Using Dynare:Troubleshooting

Wenddy Xu*
SFU and CIMERS
xuweny87@hotmail.com

October 28, 2019

Contents

To make sure this section is as user friendly as possible, the best is to compile what users have to say! Please let me know what your most common problem is with Dynare, how Dynare tells you about it and how you solve it. Thanks for your precious help!Main sources: Dynare Forum(`forum.dynare.org`).

I. Model Comparison

1. same data for models, same priors for models, but additional parameters for one, not for another. Then does that bias the posterior odds ratio/model comparison?
   Reply: For bayesian model comparison, models don't need to be nested and there is a natural degrees of freedom correction. Hence, as long as you use the same data, having different parameters does not at all.

2. Based on the formula of posterior odds ratio, since same data and prior, it is enough to compare marginal densities?
   Reply: That is sufficient. The prior over parameters does not matter, but the prior odds ratio over the models(see Koop(2003):Bayesian Econometrics). If you a prior assign equal probability to all models(0.5 for two models), you simply compare the marginal data densities.

3. From the estimation results:"Log data density" was compared using Modified Harmonic Mean and "Log data density [Laplace approximation]" via Laplace, right?
   Reply: Both the Laplace approximation and the modified harmonic mean estimation are ways to compute the marginal data density. As computing the marginal data density involves solving a complicated integral, these two methods for trackling the issue have been proposed. In theory, they should yield identical results as they measure the same thing. In practice, both involve approximations and may yield differing results.
   It is hard to tell which one to prefer. Footnote of SW(2007) state:" As discussed in John Greweke(1998), the MH-based sample of the posterior distribution can be used to evaluate the marginal likelihood of the model. Following Geweke(1998), we calculate the modified harmonic mean to evaluate the integral over the posterior sample. An alternative approximation is the Laplace approximation around the posterior mode, which is based on a normal distribution. In our experience, the results of both approximations are very close in the case of our estimated DSGE model. This is not too surprising, given the generally close correspondence between the histograms of the posterior sample and the normal distribution around the estimated mode for the individual parameters. Given the large advantage of the Laplace approximation in terms of computational costs, we will use this approximation for comparing alternative model specifications in the next section."

4. The decision rule for model comparison using marginal densities is: higher = better? e.g. model A(-950), model B(-1000), then choose model A.
   Reply: The marginal data density to be as high as possible. The logarithm is monotonic transformation, so we also want the log marginal densities to be as high as possible.

5. Do one need to make any transformation on these values first?
   Reply: It is often easier to compare models using posterior mode probabilities, see Koop(2003,p4).

6. Additional shocks matter? Reply: No matter. What matters is that you estimate those shocks' standard deviation. But this just another parameter. Note that: you do not need the same prior.

7. In the model_comparison command,

   - Bayes_Ratio refers to the ratio of Marginal Data Densities often called Bayes Factor[1].
   - Posterior_Model_Probability gives you the posterior probability of the models. It is not identical to the Posterior Odds Ratio, because the Odds Ratio gives you the relative instead of the absolute probabilities. But given that the probability must sum to 1, you can compute them.
     For example, if you compare two models and the Model 1 has a Posterior_Model_Probability of 2/3, then the other model has to have one of 1/3 and Posterior Odds Ratio is (2/3)/(1/3)=2:1.
   - Marginal Data Density(MDD) measure fit relative to other models on the same data. Note that it is pretty meaningless on its own.
   - The acceptance rate only tells you about efficiency of the MCMC. Note that it does not tell you about convergence.
   - The Geweke diagnostics for convergence.

II. Parameter Estimation

1. Whether it is possible to perform the parameter estimation using a very small sample with annual data?
   Reply: That is not ideal, but doable. The prior distribution will most probably play an important role in the case, i.e. the data will not be very informative.

