Introduction to Dynare

Alexander Kriwoluzky[1]

[1]Humboldt Universität zu Berlin
kriwoluz@wiwi.hu-berlin.de

July 12, 2007

Outline

- Download Dynare and additional information and a wonderful user guide from `http://www.cepremap.cnrs.fr/dynare/`.
- This material and all example files are also available on my webpage.

How does Dynare work?

- The user writes a mod-file.
- Then types `dynare filename` in the matlab command window.
- Dynare produces an m-file from it.
- It solves non-linear models with forward looking variables.
- It estimates the parameters of those models.

Overview

A Dynare code that solves a non-linear model consists of the following parts:

- Declaration of the variables.
- Declaration of the parameters.
- The equations of the model.
- Steady state values of the model.
- Definition of the properties of the shocks.
- Setting of additional options for the execution commands.

Example: Slightly extended Real Business Cycle model.

Model description I

$$\max E\left[\sum_{t=0}^{\infty}\beta^t e^{\eta_{b,t}}(\log c_t - An_t)\right] \tag{1}$$

s.t.

$$c_t + x_t = y_t \tag{2}$$

$$y_t = e^{\eta_{z,t}} k_{t-1}^{\theta} n_t^{1-\theta} \tag{3}$$

$$e^{\eta_{x,t}} x_t = k_t - (1-\delta)k_{t-1} \tag{4}$$

Model description II

$$\eta_{a,t} = \rho_a \eta_{a,t-1} + \epsilon_{a,t}, \ \epsilon_{a,t} \sim N(0, \sigma_a^2) \ i.i.d. \tag{5}$$

$$\eta_{b,t} = \rho_b \eta_{b,t-1} + \epsilon_{b,t}, \ \epsilon_{b,t} \sim N(0, \sigma_b^2) \ i.i.d. \tag{6}$$

$$\eta_{x,t} = \rho_x \eta_{x,t-1} + \epsilon_{x,t}, \ \epsilon_{x,t} \sim N(0, \sigma_x^2) \ i.i.d. \tag{7}$$

The structural shocks are assumed to be uncorrelated.

Outline

Declaration of the variables

- Endogenous and exogenous variables are declared separately.
- For endogenous variables use: '*var*'.

- Example:
  `var n y c k x eta_b eta_a eta_x;`
- Exogenous variables are declared with: '*varexo*'.

- Example:
  `varexo eps_b eps_a eps_x;`

Properties of the shocks

- The variances and covariances of the shocks are defined within the commands '*shocks*' and '*end*'
- The command '*var eps_b; stderr 0.02;*' sets $\sigma_{\epsilon,b} = 0.02$
- The covariances between two shocks can be declared as: '*var eps1 eps2 = phi*'
- Example:

```
shocks;
var eps_b; stderr 0.02;
var eps_a; stderr 0.02;
var eps_x; stderr 0.02;
end;
```

Declaration of the parameters

- The parameters of the model are defined with the command '*parameters*'
- Example:
  `parameters A theta delta beta rho_b rho_a rho_x;`
- Afterwards they are calibrated:
  Example:
  `A=2.3; theta=0.36; betta = 0.99; delta = 0.025;rho_b =0.5; rho_a =0.5; rho_x =0.5;`

Outline

Declaration of the model

- The equations of the model are defined within the commands '*model*' and '*end*'.
- Different time indices are abbreviated as
  - $x_t = x$
  - $x_{t+1} = x(+1)$
  - $x_{t-1} = x(-1)$
- In case the model consist of linear equations use '*model (linear)*' as opening command.

