Hessian Based MCMC for Linear DSGE Models

JOB MARKET PAPER (DRAFT)

Ed Herbst[*]

October 22, 2010

Abstract

We propose a block Metropolis-Hastings algorithm for DSGE models wherein directed proposals are generated via a single-step of Newton's method. Additionally, we suggest a way to select blocks of parameters based on the local curvature of the posterior. First, despite the additional computational burden, the block Metropolis-Hastings algorithm can be more efficient; it can more quickly (in a wall time sense) produce effectively independent samples. Second, I also show that the block Metropolis-Hastings algorithm is less likely to get stuck on local modes or difficult points. That is, the block Metropolis-Hastings algorithm can succeed where the Random Walk Metropolis is known to fail. Specifically, I show that this is the case for a basic RBC model. Third, I find that constructing blocks based on the curvature of the posterior tends to produce uncorrelated blocks, which is the most statistically efficient way to construct a block sampler. Fourth, I derive an analytic expression for the exact Hessian of the log-likelihood of a linear DSGE model. This improves both numeric accuracy and computational speed in estimating DSGE models.This improves both numeric accuracy and computational speed in estimating DSGE models.

[*] Department of Economics, University of Pennsylvania; 160 McNeil Building, 3718 Locust Walk, Philadelphia, PA 19104; *herbstep@sas.upenn.edu*

1 Introduction

Dynamic Stochastic General Equilibrium (DSGE) models are an increasingly important tool for academic economists and policymakers. These optimization-based models have an internal coherence that allows one to evaluate policy changes and propagation mechanisms and, more recently, have served as the basis for forecasting models. Bayesian techniques, in which the likelihood function implied by the model is combined with a prior distribution for model parameters to yield a posterior distribution for the model parameters, are often used to estimate these models. A survey of eight top journals in economics from 2005 to 2010 indicates that about 70% of all DSGE model estimation is done using Bayesian methods.[1]

While the technical details of the estimation of DSGE models are left to a later section, the key problem in Bayesian inference is accessing the posterior distribution. For DSGE models, the posterior is a highly nonlinear and analytically intractable distribution. Markov Chain Monte Carlo (MCMC) techniques allow one to simulate a sequence of draws whose distribution closely resembles the posterior distribution. The approximation becomes better as the sequence becomes longer. A simulation technique known as the Metropolis-Hastings algorithm can be used to construct this sequence. A known "proposal" distribution is specified; candidates are draws from the proposal and accepted into the sequence with a probability chosen to ensure convergence. Different proposal distributions yield different varieties of Metropolis-Hastings.

For DSGE models, the Metropolis-Hastings method used in roughly 95% of papers[1] using Bayesian estimation is the Random Walk Metropolis-Hastings algorithm, first used by Schorfheide (2000). The proposal for next draw in the sequence is a random perturbation about the current state in the sequence. The RWMH is simple to implement, but there are two related problems associated with using the RWMH for large and complex distributions. First, the RWMH can be quite slow to converge to the posterior distribution; the draws are very highly correlated. This means that estimates constructed using the draws have high variance and that many draws are needed for effectively independent samples. Second, the RWMH can get stuck in local modes and fail to explore the entire posterior distribution. In this case the output from the simulator can completely fail at replicating true features of the distribution. In this case, inference (e.g., variance decompositions, impulse responses, forecasts, et cetera) based on the output from the RWMH is incorrect.

To alleviate these problems, I propose a block Metropolis-Hastings algorithm

[1] Authors calculations.

for DSGE models wherein directed proposals are generated via a single-step of Newton's method. I use Newton's method to push the proposal into regions with higher posterior density. Additionally, I suggest a way to select blocks of parameters based on the local curvature of the posterior. In this way, information about the local slope and curvature of posterior is embedded into the proposal, thus allowing for a proposal distribution that more closely resembles the posterior locally. I compare this algorithm to existing algorithms using a several statistical models, a basic RBC model, and the Smets-Wouters model. I find that: First, despite the additional computational burden, the block Metropolis-Hastings algorithm can be more efficient; it can more quickly (in a wall time sense) produce effectively independent samples. Second, I also show that the block Metropolis-Hastings algorithm is less likely to get stuck on local modes or difficult points. That is, the block Metropolis-Hastings algorithm can succeed where the Random Walk Metropolis is known to fail. Specifically, I show that this is the case for a basic RBC model and a "news" shock model. Third, I find that constructing blocks based on the curvature of the posterior tends to produce uncorrelated blocks, which is the most statistically efficient way to construct a block sampler. Fourth, I derive an analytic expression for the exact Hessian of the log-likelihood of a linear DSGE model. This improves both numeric accuracy and computational speed in estimating DSGE models.