[1] Bayes Factor,Wikipedia,
In statistics, the use of Bayes factors is a Bayesian alternative to classical hypothesis testing. Bayesian model comparison is a method of model selection based on Bayes factors. The aim of the Bayes factor is to quantify the support for a model over another, regardless of whether these models are correct.
Definition
The Bayes factor is a ratio of the likelihood probability of two competing hypothesis, usually a null and an alternative. The posterior probability $Pr(M|D)$ of a model M given data D is given by Bayesi theorem:

$$Pr(M|D) = \frac{Pr(D|M)Pr(M)}{Pr(D)}$$

The key data-dependent term $Pr(D|M)$ is a likelihood, and represents the probability that some data are produced under the assumption of this model, M; evaluating it correctly is the key to Bayesian model comparison.
Given a model selection problem in which we have to choose between two models on the basis of observed data D, the probability of the two different models M_1 and M_2, parameterised by model parameter vectors θ_1 and θ_2 is assessed by the Bayes factor K given by

$$K = \frac{Pr(D|M_1)}{Pr(D|M_2)} = \frac{\int Pr(\theta_1|M_1)Pr(D|\theta_1, M_1)d\theta_1}{\int Pr(\theta_2|M_2)Pr(D|\theta_2, M_2)d\theta_2} = \frac{Pr(M_1|D_1)}{Pr(M_2|D_2)}\frac{Pr(M_2)}{Pr(M_1)}$$

When the two models are equally probable a priori, so that $Pr(M_1) = Pr(M_2)$, the Bayes factor is equal to the ratio of posterior probabilities of M_1 and M_2. If instend of the Bayes factor integral, the likelihood corresponding to the maximum likelihood estimate of the parameter for each model is used, then the test becomes a classical likelihood-ratio test. Unlike a likelihood-ratio test, this Bayesian model comparison does not depend on any single set of that is automatically, and quite naturally, includes a penalty for including too much model structure. It thus guards against overfitting. For models where an explicit version of the likelihood is not available or too costly to evaluate numerically, approximate Bayesian computation can be used for model selection in a other approaches are:

- to treat comparison as a decision problem, computing the expected value or cost of each model choice;
- to use minimum message length(MML).

2. The model is stochastically singular.
   Reply: Either that the number of observation variables is greater than the number of shocks, or observation imply an exact linear combination(more likely). Either that or measurement errors.

3. Based on bayesian method, we have the formula:

   (a) log(posterior density)=log(likelihood)+log(prior density)

   (b) combining with Dynare, then log-post(blue line) is log(posterior density) and log-lik kernel(green line) is log(likelihood).

   (c) right?

   Reply: Right

4. (1)If the mode_check plot show that mode of log-post and log-lik kernel are identical, then we can consider this mode as the mean of proposal density for the MH algorithm. Otherwise, we have to find the mode again as long as mode of log-post and log-lik kernel are identical. Right? (2)If we can not find the truly global mode, then the MH algorithm can not converge. If such, how to find the truly global mode?
   Reply: From the Bayes theorem, we know that the posterior density is equal to the likelihood times the prior density divided by the marginal density of the data:

$$p(\theta|y_T) = \frac{p(y_T|\theta)p(\theta)}{p(y_T)}$$

   The numerator is posterior kernel (it's not a density, because it does not integrate to 1). If you are only concerned by the inference about the parameters(θ), you do not need to worry the denominator, all the information about the parameters is embodied in the posterior kernel. The mode_check option will return plots of the log of the posterior kernel and of the log likelihood as a function of one parameter, keeping the other parameters constant (equal to the estimated posterior mode). So if you has two parameters θ_1 and θ_2, and if your estimation of posterior mode is $(\hat{\theta}_1, \hat{\theta}_2)$, you will have the plots for:

$$f(\theta_1) == logp(y_T|\theta_1, \hat{\theta}_2) + logp(\theta_1, \hat{\theta}_2)$$

   and

$$g(\theta_1) == logp(y_T|\theta_1, \hat{\theta}_2)$$

   with θ_1 taking values around $\hat{\theta}_1$. and

$$f(\theta_2) == logp(y_T|\hat{\theta}_1, \theta_2) + logp(\hat{\theta}_1, \theta_2)$$

   and

$$g(\theta_2) == logp(y_T|\hat{\theta}_1, \theta_2)$$

   with θ_2 taking values around $\hat{\theta}_2$.
   So these curves are not the full multi-dimensional log likelihood and posterior kernel, but only two-dimensional cuts through these objects. Note also that because the prior density is the difference between the two objects, the log likelihood is typically smaller equal than posterior kernel (if the log prior density is positive), i.e. the graph of the likelihood would be below the posterior density and potentially not in the picture. For that reason, the likelihood is shifted