The equations of the model

Example:

```
model(linear);
# y_k = (1/theta)*(1/beta-1+delta);
# c_k = y_k-delta;
n=y-c;
c=-eta_b(+1)+eta_b+(1-delta)*beta*eta_x(+1)-eta_x+c
y=eta_a+theta*k(-1)+(1-theta)*n;
k=delta*x+(1-delta)*k(-1)+delta*eta_x;
x=(y_k/delta)*y-(c_k/delta)*c;
eta_b=rho_b*eta_b(-1)+eps_b;
eta_a=rho_a*eta_a(-1)+eps_a;
eta_x=rho_x*eta_x(-1)+eps_x;
end;
```

What is ' #'

allows for additional constants or matlab functions. For example:

- `# n_bar = 1/3`
- `# n_bar=Calculate_n_bar(input)`

The steady state of the model I

- Dynare solves for the steady state of the model. It just needs initial starting values.
- These are specified within the commands '*initval*' and '*end*'.
- Then: '*steady*'.
- This routine is very sensitive to your guess.
- The best guess is the analytically calculated steady state.

The steady state of the model: Example

```
initval;
 y = 0.9916
 n = 0.2875;
 c = 0.6656;
 k = 5.0419;
 x = 0.2419;
 eta_b = 0; eta_a = 0;
 eta_x = 0; eps_a = 0;
 eps_x = 0; eps_b = 0;
end;
steady;
```

Outline

The command *stoch_simul*

- The command '*stoch_simul*' starts the solution routine. - It is the '*Do it*' function in Dynare.
- It computes a Taylor approximation around the steady state of order one or two.
- It simulates data from the model.
- Furthermore moments, autocorrelations and impulse responses are computed.

Options

Options set outside the command '*stoch_simul*':

- '*check*' - computes and displays the eigenvalues of the model. Example: `check;`
- '*datatomfile*' - saves the simulated data in a m file. Example: `datatomfile('simuldata',[])`

Options for '*stoch_simul*'

Additional options can be set in brackets after *'stoch_ simul'*:

- '*periods*' - specifies the number of simulation periods `periods=1000;`
- '*irf*' sets the number of periods for which to compute impulse responses.
- '*nomoments*', '*nocorr*', '*nofunctions*': moments, correlations or the approximated solution are not printed.
- '*order=1*' sets the order of the Taylor approximation (default is two).
- Example: `stoch_simul(irf=20, order=1, nomoments);`
- Have a look at the Dynare manual for complete description.

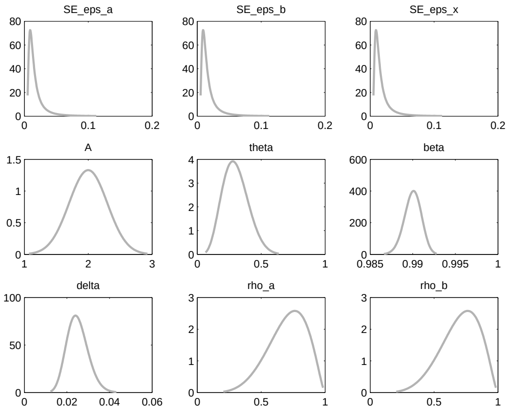
Prior distribution of the parameters

- For each parameter to be estimated a conjugate prior distribution has to be defined.
- There are four common prior distributions used in the literature:
  - *Beta* distribution for parameters between 0 and 1.
  - *Gamma* distribution for parameters restricted to be positive.
  - *InverseGamma* distribution for the standard deviation of the shocks.
  - *Normal* distribution.

Prior distribution declaration

```
estimated_params;
A, normal_pdf, 2, 0.3;
theta, beta_pdf, 0.3, 0.1;
beta, beta_pdf, 0.99, 0.001;
delta, beta_pdf, 0.0025, 0.005;
rho_a, beta_pdf, 0.7, 0.15;
rho_b, beta_pdf, 0.7, 0.15;
rho_x, beta_pdf, 0.7, 0.15;
stderr eps_a, inv_gamma_pdf, 0.02, inf;
stderr eps_b, inv_gamma_pdf, 0.02, inf;
stderr eps_x, inv_gamma_pdf, 0.02, inf;
end;
```

Plot of prior distribution



The *estimation* command

The command *estimation* triggers the estimation of the model:

1. The likelihood function of the model is evaluated by the Kalman Filter.
2. Posterior mode is computated.
3. The distribution around the mode is approximated by a Markov Monte Carlo algorithm.
4. Diagnostics, impulse response functions, moments are printed.

