Macroeconomics II

Lesson 03 — Dynare, Solving the RBC, and Assessment

Rafael Serrano-Quintero

Department of Economics
University of Barcelona

Preliminaries

Dynare

- To solve the model on the computer we will rely on Dynare.
- Dynare is a software that uses Matlab (or Octave) as an interface. It is specifically designed to solve, simulate, and even estimate DSGE and OLG models.
- Here you have a quickstart guide with installation instructions for Windows and macOS.
- For GNU/Linux users with a Debian based distro `sudo apt install dynare` followed by `sudo apt install dynare-matlab` should work.
- I will be working with Matlab R2020b and Dynare 4.6.4 which is the latest stable release.

Dynare

- Dynare works with a particular file extension `.mod` or mod-files.

- These files contain the instructions for Dynare with the equations of our model, parameters, and set of instructions to compute.

- The structure of a mod-file is typically distributed into five blocks
  1. **Preamble:** declare variables and parameters and assign values.
  2. **Model:** the equations of the model.
  3. **Steady state:** either tells Dynare to find it or provide the starting point for the simulations or IRFs.
  4. **Shocks:** define the stochastic innovations (i.e. ε_t in our model)
  5. **Computation:** the particular computations we want Dynare to perform.

Dynare

- Solve the baseline RBC with general preferences

$$u(c_t, h_t) = \frac{\left(\frac{c_t}{X_t}\right)^{1-\sigma} - 1}{1-\sigma} + \frac{\xi}{1-\eta}\left((1-h_t)^{1-\eta} - 1\right)$$

- Note that the model from the previous lessons is obtained by setting $\sigma = \eta = 1$.

- In the next slides I repeat for convenience the system of equilibrium conditions we will introduce in Dynare.

Competitive Equilibrium

Characterization of the Competitive Equilibrium (I)

$$\alpha \frac{y_t}{k_t} = r_t^k \tag{EQ1}$$

$$(1-\alpha)\frac{y_t}{h_t} = w_t \tag{EQ2}$$

$$\frac{\xi}{(1-h_t)^\eta} = \frac{w_t}{c_t^\sigma} \tag{EQ3}$$

$$\frac{1}{c_t^\sigma} = \frac{\beta}{(1+g)(1+n)}\mathbb{E}_t\left(\frac{1+r_{t+1}}{c_{t+1}^\sigma}\right) \tag{EQ4}$$

$$r_t = r_t^k - \delta \tag{EQ5}$$

Competitive Equilibrium

Characterization of the Competitive Equilibrium (II)

$$(1+n)(1+g)k_{t+1} = (1-\delta)k_t + i_t \tag{EQ6}$$

$$h_t + l_t = 1 \tag{EQ7}$$

$$\lim_{t\to\infty} \beta^t \frac{k_t}{c_t^\sigma} = 0 \tag{EQ8}$$

$$y_t = c_t + i_t \tag{EQ9}$$

$$y_t = A_t k_t^\alpha h_t^{1-\alpha} \tag{EQ10}$$

$$\log(A_t) = (1-\rho)\log(A) + \rho\log(A_{t-1}) + \varepsilon_t \tag{EQ11}$$

Remark 1

The TVC is reported for completeness, but we are not going to use it in the numerical approach.

Preparing a mod-file

Preparing a mod-file

Dynare — The Preamble

- First step is to tell Dynare which are the variables and parameters of the model.
- The first **non-comment command** in your mod-file should be `var` followed by the **endogenous variables** names.
- Avoid names that are used as commands or built-in functions of Dynare (e.g. `Sigma_e` or `sigma_e`). Same for parameters such as `alpha`, you can use `alph` instead. Same for names such as `i`
- Commands in mod-files are denoted by `//`. Blocks are ended with ;

Preparing a mod-file

Dynare — The Preamble

```
var y               //output
    c               //consumption
    x               //investment
    h               //hours worked
    lab_prod        //labor productivity
    w               //wage rate
    r               //net rate of return
    a               //TFP (exogenous state)
    rk              //rental rate of capital
    k               //capital stock
    l;              //leisure
varexo epsi;
```

- Note we include `a`.
- Note also investment is denoted `x` and there is `lab_prod` defined as y_t / h_t.
- `epsi` corresponds ε_t denoted as exogenous variable with `varexo`.

