Session: Using DSGE models with Dynare

Aurélien Poissonnier[1]

[1]Insee-European Commission

Applied Macroeconometrics - ENSAE

Plan

Introduction: what is Dynare

Main building blocks in a dynare code

Elementary outputs

Simulating models

Estimation of models

Optimal policy

Wrapping-up

Plan

Introduction: what is Dynare

Main building blocks in a dynare code

Elementary outputs

Simulating models

Estimation of models

Optimal policy

Wrapping-up

Dynare

"Dynare is a software platform for handling a wide class of economic models, in particular dynamic stochastic general equilibrium (DSGE)".

Developed by a team of researcher and hosted at Cepremap.
Runs on Matlab, Octave, C++ and currently being developed for Julia.
Has become a must in macro-economists' toolbox.

Stéphane Adjemian et al. (Apr. 2011). *Dynare: Reference Manual Version 4*. Dynare Working Papers 1. CEPREMAP

Plan

Introduction: what is Dynare

Main building blocks in a dynare code

Elementary outputs

Simulating models

Estimation of models

Optimal policy

Wrapping-up

Main blocks

How to declare a standard DSGE model in Dynare.

Remember from Lecture 4 and (Villemot, 2011), the general formula for a DSGE model:

$$E_t f(y_{t+1}, y_t, y_{t-1}, v_t) = 0 \tag{1}$$

Quite logically then, you should declare:

- the endogenous variables (y)
- the exogenous variables (v)
- the model (f), i.e.
  - $\longrightarrow$ the parameters
  - $\longrightarrow$ the equations

E_t is implicit in Dynare and solved under rational expectations

Main blocks

Therefore a code includes:

`var` list of vars ;

`varexo` list of shocks ;

`parameters` list of parameters ; (possibly followed by calibrations `beta =0.99;`)

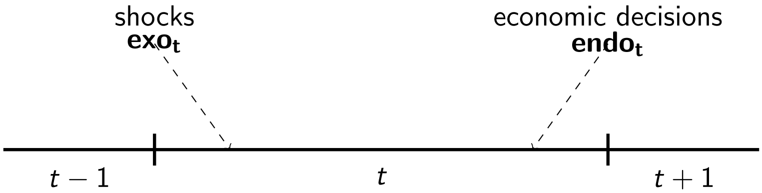
`model;` list of equations `end` ;

More options

Dynare allows for a lot of useful options. Variables and parameters can be assigned long names or Latex names. So can equations.
You can use calibrated parameters to calibrate new ones (e.g. `beta = 0.99; gamma = beta^ 2;`) The model can be declared linear (to avoid the computation of its linearization)...
Check the reference manual for more (Adjemian et al., 2011).

Timing convention

When writing the equations of a model be aware that:



This timing convention implies in particular that stock variables are stocks at the end of the period.
e.g. for capital, it implies that output (Y_t) depends on capital at $t-1$, a part of which will be used as investment (I_t) to increase capital (K_t).

Plan

Introduction: what is Dynare

Main building blocks in a dynare code

Elementary outputs

Simulating models

Estimation of models

Optimal policy

Wrapping-up

Steady state

`steady;`

This command computes the steady state of your model, i.e returns the value of $\bar{y}$ such that

$$f(\bar{y}, \bar{y}, \bar{y}, 0) = 0 \tag{2}$$

Test

In lecture 4 we exemplified this command on a small neo-Keynesian model.
Let's try on a small RBC model (Collard, 2001). (`example2.mod`)

The solution is stored in `oo_.steady_state`.
Specifying the initial value to search for the steady state of a non linear model may help the algorithm (`initval; ...`). NB: this command finds different uses in different contexts.
Check the reference manual (Adjemian et al., 2011, chap 4.10).

Checking the rank condition

`check;`

This command computes the eigenvalues of the (linearized) system, which is key to solve the model under rational expectations (see Lecture 4).

The solution is stored in `oo_.dr.eigval`.

Dynare automatically checks for the rank condition.

Test

In lecture 4 we exemplified this command on a small neo-Keynesian model as well.

