Estimating DSGE Models with Dynare

Fabio Canova
BI Norwegian Business School, CAMP, FSBF, and CEPR
April 2017

Outline

- The syntax

- Some examples

- Model_comparison and shock_decomposition.

1 Syntax

- Declaration of the observable variables
- Declaration of the estimated parameters
- Estimation command
- Model_comparison and shock_decomposition commands

Declaration of observable variables

To declare the variables needed for estimation use

varobs

- The variables you declare must be in the same order as the you have them in your data file. So if the data has output, consumption and investment they should be listed as:

Example 1.1 *varobs y c i;*

- If you have some idea of how trends in the observable variables relate to the parameters of the model you can use the command:

observation_trends;

The block of commands starts with "observation_trends" and terminates with "end". In the middle you need to specify the variables and how the trends in these variables are related to the parameters of the model.

Example 1.2 *observation_trends,*
y (mu/psi);
end;

• This command specifies only **linear** trends.

• Variables shouldn't be centered (demeaned) when observation_trends is used.

Declaration of estimated parameters

To declare which parameters are to be estimated by classical or Bayesian maximum likelihood use

estimated_params;

- The first part of the line consists of one of the three following alternatives:

i) stderr variablename: Indicates that the standard error of either the exogenous variable (variablename), or of the observation/measurement error associated with the endogenous observed variable (variablename), is to be estimated.

ii) corr variablename1, variablename2: Indicates that the correlation between the exogenous variables (variablename1,variablename2), or the correlation of the observation/measurement errors associated with endogenous observed variables (variablename1,variablename2), is to be estimated. Correlations set by previous shocks-blocks or estimation-commands are kept at the value set prior to estimation if they are not estimated again. Thus, the treatment is the same as in the case of calibrated parameters.

iii) parametername: The name of a model parameter to be estimated.

- In the block within "estimated_params" and "end" you need to specify, a) the parameters to be estimated, b) the initial conditions and c) the upper and lower limits, where the likelihood routine will search for the maximum.

Example 1.3 *estimated_params,*
alpha, 0.3, 0.01, 0.5;
zeta, 0.75, 0.2, 0.9;
end;

- If you do not specify the upper and lower bounds, estimation will be unconstrained.

- With maximum likelihood, parameters often end up on the boundary. To avoid this you may want to estimate a transformation of the parameter which has double infinity support.

To do this use the character # to tell dynare how the transformed parameter is related to the original parameter.

Example 1.4 *parameters gamma;*

```
model;
# sig= 1/(1+gamma)
y = y(+1)+sig*x;
end;

estimated_params,
gamma, 0.3, 0.01, 0.5;
end;
```

In this example, you estimate the sig even if the relevant model parameter is gamma. The first line indicates how sig and gamma are related.

For Bayesian estimation there are other columns that need to be specified in the estimated_params command. They describe:

i) The shape of the prior distribution. This could be beta (beta_pdf), gamma (gamma_pdf) normal (normal_pdf); uniform (uniform_pdf); inverse gamma (inv_gamma_pdf); or generalized versions of the beta and the gamma distributions.

ii) The mean of the prior distribution.

iii) The standard deviation of the prior distribution.

iv) Two other parameters define the third and the forth parameters of the generalized beta and generalized gamma distributions (the defaults are, respectively, 0 and 1).

v) A value which gives the jump that the MH algorithm will do for this parameter. This value can be used to accelerate convergence in some dimensions and overrides the value given mh_jump option in the estimation command.

Note that it is possible to specify only a subset of the parameters for Bayesian estimation. However if the value in v) needs to be inputted, the values in iv) need to be specified.

Examples of various syntax options:

Example 1.5 *estimated_params,*
gamma, 0.3, 0.01, 0.5, normal_pdf, 0.2, 0.5, 0, 1, 0.002;
end;

```
estimated_params;
gamma, normal_pdf, 0.2, 0.5, 0, 1, 0.002;
end;

estimated_params;
gamma, normal_pdf, 0.2, 0.5;
end;

estimated_params;
stderr e_m, inv_gamma_pdf, 0.008862, inf;
end;
```

Clearly it is possible to estimate with classical ML within a Bayesian framework. The following example, set up the estimated_parameter command to do this.