upwards to have the same maximum as the posterior kernel in order to better compare them. There is absolutely no reason to seek for an estimation where these cuts through the objects are identical. Indeed, if the prior brings some information to the inference about parameters, the curves have to be different, and in general the mode of the posterior kernel (or density) does not match the mode of the likelihood.
The last statement about the importance of finding the posterior mode is wrong. The MCMC will converge even if it starts from an initial state different from the (global) posterior mode, as long as for the initial state the posterior density is strickly positive, the jumping proposal density covariance is positive definite, and that all the usual assumptions on the posterior density are sutisfied. You will typically need a lot more iterations if the initial state is in a low posterior density region, but the MCMC will eventually converge to the posterior distribution.
Actually, as long as mh_nblocks is greater than one (the default being two). Dynare does not start the MCMC from the estimated posterior mode. Rather, the initial stae of each chain is chosen randomly to be overdispersed around the estimated posterior mode (the distance to the estimated posterior mode is controlled by option mh_init_scale).
The form of the proposal (or jumping) distribution matters more than the initial state of the chain(s) for determining the number of iterations needs to ensure convergence. It is essentially in this respect that the estimation of the posterior mode is important, because we use the inverse Hessian at the estimated mode as an approximation of the posterior covariance matrix.

5. In An and Schorfheide(2007) "Bayesian Analysis of DSGE model", for random-walk metropolis algorithm[2], they chose two jump parameters:mh_init_scale=1 and mh_jscale=0.3 (the rejection rate is about 45%). So is there any general rule to choose mh_init_scale and mh_jscale? In Dynare, the default value for mh_init_scale=2*mh_jscale, this jump factor should be adjusted, rihgt?
Reply:

   - mh_jscale should be adjusted to give an acceptance rate of 23%. For a multivariate normal posterior, this would be the most efficient choice to get quick convergence. For mh_init_scale, there is less good guidance. It should be bigger than mh_jscale to be overdispersed, but typically not too big, 2*mh_jscale is usually a good compromise.
   - The target 23% acceptance rate is at best a rule of thumb. To our knowledge, there is no proof that this would be optimal for a particular DSGE model. So We do not think that we should be obsessed by this target.
   - The choice for the parameter mh_init_scale does not affect the convergence of the MCMC. It is here to ensure that the initial conditions of the chains(if mh_b=nbloks¿1) are different enough. This parameter only affects the convergence diagnostics.

[2]As the name suggests, the random walk MH algorithm specifies the candidate generating density as a random walk:

$$\Phi^{G+1} = \Phi^{G} + e$$

where Σ the variance of e_t is set by the researcher. $\Sigma = \hat{\Omega} \times \lambda$, where $\hat{\Omega}$ is the estimated variance, λ is the scaling factor. A higher value for Σ could mean a lower rate of acceptances across the MH iterations (i.e. the acceptance rate is defined as the number of accepted MH draws divided by the total number of MH draws) but would mean that the algorithm explores a larger parameter space. In contrast, a lower value for Σ would mean a larger acceptance rate with the algorithm considering a smaller number of possible parameter values. The general recommendation is to choose Σ such that the acceptance rate is between 20% to 40%. We consider the choice of Σ in detail in the examples described below.
See Chib and Ramamurthy (2010), Blake and Mumtaz(2017) for a more efficient version of the basic algorithm described above.

- It works quite well in practice. Anything in the range of 20 to 30 percent should be ok!

6. How to adjust the jump scale to have a "reasonable" acceptance ratio of between 0.2 to 0.4 suggested by the literature?
   Reply: The tuning the scale parameter so that the acceptance ratio is between 0.2 to 0.4 is only a heuristic.From very simple models, we know that the acceptance rate should depend on dimension of the problem (the number of estimated parameters), and should be in this region. You can find an introduction on this in the book by Robert and Casella "Monte Carlo Statistical Methods"(Chapter 7, in particular section 7.8.4).
   We only know for sure that acceptance ratios close ro 0 or 1 are bad. I usually target one third. But we cannot be sure, controlling only one parameter, that the acceptance rates will be the same across chains. Normally in the end, if the chains are long enough the acceptance ratios should be similar across chains. A small acceptance rate does not mean that the MCMC is trapped in a low density region. Imagine that the current state of the MCMC rate is the posterior mode. If the jumps provided by the proposal distribution are large it is very likely that all the proposals will be rejected (resulting in a low acceptance rate).

