

## APPENDIX: INFERENCE FOR THE HAAVELMO MODEL

CHRISTOPHER A. SIMS

The model was estimated using annual, chain-indexed, real GDP component data for the US, 1929-40. The code names of the variables in the Federal Reserve Bank of St. Louis data base are PCECCA (C), GPDICA (I), and GCECA (G). The likelihood was formed conditional on the initial observation, so the sample was 1930-1940. With the model rewritten as

$$\Gamma_0 y_t = c + \Gamma_1 y_{t-1} + \zeta_t, \quad (1)$$

the log likelihood is

$$T \log(\Gamma_0) - .5 \sum_{i=1}^3 \frac{\zeta_{it}^2}{2\sigma_i^2}, \quad (2)$$

where of course we hold  $y_t$  and  $y_{t-1}$  fixed and substitute  $\Gamma_0 y_t - \Gamma_1 Y_{t-1} - c$  for  $\zeta_t$  in constructing the likelihood. To reduce the system from the original four equations to three, we use the accounting identity to eliminate  $Y_t$ . This form allows us to integrate the  $\sigma_i^2$  terms out of the likelihood analytically. As a function of each of the three  $\sigma_i^2$  terms, the likelihood is proportional to an inverse-gamma distribution. Integrating them out, treating the prior as flat in  $1/\sigma_i^2$  for each  $i$ , leaves us with, as log posterior density under a flat prior,

$$T \log |G_0| - (\frac{1}{2}T + 1) \sum_i \left( \log(\frac{1}{2} \sum_t \zeta_{it}^2) \right) - \frac{T}{2} \log(2\pi) + 2 \log \gamma(\frac{1}{2}(T + 1)).$$

What is reported in the text as the maximum likelihood estimate is actually the peak of this posterior with the  $\sigma_i^2$  terms integrated out. It was calculated using the `csmnwel.R` software available on my website ([sims.princeton.edu/yftp/optimize](http://sims.princeton.edu/yftp/optimize)) and the `hvlh.R` program listed below for likelihood evaluation. Note that in searching for the peak of the

---

*Date:* January 2, 2012.

This work [partially] supported by the Center for Science of Information (CSoI), an NSF Science and Technology Center, under grant agreement CCF-0939370. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF)©2012 by Christopher A. Sims. This document is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License. <http://creativecommons.org/licenses/by-nc-sa/3.0/>.

likelihood, the bounds on parameters were imposed by use of a penalty function. Convergence was rapid.

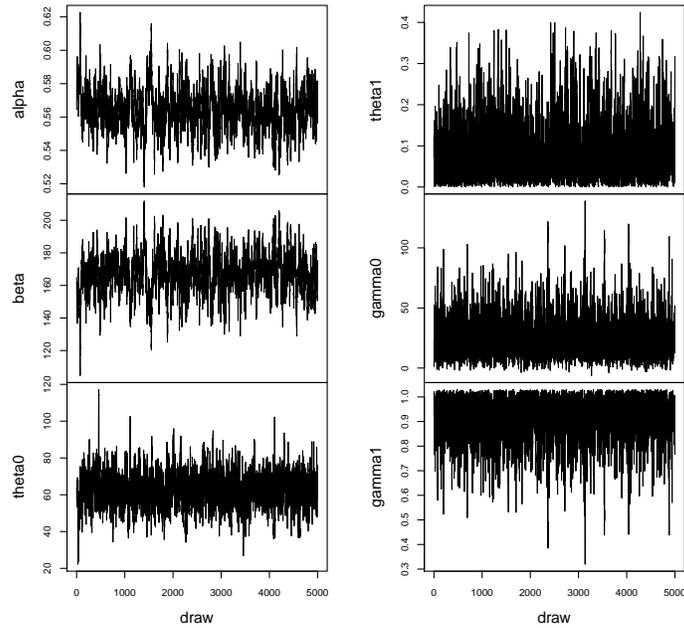
Sampling from the posterior was carried out with a simple Metropolis algorithm. The bounds on the parameters were enforced by simply rejecting draws that violated the constraints. (Of course, in a Metropolis algorithm, this entails repeating the previous draw in the Monte Carlo sample.) Convergence for this was not so rapid. Initially, a normal jump distribution with covariance matrix .09 times the approximate inverse Hessian at the likelihood peak was used. This produced extremely slow convergence for the  $\theta_0$  and  $\theta_1$  parameters, because the Hessian implied standard deviations for them that were much too small. After several runs of 50000 draws (each run took about 30 seconds), a new covariance matrix estimate was formed from the last of these runs, and the Metropolis algorithm was restarted with this matrix replacing the one based on the Hessian. The estimates seem reasonably converged on the basis of 100,000 draws generated this way. It should be noted that, compared to what emerged from the first 50,000 draws using the Hessian, there is no qualitative change. The estimated distribution for  $\theta_1$  extends to somewhat higher values, as does the distribution of the multiplier, but the shapes remain roughly the same. The other parameters are largely unaffected.

The effective sample sizes, parameter by parameter, as calculated by the `effectiveSize` function in the R `coda` package, are

| $\alpha$ | $\beta$ | $\theta_0$ | $\theta_1$ | $\gamma_0$ | $\gamma_1$ |
|----------|---------|------------|------------|------------|------------|
| 298.2    | 285.1   | 798.3      | 1212.1     | 1532.3     | 1515.1     |

These values are small relatively to the total of 100,000 draws, but they suggest the estimates are reasonably well converged. Note that the  $\theta_i$ 's, which caused the initial problems, now look well-behaved, while  $\alpha$  and  $\beta$ , which in the initial runs had larger effective sample sizes, now look the least well converged. Trace plots appear below.

Trace plots, 100,000 draws thinned by factor of 20



## LIKELIHOOD CODE

```

hvlh <- function(par, data) {
  alph <- par[1]
  bet <- par[2]
  thet0 <- par[3]
  thet1 <- par[4]
  gam0 <- par[5]
  gam1 <- par[6]
  T <- dim(hvdata)[1]
  ## c i g, but data are i, g, c, so need to permute
  y <- apply(data, 1, sum)
  G0 <- matrix(c(1-alph, -alph, -alph, -thet1, 1, 0, 0, 0, 1), 3, byrow=TRUE)
  G1 <- matrix(0,3,3)
  G1[2,1] <- -thet1
  G1[3,3] <- gam1
  iperm <- c(2,3,1)
  G0 <- G0[ , iperm]
  G1 <- G1[ , iperm]
  const <- c(bet, thet0, gam0)
  u <- data[-1, ] %*% t(G0) - data[-T, ] %*% t(G1) - t(matrix(const, nrow=3, ncol=T-1))
  ss <- apply(u^2, 2, sum)
  ## Note: this is log of flat-prior posterior with 1/sigsq terms integrated out (though not normalized).
  llh <- (T-1) * sum(log(svd(G0)$d)) - ((T-1)/2 + 1) * sum(log(ss/2)) - (T-1) * log(2 * pi)/2 + 2 * lgamma((T+1)/2)
  ## penalize negative multiplier or negative thetal
  ## llh <- llh - 1000*min(0,1 - alph * (1 + thet1)^2 - 1000 * min(0, thet1)^2
  ## browser()
  ## for MCMC, just zero out negative multiplier or thetal < 0
  attr(llh, "bad") <- alph *(1 + thet1) > 1 || thet1 < 0 || gam1 > 1.03
  return(-llh)
}

```

DEPARTMENT OF ECONOMICS, PRINCETON UNIVERSITY

*E-mail address:* sims@princeton.edu