Some options

- *datafile= FILENAME* specifies the filename.
- *nobs* number of observation used.
- *first_obs* specifies the first observation to be used.
- *mode_compute* specifies the optimizer. For example:
    - 0: switch mode computation off
    - 1: fmincon
    - 4: csminwel
- *nodiagnostic*
- Further options to be explained later.

Outline

Steady state computation: Do it yourself

You can write your own routine that computes the steady state:

- A simple Matlab file that has to be called with the name of your Dynare file followed by `_steadystate`.
- For example the function that returns the steady state vector for a model in a file called `DSGE_exampel.mod` has to be called:
  `DSGE_exampel_steadystate.m`

Steady state matlab function

- The matlab function that computes the steady state for the file *three_shock* is called:
  `function [ys check] = three_shock_steadystate (junk, ys)`
- '*ys*' is the vector containing the steady state values - the variables have to be ordered alphabetically!
- '*check*' can simply be set zero.
- Do not forget to declare the parameters necessary to solve the steady state as global.

Example code

```
function [ys check] = three_shock_steadystate
(junk, ys)
global theta A delta beta
y_over_k=(1/beta-1+delta)/theta;
x_over_k=delta;
c_over_y=y_over_k - x_over_k;
n=((1-theta)/A)*(c_over_y)^(-1);
k=(y_over_k)^(1/(theta-1))*n;
y=y_over_k*k;
c=c_over_y*y;
x=x_over_k*k
eta_a=0;eta_b=0;eta_x=0;
ys=[c; eta_a; eta_b; eta_x; k; n; x; y];
check=0;
```

Outline

The dataset

Observed variables are declared after *varobs*. You can include the dataset in the following ways:

- As matlab savefile (*.mat). Names of variables have to correspond to the ones declared under *varobs*.
- As m-file. Again names of variables have to correspond to the ones declared under *varobs*.

Matching data to the model

The variables of the model are often log deviations from the steady state with zero mean and no growth trend. To fit the model and the data:

- Detrend the data before by HP filter or a linear detrending.
- Compute first differences of the dataset and fit the model by:
  - Declaring additional endogenous variables, for example: `var y_obs.`
  - Augmenting the model block with observation equations, e.g.:

$$y\_obs = y - y(-1)$$

Outline

Summary of variables

```
Number of variables:  8
Number of stochastic shocks:  3
Number of state variables:  4
Number of jumpers:  4
Number of static variables:  2
```

Further output

- Matrix of Covariance of exogenous shocks
- Recursive law of motion
- Moments, Correlation and Autocorrelation of simulated variables

Stored output

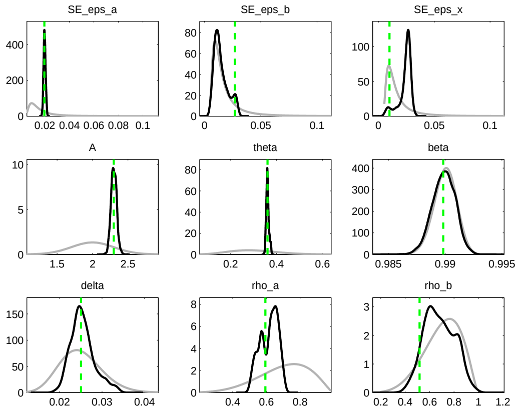
All results are stored in *filename*_results.mat:

- dr_ contains e.g.:
  - Recursive law of motion (*ghx*, *ghu*).
  - Eigenvalues
  - Steady state (*ys*)
- oo_ contains e.g.:
  - Posterior mode and std
  - Marginal density
  - Smoothed shocks