Example 1.6 *estimated_params,*
gamma, 0.3, uniform_pdf, 0.2, inf;
end;

Estimation

• The command for classical or Bayesian likelihood estimation is the same. Some options are different.

The command to be used for estimation is

estimation(options) variable_name;

The options common to both approaches are

i) datafile=file_name, where the file will typically be a matlab file of either ".m" or ".mat" type. You can use a ".xls." files but you have to add two options: xls_sheet; xls_range to tell dynare in which sheet of the file the data is and what is the range you want to use.

ii) nobs=[a:b]. To be used if you want to use only a subset of the observations in the file from a to b.

iii) first_obs=a, where a is the number of the first observation to be used (default is a=1).

iv) prefilter. If this option is used, Dynare will demean the data prior to estimation.

v) presample=a, where a is the number of observations to be skipped before evaluating the likelihood. These first observations are used as a training sample (default is a=0).

vi) loglinear. This will trigger Dynare to compute estimates for the log-linear version of the model. If it is not specified and the equations of the model are inputted nonlinearly, Dynare will compute a linear approximation.

vii) nograph. No graphs produced will be plotted.

viii) nodisplay. Do not display the graphs, but still save them to disk (unless nograph is used).

ix) conf_sig=a, where a is a number defining the confidence interval used for classical forecasting after estimation. The default is a=0.90, more common is to set a=0.95 or a=0.68.

x) mh_conf_sig=a, where a is the credible/HPD interval used for the computation of posterior statistics. The default is a=0.9, more common is to set a=0.95 or a=0.68.

xi) sub_draws=a, where a is number of draws from the Metropolis iterations used to compute posterior distribution of various objects. sub_draws should be smaller than the total number of available Metropolis draws available. The default is min(1200,0.25*Total number of draws). A good choice is 0.10*Total number of draws.

xii) mh_recover. Attempts to recover a Metropolis-Hastings simulation that crashed prematurely. Shouldn't be used together with load_mh_file.

xiii) mode_compute=a. This option chooses the routine to compute the mode of the likelihood or of the posterior. a can take 7 different values. If a=0 the mode is not computed; a=1 uses the matlab function fmincon.m; (constrained optimization); a=3 uses the matlab function fminunc.m (unconstrained optimization); a=4 C.Sims' routine csminwell.m; a=5 M.Ratto's routine newrat.m; a=6 a monte carlo based optimization routine (very time intensive) a=7 uses the matlab routine fminsearch.m (a simplex algorithm), a=8 uses Dynare implementation of the Nelder-Mead simplex based optimization routine, a=9 uses the CMA-ES (Covariance Matrix Adaptation Evolution Strategy) algorithm, a=10 uses the simpsa algorithm (a combination of simplex and simulated annealing algorithms).. Refer to the Dynare manual for the options specific to each routine.

This is one of the most important options. It needs to be selected carefully since poorly specified models may not be estimable with some of these routines. The default is a=4; a=5 is typically good but slow; a=7,or a=8 should be used to get some good initial estimates (they do not compute standard errors) to be used with a=4 or a=5.

xiv) mcmc_jumping_covariance=option. Tells Dynare which covariance to use for the proposal density of the MCMC sampler. The available options are:

- Hessian: Uses the Hessian matrix computed at the mode.

- prior_variance: Uses the prior variances, no infinite prior variances are allowed in this case.

- identity_matrix.

- filename: Loads an arbitrary user-specified covariance matrix from filename.mat. The covariance matrix must be saved in a variable named jumping_covariance, must be square, positive definite, and have the same dimension as the number of estimated parameters.

xv) mode_check. It is used to diagnose problems with the computation of the mode of the function. It will plot how the (negative) of the log-likelihood or of the posterior look like in the neighborhood of the mode.

Problems exist if the mode is on the boundary of some parameter or if it is (considerably) away from the extremum of the function (for example, if the function looks unbounded in that direction).

xvi) lik_init=a This specify how to initialize the Kalman filter used to compute the likelihood function. If a=1, the initialization is done using the steady state distribution. If a=2, the initialization is done by setting the mean of the initial value to zero and the variance of the initial value to 10 for each state variable. The default is Lik_init=1; but Lik_init=2 is preferable is the data look nonstationary (possibly after demeaning and detrending).

