ECON 747 – LECTURE 3: DYNARE

Thomas Drechsel

University of Maryland

Spring 2021

RECAP

- What **characterizes** a solution to a DSGE model?
  - Optimality conditions, constraints and stochastic processes
- What **is** a solution?
  - Policy functions
- How do we **get to** a solution?
  - Different ways... a bit of an art!
  - This is the part we use Dynare for today
  - **We want to understand what Dynare does and not use it as a black box**

WHAT IS DYNARE?

- Dynare is a essentially a big collection of Matlab functions that have been written for us by some dedicated macroeconomists
- All we need to do is to download these functions and tell Matlab which folder they are located in; Matlab then understands Dynare-related commands
- It's freely available on **www.dynare.org**
- Online you can also find a great reference manual and an active forum

WHAT CAN DYNARE DO?

- Dynare can solve DSGE models
- Dynare can also certain model types that we will not cover in detail in this course, such as OLG models
- Dynare can also *estimate* DSGE models: maximum likelihood, Bayesian methods
  - We will look at this in some detail in a few weeks
  - You will estimate a model in one of your assignments
- We will use Dynare to solve for the policy functions

DYNARE AND DIY

- Apart from letting Dynare solve for the policy rules, you should try to do as much as possible yourself!
- For example: Dynare can simulate the economy using the shocks and policy rules, but we can easily do that ourselves once we know the policy rules
- You will do that in your assignment
- You should also aim to calculate things like implied moments and IRFs yourself, to make sure you understand how they are constructed

How does Dynare solve for the policy rules?

- Dynare uses a technique called **perturbation**
- Remember: there are other methods, which may be better for different purposes
  - For example, for models with discrete choices (but fewer state variables) value function iteration is more applicable
- Idea of perturbation
  - Given that the steady state has been calculated
  - Take a Taylor expansion of the policy rules
  - Need to find the coefficients of this Taylor expansion
  - This can be done *sequentially*

PERTURBATION

- I will walk you through the main idea behind perturbation techniques formally
- Some useful references:
  - Textbook by Judd (1998)
  - Notes by Wouter Den Haan, available at http://www.wouterdenhaan.com/numerical/perturbation.pdf

- Economic model:

$$\mathbb{E}_t f(x_{t+2}, x_{t+1}, x_t) = 0$$

- Let's focus on a simple case
  - No uncertainty
  - x_t is a scalar

- A simple RBC model without shocks fits that description. Take the Euler equation and substitute the budget constraint:

$$\begin{aligned} &\left\{K_t^\alpha + (1-\delta)K_t - K_{t+1}\right\}^{-\sigma} \\ &= \beta\left(\left\{K_{t+1}^\alpha + (1-\delta)K_{t+1} - K_{t+2}\right\}^{-\sigma}\left[\alpha K_{t+1}^{\alpha-1} + (1-\delta)\right]\right) \end{aligned}$$

PERTURBATION

- The solution to $f(x_{t+2}, x_{t+1}, x_t) = 0$ is the policy function

$$x_{t+1} = g(x_t)$$

- Denote $F(x)$ as

$$F(x_t) = f(g(g(x_t)), g(x_t), x_t)$$

- Given that $g(\cdot)$ is the solution, $F(x_t) = 0$
- $F(x_t) = 0$ is an identity, it holds for all x_t, that is, *everywhere in the state space*

PERTURBATION

- Suppose we have calculated the steady state $\bar{x}$
  - This is a truly nonlinear problem and can be very difficult
  - More on this will follow further below
- The steady state is a fixed point for the policy rule

$$\bar{x} = g(\bar{x})$$

- We want to find the function $g(\cdot)$
- Perturbation $\rightarrow$ use Taylor approximation of $g(\cdot)$ around the point $x_t = \bar{x}$

PERTURBATION

$$g(x_t) \approx g(\bar{x}) + (x_t - \bar{x})g'(\bar{x}) + \frac{1}{2}(x_t - \bar{x})^2 g''(\bar{x}) + \dots$$

$$g(x_t) \approx \bar{x} + (x_t - \bar{x})\boldsymbol{g'(\bar{x})} + \frac{1}{2}(x_t - \bar{x})^2\boldsymbol{g''(\bar{x})} + \dots$$