Preparing a mod-file

Dynare — The Preamble

- In Dynare, it is convenient to declare **predetermined variables** explicitly.
- A predetermined variable in our model is k_{t+1} which is known at date t.
- Declaring predetermined variables allows Dynare to understand $\mathbb{E}_t(k_{t+1}) = k_{t+1}$.
- The proper syntax is `predetermined_variables k`.
- Note this is done **after** declaring `k` as a variable.

Preparing a mod-file

Dynare — The Preamble

- The parameters of the model are declared in the block of parameters as

```
parameters alph A n g delt rho sig_e bett xi sig eta;
```

- Recall the naming convention for $\{\alpha, \beta, \sigma\}$.

- After this command you specify the values of these parameters like

```
alph = 1/3;
```

- This is a potential approach, but we will use an alternative approach.

- We are going to set the values in a script called `calibration.m` save the values, and tell Dynare to get the values from this file. Why are we doing this?

Feeding Parameter Values

We feed the parameter values from a file `params.mat` to Dynare as follows

```
load params;
set_param_value('alph',alph);
set_param_value('A',A);
set_param_value('n',n);
set_param_value('g',g);
set_param_value('delt',delt);
set_param_value('rho',rho);
set_param_value('sig_e',sig_e);
set_param_value('bett',bett);
set_param_value('xi',xi);
set_param_value('sig',sig);
set_param_value('eta',eta);
```

Preparing a mod-file

- In this block, we write the equilibrium conditions of the model.
- We initialize the block with the `model;` keyword and finish it with command `end;`
- The **timing convention** of Dynare:
  - Dynare interprets $x(-1)$ as a variable **chosen at date** $t-1$ and $x(+1)$ as chosen at $t+1$.
  - However, if x is a **predetermined variable** then $x(+1)$ is interpreted as chosen at time t.
  - Dynare interprets all variables $x(+1)$ as expectations unless declared as predetermined.

Preparing a mod-file

Dynare — The Model

The Model Block

```
model;
     alph*(y/k) = rk;
     (1-alph)*(y/h) = w;
     y = a*(k^(alph))*(h^(1-alph));
     xi/((1-h)^(eta)) = w*c^(-sig);
     (1+n)*(1+g)*k(+1)=(1-delt)*k+x;
     h+l = 1;
     r = rk-delt;
     c^(-sig) = (bett/(1+g))*(1 + r(+1))*c(+1)^(-sig);
     y = c + x;
     lab_prod = y/h;
     log(a) = (1 - rho)*log(A) + rho*log(a(-1)) + epsi;
end;
```

Preparing a mod-file

- Dynare will now linearize the model using a Taylor expansion around the steady state.
- To perform log-linearization we need to rewrite the model by substituting each variable `u` with `exp(u)`.
- You want to do this when you want the IRFs in percentage changes or when you want to mix linearization and log-linearization.
- In the next slide I rewrite the model combining both treating r in levels while all other variables are in logs.

Mixing Linearization and Log-Linearization

```
model;
     alph*(exp(y)/exp(k)) = exp(rk);
     (1-alph)*(exp(y)/exp(h)) = exp(w);
     exp(y) = exp(a)*(exp(k)^(alph))*(h^(1-alph));
     xi/((1-exp(h))^(eta)) = exp(w)*(exp(c))^(-sig);
     (1+n)*(1+g)*exp(k(+1))=(1-delt)*exp(k)+exp(x);
     exp(h)+exp(l) = 1;
     r = exp(rk)-delt;
     c^(-sig) = (bett/(1+g))*(1 + r(+1))*exp(c(+1))^(-sig);
     exp(y) = exp(c) + exp(x);
     exp(lab_prod) = exp(y)/exp(h);
     a = (1 - rho)*log(A) + rho*a(-1) + epsi;
end;
```

Preparing a mod-file

Dynare — The Steady State

Two approaches

1. Provide Dynare with initial values and let Dynare solve for it.

2. Provide an external function where we compute the steady state by hand.

- We will go through the second way for a particular reason:
  - Flexibility!

- The function needs to be named in a particular way. Our mod-file is called `basic_RBC.mod`, then, the steady state function needs to be called `basic_RBC_steadystate.m`

- The structure of the function must be

```
function [ys,params,check] = basic_RBC_steadystate(ys,exo,M,options_)
```

Preparing a mod-file

Dynare — The Steady State

To assign numerical values on the parameters:

```
NumberOfParameters = M_.param_nbr;
for ii = 1:NumberOfParameters
    paramname = M_.param_names{ii};
    eval([paramname ' = M_.params(' in2str(ii) ');']);
end
```

- Bear in mind that the steady state needs to be consistent with the model (i.e. if the model is in levels or in logs).