The options specific for Bayesian estimation are

xvii) plot_priors=a. If a=0 no plot is made; if equal to 1 plots are made.

xviii) mh_replic=a, where a is the number of replications computed in each chain of the MH algorithm.

xix) mh_nblocks=a, where a is number of chains computed (usually 2,3 are sufficient).

xx) mh_drop=a, where a is the number of elements of the chain dropped before computing posterior estimates. The default is a=0.5, but this may be too small when the model is of large dimension or convergence is problematic.

xxi) mh_jscale=a, where a controls the variance of the jump in the MH algorithm. This is a crucial parameter you need to set it to hit an acceptance rate of 25-40 percent. The default is a=0.2, but this is almost never a good value.

xxii) mh_init_jscale=a, value for the variance of the jump in the MH algorithm for the first draw. The default is a=2*mh_jscale.

xxiii) mode_file=filename. To start the MH algorithm you do not need to run the mode computation algorithm all the times. If you have computed the mode, you can store it in a file and call it within Dynare using this option.

xxiv) load_mh_file. This option will allow you to restart your MH algorithm using the files you have created in another run. For example, it can be used to run multiple chains at different times.

xxv) nodiagostics. If used no convergence diagnostics are computed. It can be used to suppress a lot of output once you have figured out that the chains have actually converged.

xxvi) dsge_var. It triggers the estimation of a dsge-var as in Del Negro and Schorfheide (2004). The weight between of the DSGE data will be estimated. The prior for this parameter needs to be specified in the estimated_params command. If rather than estimating, you want to calibrate this parameter, use dsge_var=a, where a is the calibrated weight.

xxvii) dsge_varlag=a, where a is the number of lags used to estimate a DSGE-VAR model (default is a=4).

The options for processing output are

xxviii) bayesian_irf=a. Computes the distribution of impulse responses using posterior draws of the parameters. a regulates the horizon length of the impulse responses. The default is a=40.

xxix) moments_varendo. Computes posterior moments for the endogenous variables.

xxx) conditional_variance_decomposition = a or [a b] or [a:b]. Computes the posterior distribution of the conditional variance decomposition for the specified period(s). Uses theoretical decision rules. Can't be used in conjunction with the option periods.

xxxi) forecast=a. Computes the distribution of forecasts up to horizon a.

xxxii) smoother. This option allows to compute the smoothed version of the endogenous variables and the shocks.

xxxiii) kalman_algo=a. If a=0 automatically uses the Multivariate Kalman Filter for stationary models and the Multivariate Diffuse Kalman Filter for non-stationary models (for further options refer to the Dynare manual).

xxxiv) filter_covariance; filter_steps_ahead=[a:b]; filter_decomposition: produce and save i) one step ahead error of covariance matrices; ii) [a:b] step ahead filtered values, and iii) shock decomposition of the [a:b] step ahead filtered values.

Example 1.7 *(Classical ML estimation)*

```
estimated_params;
stderr e_rn, 0.0025,0.001,0.5;
stderr e_g, 0.01,0.00001,0.5;
stderr e_a, 0.01,0.00001,0.5;
alpha, 0.2,0.1,0.3;
delt, 0.025,0.01,0.04;
rho_a, 0.99,0.95,0.995;
rho_g, 0.95,0.9,0.99;
```

```
psi, 0.25,0.1,0.4;
end;

varobs y rn pi;

estimation(datafile=bgg_estimation_data,mode_compute=5) y rn pi;
```