- Since the steady state is known, only the **bold terms** (coefficients of the expansion) are unknown!
- The idea of perturbation is to solve for them *sequentially*, starting with $\boldsymbol{g'(\bar{x})}$

- Since $F(x) = 0 \ \ \forall x$, we have that $F'(x) = 0 \ \ \forall x$
- Differentiate $F(x_t) = f(g(g(x_t)), g(x_t), x_t)$:

$$\begin{aligned} F'(x_t) &= \frac{\partial f}{\partial x_{t+2}} \frac{\partial x_{t+2}}{\partial x_{t+1}} \frac{\partial x_{t+1}}{\partial x_t} + \frac{\partial f}{\partial x_{t+1}} \frac{\partial x_{t+1}}{\partial x_t} + \frac{\partial f}{\partial x_t} \\ &= \frac{\partial f}{\partial x_{t+2}} \frac{\partial g(x_{t+1})}{\partial x_{t+1}} \frac{\partial g(x_t)}{\partial x_t} + \frac{\partial f}{\partial x_{t+1}} \frac{\partial g(x_t)}{\partial x_t} + \frac{\partial f}{\partial x_t} \end{aligned}$$

- Remember: F and f are known functions

- Evaluate the derivative at the steady state

$$\begin{aligned} F'(\bar{x}) &= \frac{\partial f_{(1)}}{\partial \bar{x}} \frac{\partial g(\bar{x})}{\partial \bar{x}} \frac{\partial g(\bar{x})}{\partial \bar{x}} + \frac{\partial f_{(2)}}{\partial \bar{x}} \frac{\partial g(\bar{x})}{\partial \bar{x}} + \frac{\partial f_{(3)}}{\partial \bar{x}} \\ &= \frac{\partial f_{(1)}}{\partial \bar{x}} \left(\frac{\boldsymbol{\partial g(\bar{x})}}{\boldsymbol{\partial \bar{x}}} \right)^{\mathbf{2}} + \frac{\partial f_{(2)}}{\partial \bar{x}} \frac{\boldsymbol{\partial g(\bar{x})}}{\boldsymbol{\partial \bar{x}}} + \frac{\partial f_{(3)}}{\partial \bar{x}} \end{aligned}$$

- But this is the term $\boldsymbol{g'(\bar{x})}$ we were looking for above. Everything else is known!
- $F'(\bar{x}) = 0$ is an exact quadratic relation in the first coefficient of the Taylor expansion!
- Concavity of utility and production functions will imply one explosive and one stable solution

PERTURBATION

- Given the solution for $\boldsymbol{g'(\bar{x})}$, one can use $F''(\bar{x}) = 0$ to obtain an equation for $\boldsymbol{g''(\bar{x})}$, the coefficient on the second order term of the policy rule
- Believe it or not, this equation is actually linear
  - You can try this out as a homework

PERTURBATION: ADDING UNCERTAINTY

- If the model has uncertainty, we also perturb the standard deviation of the shock around a steady state with zero uncertainty
- In an RBC model with stochastic TFP, we derive a Taylor expansion around

$$(k_t, z_t, \sigma_z) = (\bar{k}, \bar{z}, 0)$$

- Note: in a first order approximation the *amount* of uncertainty does not matter, σ_z does not show up in the policy rule parameters ("certainty equivalence")
- In a second order approximation, σ_z shows up only in the constant

PERTURBATION: SOME FINAL REMARKS

- The sequential procedure shown above is very mechanical: a computer can do this for complicated f's and for many state variables!
- But: this requires certain conditions on f, for example differentiability
- Keep in mind that the approximation is around the steady state of the model
- Beware: choosing a higher order does not generally mean the approximation gets more accurate
- Note that x_t could be a transformation, for example the log of capital: for the appropriately transformed f, Dynare works just as well

DYNARE: MOD FILES

- The main unit in Dynare is the .mod file
- It is the file in which you write down your DSGE model
- The model file consists of different blocks
  - Variable block
  - Parameter block
  - Model block
  - Steady state block
  - Shocks block
  - Solution (or estimation) block
- We will go through these different blocks now
- I will upload an "empty" .mod file for you, which you can use as a starting point for your assignments.

