A Course on DSGE Models with Financial Frictions
Part 2: Simple RBC & Dynare Introduction

Stelios Tsiaras

European University Institute

April 19, 2021

Recap and Way Forward

- DSGE models overview
- Brief description of DSGE solution methods
- Next: Build a simple RBC model and solve it with Dynare

A simple Business Cycle Model

- There is a representative household
- Chooses optimally consumption and labour subject to its budget constraint
- Firms produce output according to a production technology and choose labour and capital inputs to minimize cost
- Labour, capital and output markets clear

Summary

- Households choose hours worked (H_t) and consumption (C_t) to maximize their utility

- Their utility is:

  $$U = U(C_t, L_t)$$

  All variables are expressed in real terms relative to the price of retail output

- In a stochastic environment, the **value function** of the representative household at time t is given by

  $$V_t = \mathbb{E}_t\left[\sum_{s=0}^{\infty} \beta^s U(C_{t+s}, L_{t+s})\right] \ ; \ \beta \in (0,1) \tag{1}$$

The Household Optimization Problem

- Household chooses $\{C_t\}$, labour supply $\{H_t\}$, capital stock $\{K_t\}$ and investment $\{I_t\}$ to maximize V_t given by (1) given the budget constraint:

$$B_t = R_{t-1}B_{t-1} + r_t^K K_{t-1} + W_t H_t - C_t - I_t - T_t \tag{2}$$

- B_t is the value of the stock of one-period bonds (price×number of bonds) at the end of period t.
- r_t^K is the rental rate for capital, W_t is the wage rate and R_{t-1} is the interest rate set in period $t-1$ paid in period t on bonds held at the end of period $t-1$. Note K_t is *end-of-period* capital stock.
- Capital stock accumulates according to

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

Solution to the Household Optimization Problem

First order conditions are

$$\begin{aligned} \text{Euler Consumption} \quad &: \quad R_t \mathbb{E}_t \left[\Lambda_{t,t+1} \right] = 1 \\ \text{Labour Supply} \quad &: \quad \frac{U_{H,t}}{U_{C,t}} = -W_t \\ \text{Capital Supply} \quad &: \quad \mathbb{E}_t \left[\Lambda_{t,t+1} R^K_{t+1} \right] = 1 \end{aligned}$$

where the gross return on capital is given by

$$R^K_t = \left[r^K_t + (1 - \delta) \right]$$

and $\Lambda_{t,t+1} \equiv \beta \frac{U_{C,t+1}}{U_{C,t}}$ is the *real stochastic discount factor* $[t, t+1]$.

Solution to the Household Optimization Problem

- Then we have the arbitrage condition

$$1 = R_t \mathbb{E}_t \left[\Lambda_{t,t+1} \right] = \mathbb{E}_t \left[\Lambda_{t,t+1} R^K_{t+1} \right]$$

- In our financial frictions models $R_t \mathbb{E}_t \left[\Lambda_{t,t+1} \right] \neq \mathbb{E}_t \left[\Lambda_{t,t+1} R^K_{t+1} \right]$

Production Side and Closing the Model

- Output and the firm's behaviour is summarized by:

$$\begin{aligned} \text{Output} &: \quad Y_t = A_t K_t^{\alpha} H_t^{1-\alpha} \\ \text{Labour Demand} &: \quad \frac{\alpha Y_t}{H_t} = W_t \\ \text{Capital Demand} &: \quad \frac{(1-\alpha) Y_t}{K_{t-1}} = r_t^K \end{aligned}$$

- The model is completed with an output equilibrium and a balanced budget constraint with lump-sum taxes (T_t).

$$\begin{aligned} Y_t &= C_t + G_t + I_t \\ G_t &= T_t \end{aligned}$$

where G_t is government spending and A_t follows an AR(1) process:
$\ln A_t - \ln \bar{A}_t = \rho_A (\ln A_{t-1} - \ln \bar{A}_{t-1}) + \epsilon_{A,t}$