Example 1.8 *(Bayesian ML estimation)*

```
estimated_params;
phi,NORMAL_PDF,2,0.05;
nu,NORMAL_PDF,2,0.05;
psix,NORMAL_PDF,0.5,0.25;
psir,BETA_PDF,0.80,0.25;
psip,Normal_PDF,1.7,1.95;
```

```
rhog,BETA_PDF,0.6,0.25;
rhou,BETA_PDF,0.6,0.25;
stderr eg,INV_GAMMA_PDF,0.01,0.05;
stderr eu,INV_GAMMA_PDF,0.01,0.05;
stderr er, INV_GAMMA_PDF,0.01,0.05;
end;

varobs r x p;

estimation(datafile=rawdata_US_1948Q1_2004q4,first_obs=117,
nobs=100,mode_compute=5,mode_check,mh_replic=10000,mh_nblocks=2,
mh_jscale=0.65,mh_init_scale=0.5,bayesian_irf,moments_varendo) r, x, p ;
```

The computations that dynare reports after the estimation, are obtained using either the mode of the likelihood (classical) or the posterior mean (bayesian).

The mode computation produces a marginal likelihood computed with a Laplace approximation. This is stored in
oo_.MarginalDensity.LaplaceApproximation. The MH algorithm also compute an estimate of the marginal likelihood using an harmonic mean estimator. This is stored in
oo_.MarginalDensity.ModifiedHarmonicMean.

Example 1.9 *Results of the estimation of Schorfheide and Lubik (2007) open economy model:*

```
Actual dxnorm 5.9677e-06
FVAL 957.15
Improvement 9.7084e-08
Ftol 1e-05
Htol 1e-05
Gradient norm 638.0239
Minimum Hessian eigenvalue 0
Maximum Hessian eigenvalue 1.137610391340715e+23
Estimation successful.
```

RESULTS FROM POSTERIOR ESTIMATION (MODE ESTIMATION)

parameters	prior	mean mode	s.d.	prior	pstdev
alpha	0.200	0.1510	0.0458	beta	0.0500
tau	0.500	0.5152	0.0823	beta	0.2000
rhor	0.500	0.0037	0.0029	beta	0.2000
rhoq	0.400	0.4768	0.2382	beta	0.2000
rhophistar	0.800	0.6897	0.0839	beta	0.1000

sd of shocks	prior mean	mode	s.d.	prior	pstdev
ephistar	1.650	1.6338	0.0997	invg	0.1296
er	1.500	2.6965	0.0148	invg	0.1179
eq	4.500	3.4294	0.2020	invg	0.3536
eystar	4.500	4.3814	0.3332	invg	0.3536
ez	3.000	5.3915	0.0113	invg	0.2357

Log data density [Laplace approximation] is -978.384175.

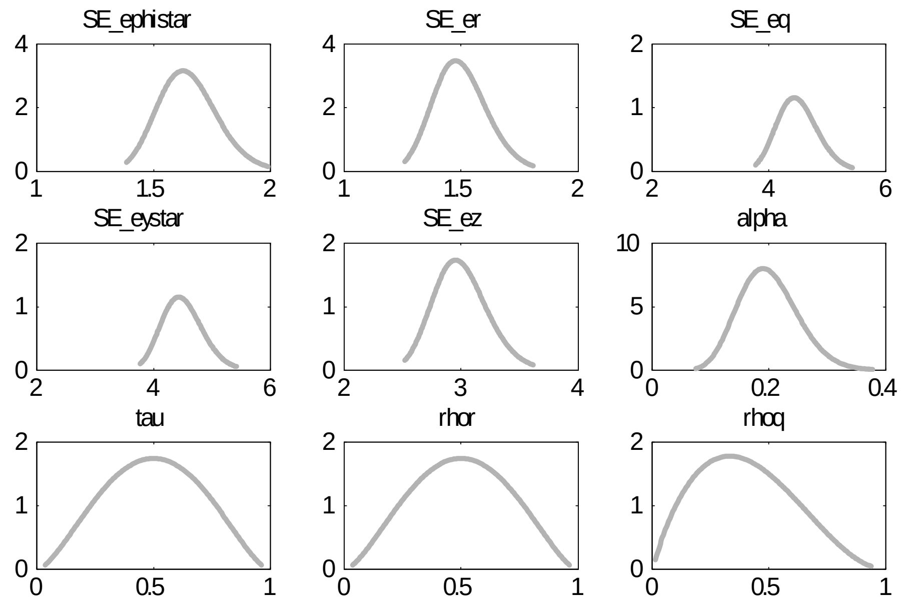
```
Estimation::mcmc: Multiple chains mode.
Estimation::mcmc: Number of mh files: 1 per block.
Estimation::mcmc: Total number of generated files: 2.
Estimation::mcmc: Total number of iterations: 10000.
Estimation::mcmc: Current acceptance ratio per chain: Chain 1: 21.5578 %
Chain 2: 22.0278 %

Estimation::mcmc: Total number of MH draws: 10000.
Estimation::mcmc: Total number of generated MH files: 1.
Estimation::mcmc: I'll use mh-files 1 to 1.
Estimation::mcmc: In MH-file number 1 I'll start at line 5000.
Estimation::mcmc: Finally I keep 5000 draws.
```

