Finance 937, Fall 2019
Problem Set #3,
due Friday, 12/27/2019 at 6pm

Projection Methods

Consider the Kiyotaki-Moore model as discussed in class (see lecture slides), with two important differences:

1. The credit constraint for the farmer is given by

$$-b^F_{t+1} \leq \theta q_t k^F_{t+1}, \tag{1}$$

i.e. the constraint limits the present value of debt in terms of the market value of capital purchased today.

2. The farmer faces the following capital adjustment cost

$$\Phi(k^F_{t+1}, k^F_t) = \frac{\phi}{2}\left(\frac{k^F_{t+1}}{k^F_t} - 1\right)^2 k^F_t.$$

With this problem set, you receive Matlab code solving the model.

1. Familiarize yourself with the code.

(a) Read the included 'readme' file.

(b) Run the code for the "benchmark" parameter combination, which is set in `experef`. To run the code, first create the experiment environment file using `main_setup`. Then run `main_execute` to start the solution algorithm.

(c) Run the simulation script `sim_stationary` on the model solution file. What do you learn from the output of the simulation?

(d) The files `sim_trans` and `plot_trans` can be used to produce generalized IRFs of the model.

(e) If you wanted to run the code for a tighter credit constraint ($\theta = 0.8$), what would you probably need to change in addition to the parameter itself? [Feel free to solve the model with this changed parameter, but not required.]

2. Consider the following change to the model. Instead of the credit constraint in (1), the farmer's optimization problem at time t now needs to satisfy

$$\theta(a(x_{t+1}) + q(x_{t+1}))k^F_{t+1} + b^F_{t+1} \geq 0 \ \ \forall x_{t+1} \in \mathcal{X}_{t+1}, \tag{2}$$

where $\mathcal{X}_{t+1} = \{X_{t+1}|X_t\}$ is the set of all possible aggregate states X_{t+1} tomorrow conditional on aggregate state X_t today. Recall that k^F_{t+1} and b^F_{t+1} are decision variables of the farmer at time t, so not stochastic as of time t. However, productivity a_{t+1} and capital price q_{t+1} are functions of tomorrow's aggregate state and thus random given time-t information.

(a) How is the constraint (2) different from the original formulation (1)? What is the economic interpretation?

(b) What is the new non-stochastic steady-state of the model?

(c) How can you rewrite the constraint such that it becomes a time-t measurable restriction on the farmer's optimization problem that can be included in our algorithm? [Hint: if the constraint is binding for the worst possible realization of $a(x_{t+1}) + q(x_{t+1})$ tomorrow, will it be binding in any other state?]

(d) Create a new version of the code and make the necessary changes to incorporate the different constraint.

- The parameter `min_q` in the old code points you in the right direction.
- Note that the code uses an analytical Jacobian that is created in `constructJacobian`. Make sure to reflect any changes you make in `calcStateTransition` and `calcEquations` also in `constructJacobian`. (You can turn off the analytical Jacobian and use numerical differentiation by setting the relevant flag in `main_setup`).

(e) Numerically solve the model with the new constraint.

- It may be helpful to slightly reformulate the "leverage" state variable of the farmer such that it works better with the new constraint.
- If you have a hard time getting the code to converge at first, you may want to consider adjusting the grid bounds of your state variables, or trying different parameters that are better behaved (e.g. a lower value of θ).

(f) What are the main economic effects of the different formulation of the constraint? Use the new model's simulation output and IRFs in your analysis.