Let's also try on a small RBC model (Collard, 2001).

(`example2.mod`)

Check the reference manual (Adjemian et al., 2011, chap 4.11).

Simulated and theoretical moments

To compute simulated or theoretical moments and other statistics linked to the stochastic part of the model, we need to specify the exogenous variables (v). This is done with a "shocks" block.

```
 shocks;
 var  an exovar name = a value for the variance ;
...
 end;
```

One can also specify correlations between exogenous variables. By assumption, exogenous variables follow normal distributions with zero mean. Check the reference manual (Adjemian et al., 2011, chap.4.8).

Simulated and theoretical moments

`stoch_simul(irf=0);`

This command computes many interesting results, among which the moments of the endogenous variables.

`stoch_simul(irf=0, periods=0);` computes the theoretical moments

`stoch_simul(irf=0, periods=1000);` computes the empirical moments on 900 simulated periods (see the option `drop` for why 900 and not 1000)

Test

`smallneoK.mod` or `example2.mod`

Many options are available, such as a first, second or third order approximation of the model. Check the reference manual (Adjemian et al., 2011, chap.4.13).

Variance decomposition

`stoch_simul(irf=0);` also returns the variance decomposition (see also Lecture 3 on SVAR).

Closely related is the conditional variance decomposition (the variance of the forecast error k steps ahead). It can be returned using `stoch_simul(irf=0, conditional_variance_decomposition = [1:3]);` for 1 to 3 steps ahead forecast errors.

Policy function

`stoch_simul(irf=0);`

`stoch_simul(irf=0);` also returns the policy function (or decision rule).

Remember from Lecture 4 and (Villemot, 2011), this is solving the model for unobserved expectations using the rational expectation assumption:

$$y_t = g(y^-_{t-1}, v_t) \tag{3}$$

This takes the form of a linear model (matrices) for a first order approximation of the model. Check the reference manual (Adjemian et al., 2011, chap.4.13.3, 4 & 5) to see under what form Dynare returns the solution to a second and third order approximation.

Plan

Introduction: what is Dynare

Main building blocks in a dynare code

Elementary outputs

Simulating models
- Stochastic simulations
- Deterministic simulations

Estimation of models

Optimal policy

Wrapping-up

Plan

Simulating models

Stochastic simulations

Deterministic simulations

IRF

stoch_simul;

As you probably guessed `stoch_simul;` (withouth the option `irf=0`) returns the impulse response functions of the endogenous variables to shocks on the exogenous variables.

The outputs are stored in `oo_.irfs`.endo_exo.
Graphs are printed (and saved).
You can adjust the list of (endogenous and exogenous) variables to be plotted.
Once again, check the reference manual (Adjemian et al., 2011, chap.4.13.1).

Test

Let's try on `example2.mod` and `NK_baseline.mod` for a change.
(Note the existence of a file `NK_baseline_steadystate.m`)

Plan

Simulating models

- Stochastic simulations
- Deterministic simulations

Deterministic simulations

The above exercises are related to solving a DSGE model under rational expectations and putting it under the form of a constrained structural VAR. As anticipated in Lecture 4, this allows to perform exercises similar to those with VAR (e.g. IRF, variance decomposition).

Instead, one could work with

$$\cancel{E_t} f(y_{t+1}, y_t, y_{t-1}, v_t) = 0 \tag{4}$$

that is assuming perfect foresight of the sequence of v (and not rational expectations).

Deterministic simulations

The above exercises are related to solving a DSGE model under rational expectations and putting it under the form of a constrained structural VAR. As anticipated in Lecture 4, this allows to perform exercises similar to those with VAR (e.g. IRF, variance decomposition).

Instead, one could work with

$$\cancel{E_t} f(y_{t+1}, y_t, y_{t-1}, v_t) = 0 \tag{4}$$

that is assuming perfect foresight of the sequence of v (and not rational expectations).

Before that, which of the above exercises is not just a rational expectation exercise?