VARIABLE AND PARAMETER BLOCKS

`var` *list variable names, endogenous variables and exogenous variables (disturbances)* ;

`varexo` *list shock names* ;

`parameters` *list parameter names* ;
list parameter values ;

- Parameter valyes can also be loaded from an external file, using the syntax

  `load` filename ; `set_param_value('`parametername`'`, parametername`)`

MODEL BLOCK

Starts with `model;`

Ends with `end;`

In between, type equations ending with `;`

- `x(-1)` for predetermined variables
- `x(+1)` for expectations
- You can type an equation over several lines if you do not end it with `;`

STEADY STATE BLOCK: THE HARD PART

- Since Dynare linearizes around the deterministic steady state, this steady state needs to be calculated
- Two options:
  1. Let Dynare calculate the steady state numerically
  2. Calculate the steady state with pen and paper and tell Dynare what it is
- Calculating the steady state is a nonlinear problem. It is difficult for a computer. You should always try option 2 first.

STEADY STATE BLOCK: OPTION 1

If we want Dynare to calculate the steady state numerically (option 1 on previous slide), we need to add the following block:

Start with `initval;`

Ends with `end;`

In between, add initial values for all variables

- If a variable does not appear, Dynare will assume it is zero in steady state
- A steady state value can depend on numerical values, parameters, and on other state state variables (if they have been given above a given steady state variable)

STEADY STATE BLOCK: OPTION 2

- The preferred way (not always possible) is to calculate the steady state ourselves
- What we need to do is to create a separate .m file that has the same name as the mod file and ends in `_steadystate`
- Inside this file, we can provide Dynare with the steady state calculations (based on model parameters)
- I will upload an "empty" steady state file together with the mod file

SHOCKS BLOCK

Starts with `shocks;`

Ends with `end;`

In between, declare shock standard deviations (as numerical values or by referring to a parameter name). Use syntax

`var` shock name `;` `stderr` standard deviation `;`

SOLUTION (OR ESTIMATION) BLOCK

At the bottom of the mod file add

```
stoch_simul(order=1,IRF=20) ;
```

Inside the brackets, many more options can be specified, mostly related top which output you want Dynare to report

After the brackets you can list individual variables if you want the output created only for these variables. Otherwise Dynare will do it for all variables.

RUNNING THE MOD FILE

In Matlab make sure you are in the right directory, then just type

```
dynare modfilename.mod
```

If you embed a mod file into a bigger matlab program, it may be useful to use

```
dynare modfilename.mod noclearall
```

OUTPUT OF DYNARE

Depending on the settings inside the `stoch_simul` command Dynare creates for us

- Policy rules
- Implied moments (means, standard deviations, correlations)
- Impulse response functions
- ...

Dynare writes these results to the screen, but also stores lots of stuff in additional files

OUTPUT OF DYNARE: POLICY RULES

POLICY AND TRANSITION FUNCTIONS

	c	k	z
Constant	1.972191	21.436072	1.000000
k(-1)	0.032524	0.977577	0
z(-1)	0.424665	1.832618	0.900000
ez	0.471850	2.036243	1.000000

How to read these policy functions?

OUTPUT OF DYNARE: POLICY RULES

$$
\begin{aligned}
C_t - C_{ss} &= a_{ck}(K_t - K_{ss}) + a_{cz}(Z_{t-1} - Z_{ss}) + a_{ce}(\varepsilon_{z,t} - 0) \\
K_{t+1} - K_{ss} &= a_{kk}(K_t - K_{ss}) + a_{kz}(Z_{t-1} - Z_{ss}) + a_{ke}(\varepsilon_{z,t} - 0) \\
Z_t - Z_{ss} &= a_{zk}(K_t - K_{ss}) + a_{zz}(Z_{t-1} - Z_{ss}) + a_{ze}(\varepsilon_{z,t} - 0)
\end{aligned}
$$

