Christiano
Assignment 9

Tutorial on Forecasting, Output Gap Estimation, DSGE Model Estimation and the MCMC Algorithm Using Dynare

As a reminder of example 5 from the model-solution handout, following are the equations of the Clarida-Gali-Gertler model.

$$\begin{aligned}
\beta E_t\pi_{t+1} + \kappa x_t - \pi_t &= 0 \text{ (Calvo pricing equation)} \\
-\left[r_t - E_t\pi_{t+1} - rr_t^*\right] + E_t x_{t+1} - x_t &= 0 \text{ (intertemporal equation)} \\
\alpha r_{t-1} + u_t + (1-\alpha)\phi_\pi \pi_t + (1-\alpha)\phi_x x_t - r_t &= 0 \text{ (policy rule)} \\
rr_t^* - \rho \Delta a_t - \frac{1}{1+\varphi}(1-\lambda)\tau_t &= 0 \text{ (definition of natural rate)} \\
y_t^* &= a_t - \frac{1}{1+\varphi}\tau_t \text{ (natural output)} \\
x_t &= y_t - y_t^* \text{ (output gap)}
\end{aligned}$$

The baseline model parameters are:

$$\begin{aligned}
\beta &= 0.99,\ \phi_x = 0.15,\ \phi_\pi = 1.5,\ \alpha = 0.8,\ \rho = 0.9,\ \lambda = 0.5,\ \delta = 0.2, \\
\varphi &= 1,\ \theta = 0.75,\ \sigma_a = \sigma_\tau = \sigma_u = 0.02.
\end{aligned}$$

You will need the Dynare files, ccgsim.mod and ccgest.mod, as well as the MATLAB m files, plots.m, analyzegap.m, suptitle.m, compareMCMCLaplace.m and HPFAST.m, to do this assignment.

The HP filter is defined as follows:

$$\min_{\{y_t^T\}_{t=1}} \sum_{t=1}^{T} \left(y_t - y_t^T\right)^2 + \lambda \sum_{t=2}^{T-1} \left[\left(y_{t+1}^T - y_t^T\right) - \left(y_t^T - y_{t-1}^T\right)\right]^2$$

The parameter, λ, controls how 'smooth' y_t^T is. If $\lambda = 0$, then $y_t = y_t^T$. If $\lambda = \infty$, then y_t^T is a time trend (i.e., a line whose second derivative is zero).

In the analysis of business cycle data, it is customary to set $\lambda = 1600$. The MATLAB m-file, [y_hp,y_hptrend]=HPFAST(y,lambda) takes y as input and puts out y_hp=$y_t - y_t^T$, y_hptrend=y_t^T.

This assignment accomplishes three things: (i) it evaluates the common practice of estimating the output gap using the HP filter; (ii) it asks the student to estimate the parameters of a DSGE model by Bayesian methods; and (iii) it explores the MCMC algorithm as a device for approximating a posterior distribution.

1. For our first exercise, we explore the MCMC algorithm and the Laplace approximation in a simple example. We ask two questions: (i) what value of k generates the best approximation with the MCMC algorithm, with the least number of replications (the conventional recommendation is that k be selected to obtain an acceptance rate of roughly 20 percent); and (ii) how well does the Laplace approximation work. Technical details about the MCMC algorithm and the Laplace approximation are discussed in the lecture notes on the econometrics of DSGE models.

   Hopefully, it is apparent that the MCMC algorithm is quite simple, and can be programmed by anyone with just a small amount of experience with MATLAB. A useful exercise to understand how the algorithm works, is to see how well it approximates a simple known function. Thus, consider the Weibull probability distribution function (pdf),

$$g(\theta) = ba^{-b}\theta^{b-1}e^{-\left(\frac{\theta}{a}\right)^2},\ x \geq 0,$$

   where a, b are parameters. (For an explanation of this pdf, see the MATLAB documentation of $[g] = wblpdf(\theta, a, b)$.) Consider $a = 10$, $b = 20$. Graph this pdf over the grid, $[7, 11.5]$, with intervals 0.001 (i.e., graph g on the vertical axis, where $g = wblpdf(x, 10, 20)$, and x on the horizontal axis, where $x = 7 : .001 : 11.5$).