- The command to compute the steady state is `steady`;

Preparing a mod-file

Dynare — The Shocks

- Recall we defined as an exogenous variable ε_t the innovations of the model.
- We named that `epsi` and recall that in the model $\varepsilon_t \underset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_\varepsilon^2)$.
- We need to tell Dynare now exactly this in a block of shocks.

```
shocks;
    var epsi;
    stderr (100*sig_e)
end;
```

Preparing a mod-file

Dynare — Solution and Simulation

- The `stoch_simul` command will solve and simulate the model. This computes the policy functions and generates IRFs and unconditional moments.

- The output of running this command is stored in `oo_` which is a struct array.

- The command accepts multiple options. Some useful ones
  - `hp_filter=1600` computes theoretical moments for HP filter data with $\lambda = 1600$.
  - `order=1` the order of the Taylor expansion.
  - `irf=60` plot the IRFs for 60 periods.
  - `periods=100` compute moments based on the simulation for 100 periods.

Dynamics of the Basic RBC

Short-run — IRFs for a Shock to TFP

Computation of IRFs

- Dynare uses Sims' method to compute policy functions.

- To do so, first Dynare expresses the model in state space form as

$$
\begin{aligned}
s_t &= \mathbf{A}s_{t-1} + \mathbf{B}\varepsilon_t \\
x_t &= \Phi s_t
\end{aligned}
$$

  where s_t is the $(m \times 1)$ vector of states, x_t is the $(n \times 1)$ vector of controls (both dynamic and static).

- Let $\mathbf{C} = \Phi_{n\times m}\mathbf{A}_{m\times m}$ and $\mathbf{D} = \Phi_{n\times m}\mathbf{A}_{n\times w}$. The full system is written as

$$
\begin{aligned}
s_t &= \mathbf{A}s_{t-1} + \mathbf{B}\varepsilon_t \\
x_t &= \mathbf{C}s_{t-1} + \mathbf{D}\varepsilon_t
\end{aligned}
$$

Short-run — IRFs for a Shock to TFP

Computation of IRFs

- IRFs are computed assuming first the economy is at the steady-state equilibriuym $x_t = x_s$ and then, at period $t = \tau$ it is hit by an innovation.

- Then, $\varepsilon_t = 0 \forall t \neq \tau$. On impact, the IRFs are computed as

$$
\begin{aligned}
x_t - x_s &= \mathbf{D}\varepsilon_t && \text{for } t = \tau \\
x_t - x_s &= \mathbf{C}s_{t-1} && \forall\, t \geq \tau + 1
\end{aligned}
$$

- That is, innovations only hit once and we compute the transition back to the steady-state.

Short-run — IRFs for a Shock to TFP

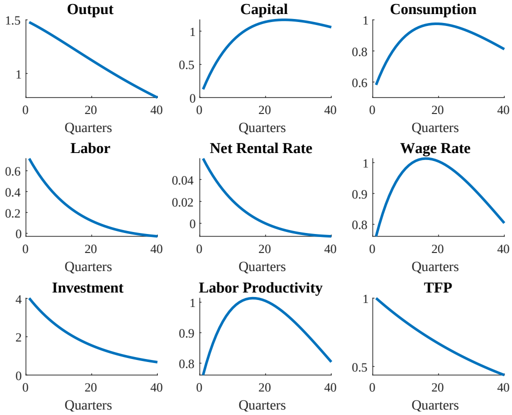

Figure 1: IRFs of a Technology Shock

- On impact $\downarrow$ mc $\Rightarrow \uparrow \{k, h\}$
- For fixed supply $\Rightarrow \uparrow \{r^k, w\}$
- $\uparrow$Households wealth $\Rightarrow \uparrow \{c_t\}$ for more than one period (consumption smoothing).
- Also $\uparrow$ investment (more profitable, because $\uparrow r^k$)
- $\uparrow$ Labor supply because substitution effect $>$ income effect.

Long-Run — Simulated Series

- To analyze the long-run dynamics of the model we can follow a similar approach to IRFs.

- We need to draw a sequence of innovations $\{\varepsilon_t\}_{t=0}^{T}$.