7. (1)Dynare compute the RW Metropolis-Hasting acceptance ratio as R =number of accepted draws/number of proposals. Right?(2)Normally, we prefer that the acceptance ratio should not be very close to 0 or 1, an ideal ratio is around 1/3. To reach the ideal ratio, we should adjust the jump scale (in Dynare, we adjust by using mh_jscale). Based on that, one set my jump scale with 0.3. Moreover, one set number of the MH chain with 10 and number of iteration with 200,000. So then lauching Dynare, I found that: for the first MH chain, Dynare report the RW Metropolis-Hasting(1/10) Current acceptance ratio of around 0.256. However, the RW MH(2/10) Current acceptance ratio reduces a lot to 0.129 in the second MH Chain. The acceptance ratio of 0.129 is very far from the ideal ratio of 1/3. Even though I do not change the jump scale. The jump scale is still the same as 0.3 in the first MH Chain. Why the acceptance ratio is not identical for all MH Chain with the same jump scale?
   Reply:

   - The understanding of the acceptance ratio is correct.
   - This is simply impossible. You cannot tune a single parameter so that all the chains have the same acceptance ratios. The fact that the acceptance rate is significantly lower in one of the chains is a matter of chance(the initial condition of this chain may be far from the initial conditions of the other chains, and the proposals are obviously different). Frankly I would not bother about this. As long as all the acceptance ratios are strickly positive, there is no trouble.
   - I trend to somewhat disagree with the above answer. If your acceptance rates very different, it suggests that you are not sampling from the same distribution, i.e. at least one chain does not sample from the ergodic distribution we are interested in. As the above, the difference may just be due to the initial convergence to the ergodic distribution. But this then suggests that the burnin is very large relative to the actual draw, i.e. your chain are too short. For that reason, I would be extremely careful in checking convergence (and the initial mode).
     Also, you might want to consider fewer, but longer chains in this case.
   - This is most probably the problem, 200,000 is not enough. At this stage, i.e. as long as the chains did not converge to the (common) ergodic distribution, the total number of draw of 200,000 is not very pertinent, only the number of iteration per chain matters.

As the above suggested that reduce the number of chains (I never use more than 5), or run the estimation with the parallel capabilities of Dynare.

8. A common problem that plogues the estimation of DSGE models is the computation of the inverse of the Hession at the posterior mode, that is not positive definite. I am aware that the option mode_compute=5 offers a different way of computing the Hessian, optim={'Hessian',2}. This seems to work well, from what I have seen. Do other optimisers also offer similar options?
   Reply:

   - I believe the answer is no. However, mode_compute=6, which is based on MCMC, provides an estimate of the posterior covariance matrix not based on the inverse of the Hessian matrix(we use MCMC draws, so it works as long as the acceptance ratio is not too close to 0). Also, it is not mandatory to estimate the posterior mode(with Hessian at the estimated mode) for running a metropolis. You can optionally use another covariance matrix for the jumping distribution.
   - mode_compute =5 with optim={'Hessian',2} relies on the outer product of gradients and requires the use of a univariate Kalman fiter. Therefore, something similar is not avaiable for other optimizers. As the above indicated, mode_compute=6 also differs.
     The reason for using the inverse Hessian at the mode is that this is the most efficient choice if the posterior would be normal. But any positive definite matrix for the proposal density will do. You could provide any arbitrary matrix via command.
   - We use the draws of the Metropolis (run in mode_compute=6) to estimate the posterior covariance matrix. We need to have enough variability in these draws.If all the draws were identical (or more generally if the number of parameters) the sample covariance matrix would obviously not be full rank. That's why we target, by default, an acceptance rate of 1/3 (which is the value commonly considered in the literature.
     The optimization routine has several steps:

     1 we run a Metropolis, where we tune the scale parameter so that the acceptance rate is close to the target. In this first step the draws are not used to estimate the covariance matrix, we only continuously keep record of the vector of parameters which gives the best value for the objective. In this first step the covariance matrix if the jumping distribution is an identity matrix.