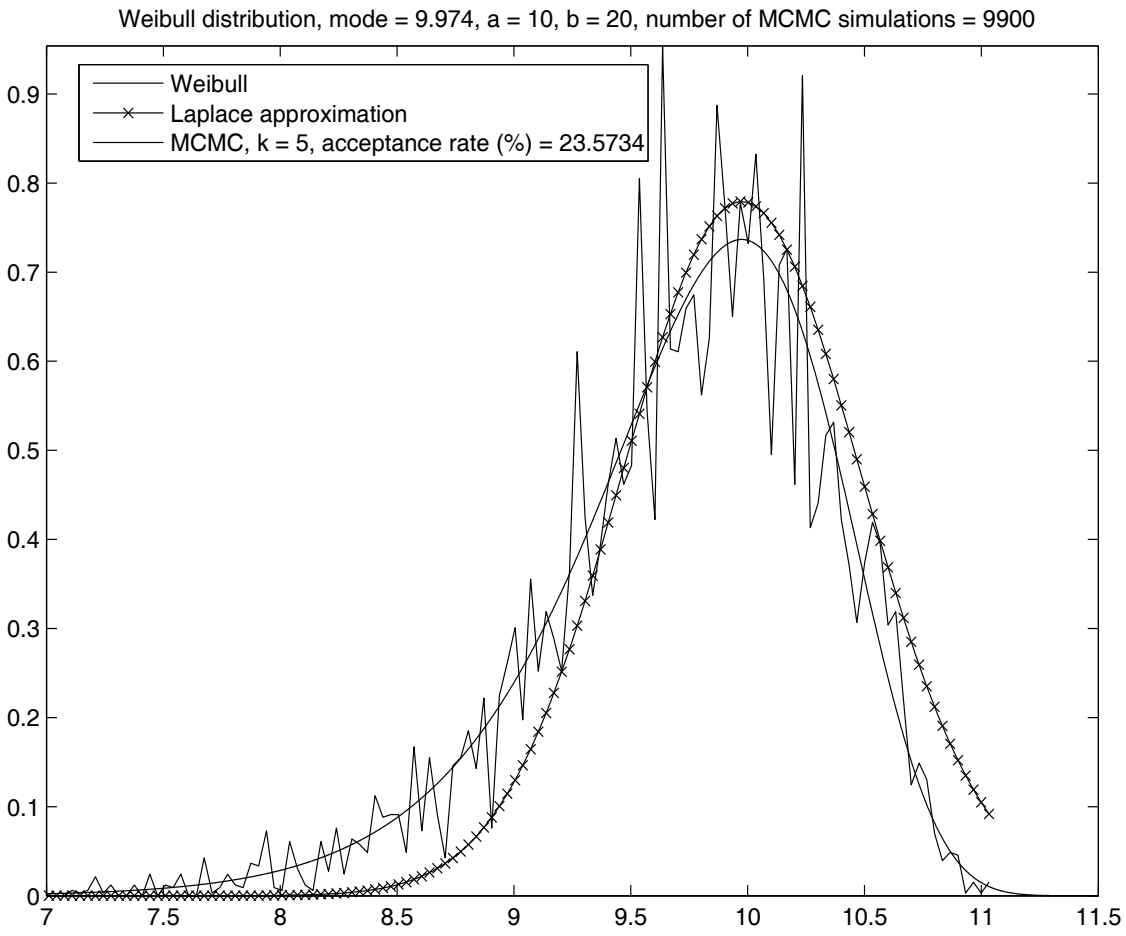
   (a) Compute the mode of this pdf by finding the element in your grid with the highest value of g. Compute the second derivative of the Weibull at the mode point numerically, using the formula,

$$f''(x) = \frac{f(x + 2\varepsilon) - 2f(x) + f(x - 2\varepsilon)}{4\varepsilon^2},$$

for ε small (for example, you could set $\varepsilon = 0.000001$.) Here, x plays the role of θ^* and f plays the role of the MATLAB function, *wblpdf*. Set $V = -f''(\theta^*)^{-1}$.

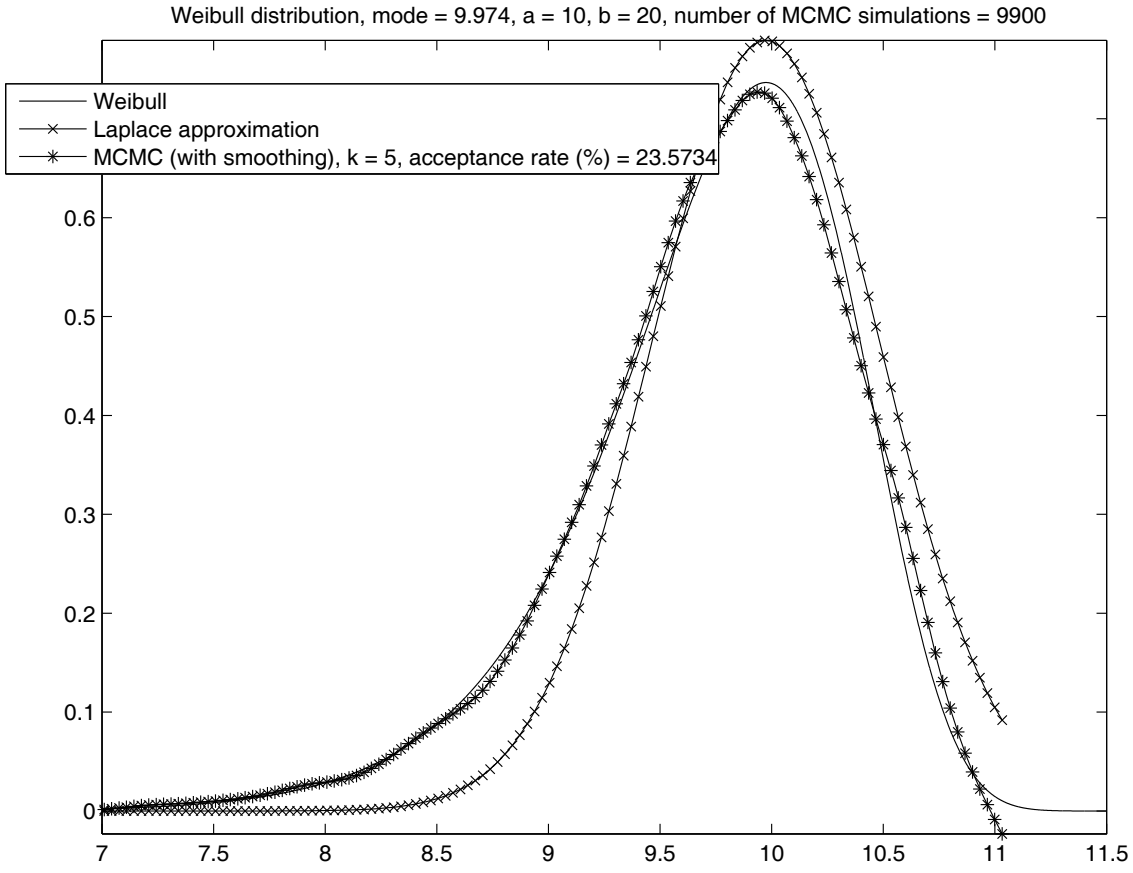
(b) Set $M = 1,000$ and set $k = 1$. Graph the density function implied by the MCMC approximation, the actual density function, and the one implied by the Laplace approximation. The value of k, $k = 1$, produces a large acceptance rate (I obtained an acceptance rate of 69%). In any case, the MCMC algorithm produces a poor approximation with such a low value of M.