ESTIMATION RESULTS

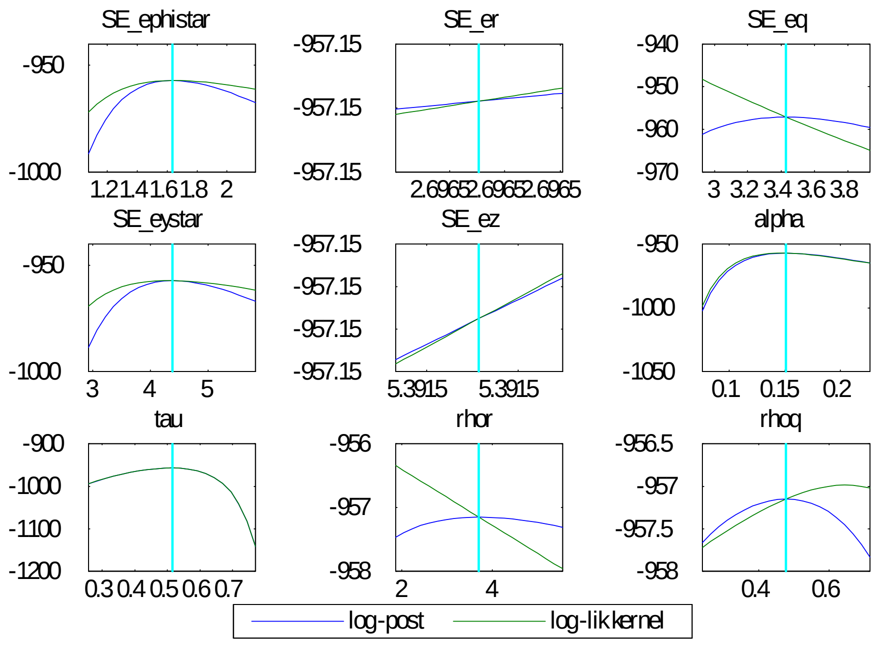
Log data density is -979.856512.

parameters	prior mean	post. mean	90% HPD	interval	prior	pstdev
alpha	0.200	0.1547	0.0821	0.2173	beta	0.0500
tau	0.500	0.5052	0.3722	0.6233	beta	0.2000
rhor	0.500	0.0064	0.0005	0.0127	beta	0.2000
rhoq	0.400	0.4504	0.1388	0.7492	beta	0.2000
rhophistar	0.800	0.6822	0.5465	0.8258	beta	0.1000