- Assuming the economy starts at the steady state, the series are computed

$$
\begin{aligned}
(s_t - s_s) &= \mathbf{B}\varepsilon_t \\
(x_t - x_s) &= \mathbf{D}\varepsilon_t
\end{aligned}
$$

  at the **impact of the shock** $(t = \tau)$ when the state $s_{t-1} = s_s$ and then

$$
\begin{aligned}
(s_t - s_s) &= \mathbf{A}(s_{t-1} - s_s) + \mathbf{B}\varepsilon_t \quad \forall t \geq \tau + 1 \\
(x_t - x_s) &= \mathbf{C}(s_{t-1} - s_s) + \mathbf{D}\varepsilon_t \quad \forall t \geq \tau + 1
\end{aligned}
$$

Long-Run — Simulated Series

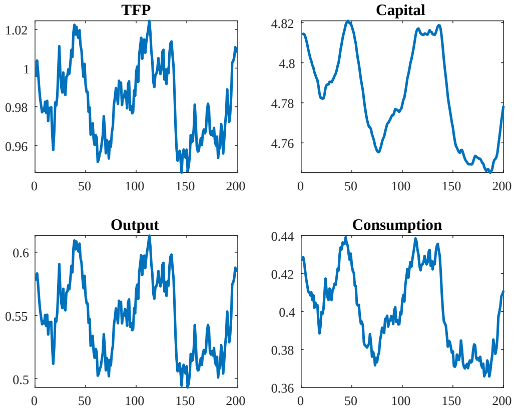

Figure 2: Detrended Series

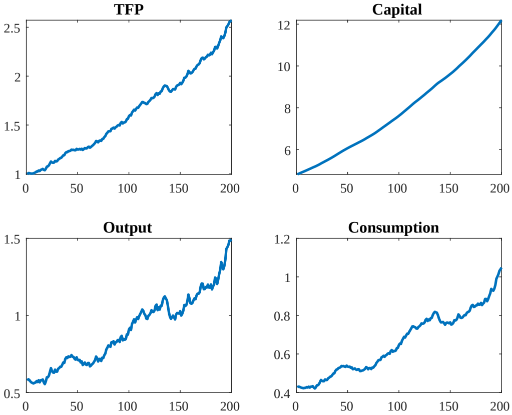

Figure 3: Per Capita Series

Random Walk Technology

Random Walk Technology

- So far, we have assumed $0 < \rho < 1$, what if we would assume $\rho = 1$?
- This would imply that every shock ε_t has a permanent effect on $A_t \Rightarrow$ that the steady state equilibrium **depends upon the realization of the innovations**.
- Then, we cannot use the log-linearized model we used before since the steady-state would not be stationary!
- We need to **re-scale variables** (similar to detrending) and then recover the responses of unscaled variables.
- Basically, remove the unit roots of A_t.

Random Walk Technology

- We know that along the BGP $\gamma_k = \gamma_y$ then, from the production function

$$\frac{y_t}{A_t^{\frac{1}{1-\alpha}}} = \frac{y_{t-1}}{A_{t-1}^{\frac{1}{1-\alpha}}}$$

- Rescale all **growing** variables by the factor $A_{t-1}^{\frac{1}{1-\alpha}}$

$$\tilde{y}_t = \frac{y_t}{A_t^{\frac{1}{1-\alpha}}} \quad \tilde{c}_t = \frac{c_t}{A_t^{\frac{1}{1-\alpha}}} \quad \tilde{i}_t = \frac{i_t}{A_t^{\frac{1}{1-\alpha}}} \quad \tilde{k}_t = \frac{k_{t+1}}{A_t^{\frac{1}{1-\alpha}}} \quad \tilde{w}_t = \frac{w_t}{A_t^{\frac{1}{1-\alpha}}}$$

- Using these definitions and the production function

$$\tilde{y}_t = \left(\frac{A_t}{A_{t-1}}\right)^{\frac{\alpha}{\alpha-1}} \tilde{k}_t^{\alpha} h_t^{1-\alpha} = \left(\gamma_t^a\right)^{\frac{\alpha}{\alpha-1}} \tilde{k}_t^{\alpha} h_t^{1-\alpha}$$

where $\gamma_t^a = \exp\{\varepsilon_t\}$.

Random Walk Technology

- When $\sigma \neq 1$ and $\eta \neq 1$ we need to adjust the instantaneous utility function

$$u(c_t, h_t) = \frac{\left(\frac{c_t}{A_t^{\frac{1}{1-\alpha}}}\right)^{1-\sigma} - 1}{1-\sigma} + \frac{\xi}{1-\eta}\left((1-h_t)^{1-\eta} - 1\right)$$

- We can interpret A_t as exogenous and stochastic habits. A_t can be interpreted as a habit stock in terms of technology level.