Outline

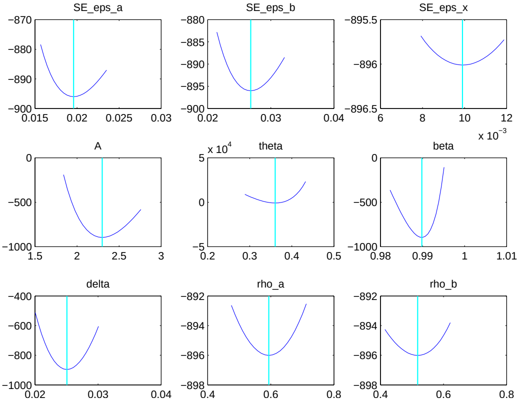
1. Dynare: Introduction.
2. Structure of a mod-file
   - Declaration of the variables, parameters and shocks
   - The model
   - Solving and estimating the model
3. Additional insights I
   - Steady state computation
   - The dataset
4. Dynare output
   - Characteristics of the model
   - Figures and how to produce them
5. Additional insights II

Prior vs. Posterior



Mode check

Add the option *mode_check*.

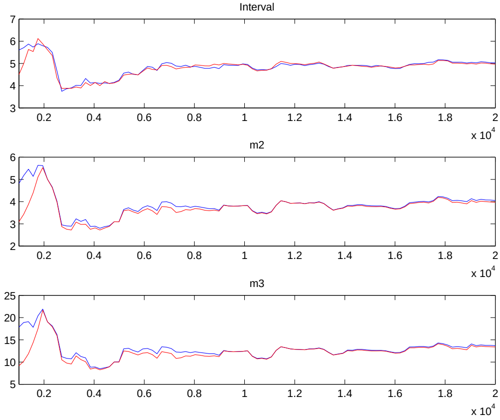


Convergence of the Markov chain

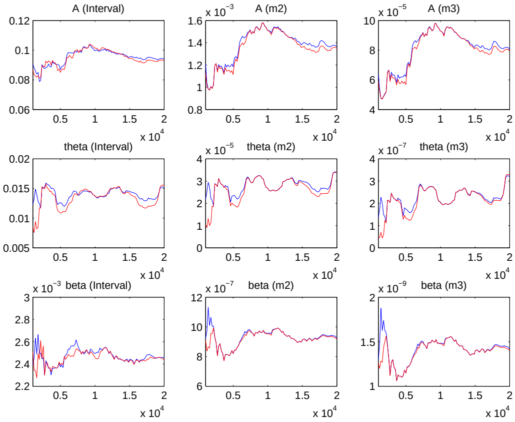
Using simulation methods it is important to insure convergence of the proposal distribution to the target distribution:

- Dynare runs different, independent chains. Default=2. Set the number of chains by: *mh_blocks*.
- Longer chains are more likely to have converged. Set the number of draws by: *mh_replic*
- The first draws should be discarded. Set the percentage of discarded draws by: *mh_drop*.