Declaring shocks differently

A completely different perspective: you don't calibrate standard deviations but values of the shocks v in specific periods. Dynare stacks the equations $f(y_{t+1}, y_t, y_{t-1}, v_t) = 0$ at the different periods and solve for y (non linear problem).

You should give a starting and ending value for y.

`initval ;`, `endval ;`, `steady ;`, `histval ;`
`resid ;` may also be useful.
Check the reference manual (Adjemian et al., 2011, chap.4.8) to be sure about what you ask Dynare to do as these commands can have different behaviors in different contexts!

Declaring shocks differently

You declare shocks such as

```
shock ;
var x ;
periods 1:3 ;
values 1 ;
end ;
```

Check the reference manual (Adjemian et al., 2011, chap.4.7) to be sure you understand the difference between shock declaration in the stochastic and deterministic context.

You run the simulation

`simul;` in a condensed way

or `perfect_foresight_setup;` & `perfect_foresight_solver;`

Results are stored in `oo_.endo_ simul` & `oo_.exo_ simul`

Test

Let's try on `ramst.mod`.

Exercise

On `NK_baseline.mod`, code a simulation for an economy where the central banker would change the inflation target (e.g. to move away from the risk of ZLB).
NB: You may want to change a parameter into an exovar (e.g. to simulate a transition between two steady states linked to two calibrations). There is a special command for this (`change_type`)

Plan

Introduction: what is Dynare

Main building blocks in a dynare code

Elementary outputs

Simulating models

Estimation of models

- Observed variables
- Estimation
- Prior distribution
- Posterior distribution
- Shock decomposition
- Replication code for (Smets and Wouters, 2007)
- Model Comparison

Plan

Estimation of models
- Observed variables
- Estimation
- Prior distribution
- Posterior distribution
- Shock decomposition
- Replication code for (Smets and Wouters, 2007)
- Model Comparison

Observed variables

`varobs;`

To estimate a model you need to declare which endogenous variables are observed with `varobs` a list of endovars `;`.

Thanks to the Kalman filter, you can estimate the model without observing all the data.

You cannot have more observed variables than exogenous variables (stochastic singularity).

You point Dynare to the observed data with `estimation(datafile=...);`

Plan

Estimation of models

- Observed variables
- Estimation
- Prior distribution
- Posterior distribution
- Shock decomposition
- Replication code for (Smets and Wouters, 2007)
- Model Comparison

Estimation

`estimation(...);`

This is a central command of Dynare.

It allows in particular for MLE and Bayesian estimations of models.

We will see two examples `fd2000.mod` and (Smets and Wouters, 2003) (see also Lecture 6).

Check the reference manual (Adjemian et al., 2011, chap.4.14). It includes 28 pages of options.

Plan

Estimation of models
Observed variables
Estimation
Prior distribution
Posterior distribution
Shock decomposition
Replication code for (Smets and Wouters, 2007)
Model Comparison

Prior distribution

`estimated_params;`

For a Bayesian estimation you can declare your priors about estimated parameters.
You can specify them choosing from various distributions.
If you don't specify any prior, you must calibrate the non estimated parameters.

Test

Let's have a look in `fd2000.mod`.

For a maximum likelihood estimation, you can force the estimated parameters within chosen intervals with
`astimated_params_bounds ;`

Plan

Estimation of models

- Observed variables
- Estimation
- Prior distribution
- Posterior distribution
- Shock decomposition
- Replication code for (Smets and Wouters, 2007)
- Model Comparison

On top of the estimated mode (and an approx of the distribution around it), you can estimate the posterior distribution with a specific algorithm (Monte Carlo Markov Chain - Metropolis-Hastings, see Lecture 6).

The `estimation` command has a series of specific options `mh_replic`, `mh_nblocks`, `mh_jscale`, once again check the reference manual (Adjemian et al., 2011, chap.4.14)

Test

Let's estimate the posterior distribution of the parameters on simulated data with `fd2000.mod`.

On top of the estimated mode (and an approx of the distribution around it), you can estimate the posterior distribution with a specific algorithm (Monte Carlo Markov Chain - Metropolis-Hastings, see Lecture 6).