The Steady State

- We assume a **zero-growth** steady state and a CRRA utility $U = \ln C_t - \omega \frac{H_t^{1+\phi}}{1+\phi}$ where $\omega > 0$ indicates how leisure is valued relative to consumption, and $\phi > 0$ is the inverse of the labour supply elasticity
- $\bar{A}_t = \bar{A}_{t-1} = A$, say and $\bar{G}_t = \bar{G}_{t-1} = G$. $K_t = K_{t-1} = K$ etc
- The zero-growth steady state in recursive form is given by:

$$\begin{aligned}
R &= \frac{1}{\beta} \\
\frac{K}{Y} &= \frac{(1-\alpha)}{R-1+\delta} \\
\frac{I}{Y} &= \frac{\delta K}{Y} = \frac{(1-\alpha)\delta}{R-1+\delta} = \frac{(1-\alpha)\delta}{R-1+\delta} \\
\frac{C}{Y} &= 1 - \frac{I}{Y} - \frac{G}{Y} = 1 - \frac{I}{Y} - g_y
\end{aligned}$$

Zero Growth Steady State in Recursive Form (cont)

$$\begin{aligned}
H &= \left(\frac{\alpha}{C/Y}\frac{1}{\omega}\right)^{\frac{1}{1+\phi}} \\
Y &= (AH)^{\alpha}K^{1-\alpha} = (AH)^{\alpha}\left(\frac{K}{Y}\right)^{1-\alpha}(Y)^{1-\alpha} \\
&\Rightarrow Y = (AH)(K/Y)^{\frac{1-\alpha}{\alpha}} \\
G &= g_y Y = T \\
W &= \alpha\frac{Y}{H} \\
I &= \frac{I}{Y}Y\,;\; C = \frac{C}{Y}Y\,;\; K = \frac{K}{Y}Y \\
A &= 1
\end{aligned}$$

Solving the Model with Dynare

- Dynare uses a technique called **perturbation**. For more: Judd (1998)
- Computes first, second and third order Taylor series approximation of the policy rules around the steady state

It can also:

- Compute the steady state (numerically) of the model
- Compute the solution of deterministic models
- Estimate (either by maximum-likelihood(ML) or Bayesian approach) parameters of DSGE models and their distribution

Dynare Starters

- It is a big collection of Matlab functions that use Matlab in order to solve the model with perturbation

- For more related to Matlab, check the Mathworks forum (or Google)

- We just have to set the external path of Matlab to the Dynare folder

- Download it at `https://www.dynare.org`

- Install it. Go to Matlab on menu File/Set Path to add the path to the Dynare subdirectory (to store all the subroutines), e.g. the path would be set to $c:\backslash dynare\backslash 4.4.y\backslash matlab$

Dynare Model [`.mod`] file

- The .mod file is the file where you write down your DSGE model

- It includes several blocks
  - Variable block
  - Parameter block
  - Parameter values block
  - Model block
  - Steady state block
  - Shocks block
  - Solution (or estimation) block

  Dynare Manual: Adjemian *et al.* (2011)

Variables and Parameters Block

- `var` block: Names of the endogenous variables
  example:
  ```
  var K C G A;
  ```

- `varexo` block: Names of the shocks
  example:
  ```
  varexo epsA epsG;
  ```

- `parameters` block: Names of the parameters ; Values of the parameters
  example:
  ```
  parameters alpha beta delta ;
  alpha=0.3;
  beta=0.99;
  delta=0.025;
  ```

Model Block

- Starts with `model;` and ends with `end;`

- Type equations ending with ;
  - `x(-1)` for predetermined variables. The variable is decided in $t-1$ (predetermined), e.g. the capital stock, write it as `x(-1)` instead of`x`
  - `x(+1)` for expectations

  example:

  ```
  K = (1-delta)*K(-1)+I;
  ```

Shocks Block

- Starts with `shocks;` and ends with `end;`

- In between declare shock standard deviations
  example:
  ```
  shocks;
  var epsA;
  stderr 0.02;
  end;
  ```

- The variances (and covariances) of the shocks are defined within these commands

- Sets the std. error of this exogenous variable = 0.02

Some info

- **Note** that each instruction of the .mod file must be terminated by a semicolon (;)
- Also Dynare uses 2 forward slashes (//) to comment out any line (whereas MATLAB uses %). (Note: for Dynare the two are equivalent!)
- There need to be as many equations as your endogenous variables declared (except for optimal policy)
- Names are case sensitive
- The stability “Blanchard-Kahn” conditions are met only if the number of jumpers equals the number of eigenvalues greater than one. (See Topic 2).