(c) Consider $M = 10,000$. The acceptance rate (69 percent) is quite a bit higher than what is recommended. To get the acceptance rate down, consider a higher value of k, say $k = 5$. In this case, I obtained an acceptance rate near the recommended one of 20 percent. The results are as follows:



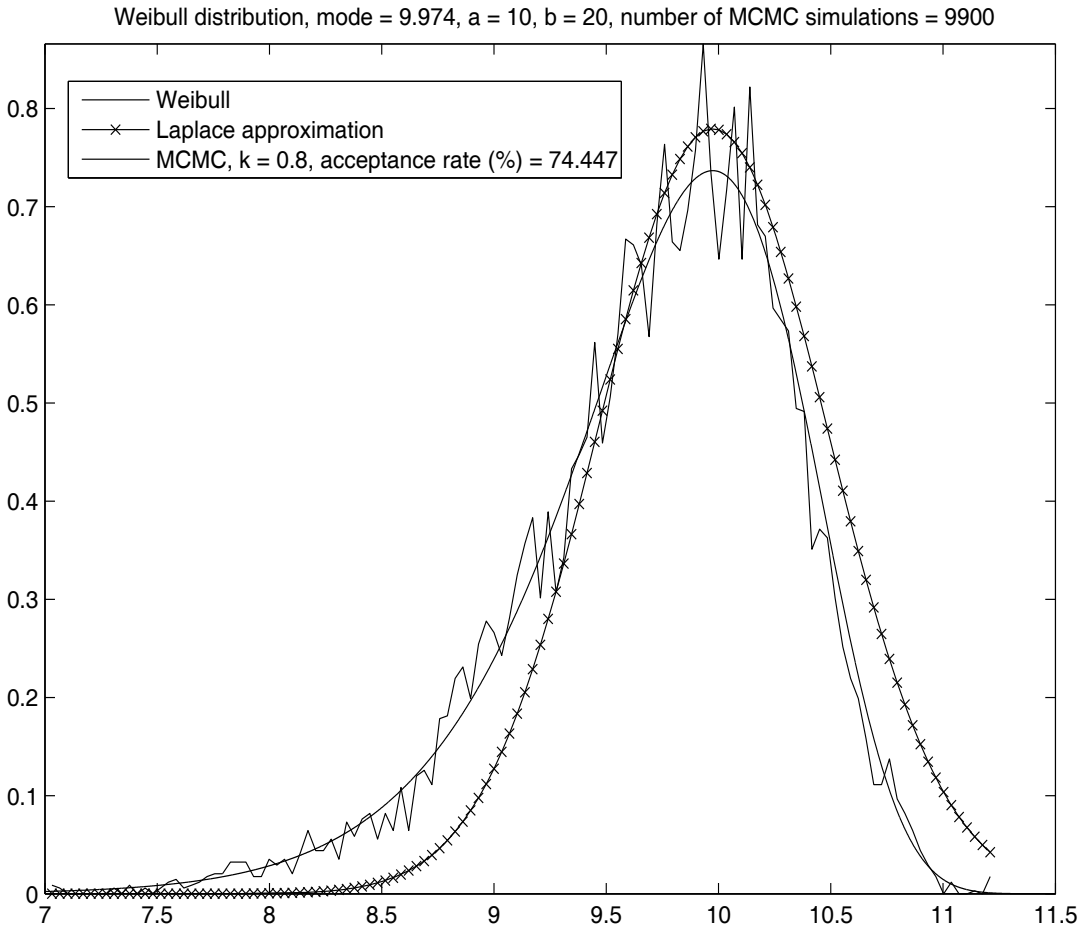
Note how 'choppy' this distribution looks. With a larger M the MCMC approximation would be smoother. Alternatively, one could smooth the approximation by filtering the results. For example, when the results are filtered with the HP filter with a smoothing parameter of $\lambda = 100$, I obtained the following result[1]:

[1]Roughly the same results were obtained using the MATLAB kernel smoothing routine, ksdensity, using its default settings. That is, letting $\bar{\theta} = \begin{bmatrix} \theta(1) & \cdots & \theta(M) \end{bmatrix}$ denote the draws from the MCMC algorithm, I computed [f,xi]=kdensity($\tilde{\theta}$) and plotted the result using plot(xi,f).