     2 we run a Metropolis to estimate the covariance matrix. Again we continuously keep record of the vector of parameters which gives the best value for the objective.

     3 We return to 1, with the new estimate of the posterior covariance matrix.
     By default, we iterate on these steps two or three times(I do not remeber, look at the reference manual). And in the last step we run a last metropolis, where we only update our estimate of the posterior mode. In this last round we decrease slowly the size of the jumps.

9. Why the Hessian found by the numerical optimisers is not positive definite?
   Reply: In Dynare we do not use the Hessian matrices returned by the optimizers. For instance, mode_compute=4, the default algorithm derived from Chris Sims code, returns a crude estimate of the Hessian that is not used by Dynare (see any presentation of the BFGS, there is nice page on wikipedia, algorithm which is close to what is done here). Instead we compute the Hessian with finite difference (expect mode_compute=6 and mode_compute=5

which uses a gradient the outer product approach), by calling hessian.m function.
The main culprit is that the optimization routine failed in finding a (local) minimum of minus the likelihood (or posterior kernel). That's why you need to play with other optimization routine and/or the initial guesses. Another culprit, may be the noise in the objective function. In this case you have to change the length of the steps in the finite difference routine (controlled by options_.gstep).
Two additional issue are:

- the Hessian only needs to be positive definite at an interior solution. If you have a corner solution, i.e. you are at the bound of your prior parameter space, there will be a problem.
- if a parameter is not identified or if there is collinearity in the jacobian of the likelihood, the Hessian will also be non-positive definite.
  That is why a look at the mode_check plots is often revealing to see many pathological issues simply due to the finite difference approximation to the Hessian.

10. some parameters are at the prior bounds.
    Reply: some potential solution are:

    - Check your model for mistakes;
    - Check whether model and data are consistent (correct observation equations);
    - Shut off prior_trunc;
    - Change the optimization bounds;
    - Use a different mode_compute like 6 or 9;
    - Check whether the parameters estimated are identified;
    - Check prior shape (e.g. Inf density at bounds);
    - Increase the informativeness of the prior.

11. Log Data Density[Laplace approximation] is NaN. Error using chol: Matrix must be positive definite.
    Reply:

    - First of all, take a look at the mode_check plots if there is anything strange (in particular if there are any horizontal lines indicating non-identifiability);
    - Second, wrong observation equations;
    - It may also be that the computed posterior mode is not a true local maximum; in that case you may want to try other initial values, or another optimizer.

12. In historical and smoothed variables plot, two lines are different, if no measurement error?
    Reply:

    - It is up to you to decide if your model has measurement errors. If you decide to add measurement errors, then you will observe gaps between historical data and smoothed variables. The smaller are these differences, the better is the fit. Obviously this comparison does not make sense if you do not have measurement errors. It may happen that, even without measurement errors, you observe differences between historical and smoothed variables (typically different means). This may reflect a problem in the specification of your measurement equations (missing constant) or a bug in Dynare.

- You can also get a gap between two lines in case of stochastic singularity. This could particularly occur before Dynare 4.5 as the use_univariate_filters_if_singularity_is_detected option was enabled by default.

13. After estimation, the absolute value of standard deviation of observation are quite high which are much higher than the data standard deviation. However, the relative standard deviation are similar. I take log-linearization of the model by hand, so every variable in the code represents percentage deviation from steady state(which are all 0). The measurement equation for demeaned growth rate data:

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

$$c\_obs = c - c(-1) + y\_ME$$

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

$$h\_obs = h$$

The measurement equation for one-side HP filtered data(cyclical component):

$$y\_obs = y + y\_ME$$

$$c\_obs = c$$

$$i\_obs = i$$

$$h\_obs = h$$

both model second moments are much larger than corresponding data second0 moments.
Reply: Unfortunately, this is not entirely unheard of. In Born, Peter and Pfeifer(2013) "Fiscal news and macroeconomic volatility", they use the endogenous_prior option for this reason. You should try whether this helps. If yes, the actual estimation might be fine.

