Structural Macroeconometrics

Chapter V:
Dynare

Prof. Dr. Johannes Pfeifer

Kobe University

Summer 2018

Dynare

Preprocessor and collection of Matlab (and GNU Octave) routines that

1. computes the solution of deterministic models (arbitrary accuracy)
2. computes first- and higher-order approximation to solution of stochastic models
3. estimates (maximum likelihood or Bayesian approach) parameters of DSGE models, for linear and non-linear models
4. checks for identification of estimated parameters
5. computes optimal policy
6. performs global sensitivity analysis of a model
7. solves problems under partial information
8. estimates BVAR and Markov-Switching Bayesian VAR models
9. provides macro language and reporting facility

Dynare

- Latest version: Dynare 4.5.5

  $\rightarrow$ download from `http://www.dynare.org`

- Online forum at `https://forum.dynare.org`

- Report bugs at `https://github.com/DynareTeam/dynare`

Structure of typical Dynare $.mod$ file for stochastic models

- Declare variables, shocks, and parameters ($+$ values) of the model
- Specify model equations
- Specify steady state and let Dynare check stability conditions

  $\rightarrow$ Dynare can also search for steady state numerically
- Dynare

  $\rightarrow$ linearizes model around steady state

  $\rightarrow$ solves for recursive equilibrium laws of motion
- Plot impulse responses, conduct stochastic simulations, ...

Structure of typical Dynare *.mod* file: declarations

- `var ;`

  → define endogenous model variables

- `varexo ;`

  → define exogenous model variables, i.e. shocks

- `parameters ;`

  → define list of parameters

  → assign parameter values

- All command lines are ended with a semicolon
- Note: For all three types, you can specify LaTeX names with $$ and long names for Dynare output with `(long_name='name here')`

Structure of typical Dynare $.mod$ file: declarations

- Declaration of variables in the model

```
var y $y$ (long_name='output')
c $c$ (long_name='consumption')
;
```

- Declaration of shocks in the model

```
varexo eps_z ${\varepsilon_z}$ (long_name='TFP shock')
eps_g ${\varepsilon_g}$ (long_name='government spending shock')
;
```

- Declaration of parameters in the model

```
parameters beta ${\beta}$ (long_name='discount factor')
psi ${\psi}$ (long_name='labor disutility parameter')
;
```

- Setting of parameter values

```
sigma=1;
alpha= 0.33;
```

Structure of typical Dynare $.mod$ file: model equations

- Model block starts with:

  ```
  model;
  ```

  and ends with:

  ```
  end;
  ```

- The same general syntax applies to other blocks like `initval;` or `steady_state_model;`
- In between the model-block you write down the necessary equations defining the dynamic equilibrium of the model

  $\rightarrow$ FOCs, constraints, additional variable transformations
- Remember: you need as many equations as endogenous variables
- Note: you can name the equations with `[name='name here']`

Structure of typical Dynare $.mod$ file: model equations

- Block of model equations

```
model;
[name='Euler equation']
c^(-sigma)=beta/gammax*c(+1)^(-sigma)* (alpha*exp(z(+1))*(k/l(+1))^(
alpha-1)+(1-delta));
[name='Labor FOC']
psi*c^sigma*1/(1-l)=w;
[name='Definition log output']
log_y = log(y);
end;
```

Structure of typical Dynare $.mod$ file: steady state

- If not told differently, Dynare tries whether 0 is the steady state, and if not, starts a numerical solver to find the steady state
- As 0 is often infeasible for nonlinear models, always provide explicit steady state information
- There are three ways to do this:
  1. compute the steady state analytically and provide it to Dynare using the `steady_state_model`-block
  2. provide initial guesses to start the numerical optimizer using the `initval`-block (the better the higher the chances of finding the steady state; economic intuition provides good guidance, e.g. $I < C < Y < K$, with $I = \delta K$.)
  3. use an explicit steady state file (see e.g. the `NK_baseline.mod` in the Dynare examples folder), which offers a lot of flexibility
- Note: linearized models typically do not spare you from computing the steady state, they just shift the problem to computing parameters