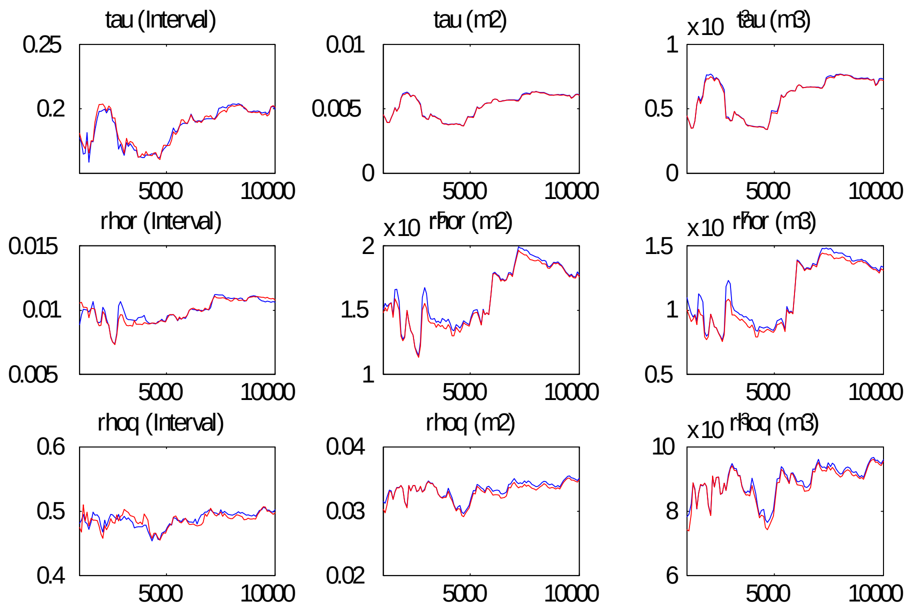
sd of shocks	prior mean	post. mean	90% HPD	interval	prior	pstdev
ephistar	1.650	1.6434	1.4893	1.7848	invg	0.1296
er	1.500	2.6549	2.5769	2.6965	invg	0.1179
eq	4.500	3.4509	3.0921	3.7509	invg	0.3536
eystar	4.500	4.4533	3.9527	5.0329	invg	0.3536
ez	3.000	5.3811	5.3670	5.3915	invg	0.2357


SE_ephistar
SE_er
SE_eq
SE_eystar
SE_ez
alpha
tau
rhor
rhoq
4
2
0
1
1.5
2
6
10
5
0.2
0.4
3
0.5

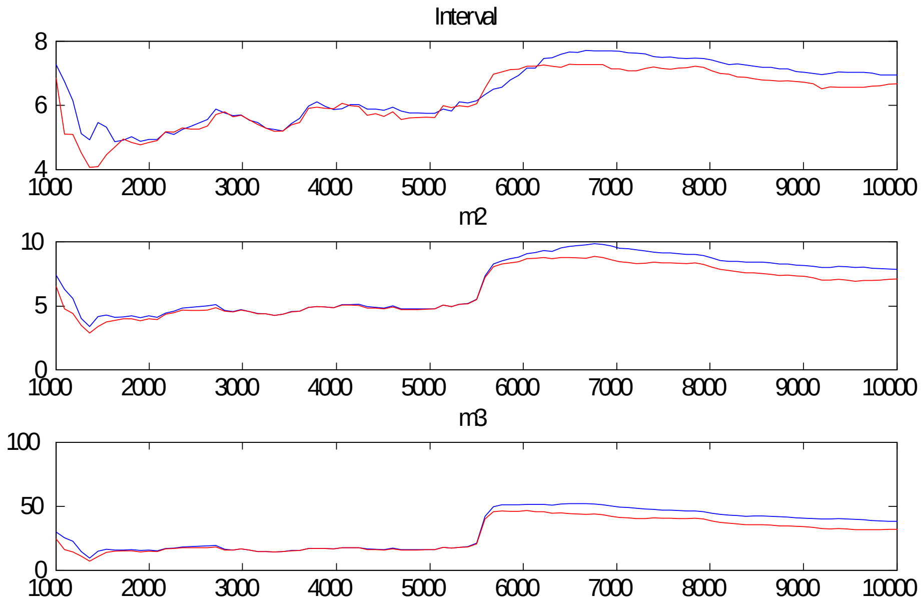
Priors


SE_ephistar
SE_er
SE_eq
SE_eystar
SE_ez
alpha
tau
rhor
rhoq
log-post
log-lik kernel

