Some exercises related to models with heterogeneous agents.

Part A: Solving a KS-type model with aggregate & idiosyncratic risk

This question continues part D of the last assignment. The model remains identical, that is, corresponds to the one specified in the file simpleheteroB.mod.

1. Second-order Xpa solution. Obtain the second-order Xpa solution.
   - For sure you will need to include the mean of capital and the mean of capital squared as state variables. Do you need any other state variables?
   - Use the first-order solution for *Ka*, i.e., mean capital, as the initial guess for the second-order solution. Make up something for the other aggregate state variable (using small coefficients to begin with).
   - Get a second-order solution for *k*.
   - This is quite messy because Dynare reports the solution as a function of *z*(-1) and the innovation *e2* (and other stuff) What is the solution as a function of *z*?
   - Aggregate this equation to get the new policy rule for *Ka*.
   - How do you get the laws of motion for the other aggregate state variables?

2. Accuracy. Do the standard accuracy test.
   a) Generate a time series for aggregate productivity.
   b) Using only laws of motions for aggregate variables generate a time series for aggregate capital and the other aggregate state variables.
   c) Use a cross-section of 10,000 agents (or more if you have a quick computer) and using only laws of motions for individual variables generate a panel time series. If you would need the value of an aggregate variable, then explicitly calculate it from the cross-section. Do not even think of touching the laws of motion for the aggregate variables here.
   d) Compare the two (e.g. by plotting them).
   e) Using the time series generated in step c calculate new laws of motion for the aggregate laws of motion (e.g. linear or quadratic). Are they different?

3. Not a polynomial in levels. In this exercise, use a first-order approximation.
   a) Rewrite simpleheteroB.mod to get a log-linear approximation for individual *k*.
   b) Note that the mean of ln(*k*) is not equal to the log of *Ka*, so we cannot explicitly calculate, at least not directly. Instead use the solution of the individual policy rule to get a log-linear approximation to the individual policy rule. Use this to explicitly aggregate.

4. Accuracy. As in #2 but now for the solution in #3.

Part B: Solving a monetary model with heterogeneous agents

In the file moneypartial.mod you can find the partial equilibrium version of the model discussed in the slide. The goal is to come up with a first-order approximation of the full equilibrium model.

Idea: The idea is to write a Matlab program (like solvehetero.m) that does the following:

- start with a guess for the laws of motion of the aggregate variables (inflation, bond price and aggregate real money balances)
- solve the individual problem
- explicitly aggregate to get new laws of motion for the aggregate variables
- iterate

Steps:

1. moneypartial.mod in some sense does most of the work. This program makes clear how to write a program such that you always impose equilibrium. You just have to run it and figure out how to get an initial guess for the laws of motion of the aggregate variables.
2. rewrite moneypartial.mod to incorporate the aggregate laws of motion. First separately check this program to make sure it runs
3. build the equilibrium *.mod file into an iterating matlab program.

Question:

1. Could you add lagged inflation to the list of aggregate state variables?