Notes on using Dynare

Advance Macroeconomics - Part II

Pablo Garcia

Introduction

What is Dynare?

Software to solve DSGE models.

Key advantages:

- Free.
- User-friendly.
- Easy to code.
- Fast.

Getting started

Download Dynare here:

- https://www.dynare.org/download/

Learn to install Dynare (and much more) here:

- https://www.dynare.org/manual/

To use Dynare, you create .mod files.

- You create these in the m-file editor, and save them as .mod files.

You run the .mod files in the Matlab command window.

The Real Business Cycle Model

We are going to solve the RBC presented in class:

$$1 = \beta E_t \frac{c_t}{c_{t+1}} \left[1 + \alpha \frac{Y_{t+1}}{K_t} - \delta \right],$$

$$K_t = (1 - \delta) K_{t-1} + I_t,$$

$$Y_t = Z_t K_{t-1}^{\alpha},$$

$$\ln Z_t = \ln Z_{t-1} + \epsilon_t,$$

$$\epsilon_t \sim N(0, \sigma^2).$$

5 endogenous variables $\{C_t, Y_t, K_t, I_t, Z_t\}$, 5 equations, 1 shock $\{\epsilon_t\}$, and 4 structural parameters $\{\alpha, \beta, \delta, \rho\}$.

Writing the Dynare Code

Step 1: Declare Variables

We first declare the endogenous variables.

The first line of code will look like this:

```
var c y k i z;
```

We then declare the shock.

The second line of code will look like this:

```
varexo e;
```

In Dynare, each entry must end up with a semicolon.

Step 2: Declare Parameters

We begin by declaring the parameters of the model:

```
parameters alpha beta delta rho;
```

We then specify their numerical values:

```
alpha = 0.33;
beta = 0.99;
delta = 0.025;
rho = 0.95;
```

We are now ready to declare the model.

Step 3: Declare Model

To declare the model, you type in "model;" followed by the equilibrium conditions, followed by "end;".

For our RBC model, the code looks like this:

```
model;
1 = beta * (c / c(+1)) * (1 + alpha * y(+1) / k - delta);
k = (1 - delta) * k(-1) + i;
y = c + k - (1 - delta) * k(-1);
log(y) = log(z) + alpha * log(k(-1));
log(z) = rho * log(z(-1)) + eps;
end;
```

Step 4: Compute the Steady State

Dynare solves for the steady state numerically for you.
But you need to give it a good guess of the steady state. To do so, you type in "initval;" followed by the guess, followed by "end;".

```
initval;
z = 1;
k = (alpha / (1/beta + delta - 1)) ^ (1/(1 - alpha));
c = k ^ alpha - delta * k;
y = z * k ^ alpha;
i = delta * k;
end;
```

Then, you type "steady;". Dynare will compute the steady state.

```
steady;
```

Step 5: Get information about the model

We now ask Dynare to compute the eigenvalues of the model linearised around the steady state.

A necessary condition for the uniqueness of a stable equilibrium (around the steady state) is that there are as many eigenvalues larger than one in modules as there are forward looking variables in the system.

To get Dynare to check this condition, we simply type in:

```
check;
```

Step 6: Specify variance of the shock

We now specify the variance of the shock.

This part of the code starts with "shocks;", followed by a specification of the variance, followed by "end;".

```
shocks;
var eps = 0.01 ^ 2;
end;
```

Step 7: Payoff

This command is the payoff. It's the "stoch simul" command, which solves the model, produces the policy functions, and generates impulse responses functions and unconditional moments.

There are several options following this command. Here, we are going to use the most basic ones:

```
stoch_simul(order=1, irf = 40, periods = 500000);
```

For more on the "stoch simul" command, check:

- https://www.dynare.org/manual/the-model-file.html#stochastic-solution-and-simulation

Running Dynare

Running Dynare

To run the .mod file, type "dynare filename" into the Matlab command window.

For this to run, you must:

- Be in the same directory where the Dynare files are stored, or;
- Have set the path of where your Dynare files are stored, or;
- Type "addpath (directory)" where the directory is the location of your Dynare files.

I take the latter option. Hence, in the Matlab command window I type:

```
addpath /Applications/dynare/4.6.2/Matlab
dynare main
```

Dynare Output

Running Dynare I

In the basic setting, Dynare will produce:

- Policy and transition functions.
- Impulse response functions.
- Statistical moments of endogenous variables.

The first item is key, so let us be clear about it. The second and third items are self-explanatory.

Running Dynare II

As seen in class, our RBC model can be written as:

$$Y_t = \Psi S_t + \Omega \epsilon_t.$$

Here Y is the set of all endogenous variables, and S is the set of state variables. Both Y and S are centered at their steady state values (i.e. they have zero mean).

Dynare automatically sorts variables in the following order:

- Static variables (index t).
- Predetermined variables (indexes t and $t-1$).
- Forward looking variables (indexes $t+1$, t and $t-1$).

Running Dynare III

Recall that we have declared our variables as follows:

```
var c y k i z;
```

Therefore Dynare defines Y_t and S_t as:

$$Y_t = [i_t \; k_t \; z_t \; c_t \; y_t],$$
$$S_t = [k_{t-1} \; z_{t-1}].$$

Dynare reports matrices Ψ and Ω in the command window. It also saves them as oo_.dr.ghx and oo_.dr.ghu, respectively.

Final remark

Final remark

In the codes I have posted, I do the following things after running "dynare main":

- Recover and plot the policy functions as a function of the capital stock.
- Plot impulse responses in % deviations from steady state.
- Use the policy functions to simulate the economy.

I provide these codes for those of you interested in Dynare. I believe that they can help you to better understand how to code a model.

For this class, however, writing, understanding, and running the .mod file is enough.