Institute of Economic Policy I Alexander Kriwoluzky

Introduction to Dynare

Handout

1 Dynare: Introduction.

- Download Dynare and additional information and a wonderful user guide from `http://www.cepremap.cnrs.fr/dynare/`.
- How does Dynare work?
  - The user writes a mod-file.
  - Dynare produces an m-file from it.
  - It solves non-linear models with forward looking variables.
  - It estimates the parameters of those models.
- 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.

2 Example: Slightly extended RBC model

We will simulate data from the model by the *stoch_ simul* command. Afterwards the simulated dataset is used to estimate the parameters of the model.

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

s.t.

$$
\begin{aligned}
c_t + x_t &= y_t && (2)\\
y_t &= e^{\eta_{z,t}} k_{t-1}^{\theta} n_t^{1-\theta} && (3)\\
e^{\eta_{x,t}} x_t &= k_t - (1-\delta) k_{t-1} && (4)
\end{aligned}
$$

$$
\begin{aligned}
\eta_{a,t} &= \rho_a \eta_{a,t-1} + \epsilon_{a,t},\ \epsilon_{a,t} \sim N(0, \sigma_a^2)\ i.i.d. && (5)\\
\eta_{b,t} &= \rho_b \eta_{b,t-1} + \epsilon_{b,t},\ \epsilon_{b,t} \sim N(0, \sigma_b^2)\ i.i.d. && (6)\\
\eta_{x,t} &= \rho_x \eta_{x,t-1} + \epsilon_{x,t},\ \epsilon_{x,t} \sim N(0, \sigma_x^2)\ i.i.d. && (7)
\end{aligned}
$$

The structural shocks are assumed to be uncorrelated.

3 Structure of a mod-file

3.1 Declaration of the variables, parameters and shocks

- 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;`
- 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;
  ```
- 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;
```

3.2 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.

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(+1)-theta*beta*y_k*y(+1)+theta*beta*y_k*k;
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;
```

3.3 Steady state of the model

3.3.1 Steady state computation by *initval*

- 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 for initial starting values.

```
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;
```

3.3.2 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`
- 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;
```

3.4 Solving the model

- 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 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',[])`
- Additional options can be set in brackets after *'stoch_ simul'*:
  - '*periods*' - specifies the number of simulation periods. Example: `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.

3.5 Output

Dynare prints the following:

- Summary of variables:
  - Number of variables: 8
  - Number of stochastic shocks: 3
  - Number of state variables: 4 - number of predetermined variables
  - Number of jumpers: 4 - variables that appear with a lead
  - Number of static variables: 2 - variables that appear neither with lag nor lead
- Matrix of Covariance of exogenous shocks
- Recursive law of motion
- Moments, Correlation and Autocorrelation of simulated variables
- All results are stored in `filename`_results.mat:
  - The matrix dr_ contains e.g.:
    * Recursive law of motion (*ghx*, *ghu*).
    * Eigenvalues
    * Steady state (*ys*)
  - The matrix oo_ contains e.g.:
    * Posterior mode and std
    * Marginal density
    * Smoothed shocks

4 Estimating the model

4.1 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 you can do the following:

  - 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.:

$$y\_obs = y - y(-1) + log(psi)$$

4.2 Prior distribution

- 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.
- Have a look at Del Negro and Schorfheide (2007) for a discussion of prior distributions.
- See figure 1.