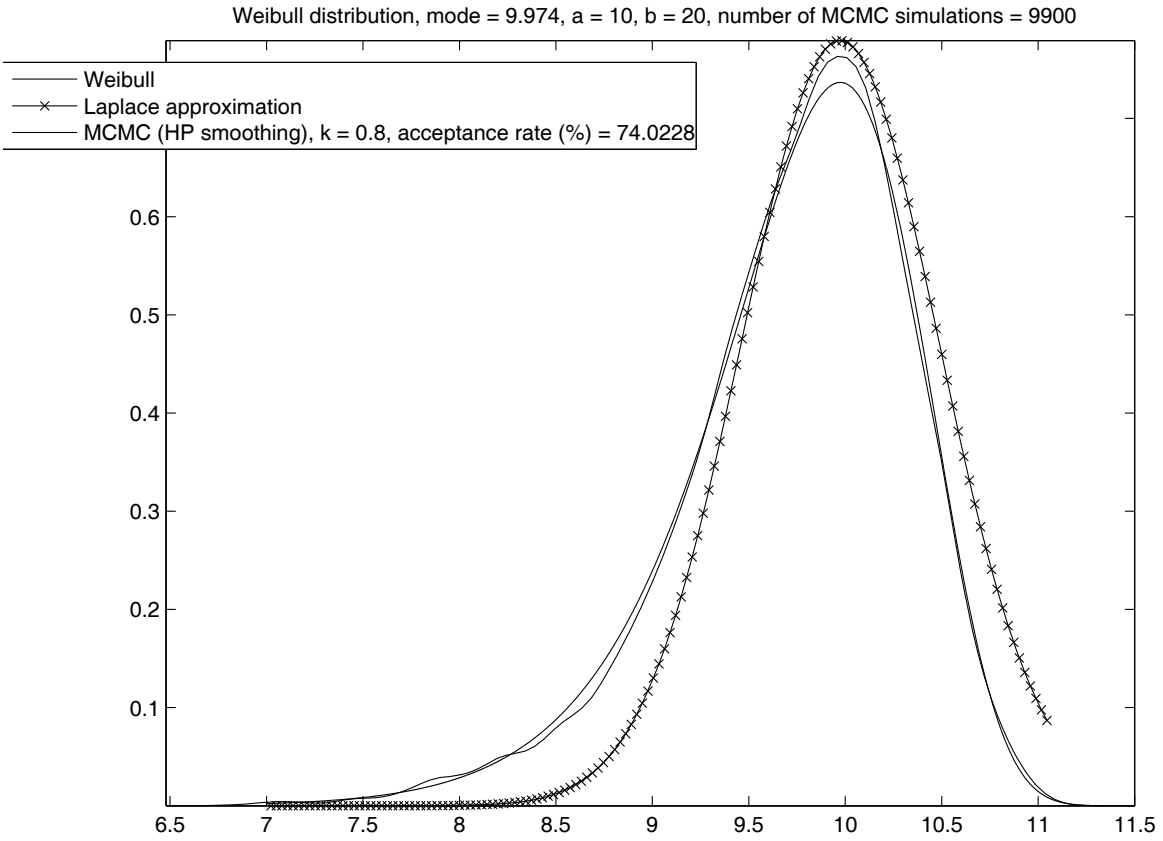


Evidently, the smoothed MCMC approximation works reasonably well. It works better than the Laplace approximation, because the Weibull distribution we are working with is slightly asymmetric, and the Normal distribution can never approximate such a distribution well.

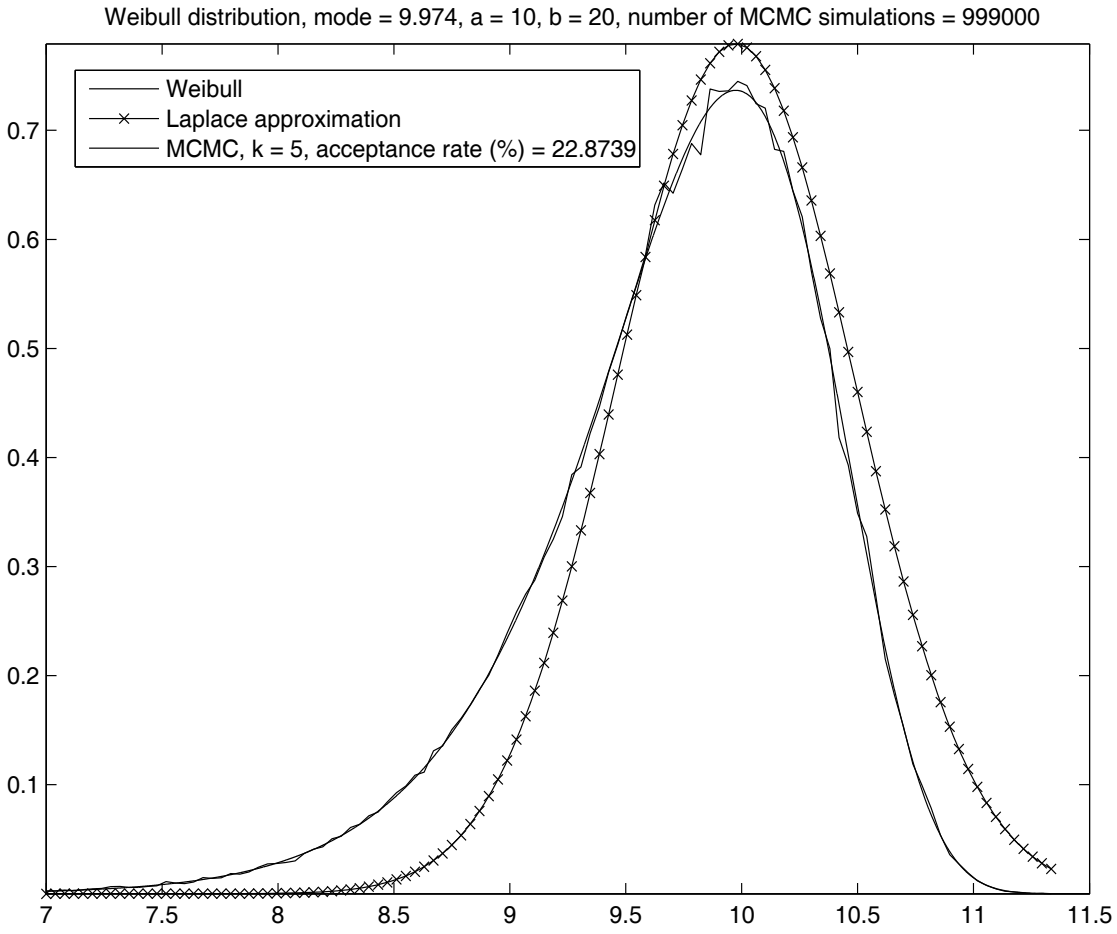
(d) To understand better the implications of the example for the optimal choice of k, consider now a higher acceptance rate. To this end, set $k = 0.8$. In this case, I obtained an acceptance rate of 74 percent:



Interestingly, the unsmoothed results seem to lie closer to the true distribution when the acceptance rate is high, because they are less noisy. However, when the results are smoothed, it appears that the results are more accurate when the acceptance rate is low. We can see this by comparing the following figure with the smoothed results obtained with $k = 0.8$:



Note how the peak of the smoothed MCMC rises well above the peak of the Weibull distribution. These results appear to support the conventional recommendation regarding k. In any case, the precise choice of k does not matter when M is very large. For example, when I set $M = 1,000,000$ with $k = 5$, I obtained the following result:



Our findings for the quality of the Laplace approximation are that it assigns too much density near the mode, and lacks the skewness of the Weibull. Still, for practical purposes the Laplace may be workable, at least as a first approximation in the initial stages of a research project. This could be verified in the early stages of the project by doing a run using the MCMC algorithm and comparing the results with those of the Laplace approximation. It's perhaps clear that the MCMC algorithm is very time intensive because M must be set high. So, if a workable alternative is available during the development stages of a project, this is useful.

2. For the remainder of this assignment, it will be useful to have a sense of how the CGG economy responds to a shock. In the parameterization above, we have specified that there are no monetary policy shocks, and the standard deviation of the other shocks is 2 percent, each.

   (a) In the case of each shock, use Dynare to compute the impulse response functions of the variables to each shock. The m file, plots.m, produces these in a format that resembles the one in the model-solving lecture notes. You may want to use these. In particular, note whether the economy over- or under- responds to the shock compared to what natural output does. What is the economic intuition in each case?

   (b) Do the calculations with $\phi_\pi = 0.99$. What sort of message does Dynare generate, and can you provide the economic intuition for it?

   (c) Return to the parameterization, $\phi_\pi = 1.5$. Now, insert r_t into the Cavlo pricing equation. Redo the calculations and note how Dynare reports indeterminacy. Provide economic intuition for your result.

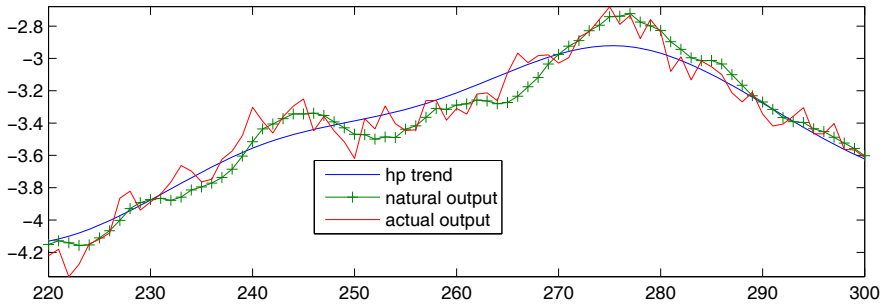
3. Generate $T = 5000$ artificial observations on the 'endogenous' (in the sense of Dynare) variables of the model. These are the variables in the 'var' list. The mod file provided, cggsim.mod, has 7 variables. Before doing the simulation, you should add the growth rate of output to the equations of the model and to the var list (call it 'dy'.) That way, Dynare will also simulate output growth. The variables simulated by Dynare are placed in the $n \times T$ matrix, $y\_$. The n rows of $y\_$ correspond to the $n = 8$ variables in var, *listed in alphabetical order* from the first to the last row. In particular, the order of the variables will not be the same as the order in which you listed them in the var statement, if you didn't enter them in alphabetical order. To verify the order that Dynare puts the variables in, see how they are ordered in lgy_ in the Dynare-created file, cggsim.m.

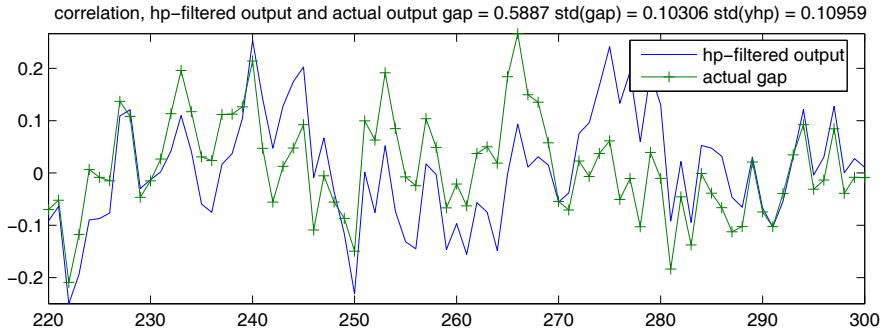
   Retrieve output growth from (the second row of) $y\_$ and get the log level of output, y, using $y =$cumsum$(dysim)$, where $dysim$ is the name I arbitrarily assigned to the second row of $y\_$. Also, retrieve x from

(the eighth row of) $y\_$ and create natural output from the relation, $y^* = y - x$.