14. After using the option, the problem has been solved. Could you explain a little bit more about endogenous_prior, what is the mechanism to decrease the second moments and is there any disadvantage using this option? I find that after using this option, the mode_check plots are a bit different, especially for the persistence of technology shock and preference shock.
Reply: See the manual 4.5.6(page78) on the option and the reference therein. The Christiano, Traband and Walentin's endogenous prior is based on "Bayesian learning". It updates a given prior using the first two sample moments.

15. Signs of the marginal likelihood(log data density)
Reply: There is no reason why the marginal likelihood should be less than one. The log data density may be positive. The reason is that the marginal likelihood may be larger than 1 so that its log is positive(Think of a normal distribution with infinitesimally small variance. In this case the density around the mean will go to infinity. Only for disrete distributions is the marginal density always negative).

16. When I start to find mode, set mh_replc=0, sometimes no matter I use mode_compute=9, the result always show (minus) the Hessian matrix at the "mode" is not positive definite. In this case ,I will run mode_compute=6 to get a "nice" mode check plot. Then I use the mode file to run mode_compute=9 again, if Hessian matrix is positive definite, then I run MCMC, like 200,000 draws with mode_compute=9. If Hessian matrix is not positive definite, I just run

MCMC with mode_compute=6 to get final results.(9 is better than 6 to run MCMC if Hessian matrix is positive definite) Are these steps reasonable? or could you give me some advices? By the way, is mode_compute=9 always better than 4 since the former one is globally finding mode, while the later one is locally?
Reply:

- The issue in the end always is to find the global mode. What you describe is a iterative procedure to find it and it sounds reasonable. However, you should keep track of whether there is actual improvement in the posterior density across runs(and particularly the MCMC). If you are unsure where the mode is and you have rather diffuse priors that do not smooth out the likelihood by much, mode_compute=4 is not advisable, because it tends to get stuck at local modes.
- Sequentially running different mode_compute makes a lot of sense. In case you did not find the true mode, you increase the likelihood of finding it. In case you already found it, running mode_compute a second time will just cost your time, but will not do any other harm as the optimizer will stay at the same mode you found before. Using the calibrated value as starting values is recommented in case your mode_compute fails because of invalid starting values.

17. initial_estimation_check
    Reply: There have been stochastic singularity, i.e. the model implies a perfect linear combination of observables. In this case, having as many shocks as observables is not sufficient. Typical examples are observing all components of the budget constraint. You need to find out where comes from. Start by dropping one observables at a time.

18. Error using chol
    Reply: Prior is two tight. Change the prior to something more sensible or at least use prior_trunc=0.

III. Simulation

1. One of the eigenvalue is close to 0/0.
   Reply: A generalized eigenvalue of 0/0 is related to Blanchard and Kahn condition, because it means that any complex number can be a generalized eigenvalues. So we don't have a unique solution in this case (the uniqueness problem here is different from the one treated by Blanchard and Kahn condition which assume that the eigenvalues are uniquely determined and discuss the uniqueness and solutions of the paths for endogenous variables). Obviously, in practice, we need to decide what is zero. By default, Dynare issue the 0/0 error message if the denominator and numerator are less than 1e-6. Adding an option for the threshold level (called qz_zero_threshold). The option can be used in check, stoch_simul and estimation commands. An example on the gist.github.com/stepan_a.

2. Collinear equations
   Reply: Unless there are unit roots involved in your model, there should not be collinearity in the first place. It would imply that some variables cannot be determined in your model.

3. Oscillating or Alternating IRFs
   Reply:

- It means that there is a complex eigenvalue leading to oscillations. Usually, when people have this problem in the forum, it comes from them ignoring the Dynare timing convention and then torturing the timing until the BK conditions are satisfied. When doing so, the correct economic timing is ignored and oscillating IRFs results.
- Most of time, this is timing problem. Wrong timing in feedback equations(wrong equations)
- Something maybe non-convergence.
- There is anything usually in the parametrization of policy rules, i.e. a fiscal rule with too much debt feedback.(very strange parameter value)

4. What does it mean the imaginary part of the eigenvalue is not equal to zero? Is that a problem? 5 looking-forward variables, 4 eigenvalues larger than 1, then changing the timing, 4 looking-forward variables, 4 eigenvalues larger than 1, BK condition is verified. Appropriate?
   Reply:

   - It is generalized eigenvalues. This is not a problem, unless you get oscillating IRFs.
   - No, this is not appropriate. There is one unique correct timing and you cannot arbitarily shift timing.