Example 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;
```

4.3 The estimation routine

- 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 include:

    - *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*
    - The Dynare userguide offers a very good description of all options available.

4.4 How to produce and interpret figures

To produce figures add the option of interest into the brackets after the *estimation* command.

4.4.1 Mode check

- Example: `estimation(mode_check)`.

- See figure 3.

- The figure plots how the objective function changes if the respective parameter is varied while the other held constant.

- Make sure to observe a minimum for each parameter.

4.4.2 Prior vs. Posterior

- For each parameter Dynare plots the prior and the posterior distribution in one figure.
- The grey line represents the prior, the black line the posterior. Both should be different from each other. In case there are not the parameter is not identified.
- The dotted green line represents the value at the posterior mode. Ideally the mode is in the center of the posterior distribution.
- See figure 2 for an example.

4.4.3 Diagnostics

Convergence of the Markov chain is important.

- 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*.
- Dynare plots one multivariate statistic (figure 4) and several univariate ones (figure 5)-depending on the number of parameters.
- In the convergence diagnostic figure the red and blue line represent specific within and between chain measures: [1]
  - Interval statistic constructed around parameter mean.
  - M2 statistic a measure of the variance.
  - M3 based on third moments.
- Both lines (red and blue) should be constant and should converge.
- The multivariate convergence statistic as an aggregate measure is based on the eigenvalues of the variance-covariance matrix.

[1] For more information on the measures see Brooks and Gelman (1998)

4.4.4 Bayesian impulse response function

- Example: `estimation(bayesian_irf, irf=10) y,n`.
- See figure 6.
- Length of IRF is controlled with *irf*.
- The variables for which IRF should be plotted can be named after the brackets.
- Dynare generates IRF for each parameter vector draw. This yields the distribution of IRFs.

4.4.5 Filtered and smoothed variables

- *filtered_vars* triggers the computation of filtered variables, i.e. forecast on past information (see figure 7):

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

- *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 (see figure 8):

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

- The plot of smoothed shocks is always produced. It also serves as a check for the estimation → the shock realizations should be around zero. (see figure 9)

4.4.6 Out of sample forecast

- Example: `estimation(forecast=10)` yield a forecast 10 periods.
- There are two confidence intervals plotted (see figure 10):
  - The green lines correspond to parameter uncertainty. The way it is computed is similar to the Bayesian IRF.
  - The red lines take the possibility of future shocks into account.

5 Additional insights

5.1 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*.
- Example: `estimation(mode_file=filename_mode, mode_compute=0, load_mh_file)`

5.2 Markov chain mechanism

The MCMC mechanism can be summarized in the following steps [2]:

- 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.
- → The average acceptance rate and therefore the speed of convergence depend on the scaling parameter c.
- Recommended is an accepted rate of about 0.23 (see Roberts et al. (1994) for a formal derivation). The optimal scale factor has to be found by trial and error.
- *mh_jscale* sets the scaling parameter.
- *mh_ init_ scale* allows for a wider distribution for the first draw.

5.3 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;*

[2] Have a look at Schorfheide (2000) and An and Schorfheide (2006) for a detailed description of the algorithm.

5.4 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&
```

5.5 Create tex-tables

- Run Dynare and stop it shortly afterwards.
- Type *lgy_*
- Then define *lgy_TeX_* in your mod-file.
- Add the list from *lgy_* to your mod-file in the following way:(*lgy_* **and** *lgy_Tex* **has to have the same order!**)

```
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');
```

- To set the names of the parameters first look up the order by typing `estim_params_.param_names`. Then define the matrix `estim_params_.tex` in you mod-file:

```
estim_params_.tex = 'A';
estim_params_.tex = strvcat(estim_params_.tex,'theta');
estim_params_.tex = strvcat(estim_params_.tex,'beta');
```

```
estim_params_.tex = strvcat(estim_params_.tex,'delta');
estim_params_.tex = strvcat(estim_params_.tex,'rho_a');
estim_params_.tex = strvcat(estim_params_.tex,'rho_b');
estim_params_.tex = strvcat(estim_params_.tex,'rho_x');
```

- Repeat this for:
  - $lgx\_ \rightarrow lgx\_TeX\_$
  - $options\_.varobs \rightarrow options\_.varobs\_TeX$
- Additionally $lgx\_TeX\_$ and $lgy\_TeX\_$ have to be defined globally: `global lgx_TeX_ lgy_TeX_ ;`
- Do not forget to add *tex* to the *estimation* options.

This yields for example:

	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 1: Results from Metropolis Hastings (parameters)

References

An, S. and F. Schorfheide (2006). “Bayesian analysis of DSGE models”. Technical report.

Brooks, S. and A. Gelman (1998). “Some issues in monitoring convergence of iterative simulations”.

