Class exercise

Solving nonlinear models by the perturbation method in Dynare.

Consider the Dynare file, rbc.mod, which is on the class website. This file can be used to solve the neoclassical model studied in the class handout on the perturbation method.

1. Dynare obtains policy rules for consumption and output as well as capital because it does not substitute out for output and consumption using the production function and resource constraint, as is done in the handout.

   (a) The program, rbc.mod, is set to compute the steady state but you have to give it a good initial guess. The program is set to provide Dynare with the actual steady state as a 'guess'. Change that guess a little. Does Dynare manage to get the steady state with an OK but not-too-bad guess? What if you make the initial guess really bad?

   (b) You should verify that, despite the difference in substitution strategy, the policy rule computed for capital by Dynare coincides with the one reported in the handout. This will not be immediately obvious, because there is another difference between what Dynare did and what I did. Dynare has a habit of defining the state as

   $$k_t, a_{t-1}, \varepsilon_t,$$

   rather than k_t, a_t. (Obviously, the *minimal* state only contains k_t and a_t, because given a_t alone, there is no additional information in a_{t-1} and ε_t. In particular, to know the position of the time t production function you only need a_t and to make a forecast about future a_t's you don't need to know anything beyond a_t itself.) Thus, before comparing the policy rules in the handout and the one produced by Dynare you need to see if you can write the solution that Dynare produced in terms of a_t only, not in terms of a_{t-1} and ε_t separately. (Hint: for this to be true requires that wherever a_{t-1} and ε_t appears, the coefficients must have the property that the coefficient on a_{t-1} equals the coefficient on ε_t times ρ.)

   (c) You can directly do the calculations I did in the handout by substituting out for consumption and output. Doing so directly is a pain and results in complicated, hard-to-read, code. Fortunately, Dynare gives you a nice way to deal with this kind of a situation without creating too much of a mess. You should delete *lc* and *ly* from the VAR list. Then, inside the model command you can insert lines that begin with # and then define *lc* and *ly*. Dynare then interprets *lc* and *ly* wherever they appear in the model definition as a short hand for what appears after *lc* = and *ly* = in the respective # statements. Careful, you can't refer to $lc\,(+1)$ if *lc* has been defined in a # statement. So, you will have to have to write a separate *lc* command to substitute out for $lc(+1)$. You could call this lc, lcp. Redo the calculations substituting out lc and ly in this way and verify that your results continue to coincide with what appears in the handout.

2. Now let's do some simulations. This involves drawing a sequence of shocks, ε_t, $t = 1, ..., T$, from Normal random number generator with mean zero and standard deviation equal to the value you assigned. You then input these shocks into the model solution to get a sequence of random variables. This is uncomplicated if you are working with the first order linearization of the policy rule:

$$\begin{aligned} k_{t+1} &= k^* + g_k (k_t - k^*) + g_a a_t, \\ a_t &= \rho a_{t-1} + \varepsilon_t. \end{aligned}$$

Let the initial date be denoted $t = 0$ and suppose k_0 and a_0 are given numbers. The linearized policy rule is the first order Taylor series expansion of the mapping from a_0 and k_0 to k_1 :

$$k_1 = k^* + g_k (k_0 - k^*) + g_a a_0.$$

The simulated value of k_2 is

$$\begin{aligned} k_2 &= k^* + g_k (k_1 - k^*) + g_a (\rho a_0 + \varepsilon_1) \\ &= k^* + g_k^2 (k_0 - k^*) + g_k g_a a_0 + g_a (\rho a_0 + \varepsilon_1), \end{aligned}$$

which is a linear map from a_0, ε_1, k_0 to k_2. Similarly, the mapping from $a_0, k_0, \varepsilon_0, ..., \varepsilon_t$ to k_{t+1} is linear for all t. So, direct simulation of the linearized policy rule preserves linearity of the map from shocks to the endogenous variables, k_t.

