

Economics 416.
 Advanced Macroeconomics
 Christiano
 Homework #5, due May 14.

1. The MCMC algorithm and the Laplace approximation. Technical details about both these objects are discussed in the lecture notes. One practical consideration not mentioned in the notes is relevant for the case in which the pdf of interest is of a non-negative random variable. Since the jump distribution is Normal, a negative x could be drawn. A ‘quick and dirty’ approach in this case is to work with the absolute value of x .

Consider the Weibull probability distribution function (pdf),

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

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 $x = 7 : .001 : 11.5$). 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 $V = -f''(\theta^*)^{-1}$.

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 , $k = 1$, seems to be a bit large. A value of 0.8 seems to work better. Note how rough the MCMC approximation is. In part, this is due to the fact that we have used a relatively small value for M . Try $M = 10,000$ and then $M = 100,000$ and $k = 0.80$.

You should find that the Laplace approximation does not closely resemble the true distribution, in part because the true distribution is not symmetric while the Laplace is symmetric.

Try out different values of k to see which produces the best results. What is the rejection rate in the jump distribution associated with the k that produces the best results? Are the results consistent with the practical advice offered for Bayesian analysis.

2. Next, use Dynare to solve the linearized CGG model that you have worked with in a previous assignment. The Dynare code is available on the course website. The code is fairly easy to use. You provide Dynare with instructions in a file with extension `.mod`. Dynare instructions include a `var` statement, which lists all the time series variables, a `varexo` statement, which lists the innovations to exogenous shocks (the shocks themselves are included in the `var` statement) and a `parameter` statement, which includes all the variables that are constants. It also includes the command, `model;` followed by the model equations, followed by the command, `end;`. After the model, there is a set of commands, beginning with `shocks;` and ending with `end;`. In between, the shock innovation variances are set. Following that, there is a command, `stoch_simul`, with some parameters. This simulates data. The code for this question is provided in `cgg.mod`. To run a dynare program, type `dynare cgg` at the MATLAB command prompt. Instructions about Dynare generally and about `stoch_simul` specifically, are provided in the manual which is included as a pdf file with the code that is attached to this homework.

Following are the equations of the Clarida-Gali-Gertler model.

$$\beta E_t \pi_{t+1} + \kappa x_t - \pi_t = 0 \text{ (Calvo pricing equation)}$$

$$- [r_t - E_t \pi_{t+1} - rr_t^*] + E_t x_{t+1} - x_t = 0 \text{ (intertemporal equation)}$$

$$\alpha r_{t-1} + u_t + (1 - \alpha) \phi_\pi \pi_t + (1 - \alpha) \phi_x x_t - r_t = 0 \text{ (policy rule)}$$

$$rr_t^* - \rho \Delta a_t - \frac{1}{1 + \varphi} (1 - \lambda) \tau_t = 0 \text{ (definition of natural rate)}$$

$$y_t^* = a_t - \frac{1}{1 + \varphi} \tau_t \text{ (natural output)}$$

$$x_t = y_t - y_t^* \text{ (output gap)}$$

The baseline model parameters are:

$$\begin{aligned} \beta &= 0.97, \phi_x = 0., \phi_\pi = 1.5, \alpha = 0, \rho = 0.2, \lambda = 0.5, \delta = 0.2, \\ \varphi &= 1, \theta = 0.75, \sigma_a = \sigma_\tau = 0.02, \sigma_u = 0. \end{aligned}$$

You will need the Dynare files, ccgsim.mod and ccgest.mod, as well as the MATLAB m files, plots.m, analyzegap.m, supitle.m and HP-FAST.m, to do this assignment. 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) 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.
 - (d) How high does ϕ_π have to be before the responses in the equilibrium with the Taylor rule resemble the responses in the natural equilibrium, to a technology shock and to a preference shock?
3. Generate $T = 200$ 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 $y = \text{cumsum}(dysim)$, 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 $\lambda = 160,000,000$. 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 actual output is somewhat more volatile than potential or natural output. As a result, the HP filter with $\lambda = 1600$ grossly over smooths the data. You can see this by graphing $y_t - y_t^T$ and the true output gap, x_t , as well as y , y^T and y^* . Compute the

correlation between $y_t - y_t^T$ and x_t . Also compute the correlation for the case where technology shocks are dominant (i.e., $\sigma_a = 2$, $\sigma_\tau = 0.02$) and for the case where preference shocks are dominant (i.e., $\sigma_a = 0.02$, $\sigma_\tau = 2$). Interpret the results. The MATLAB command for computing the correlation between two variables, w_t and u_t , is `corrcoef(w,u)`. The result of this calculation is a 2×2 matrix with unity on the diagonal and the correlation on the off-diagonal.

Often researchers use the hp-filter to estimate the ‘output gap’. Note that in this model, the hp-filtered data are a very unreliable guide about the output gap, x_t . Of course, it is possible that if this exercise had been done in a version of the model (i.e., with capital and other complications) that allow the model to fit the data well, then the hp-filter would have worked well. This is certainly worth further investigation.

4. Now we will do some estimation. First, we generate artificial data from the baseline parameterization of the model. Place the MATLAB instruction, `save data y_` at the end of `cggsim.mod` and verify that the model parameters are set at their baseline values. 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=0.9`. 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 uniform distribution as the prior on the two autocorrelations.

- (a) Set the mean of the priors over the parameters to the corresponding true values. Set the standard deviation of the inverted gamma to 10. 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. Try $M = 4,000$.
 - (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.
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`. 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`). Graph the results for the MCMC algorithm with $M = 10,000$ together with the results based on the Laplace approximation. Do they look very close? Do the same for $M = 100,000$. How do MCMC and Laplace compare now? Next, redo the charts with $M = 1,000,000$ and obtained virtually the same result as with $M = 100,000$. Have your results converged? Does it seem like the MCMC results based on $M = 100,000$ are adequate? What do you conclude about the reliability of the Laplace approximation as a guide to the nature of the posterior distribution?