Del Negro, M. and F. Schorfheide (2007). “Forming Priors for DSGE Models (and How It Affects the Assessment of Nominal Rigidities)”. CEPR Discussion Papers 6119, C.E.P.R. Discussion Papers. available at http://ideas.repec.org/p/cpr/ceprdp/6119.html.

Roberts, G., A. Gelman and W. Gilks (1994). “Weak convergence and optimal scaling of random walk Metropolis algorithms”.

Schorfheide, F. (2000). “Loss Function-Based Evaluation of DSGE Models”. *Journal of Applied Econometrics*, 15(6), 645–670.

Figures

Figure 1: Plot of prior distribution

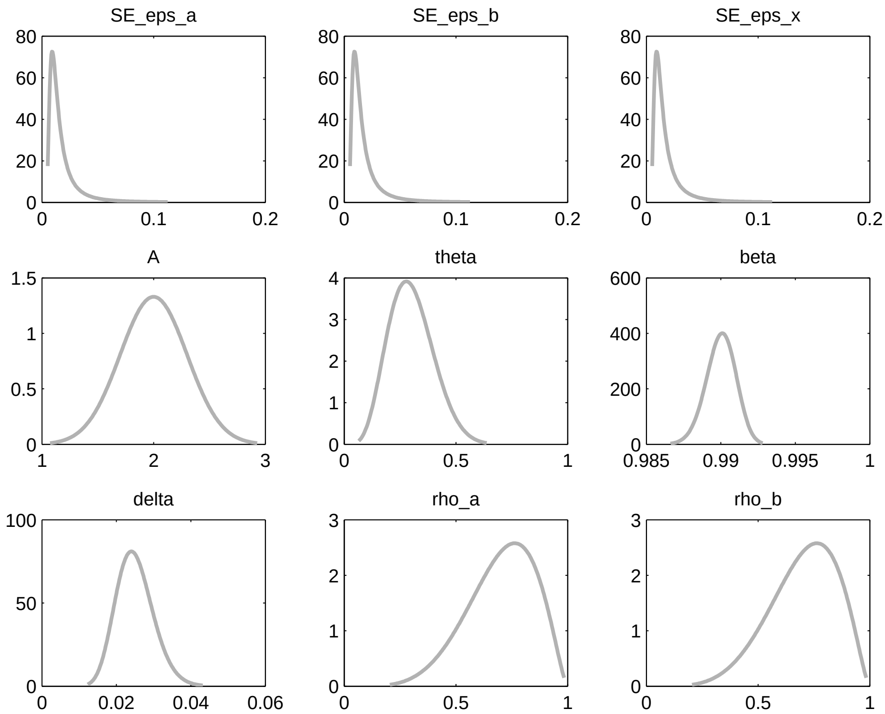

Figure 2: Prior vs. Posterior

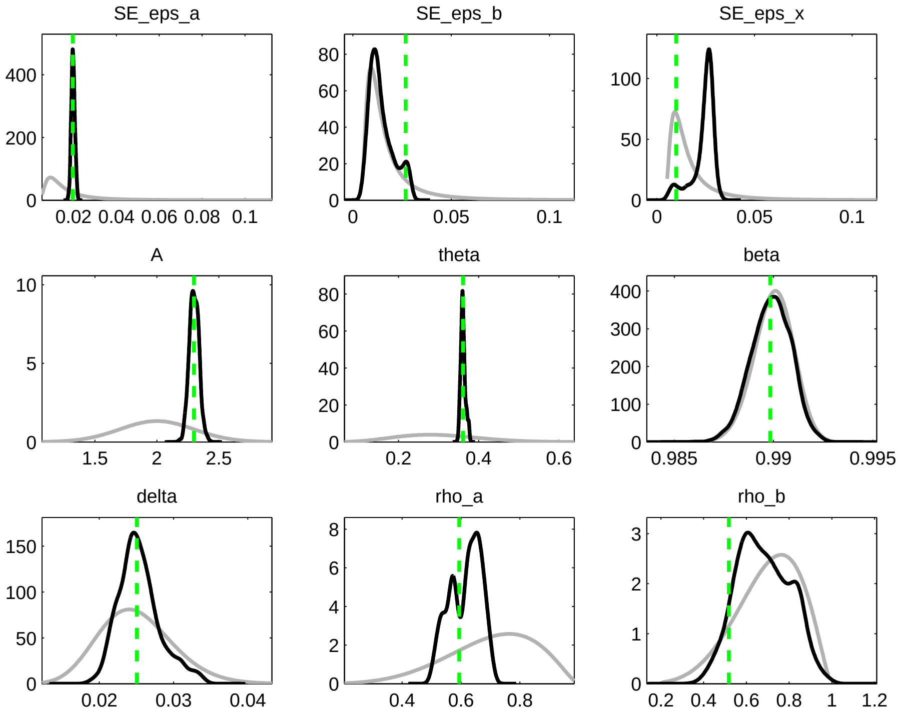

Figure 3: Mode check plots

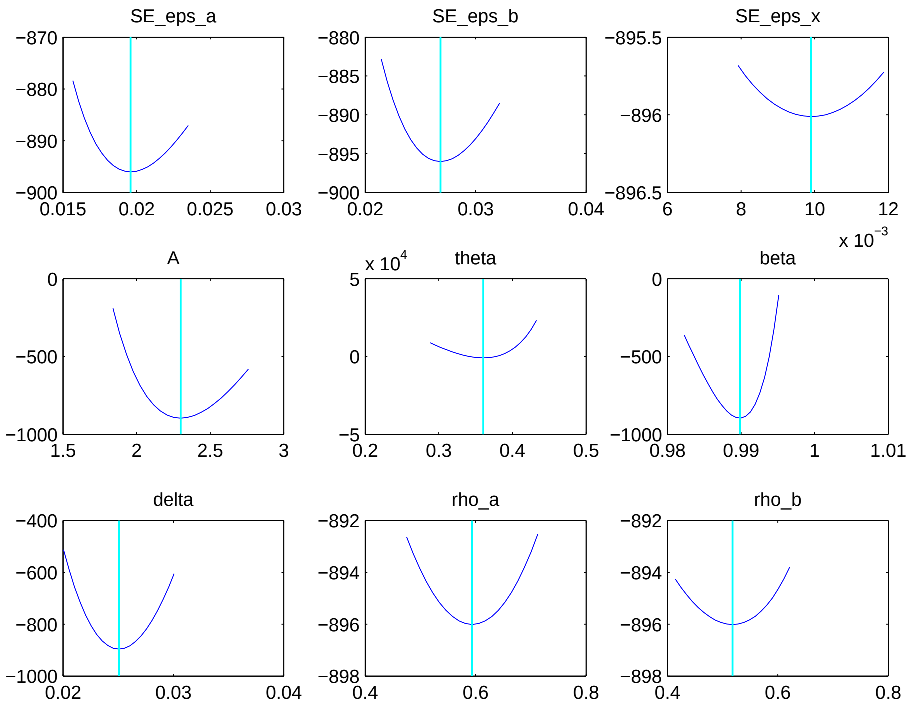