(a) Compute the HP filter of y with $\lambda = 1$ and display a graph with y and y^T. Do this also for $\lambda = 1600$ and for λ =160,000,000. Do the results accord with what you would expect, given the formula for the HP filter above?

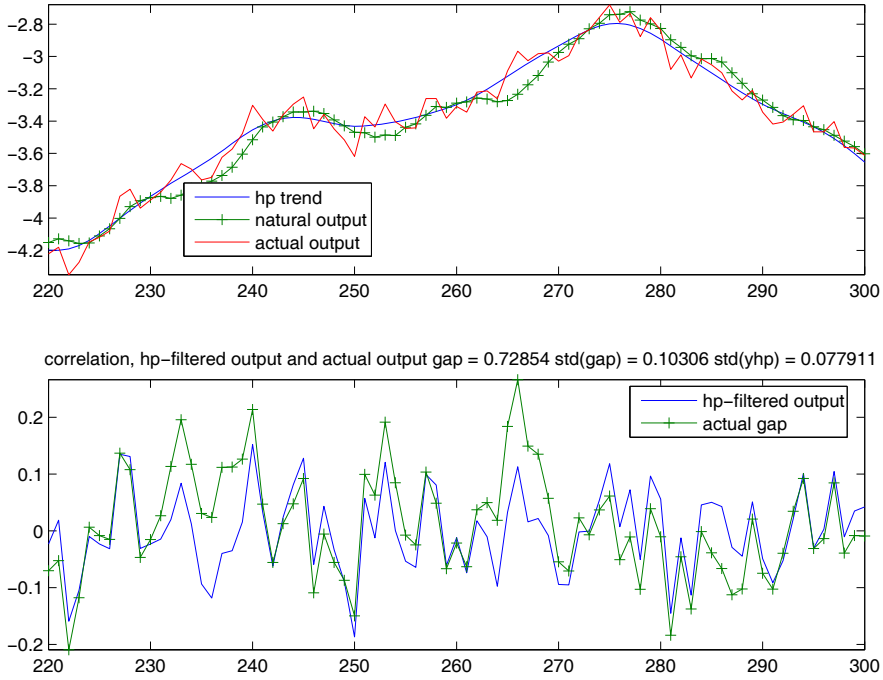
(b) Graph the HP filter trend, y^T, ($\lambda = 1600$) along with y and y^*. Note how natural output is somewhat smoother than the hp trend. That is, the hp trend with $\lambda = 1600$ oversmooths the data. Still, the two are relatively close and the two implied gaps have a correlation of 0.59. Following is a graph of the results:





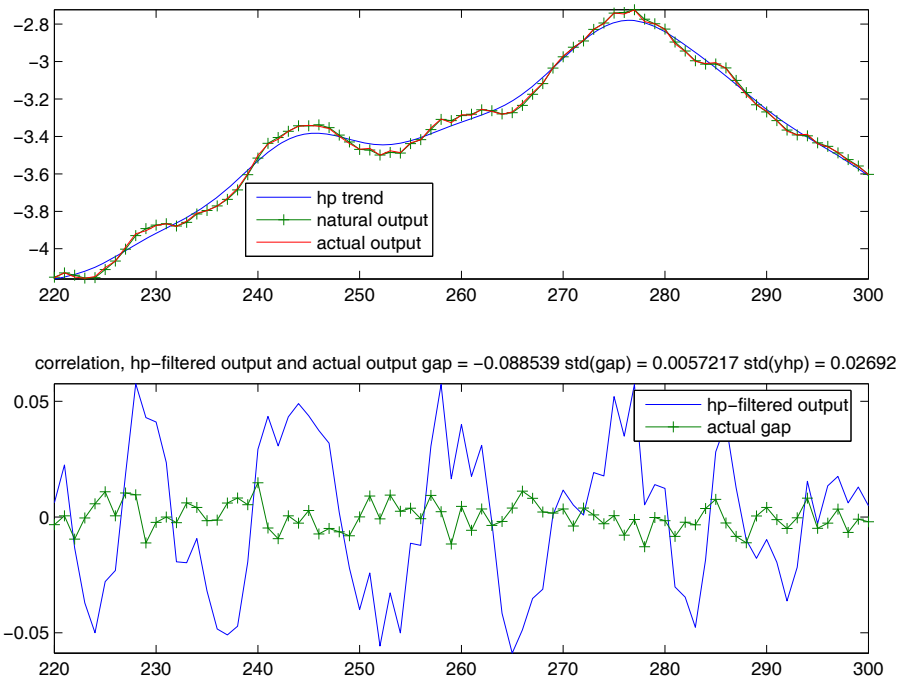
Not surprisingly, when λ is reduced, the two gap measures correspond more closely. For example, with $\lambda = 50$, one obtains the

following results:



Note how the correlation between the two gap measures has jumped from 0.59 to 0.73.

The reason that the hp filter works well as a gap measure is that the nature of the economic inefficiency in the economy is such that it over-reacts to shocks. This is evident in the impulse response functions computed in question 2 above. If ϕ_π is increased and there are no monetary policy shocks, then the economy is more efficient, and the gap is smaller. For example, when $\phi_\pi = 50$ and $\sigma_u = 0$, then one obtains the following results:



Evidently, whether the hp filter approximates the gap well depends on parameters. Since the HP filter is often used to measure gaps, it would be useful to conduct the above exercise generating data from models that fit the data well.

