data on GNP and at least 4 other variables of either consumption, investment, government purchases, imports, exports, hours, or employment.
  - (b) Write your own program to HP filter data using FORTRAN.
  - (c) Using your HP filter and FORTRAN, generate the growth and cyclical components of the log of each series. Then, using MATLAB, generate plots of the log series along with their HP trends and of the series cyclical components.
  - (d) Using FORTRAN and your HP filtered data, compute a collection of business cycle statistics. The statistics should include each variables standard deviation and the cross-correlation of each variable (including GNP itself) with GNP as well as it's first 5 lags and leads. Something similar to Table 1.1 in Chapter 1 of Cooley (1995).
  - (e) Create a pdf which contains all your figures, a table of your business-cycle statistics and one or two paragraphs (no more than 1/2 a page) commenting your findings.
2. Consider a representative agent economy where preferences are given by

$$U(c, n) = \frac{(c^\alpha(1 - n)^{1-\alpha})^{1-\sigma}}{1 - \sigma},$$

and the technology is

$$y = zk^\theta n^{1-\theta}.$$

Capital,  $k$ , depreciates at rate  $\delta$ . In the following, use  $\beta = 0.985$ ,  $\delta = 0.01$ ,  $\theta = 0.36$ ,  $\alpha = 0.6$  and  $\sigma = 1.5$ . Consider  $z$  as a parameter and include an equation to normalize  $y$  to 1 in the steady state.

- (a) Solve the Bellman equation using MATLAB and value function iterations. Use matrix operations to compute the updated value function at each iteration.
- (b) Solve the model again using value function iterations with FORTRAN and “for-loops” (You should write a one-dimensional equation solver to solve the first-order condition for labor.)
- (c) Use the fact that the value function is concave and the policy function monotonic to improve your “for-loops” algorithm.
- (d) Incorporate Howard’s improvement algorithm into your FORTRAN program.

- (e) Create a table where you report computing time for each case and for different grid sizes. Put the table in a pdf file.

3. Consider the optimal growth model

$$V(k) = \max_{k'} U(F(k) - k') + \beta V(k'),$$

where

$$U(c) = \frac{c^{1-\sigma} - 1}{1 - \sigma},$$

and

$$F(k) = k^\alpha + (1 - \delta)k.$$

Set the parameters to  $\alpha = 1/3$ ,  $\sigma = 2.0$  and  $\delta = 0.05$ . Using FORTRAN for all computation do the following:

- (a) Solve the model using value function iterations. Once you have a solution, find the parameters  $a$  and  $b$  of the function  $a + b \ln(k)$  such that this function represents your policy function as best as possible in a least square sense. In other words, “estimate” the equation:

$$k' = a + b \ln(k).$$

- (b) Now implement the following algorithm, called “policy function iterations.”

Note: You should choose the magnitude of the algorithm parameters like  $N$  and  $\lambda$ . You will need your one-dimensional non-linear equation solver from question 2. Finally, you have to choose the stopping criterion yourself, when you check for convergence. In the end, compare your results with those obtained in part (a).

- i. Choose a grid of  $N$  values for  $k$ .
- ii. Guess a policy function  $k' = a + b \ln(k)$ . Call your guess’ parameters at step  $s$   $a_s$  and  $b_s$ .
- iii. Step  $s$  : For each  $k_0$  in your grid, solve the first order condition given by

$$c_0^{-\sigma} - \beta c_1^{-\sigma} [\alpha x^{\alpha-1} + 1 - \delta] = 0,$$

for  $x$  where

$$c_0 = k_0^\alpha + (1 - \delta)k_0 - x,$$

and

$$c_1 = x^\alpha + (1 - \delta)x - a_s - b_s \ln(x).$$

- iv. “Estimate” the equation:

$$x = a + b \ln(k).$$

If  $a$  and  $b$  are “close” to  $a_s$  and  $b_s$ , you have converged to a solution. If not,

update with  $a_{s+1} = \lambda a_s + (1 - \lambda)a$  and  $b_{s+1} = \lambda b_s + (1 - \lambda)b$ , where  $\lambda \in (0, 1)$  and go to step (c).

Send me by email (kkopecky@uwo.ca) a zip or rar folder containing your pdf files and all your FORTRAN and MATLAB code. Note that all code should be neat, clean, and well-commented. Please name your submission after yourself, for example, my submission would be KarenKopecky.zip.