The Steady State Block

- It’s the most difficult and time consuming part

- There are two options
  - Let Dynare calculate the steady state (sounds good, does not always work)
  - Calculate it ourselves and then add this as a Matlab function

The Steady State Block: Option 1

- Dynare solves for the steady state of the model, it just need (good!) initial values

- Starts with `initval;` and ends with `end;`

- In between, add initial values for all variables

- Initial values can be exact numbers or functions that depend on parameters or steady state variables

- Then, *steady* command computes the steady state

- If the model is quite complicated and the initial values not close to the truth there will be problems → **Option 2**

The Steady State Block: Option 2

- Find the analytical solution for the steady state
- Import it to a Matlab function doing the computation externally with a Matlab program FILENAME_steadystate.m
- Dynare understands that this function gives the steady state of the model
- Needs a specific preamble and ending that is provided in these files

Solution Block

- `stoch_simul` starts the solution routine for stochastic models and `simul` for deterministic simulations
  example
  `stoch simul(order=1,IRF=20, periods =10000) ;`

- There are many options for the stochastic simulation (see Dynare manual for more)

- *periods* - specifies the number of simulation periods

- *irf* sets the number of periods for which to compute impulse responses

- *order* $= 1$ sets the order of the Taylor approximation (default is two)

Solve your Model

- Just type in Matlab `dynare modfilename.mod`
- Dynare output is (among many others):
  - Policy rules
  - Moments
  - Impulse response functions
  - Almost everything is in the folder `oo_`. You will find it in Matlab's workspace right after the solution takes place

Exercise - Introducing Investment Adjustment Costs

- Same problem as above BUT now we have cost $\Phi(\Xi_t)$ for any investment adjustment

$$\begin{aligned} K_t &= (1-\delta)K_{t-1} + (1-\Phi(\Xi_t))I_t \\ \Xi_t &\equiv \frac{I_t}{I_{t-1}}\,;\;\; S',\, S'' \geq 0\,;\;\; \Phi(1) = \Phi'(1) = 0 \end{aligned}$$

- I_t units of output converts to $(1-\Phi(\Xi_t))I_t$ of new capital sold at a real price Q_t

- We now have two constraints and two Lagrange multipliers: λ_t & μ_t.

- Express Q_t as μ_t/λ_t: the marginal value of capital measured in terms of consumption goods (this is Tobin's Q) and express the FOC to Investment in terms of Q_t.

Exercise - Introducing Investment Adjustment Costs

- For $\Phi(\Xi_t) = \phi_X(\Xi_t - 1)^2$ find the new equilibrium and solve the model in Dynare

- Do it first by setting $\phi_X = 2$

- Construct a [0:4] vector with step=1 grid for ϕ_X and show the different impulse responses for ouput, consumption, investment and labour hours for every different value of ϕ_X after a positive TFP shock

- Important! At each iteration set `dynare modfile.mod noclearall`. Dynare does not clear up the workspace after each iteration

Exercise - Introducing Investment Adjustment Costs

What you need In Matlab

```
% Give to Phix the value of every iteration
phiX    = Phix_value(i);
% Send the parameter value to Dynare Phix_value
save Phix_value phiX
% Run the mod file every time with noclearall option
dynare RBC1_inv_adj noclearall
```

In the parameter section of Dynare

```
load Phix_value;
set_param_value('phiX',phiX);
```

Adjemian, S., Bastani, H., Juillard, M., Karamé, F., Maih, J., Mihoubi, F., Mutschler, W., Perendia, G., Pfeifer, J., Ratto, M., and Villemot, S. (2011). Dynare: Reference manual version 4. Dynare Working Papers 1, CEPREMAP.

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