Multivariate diagnostics

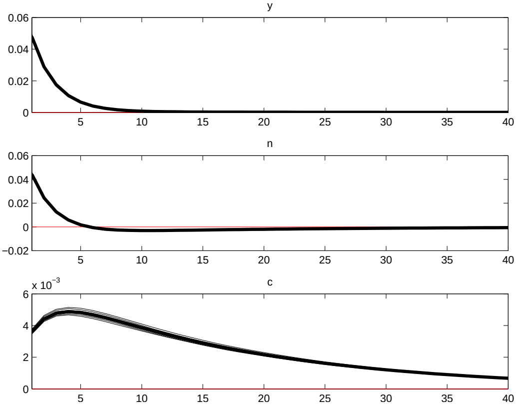


Univariate diagnostics



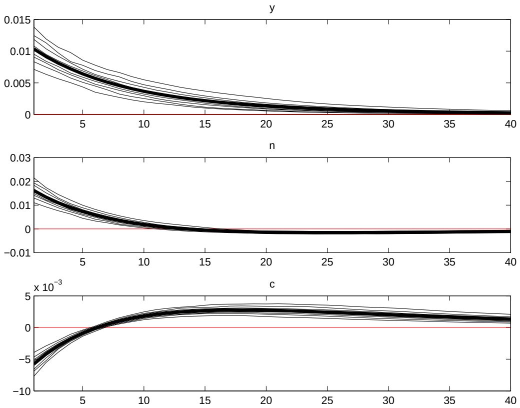
Bayesian impulse response function

Add *bayesian_irf*.



More Bayesian impulse response function

Add *bayesian_irf*.



Filtered and smoothed variables

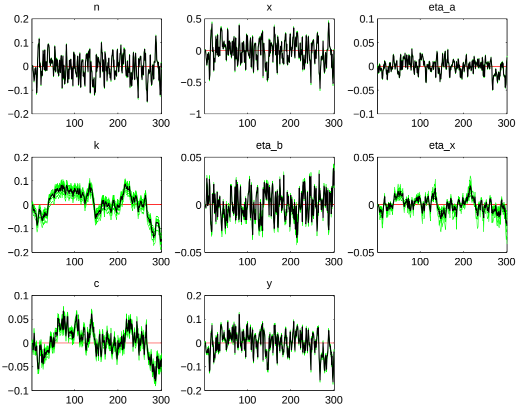
- *filtered_vars* triggers the computation of filtered variables, i.e. forecast on past information:

$$x_{t|t-1} = E[x_t|I_{t-1}]$$

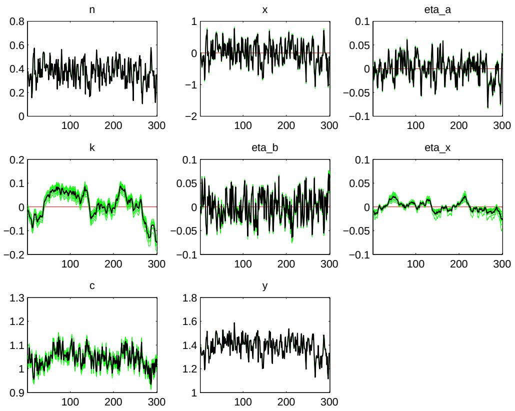
- *smoother* computes posterior distribution of smoothed endogenous variables and shocks, i.e. infers about the unobserved state variables using all available information up to T:

$$x_{t|T} = E[x_t|I_T]$$

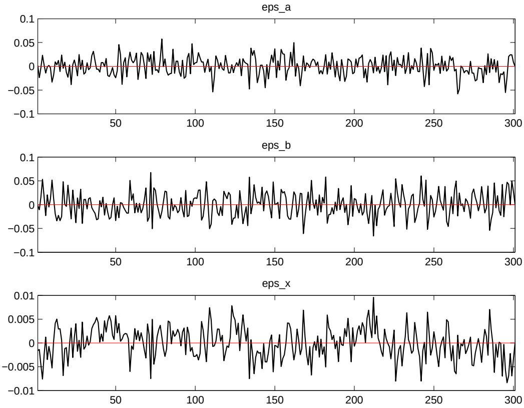
Filtered variables



Smoothed variables

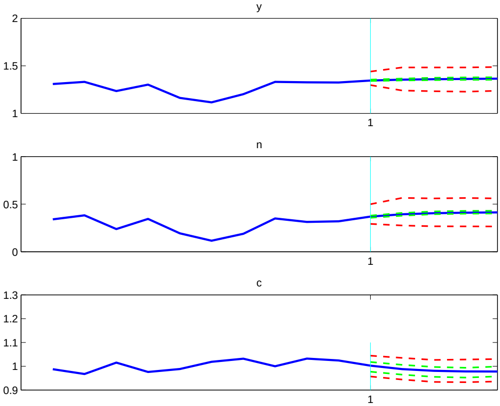


Smoothed shocks



Out of sample forecast

Add *forecast=10* to forecast 10 periods.