OUTPUT OF DYNARE: POLICY RULES

$$
\begin{aligned}
C_t - C_{ss} &= a_{ck}(K_t - K_{ss}) + a_{cz}(Z_{t-1} - Z_{ss}) + a_{ce}(\varepsilon_{z,t} - 0) \\
K_{t+1} - K_{ss} &= a_{kk}(K_t - K_{ss}) + a_{kz}(Z_{t-1} - Z_{ss}) + a_{ke}(\varepsilon_{z,t} - 0) \\
Z_t - Z_{ss} &= a_{zk}(K_t - K_{ss}) + a_{zz}(Z_{t-1} - Z_{ss}) + a_{ze}(\varepsilon_{z,t} - 0)
\end{aligned}
$$

Observations:

- Dynare reports policy functions in deviations from steady state values
- The constant gives the steady state value of a given variable (column)
- Dynare reports the coefficient on lagged exogenous variable and therefore includes the error term

OUTPUT OF DYNARE: POLICY RULES

The policy function above should be read as

$$\begin{aligned} C_t - 1.97 &= 0.03(K_t - 21.4) + 0.42(Z_{t-1} - 1) + 0.47(\varepsilon_{z,t} - 0) \\ K_{t+1} - 21.4 &= 0.98(K_t - 21.4) + 1.83(Z_{t-1} - 1) + 2.04(\varepsilon_{z,t} - 0) \\ Z_t - 1 &= 0(K_t - 21.4) + 0.9(Z_{t-1} - 1) + (\varepsilon_{z,t} - 0) \end{aligned}$$

Note that $0.42 = 0.47\rho$ and $1.83 = 2.04\rho$ because $\rho = 0.9$

A "LIVE" DEMONSTRATION

Remember the model from lecture 2

$$
\begin{aligned}
C_t^{-\sigma}\frac{1}{V_t} &= \beta\mathbb{E}_t\left(C_{t+1}^{-\sigma}\left[\alpha Z_{t+1}K_{t+1}^{\alpha-1}+(1-\delta)\frac{1}{V_{t+1}}\right]\right)\\
C_t+\frac{K_{t+1}}{V_t} &= Z_tK_t^{\alpha}+(1-\delta)\frac{K_t}{V_t}\\
Z_t &= 1-\rho_z+\rho_zZ_{t-1}+\varepsilon_{z,t}\\
V_t &= 1-\rho_v+\rho_vV_{t-1}+\varepsilon_{v,t}
\end{aligned}
$$

- The $1-\rho$ terms ensure that the disturbances Z_t and V_t are equal to 1 in steady state: $Z_{ss} = V_{ss} = 1$
- Dropping time subscripts in the Euler equation and solving for capital gives

$$K_{ss} = \left(\frac{1/\beta - (1-\delta)}{\alpha} \right)^{\frac{1}{\alpha - 1}}$$

- From the resource constraint we get

$$C_{ss} = K_{ss}^{\alpha} - \delta K_{ss}$$

MORE VARIABLES

- The model above is ready for Dynare
- But remember we have substituted out Y_t and I_t in the previous lecture. If we are interested in the variables as well, let us add the equations

$$\begin{aligned} Y_t &= Z_t K_t^{\alpha} \\ K_{t+1} &= (1-\delta)K_t + V_t I_t \end{aligned}$$

- In steady state

$$\begin{aligned} Y_{ss} &= K_{ss}^{\alpha} \\ I_{ss} &= \delta K_{ss} \end{aligned}$$

CALIBRATION

- Let's use the following calibration:

$$\begin{aligned} \alpha &= 0.3 \\ \beta &= 0.99 \\ \delta &= 0.025 \\ \sigma &= 3 \end{aligned}$$

- And for the stochastic processes

$$\begin{aligned} \rho_z &= 0.9 \\ \rho_v &= 0.9 \\ \sigma_z &= 0.1 \\ \sigma_v &= 0.1 \end{aligned}$$

Let's put this in Dynare. I will share my screen and show you how it works ...

Bibliography

Judd, K. L. (1998): *Numerical methods in economics*, MIT press.