4. Now we will do some estimation. First, we generate artificial data from the baseline parameterization of the model, except that σ_u should be set to zero (it would be good to verify that the model parameters are set at the correct values). Place the MATLAB instruction, save data y_ at the end of cggsim.mod. Also, set periods = 5000 in the stoch_simul command. Run the mod file using Dynare. This saves the simulated data. Second, open cggest.mod.

   (a) First, do maximum likelihood estimation. Use 4,000 observations to verify that everything is working properly. Consistency of maximum likelihood implies that with this many observations, the probability that the estimates are far from the true parameter values is low. Try doing the estimation when you start far from the

true parameter value, say with rho=lambda=0.2, $\sigma_a = \sigma_\tau = .1$. Despite the bad initial guess about the parameter values, you should end up roughly at the true values.

(b) Redo (a), but now with 30 observations, and you should see that maximum likelihood still works well.

5. Now do Bayesian estimation, using the inverted gamma distribution as the prior on the two standard deviations and the beta distribution as the prior on the two autocorrelations. The beta distribution is nice because it is restricted to the zero-one interval, while the inverted gamma is convenient because it is restricted to be positive. Dynare automatically displays graphs of the prior distributions, so that you can visualize their shape.

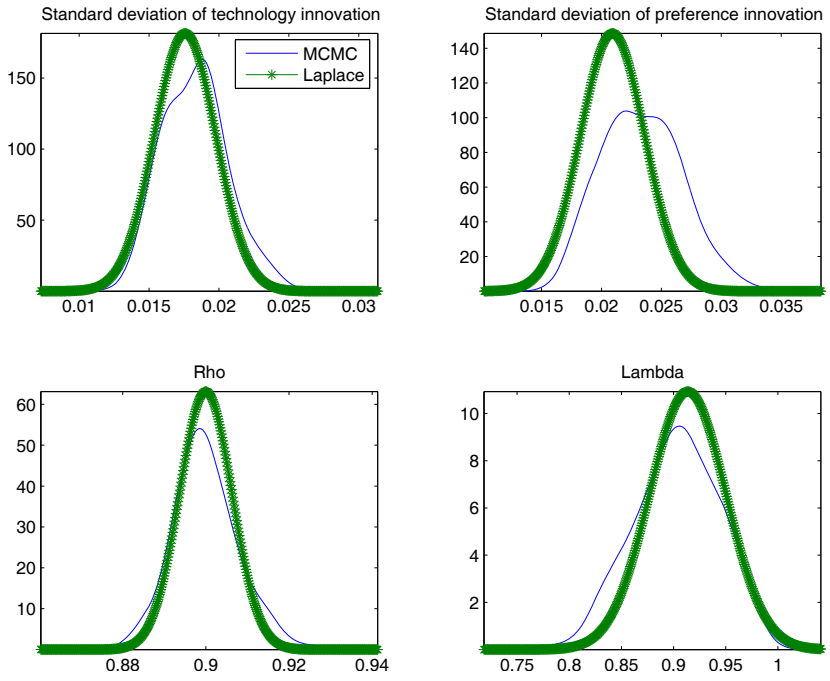
   (a) Set the mean of the priors over the parameters to the corresponding true values. Set the standard deviation of the inverted gamma and the beta to 10 and 0.04, respectively. Use 30 observations in the estimation. Adjust the value of k, so that you get a reasonable acceptance rate. I found I had to set $k = 1.2$. Have a look at the posteriors, and notice how they are tighter than the priors. An exception is λ, where the posterior is roughly the same as the prior. This suggests that the data contain very little information about this parameter. Set the number of replications in the MCMC algorith to 1,000 and compare the results with what you get with 10,000.

   (b) Now set the mean of the priors on the standard deviations to 0.1, far from the truth. Set the prior standard deviation on the inverted gamma distributions to 1. Keep the observations at 30, and see how the posteriors compare with the priors.

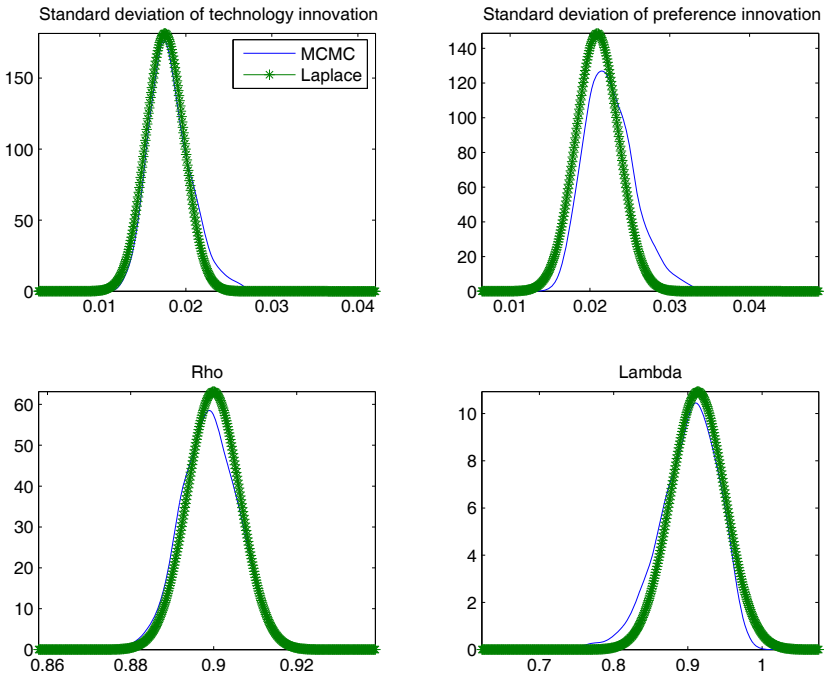
   (c) Repeat (b) with 4,000 observations. Compare the priors and posteriors. Note how the posteriors become spikes, as consistency of the maximum likelihood estimator implies (actually, there is again relatively little information about λ).