Figure 4: Plot multivariate convergence statistics

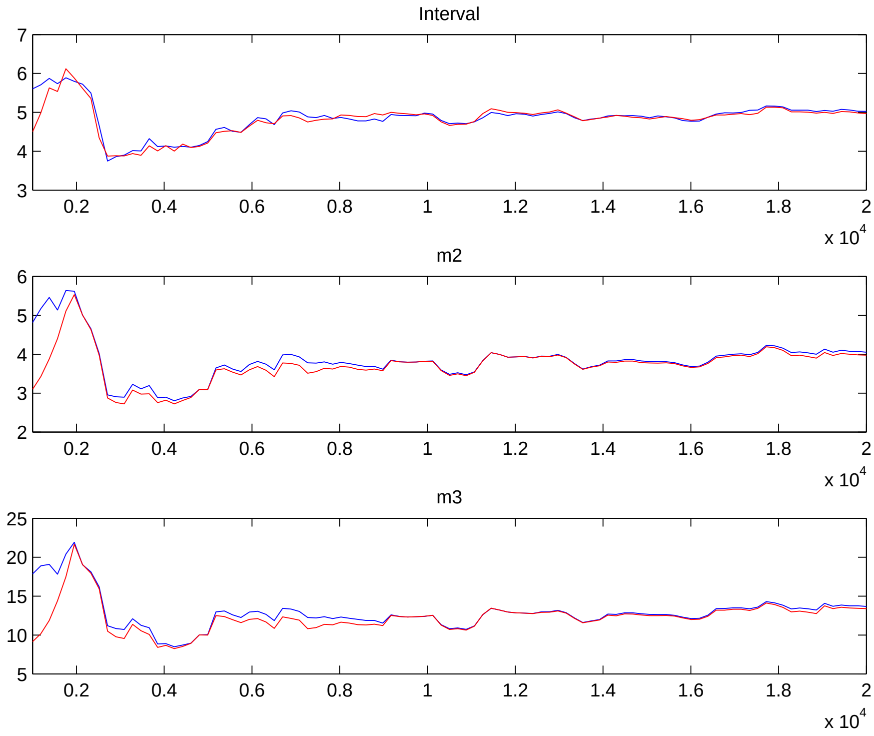

Figure 5: Example univariate convergence figure

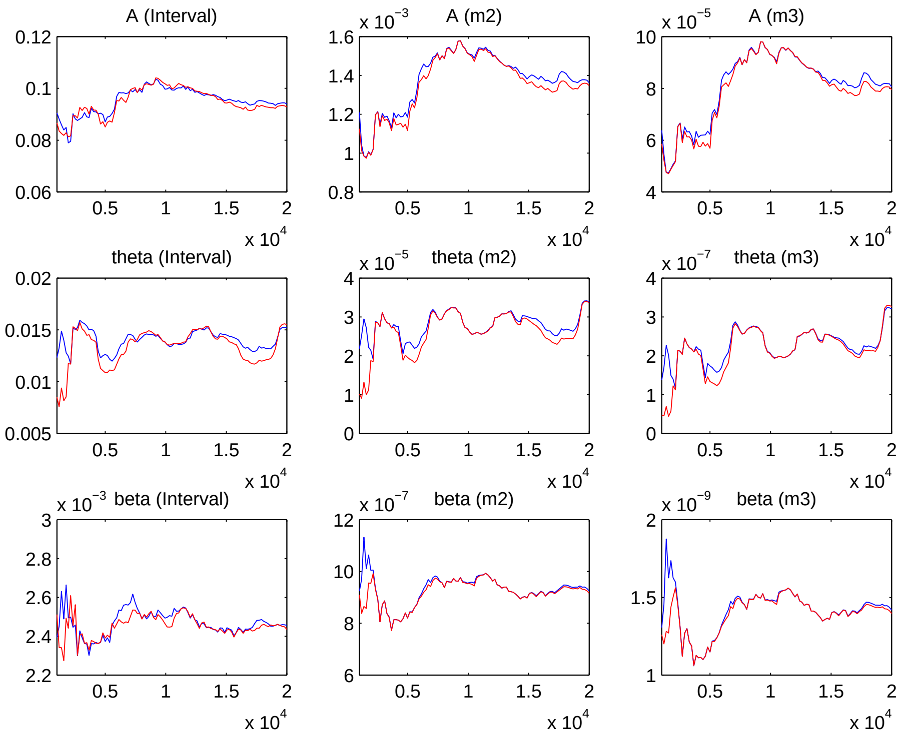

Figure 6: Bayesian IRF

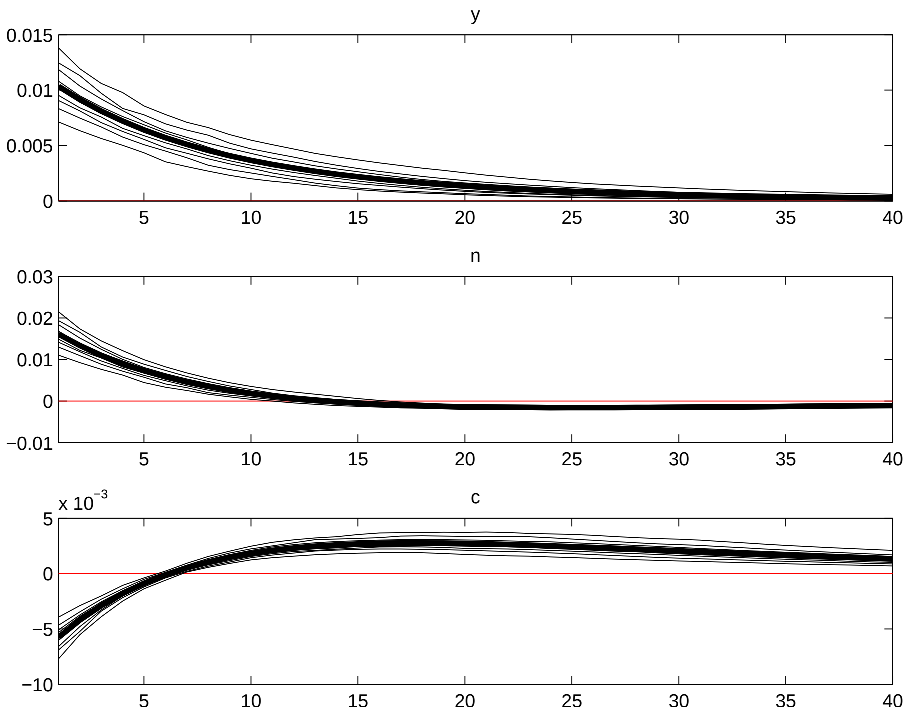