The same is not true when simulating the second (or higher) order approximation to the policy rule. If apply the procedure in the previous paragraph to simulating the quadratic approximation to the policy rule, the mapping from shocks to endogenous variables becomes highly nonquadratic. To see this, consider the following simplified representation of the second order approximation to the policy rule (I assume $\rho = 0$, leave out some terms and set $y_t = k_t - k^*$):

$$y_{t+1} = g_k y_t + \frac{1}{2} g_{kk} y_t^2 + g_a \varepsilon_t.$$

Then,

$$y_1 = g_k y_0 + \frac{1}{2} g_{kk} y_0^2 + g_a \varepsilon_0,$$

for given y_0. Going one more period:

$$\begin{aligned}
y_2 &= g_k y_1 + \frac{1}{2} g_{kk} y_1^2 + g_a \varepsilon_1 \\
&= g_k^2 y_0 + g_k \frac{1}{2} g_{kk} y_0^2 + g_k g_a \varepsilon_0 \\
&\quad + \frac{1}{2} g_{kk} \left(g_k y_0 + \frac{1}{2} g_{kk} y_0^2 + g_a \varepsilon_0 \right)^2 \\
&\quad + g_a \varepsilon_1 .
\end{aligned}$$

We haven't left the quadratic world yet. So, go one more period:

$$\begin{aligned}
y_3 &= g_k y_2 + \frac{1}{2} g_{kk} y_2^2 + g_a \varepsilon_2 \\
&= g_k^3 y_0 + g_k^2 \frac{1}{2} g_{kk} y_0^2 + g_k^2 g_a \varepsilon_0 \\
&\quad + g_k \frac{1}{2} g_{kk} \left(g_k y_0 + \frac{1}{2} g_{kk} y_0^2 + g_a \varepsilon_0 \right)^2 + g_k g_a \varepsilon_1 \\
&\quad + \frac{1}{2} g_{kk} [g_k^2 y_0 + g_k \frac{1}{2} g_{kk} y_0^2 + g_k g_a \varepsilon_0 \\
&\quad + + \frac{1}{2} g_{kk} \left(g_k y_0 + \frac{1}{2} g_{kk} y_0^2 + g_a \varepsilon_0 \right)^2 + g_a \varepsilon_1]^2 + g_a \varepsilon_2 .
\end{aligned}$$

Notice that y_3 is a function of ε_1^2. Consider the next period:

$$y_4 = g_k y_3 + \frac{1}{2} g_{kk} y_3^2 + g_a \varepsilon_3 .$$

Given that y_3 is a function of ε_1^2 we see that y_4 is a function of ε_1^4. The mapping from shocks to the y_t's is clearly not quadratic for $t \geq 4$. Indeed the powers on the shocks grows as t gets larger. In particular, the mapping from shocks to endogenous variables is not the second order approximation of the true map those shocks, evaluated around the mean value of the shocks.

Another tip off that there is something wrong with a direct simulation of the second order approximation is to note that that approximation has two steady states because $g_k < 1$ and $g_{kk} > 0$. The steady state is identified by deleting the time index and setting the shock to zero:

$$y = g_k y + \frac{1}{2} g_{kk} y^2$$

There are two values of y that satisfy this equation:

$$y = 0, \ y = 2 \frac{1 - g_k}{g_{kk}} .$$

Note that the second state is a k greater than k^*. Thus, the policy rule cuts the 45 degree line at $y_t = 0$ and then cuts it again from below. But, it is highly unstable there and could, with the right shock, drive off to plus infinity.

Clearly, direct simulation of the second order approximation of the policy rule is bound to give us junk. So, a correction has been made, which ensures that the mapping from the shocks to the endogenous variables in a simulation are the second order approximation of the true map. In this approach, explosions such as those that are possible in the naive simulation described above cannot occur. The adjustment is called *pruning.*

(a) Leave the shock standard deviation at 0.01 in rbc.mod. First do a stochastic simulation of length $T = 100$ without pruning. Then do it again, with pruning and graph gdp in both cases to compare. To ensure a good comparison, set the Dynare seed to unity in the first simulation (the one without pruning in the stoch_simul command), by including the line, *set_ dynare_ seed (1);* .Then, before the second simulation (the one with pruning), reset the Dynare seed to its initial value, by including *set_ dynare_ seed ('reset');* .

(b) Now make the shocks larger, until you find a difference. The message should be that with normal sized shocks it doesn't matter if you prune or not.

(c) Now compare the stochastic simulations when you do first and second order approximations with or without pruning. Again, you should find that it only makes a difference if you go to high values of the shock standard deviations.