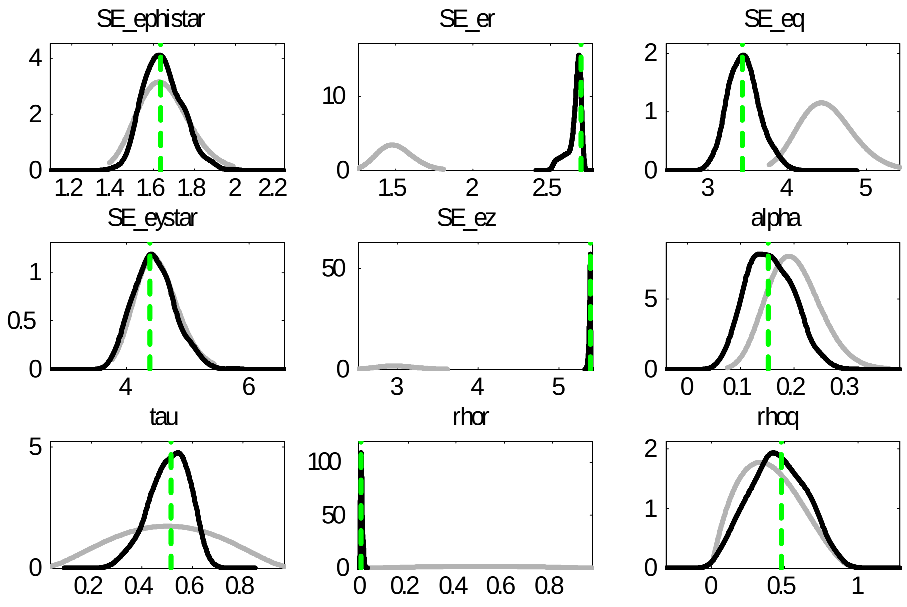
Mode check



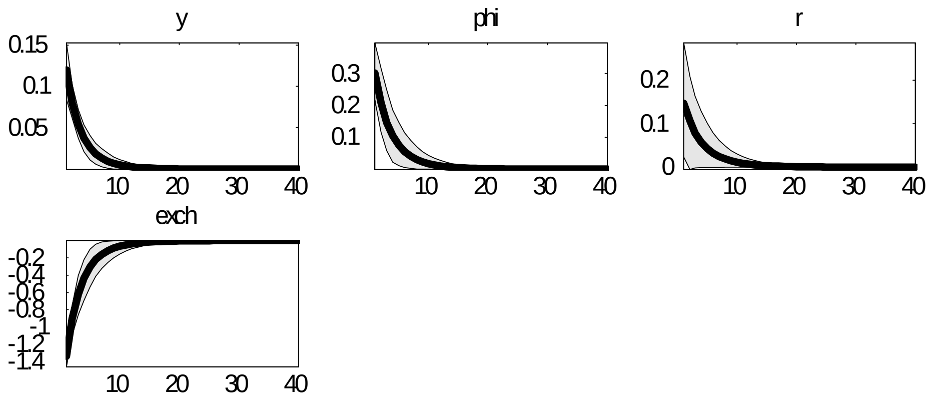
Univariate convergence diagnostics



Multivariate convergence diagnostics



Priors, mode and posteriors



Impulse responses

Model Comparison

To compute the Posterior odds ratio of two models use the command

model_comparison

This command requires you to specify the two models you consider and the prior weights.

Example 1.10 *model_comparison nk_model1 (0.5) nk_model2 (0.5)*

If you want to compare your model to a BVAR use the command

bvar_density

which compute the ML for a BVAR.

You can also run a BVAR and a DSGE jointly inside the same dynare file. The example below shows how to do it. It also shows how to forecast with a BVAR (for options see the manual).

Example 1.11 *var dx dy;*

```
varexo e_x e_y;
parameters rho_x rho_y;
rho_x = 0.5;
rho_y = -0.3;
model;
dx = rho_x*dx(-1)+e_x;
dy = rho_y*dy(-1)+e_y;
end;
estimated_params;
rho_x,NORMAL_PDF,0.5,0.1;
```

```
rho_y,NORMAL_PDF,-0.3,0.1;
stderr e_x,INV_GAMMA_PDF,0.01,inf;
stderr e_y,INV_GAMMA_PDF,0.01,inf;
end;
varobs dx dy;
check;
estimation(datafile = bvar_sample, mh_replic = 1200, mh_jscale = 1.3,
first_obs = 20);

bvar_density(bvar_prior_train = 10) 8;
bvar_forecast(forecast = 10, bvar_replic = 2000, nobs = 200) 8;
```

To compute historical decompositions use the command

shock_decomposition

The options for this command are:

i) parameter_set = a, where a could be prior_mode, prior_mean, posterior_mode, posterior_mean, posterior_median. The default is posterior_mode. When the MH algorithm is run, it uses the posterior_mean.