Figure 7: Filtered variables

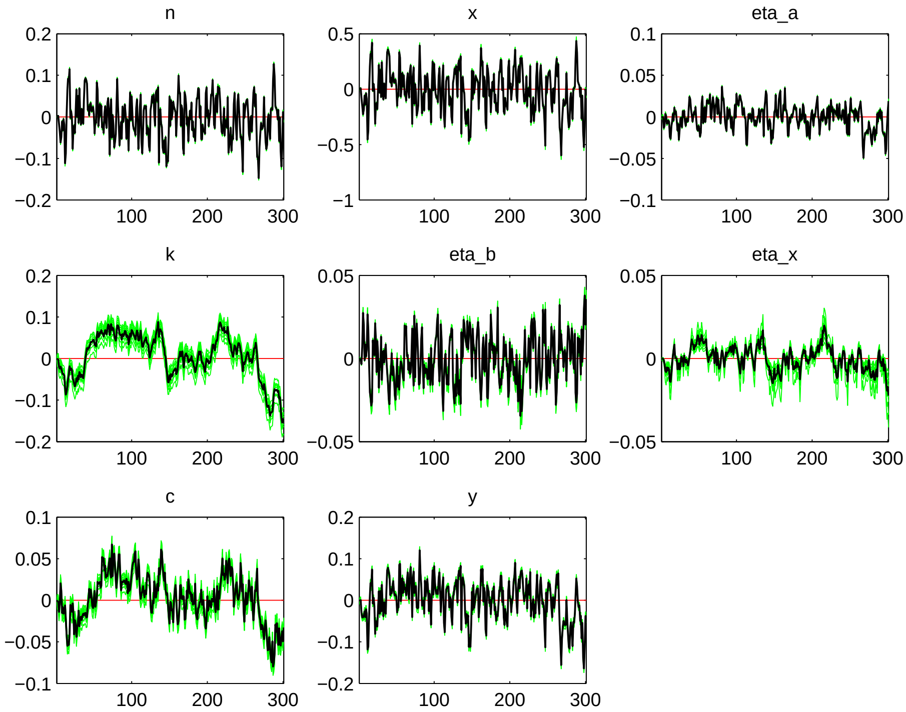

Figure 8: Smoothed variables

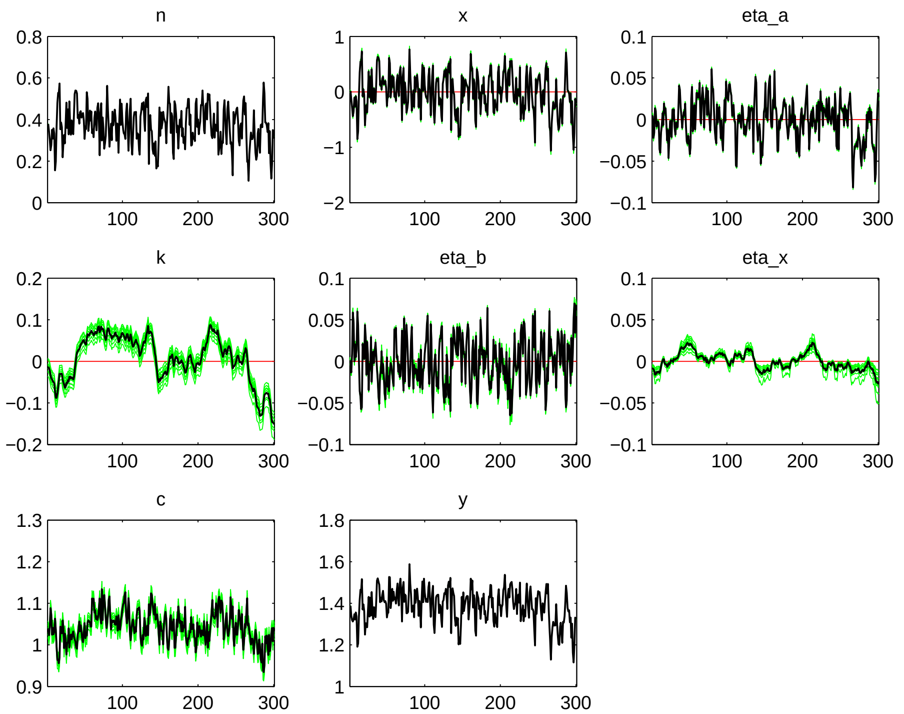

Figure 9: Smoothed shocks

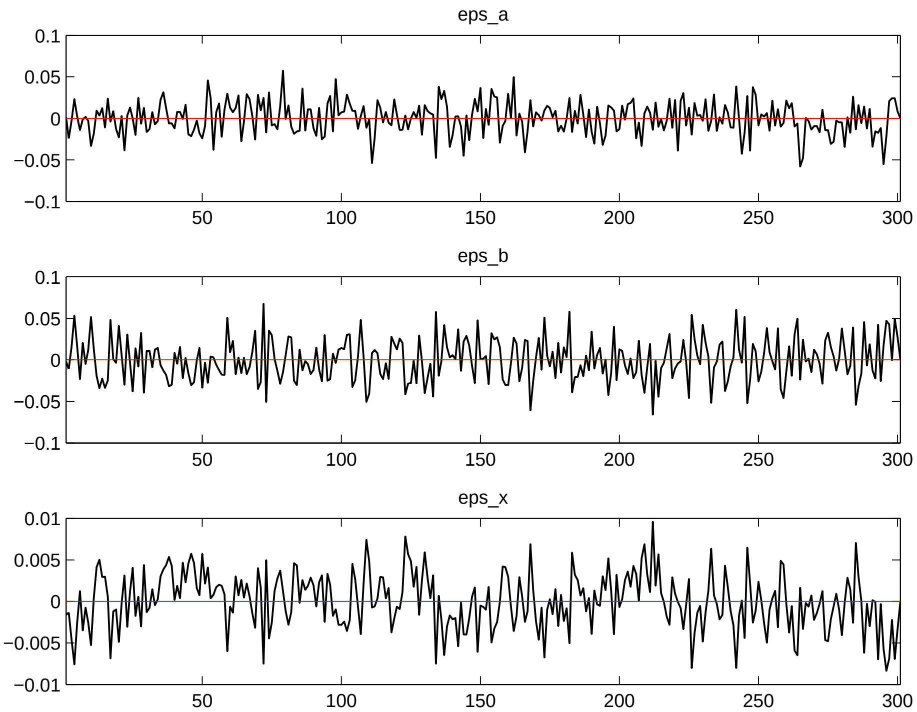

Figure 10: Out of sample forecast