Random Walk Technology

Equilibrium Conditions with Random Walk Technology (I)

$$\alpha \frac{\tilde{y}_t}{\tilde{k}_t} \left(\gamma_t^a\right)^{\frac{1}{1-\alpha}} = r_t^k \tag{RW1}$$

$$(1-\alpha)\frac{\tilde{y}_t}{h_t} = \tilde{w}_t \tag{RW2}$$

$$\frac{\xi}{(1-h_t)^\eta} = \frac{\tilde{w}_t}{\tilde{c}_t^\sigma} \tag{RW3}$$

$$\frac{1}{\tilde{c}_t^\sigma} = \frac{\tilde{\beta}}{1+g}\mathbb{E}_t\left(\frac{1+r_{t+1}}{\tilde{c}_{t+1}^\sigma}\left(\gamma_t^a\right)^{\frac{1}{\alpha-1}}\right) \tag{RW4}$$

$$r_t = r_t^k - \delta \tag{RW5}$$

Random Walk Technology

Equilibrium Conditions with Random Walk Technology (II)

$$(1+n)(1+g)\tilde{k}_{t+1} = \left(\gamma_t^a\right)^{\frac{1}{\alpha-1}}(1-\delta)\tilde{k}_t + \tilde{i}_t \tag{RW6}$$

$$h_t + l_t = 1 \tag{RW7}$$

$$\lim_{t\to\infty} \beta^t \frac{\tilde{k}_t}{\tilde{c}_t^{\sigma}} = 0 \tag{RW8}$$

$$\tilde{y}_t = \tilde{c}_t + \tilde{i}_t \tag{RW9}$$

$$\tilde{y}_t = \left(\gamma_t^a\right)^{\frac{\alpha}{\alpha-1}}\tilde{k}_t^{\alpha} h_t^{1-\alpha} \tag{RW10}$$

$$\gamma_t^a = \exp\{\varepsilon_t\} \tag{RW11}$$

Evaluation of the RBC Model

- Let's evaluate now the predictions of the RBC model for $\sigma = \eta = 1$.
- To do so, we are going to compare the moments of the model with the data moments from **?**.
- The next table presents the data moments in parenthesis.
- The moments of the model are the theoretical moments computed by Dynare.

Evaluation of the RBC Model

Table 1: Model Statistics

	Std. Dev.	Rel. Std. Dev	Autocorr	Corr. Output
Output	1.39	1.00	0.72	1.00
	(1.81)	(1.00)	(0.84)	(1.00)
Consumption	0.60	0.43	0.79	0.95
	(1.35)	(0.74)	(0.80)	(0.88)
Investment	3.75	2.70	0.71	0.99
	(5.30)	(2.93)	(0.87)	(0.80)
Hours	0.67	0.49	0.71	0.97
	(1.79)	(0.99)	(0.88)	(0.88)
Labor Productivity	0.75	0.54	0.76	0.98
	(1.02)	(0.56)	(0.74)	(0.55)
Wages	0.75	0.54	0.76	0.98
	(0.68)	(0.38)	(0.66)	(0.12)
Net Rental Rate	0.06	0.04	0.71	0.96
	(0.30)	(0.16)	(0.60)	(−0.35)
TFP	0.94	0.67	0.72	1.00
	(0.98)	(0.54)	(0.74)	(0.78)

Note: Business cycle statistics from **?**

Evaluation of the RBC Model

Good Features:

- Volatility of output similar to the U.S. data. Variance ratio $(1.39/1.81)^2 \approx 0.59 \Rightarrow 59\%$ of the variance explained by a single shock.
- Investment is $\approx 3\times$ more volatile than output.
- Consumption is less volatile than output.
- Correlation of output and inputs is positive and large.

Evaluation of the RBC Model

Not so Good Features:

- In the model, however, consumption is not volatile enough $0.43 < 0.74$.
- Strong persistence, but not as strong as in the data.
- Correlation between output and consumption, investment, hours, and labor productivity.
- Procyclicality of wages and rental rate at odds with the data.

Conclusions

- The simple RBC model can account for some features of the data, but significantly at odds with other features.

- Research in DSGE modelling has grown non-stop and is still doing so. Two main branches
  - Real Business Cycles: models in which only real variables matter.

  - New Keynesian Models: models with nominal rigidities (in prices or wages). These rigidities make nominal shocks affect real variables.

- In next lectures, we will focus on extensions of the RBC model.