Save some time

- *mode_file=filename_mode* uses former mode. This mode file is automatically generated by Dynare. Don't forget: Set *mode_compute=0*.
- Continue an old markov chain: *load_mh_file*.

Markov chain mechanism

- Given θ^{i-1}, draw the parameter vector θ from a joint normal distribution (proposal distribution):

$$\theta^i \sim \mathcal{N}(\theta^i, c^2\Sigma)$$

  where Σ denotes the inverse Hessian evaluated at the posterior mode and c a scaling factor.
- Denote the logobjective function as $l(\theta)$. The draw is then accepted with probability:

$$min(1, exp(l(\theta^i) - l(\theta^{i-1})))$$

- Repeat this until the distribution has converged to the target distribution.
- $\rightarrow$ The average acceptance rate and therefore the speed of convergence depend on the scaling parameter c.

Scaling the chain

- Recommended is an accepted rate of about 0.23. The optimal scale factor has to be found by try and error.
- *mh_jscale* sets the scaling parameter.
- *mh_init_scale* allows for a wider distribution for the first draw.

From mod to m file

Dynare produces three m-files. It is possible to set all options directly in the m-files. The m-file named as the mod-file contains all options. For example:

- *options_.mode_compute=0;*
- *options_.mode_file='filename_mode';*
- *options_.load_mh_file=1;*
- *options_.mh_jscale=0.43;*

Dynare and nohup

Dynare cannot be started by a *nohup* command. Instead:

1. Run the Dynare command on your desktop.
2. Open the now created m-file.
3. Set the folder containing Dynare on the server. For example: `path(path,'../dynare_v3/matlab')`
4. Upload the m-files and your dataset.
5. Run the m-file by the *nohup* command

Example:

```
nohup matlab -nodisplay <filename.m
>output.txt&
```

Create tex-tables

- Add *tex* to the *estimation* options.
- Additionally, the matrices *lgx_TeX_*, *lgy_TeX_*, *options_.varobs_TeX* and *estim_params_.tex* have to be defined.
- *lgx_TeX_* and *lgy_TeX_* have to be defined globally: `global lgx_TeX_ lgy_TeX_ ;`
- Run Dynare and stop it shortly afterwards.
- Type *lgy_*.

Create tex-tables II

Add the list to your mod-file in the following way:

```
lgy_TeX_ = 'c';
lgy_TeX_ = strvcat(lgy_TeX_,'eta_a');
lgy_TeX_ = strvcat(lgy_TeX_,'eta_b');
lgy_TeX_ = strvcat(lgy_TeX_,'eta_x');
lgy_TeX_ = strvcat(lgy_TeX_,'k');
lgy_TeX_ = strvcat(lgy_TeX_,'n');
lgy_TeX_ = strvcat(lgy_TeX_,'x');
lgy_TeX_ = strvcat(lgy_TeX_,'y');
```

Create tex-tables III

Repeat this for:

- *lgx_* → *lgx_TeX_*
- *estim_params_.param_names* → *estim_params_.tex*
- *options_.varobs* → *options_.varobs_TeX*

Tex output

	Prior distribution	Prior mean	Prior s.d.	Post. mean	HPD inf	HPD sup
A	norm	2.000	0.3000	2.3032	2.2346	2.3592
θ	beta	0.300	0.1000	0.3612	0.3518	0.3714
β	beta	0.990	0.0010	0.9899	0.9883	0.9914
δ	beta	0.025	0.0050	0.0256	0.0210	0.0301
ρ_a	beta	0.700	0.1500	0.5903	0.5317	0.6954
ρ_b	beta	0.700	0.1500	0.6308	0.4965	0.8801
ρ_x	beta	0.700	0.1500	0.5980	0.4231	0.6856

Table: Results from Metropolis Hastings (parameters)