The `estimation` command has a series of specific options `mh_replic`, `mh_nblocks`, `mh_jscale`, once again check the reference manual (Adjemian et al., 2011, chap.4.14)

Test

Let's estimate the posterior distribution of the parameters on simulated data with `fd2000.mod`.
Discuss the identification of the parameters.

Plan

Estimation of models

- Observed variables
- Estimation
- Prior distribution
- Posterior distribution
- Shock decomposition
- Replication code for (Smets and Wouters, 2007)
- Model Comparison

Shock decomposition

Test

Let's run `fd2000.mod` with a shock decomposition command.

Check the reference manual (Adjemian et al., 2011, chap.4.16) for more options.

Plan

Estimation of models

- Observed variables
- Estimation
- Prior distribution
- Posterior distribution
- Shock decomposition
- Replication code for (Smets and Wouters, 2007)
- Model Comparison

Smets and Wouters, 2007

Test

Let's run the Smets and Wouters model estimation for the US.

Commented in the author's code (updated by Johannes Pfeifer) are many other commands to replicate their results.

Plan

Estimation of models

- Observed variables
- Estimation
- Prior distribution
- Posterior distribution
- Shock decomposition
- Replication code for (Smets and Wouters, 2007)
- Model Comparison

Model Comparison

In a Bayesian setup, a comparison of 2 models on the same dataset is done through the posterior odds ratio (Lecture 6).

Dynare allows you to do that with the command `model_comparison` (Adjemian et al., 2011, chap.4.15).

Exercise

Using the Smets and Wouter model, compare the baseline model to the same model estimated without habit formation.

Check the reference manual (Adjemian et al., 2011, chap.4.16) for more options.

Plan

Introduction: what is Dynare

Main building blocks in a dynare code

Elementary outputs

Simulating models

Estimation of models

Optimal policy

Wrapping-up

Optimal simple rule

`osr;`

You can search for the optimal values of (policy) parameters such that they minimize a quadradic function of the endogenous variables.
(Galí, 2015, Chapter 6) shows that the welfare loss function in our `smallneoK.mod` example is a function of the variances of output gap, price inflation and wage inflation.

Test

Let's use `GaliChap6.mod` to find the coefficients of the Taylor rule such that the loss function is minimized.

We use `osr_params` to chose the parameters with respect to which we optimize.
`osr_weights` to set the weights in the quadratic function.
`osr;` to run the command.

Plan

Introduction: what is Dynare

Main building blocks in a dynare code

Elementary outputs

Simulating models

Estimation of models

Optimal policy

Wrapping-up

Applications with Dynare

What we have covered

Main code blocks, solving a model under rational expectations, stochastic vs. deterministic simulations, estimation, optimal policy rule

Also available

Sensitivity analysis, forecasting, Ramsey policy, markov switching SBVAR, DSGE-VAR, macro-processing language, time-series manipulation, Dynare and Latex, Dynare and Matlab, Dynare++

References I

Adjemian, Stéphane et al. (Apr. 2011). *Dynare: Reference Manual Version 4*. Dynare Working Papers 1. CEPREMAP.

Collard, Fabrice (2001). "Stochastic simulations with DYNARE. A practical guide. (Adapted for Dynare 4.1, 2009)". In: pp. 1 –8.

Galí, Jordi (2015). *Monetary policy, inflation, and the business cycle: an introduction to the new Keynesian framework and its applications*. Princeton University Press.

Smets, Frank and Raf Wouters (2003). "An Estimated Dynamic Stochastic General Equilibrium Model of the Euro Area". In: *Journal of the European Economic Association* 1.5, pp. 1123–1175.

Smets, Frank and Rafael Wouters (2007). "Shocks and Frictions in US Business Cycles: A Bayesian DSGE Approach". In: *American Economic Review* 97.3, pp. 586–606.

References II

Villemot, Sébastien (Apr. 2011). *Solving rational expectations models at first order: what Dynare does*. Dynare Working Papers 2. CEPREMAP.