To do unconditional forecasting with an estimated model use the forecast option in the estimation command (refer to the set of slides "Solving DSGE Models with Dynare" for commands and options related with conditional forecasting).

To do conditional forecasting need two use a combination of three commands

conditional_forecast_path

It tells dynare a) which endogenous variable you want to fix; for how many periods and at what values.

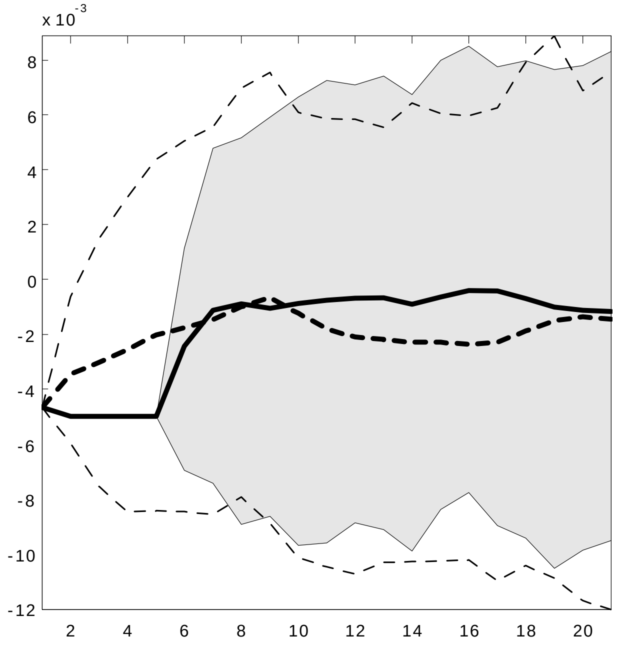
conditional_forecast

It tells dynare what parameters values it has to use (calibrated valeus, prior mean, posterior mean, etc.); what exogenous variables are adjusted to generate the path for the endogenous variables; and how many replications you are going to do to construct forecast bands.

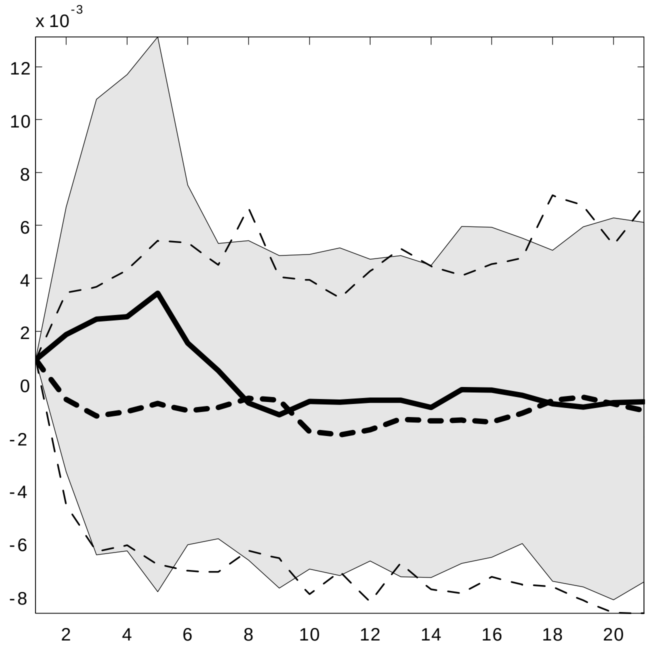
plot_conditional_forecast

It plot the results of the exercise

Example 1.12
```
conditional_forecast_paths;
var r;
periods 1:4;
values -0.005;
end;

conditional_forecast(parameter_set=posterior_mean, controlled_varexo=(er
), replic=100, periods=20);

plot_conditional_forecast x p r;
```



Conditional forecast of r, keeping it constant for 4 periods



Forecast of p, conditional on the path of r