HW 4:

International Asset Pricing with Recursive Preferences in Dynare++

Due by Dec 23rd , 2022 @ noon

Objectives:

1) efficiently solving a simple exchange economy model with two goods and two agents with EZ preferences in dynare++;
2) integrate in a main matlab file the commands required for:
    a. writing the mod file;
    b. solving the mod in dynare++;
    c. simulating the model to generate nice tables;
    d. creating nice figures.

Expected Final Output: I expect from each one of you a single matlab file called `Main_YOURNAME.m' able to produce the output detailed above (and below) just in one click. The idea is the following: in our field we often work with other co-authors. We need to maximize communication effectiveness and minimize time wasted on the codes. Hence, we need to reduce the number of files we share (that's why I am asking for just one file) and the file needs to have enough comments to be self-explanatory. The file has to be flexible, i.e., it has to allow for different calibrations.

Grading: I will judge your output from the perspective of a coauthor that is involved in a joint research project with you. Here is what coauthors expect:

- the output has to be correct, no typos or miss-computations.
- the matlab file has to contain all essential directions to understand what is going on in the code. ``All'' directions = write enough comments. ``Essential'' directions = give me only the relevant information.
- The code has to be designed to be easy and standalone.

STEP 1: the mod file

- **Warm up.** Complete the files 'Main_RS4LR_student.m' and the implied mod files. You have all parameters required for the EZ-exchange economy and the equations for the cash-flow model. Execute this mod file in dynare++ and have a look at the output collected in ` Main1.mat' (benchmark calibration) and 'Main1000.mat' (alternative calibration without LRR). If you do not get a working .mat file, have a look at the `*.log' file produced by dynare++ and shoot me an email.

STEP 2: Matlab

- **Making everything automatic in matlab. You need to create your matlab code, `MAIN_YOURNAME.m', in the spirit of what done for previous HWs.** Your code will need to have the same structure of "MAIN_RS4LR_Students.m". This is actually the main file that Ric and I used for our paper (from which I erased several lines of code).

STEP 3: produce nice figures

- Produce the mat files for specification (1) and (2) of Table II in the CC-JoF paper.
- Reproduce Figure 2 and 5.
- Since this risk sharing scheme is based on time-variation in variance, you need at least a 3rd order approximation to capture time-variation in variances. For the calibrations with IES=1.5, we use a 3rd order expansion and we find no significant gain in going to higher order. The cost, instead, is huge in terms of simulation time.