Economics 8823
Advanced Macroeconomics
Project 0
Professor Sanjay Chugh
Spring 2018

Objective

You will computationally solve the deterministic (aka non-stochastic) steady state of the basic RBC economy.

The Economy

The representative household maximizes lifetime utility

$$\max_{c_t, n_t, k_{t+1}, x_t} E_0 \sum_{t=0}^{\infty} \beta^t \ln\left[c_t - x_t \cdot \frac{\kappa}{1+1/\psi} n_t^{1+1/\psi} \right]$$

subject to its budget constraint

$$c_t + k_{t+1} - (1-\delta)k_t = w_t n_t + r_t k_t$$

and

$$x_t = c_t^{\omega} \cdot x_{t-1}^{1-\omega} .$$

The representative firm maximizes profits

$$\max_{n_t, k_t} \left[z_t f(k_t, n_t) - w_t n_t - r_t k_t \right]$$

by choosing factor inputs n_t and k_t.

The aggregate goods resource constraint is

$$c_t + k_{t+1} - (1-\delta)k_t + g_t = z_t f(k_t, n_t),$$

in which g_t denotes the government's period-t flow of expenditures. The stationary-state (aka, balanced-growth) production technology is

$$f(k_t, n_t) = z_t k_t^{a} n_t^{1-a}$$

and the steady-state level of TFP is $\bar{z} = 1$.

Numerically compute the deterministic steady-state values of c, n, k, r, w, and x (i.e., consumption, labor, the capital stock, the real interest rate, the real wage, and the term that arises from the Jaimovich-Rebelo preference specification) for this economy using the two parameter sets in Table 1**.** (Note: you are **not** solving for any dynamics, just long-run values for each parameter set.)

	Parameter Set A	**Parameter Set B**
β	0.99	0.99
δ	0.02	0.02
ψ	1	1
α	0.36	0.36
ω	1	0.3
κ	**To be determined**	**To be determined**
$\bar{g}$	**To be determined**	**To be determined**

Table 1. Parameter values.

For each parameter set, compute (i.e., calibrate) the value of κ so that $n^{SS} = 0.30$ and the value of $\bar{g}$ so that the long-run share of government expenditures in GDP is 20 percent. (That is, you will compute two different deterministic steady states.)

Also compute the steady-state level of household lifetime utility for each of the two parameter sets, and provide (brief) economic discussion/interpretation of how/why they do or do not differ.

What To Submit

- A clear, concise definition of steady-state equilibrium.

- The steady-state results should be presented in a neatly-organized, easy-to-read table.

- Include your code (i.e., all the relevant files that one would need to replicate your results).

(Some) Computational/Programming Guidance

If using Matlab, you will have to acquaint yourself with Matlab's **fsolve** function. Matlab's **fsolve** function solves a system of (possibly non-linear) equations for a specified vector (or matrix) of variables and hence is suited for computing both the deterministic steady state of models we will encounter as well as for approximating their decision rules. Also perhaps useful (though I believe not necessary for this exercise) is Matlab's **if-else** structure, which allows you to control which lines of code get executed depending on an evaluation of conditions you (the programmer) specify. Matlab's built-in HELP system is useful for learning the basics of all built-in functionality. To access Matlab's HELP feature, you can type at the command line, for example, `help fsolve` or `help if`. More generally, invoking `help xxx` returns information about Matlab's functionality xxx.

Feel free to write your code/programs in whatever way you wish (and this will surely evolve as you experiment with different programming styles over the coming weeks and months (and years…), some that will suit you and some that will not), but there are three guidelines you should (must) follow:

1. Matlab program files are called "m-files." There are two broad types of m-files that you can/will need to create: script files and function files. Your code/programs should be written as one or more than one (obviously inter-related) m-files.

2. Your programs should be written flexibly enough that you can easily run your code for different values of model parameters (the coefficient of relative risk aversion, the share of capital in the production function, the depreciation rate of physical capital, and so on). That is, none of your model parameter values should be "hard-coded" in your programs – they should all be written as "variables." If using Matlab, this requires some block/section of code where the model parameters are "defined" or "declared."

3. Your programs should be readable by someone familiar with programming (i.e., me). Towards this end, you should at least sprinkle (and maybe even heavily douse) your programs with comments (look up what "comments" are if you are unfamiliar with the idea) that describe the tasks/computations that different sections of your code perform. More importantly than making it easier for a reader of your code to understand your logic and computational structure, useful commenting makes it easier for **you** to remember/understand what **you yourself** were originally thinking when you return to your code days or weeks or months after initially writing it.