Structure of typical Dynare $.mod$ file: steady state

- Block of steady state definitions

```
steady_state_model;
gammax=(1+n)*(1+x);
delta=i_y/k_y-x-n-n*x;
// calibrate the model to steady state labor of 0.33, i.e. compute the
corresponding steady state values and the labor disutility parameter by hand
beta=gammax/(alpha/k_y+(1-delta));
l=0.33;
k=((1/beta*gammax-(1-delta))/alpha)^(1/(alpha-1))*l;
end;
```

Structure of typical Dynare $.mod$ file: shocks

- Block defining the covariance matrix of exogenous shocks

```
shocks;
var eps_z=0.68^2;
var eps_g=1.05^2;
end;
```

Structure of typical Dynare $.mod$ file: additional commands

- Commands to generate LaTeX output

```
write_latex_dynamic_model;
write_latex_parameter_table;
write_latex_definitions ;
collect_latex_files ;
```

- Commands to compute the steady state and check stability conditions

```
steady;
check;// check Blanchard-Kahn-conditions
```

- Command to
  - linearize the model around the steady state
  - compute recursive equilibrium laws of motion of the linearized model
  - analyze the model via impulse responses, stochastic simulations, moments, etc.

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

Timing conventions

- Time indices are given in parenthesis:
  - X_{t+1} is written $X(+1)$, (these are forward looking variables)
  - X_{t-1} is written $X(-1)$
  - X_t is written X (no time index needed for the current period)
- Time indices refer to the period when the value of the variable is determined, i.e. the **stock at the end of period notation**
- The value of the capital stock used in production at time t is determined by investment at time $t-1$ and therefore predetermined

$$K_t = (1-\delta)K_{t-1} + I_t \tag{1}$$
$$Y_t = K_{t-1}^{\alpha} L_t^{1-\alpha} \tag{2}$$

- To use the **stock at the beginning of period notation**

$$K_{t+1} = (1-\delta)K_t + I_t \tag{3}$$
$$Y_t = K_t^{\alpha} L_t^{1-\alpha} \tag{4}$$

  you can declare a particular variable as predetermined using the `predetermined_variables` declaration

Timing conventions: conditional expectations

- The conditional expectations operator E_t is typically not needed in Dynare as it starts from an equilibrium system

$$0 = E_t\left(f(y_{t+1}, y_t, y_{t-1}, \varepsilon_t)\right) \tag{5}$$

  and therefore assumes a conditional expectations operator in front of every equation

- `1/c=beta/c(+1)*(R(+1));`
  therefore corresponds to

$$\frac{1}{c_t} = E_t\left(\frac{1}{c_t}\right) = E_t\left(\beta\frac{R_{t+1}}{c_{t+1}}\right) = \beta E_t\left(\frac{R_{t+1}}{c_{t+1}}\right) \tag{6}$$

- Due to certainty equivalence at first order, we do not have to bother with this, but intricate issues can arise at higher order (e.g. Epstein-Weil-Zin preferences)

Log-linearization vs. linearization

- Dynare linearizes the model around the steady state
- However, we often want log-linearization to express everything in percentage deviations from steady state (or the balanced growth path)
- To obtain a log-linear solution, you can
  1. use the `loglinear` option
  2. write down the model in terms of logarithmic transformations of the original variables (exp()-substitution)
  3. append the logarithm of model variables as separate model variables (e.g. `log_y=log(y)`)

Estimation

- Set parameters to be estimated and their priors

```
estimated_params;
rhoz, beta_pdf,0.7,0.1;
 stderr eps_z, inv_gamma_pdf, 0.01, 0.1;
end;
```

- Initialize parameters at model calibration

  → otherwise initialized at prior mean

```
estimated_params_init(use_calibration);
end;
```

- Declare observed variables

```
varobs y_obs;
```

- Conduct estimation → see Dynare manual for more options

```
estimation(datafile=data_bayesian_estimation,
mh_jscale=?
mh_replic=?,
mh_nblocks=?)
end;
```