6. It is of interest to compare the posterior densities approximated by the MCMC algorithm with the Laplace approximation. Consider the

setup in 5 (a). You can recover all the information you need for these calculations from the structure, oo_. The posterior distributions of the parameters and shock standard errors are in the structure oo_.posterior_density (first column represents various values of the parameter, and second column is the MCMC estimate of the associated density). Posterior modes are in oo_.posterior_mode. Posterior standard deviations (taken from the relevant diagonal parts of the inverse of the hessian of the log criterion) appear in oo_.posterior_std (my code for recovering these objects is compareMCMCLaplace.m.) Setting $M = 1,000$, I found



Note that the Laplace and MCMC posteriors differ substantially, especially for σ_τ. I then increased the number of replications to $M = 10,000$, I found



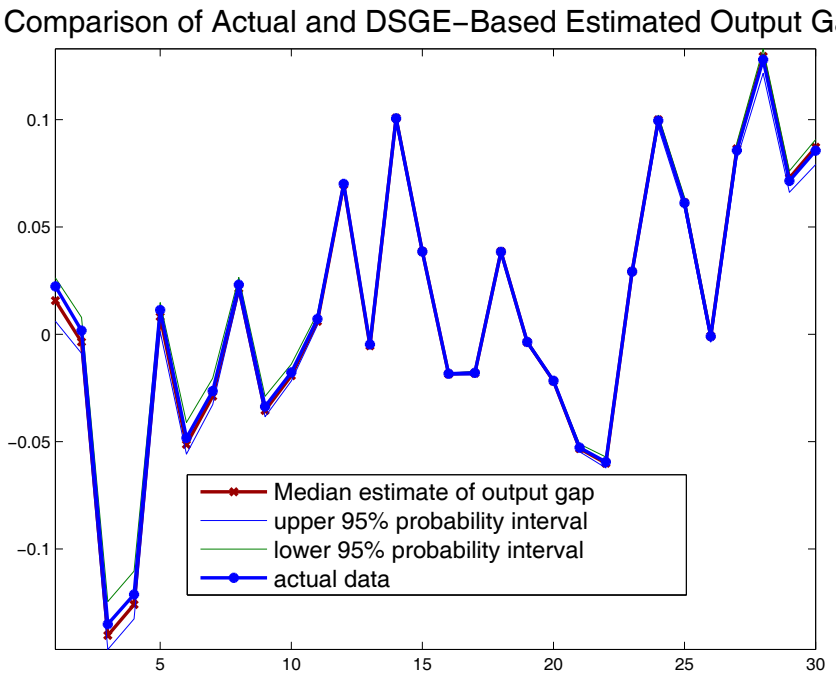
Note how much more similar the MCMC and Laplace posteriors are. The Laplace and MCMC approximations deliver very similar results, consistent with the conclusion that the Laplace approach can used in the middle of a research project, while the MCMC can be done later on. Note that in any particular project, you can check this proposition out by periodically comparing the posterior distribution obtained by the Laplace approximation with the posterior distribution obtained by MCMC.

7. The output gap is not in the dataset used in the econometric estimation. However, as noted in the handout, it is possible to use the Kalman filter to estimate the output gap (actually, everything in the state) from the available data. There are two ways to do this: 'smoothing' uses the entire data set and 'filtering' only uses the part of the dataset prior to the date for which the estimate of the gap is formed. To activate the Kalman smoother in Dynare, include the argument, smoother, in the estimation argument list. The smoothed estimates will then be placed in the MATLAB structure, oo_. This structure can be accessed either

directly from the command line, or by selecting >desktop>workspace from the pull down menu available in the command window. You will see that inside oo_ there are a number of subcategories, with output related to the Bayesian estimation.

To see how well the Dynare-estimated version of the model does at producing a good guess of the output gap, include the code, analyze.m, at the end of your mod file. This shows you how to recover the smoothed output gap from Dynare, and allows you to compare it with the actual output gap, as well as with the hp-filtered estimate of the output gap.

I obtained this result with the estimated model:



The estimate of the gap, and the actual output gap are hard to distinguish in the above graph and this is because they are very close! In addition, Dynare provides a confidence interval for the gap. Note how the confidence interval widens at the beginning and the end of the data

set. This is because there is less information there about the gap. For example, at the beginning of the data set you only have the later data to use in estimating the gap, while the earlier data are not available. (I believe that the confidence integral integrates both parameter uncertainty as well as the uncertainty you would have even if you knew the true values of the parameters.)

8. The analysis in the previous question suggests that the output gap can be estimated reliably using the estimated dsge model. However, in practice one needs the output gap in real time. For this, the smoothed estimates of the output gap are not a reliable indicator. Instead, it is useful to look at the filtered estimates. These are found by running analyze.m to line 49 (the code shows how these are recovered from oo_). Note that there is a systematic phase shift between the estimated and actual gaps. This is as expected. Turning points are hard to 'see' in real time. They become evident only after the fact.

9. It is interesting to see how the HP filter works in real time. By running analyze.m down to line 65 one obtains an estimate of this.

10. Dynare will also do forecasting. For this, one includes the argument, forecast=xx, where xx indicates how many periods in the future you want to forecast. (Put in xx=12.) To obtain the forecasts, as well as forecast uncertainty, execute the rest of analyze.m. You can see from the analyze.m code where in oo_ the forecasts as well as the forecast uncertainty is stored.