5. MS-DSGE
   Reply: Dynare does not support Markov-Switching DSGE models yet. Dynare only supports perturbation, thus requiring that all functions are sufficiently smooth. It would not work with discrete Markov Chains.
   Confusing:

   - Markov process for an exogenous variable: essentially a discretized version a continuous distribution used as a numerical trick—Markove process for shock.
   - True Markov Switching process (Junior Maih's RISE toolbox(`https://github.com/stepan-a/rise`))

IV. Steady State

V. General Issues

1. The Jacobian contains Inf. or NaN.
   Reply:

   - Just remove the check command before the model_diagnostics command and you will obtain more information.
   - Mathematically, that is not a problem, but numerically, the computer does not know how to handle the limit where you have a variable with $(s^0)^\infty$.
   - If really need to consider the limit case, you will have to do it analytically and change the equations accordingly(i.e. change a CES like production function for a CD form). Unfortunately, Dynare cannot do this for you.

2. Positive Monetary policy shock means that Nominal Rate decrease, and expansive policy?
   Reply: In standard NK models, nominal interest rate may change in either way.
   When Positive monetary policy shock with nominal interest rate falling, The mechanism is the following: Assume the initial monetary policy shock increases the nominal and thus the real interest rate. The central bank then reacts to lower output and inflation endogenously by lowering the nominal interest rate. If the response is strong enough, it overcompensates the initial increase due to the shock. The only thing you know is that the real interest rate increases, so that the shock is still contractionary, although the nominal interest rate goes down. This is one of the reasons why you cannot look at interest rates to determine whether monetary policy is expansionary.
   Due to general equilibrium effects, the nominal interest rate can go any way. You only know that the real interest rate must increase after a positive shock to the policy rate:

$$r_real = r - pi(+1);$$

   That is actually the case. The "erratic" behavior is actually not that erratic, it simply shows there is not much persistence. If you increase the persistence of the shocks and the interest smoothing, things look more "normal". What stays is that reactions in the first period are quite differently from the rest.
   In Gali's textbook, the sign of the response depends on various parameters.

3. The BK condition and the rank condition
   Reply: TBC

4. How to solve a model with heterigenous agents?
   Reply:

   - Individual's state variable includes their wealth, like asset or capital holdings. And every agent faces a probability of dying. So their assets evolve over time. And their decision rules depend on assets, maybe in a nonlinear way.
   - Taking a look at the JEDC special issue(2010,34(1)):"Computational Suite of Models with Heterogenous Agents: Incomplete Markets and Aggregate Uncertainty". The Kim, Kollman and Kim approach is easily implementable in Dynare.

5. How is heterogeneity defined?
   Reply: It is about whether aggregation is possible or not. If you cannot easily aggregate up, you need to keep track of the distribution(typically the wealth distribution). Distribution are infinite dimensional objects (you need all moments from 1 to infinity to fully characterize a distribution). Dealing with this infinite dimensional object is what makes it so hard.
   In many RA models, there is a continuum of agents, but their consumption and thus wealth is perfectly insured via Arrow-Debreu securities. In this case, every agent will always have the same wealth at the end of the period, even if there are idiosyncratic shocks that only hit some agents. Take away complete markets and some agents will have positive shocks while other will have negative ones. In this case, they will choose different assets at the end of the period. You would need to keep track of this. If you only assume two types, this is doable. If there are infinitely many, it quickly becomes intractable.

6. Sate-dependent IRFs/GIRFs. Suppose one wants to investigate the effect of a government spending shock: the size of fiscal multiplier should vary, if the economy is in a state where

output is below or higher than its steady state value.
Reply:

- In linear model, IRFs are state independent.
- For higher order approximation, Dynare generates a generalized IRFs that is state dependent. However, the GIRFs is computed at the ergodic mean. i.e. `https://site.google.com/site/pfeiferecon/RBC_state_dependent_GIRF.mod?attredirects=0`.
- In RBC model, only the state variables matter for the dynamics, a control variable is a function of the state variables. If one set states to a particular value, controls will be consistent with this value. Only states matter.