2 Background on Bayesian Estimation of Linear DSGE Models

A linearized DSGE model can be written as linear rational expectations system,

$$\Gamma_0(\theta)x_t = \Gamma_1(\theta)E_t[x_{t+1}] + \Gamma_2(\theta)x_{t-1} + \Gamma_3(\theta)\varepsilon_t \tag{1}$$

where x_t are the model variables, ε_t the exogenous shocks, θ the structural parameters of interest, and $\{\Gamma_i\}$ matrix functions that map the equilibrium conditions of the model. The solution to the system (under some bounds on the growth rate of x_t) takes the form of an AR(1),

$$x_t = T(\theta)x_{t-1} + R(\theta)\varepsilon_t.$$

The mappings from θ to T and R must be solved numerically for all models of interest. The model variables x_t are linked to observed y_t via a state space system:[2]

$$
\begin{aligned}
x_t &= T(\theta)x_{t-1} + R(\theta)\varepsilon_t, && (2)\\
y_t &= D(\theta) + Z(\theta)x_t. && (3)
\end{aligned}
$$

DSGE models are typically estimated using Bayesian methods.[3] Bayesian methods combine prior beliefs about a vector of parameters, θ, with data, $Y^T = \{y_t\}_{t=1}^T$, whose relationship with θ is embodied in a likelihood function, $p(Y^T|\theta)$. In DSGE models, the likelihood can be constructed using the state space system in (2-3). If the exogenous shocks are assumed to be independently and identically normally distributed, then the Kalman filter can be used to integrate X^T out of the likelihood induced by (2-3), delivering $p(Y^T|\theta)$. The prior and likelihood are combined using Bayes' rule to obtain the posterior distribution of θ,

$$p(\theta|Y^T) = \frac{p(Y^T|\theta)p(\theta)}{p(Y^T)}.$$

The posterior of the parameters is then used to construct the posterior distribution (or moments thereof) of objects of interest to the economist, such as impulse response functions, variance decompositions, and predictive distributions.

The key problem for the econometrician is constructing $p(\theta|Y^T)$. Since the mapping from θ to T and R cannot be known in closed form, there is no hope of deriving an analytic expression for $p(\theta|Y^T)$. However, since we are often interested in expectations of the form,

$$E_{\theta|Y}[h(\theta)] = \int h(\theta)p(\theta|Y)d\theta,$$

random draws from the posterior are sufficient for approximating this integral via Monte Carlo integration. If $\{\theta_i\}_{i=1}^N$ are independent and identically distributed draws from the posterior distribution, then

$$\hat{E}_{\theta|Y} = \frac{1}{N}\sum_1^N h(\theta_i) \longrightarrow E_{\theta|Y}[h(\theta)]$$

as $N \longrightarrow \infty$ by the Strong Law of Large Numbers. Unfortunately, generating independent draws from the posterior, as in necessary in Monte Carlo integration, is very difficult when the size of θ is large and little is known *a priori* about the posterior. To overcome this problem, the posterior is simulated using Markov Chain Monte Carlo (MCMC) techniques. MCMC methods construct Markovian sequences $\{\theta^i\}_{i=1}^N$ that converge in distribution to the posterior distribution as N

[2] We have omitted measurement errors for simplicity.

[3] For a spirited defense of the Bayesian approach to macroeconometrics, see Fernández-Villaverde (2009).

becomes large. The convergence of the above approximation is justified not by the SLLN but by the ergodic theorem. MCMC turns the standard problem in Markov Chain theory on its head: the invariant distribution is known; what is needed is the transition kernel for a Markov chain to get to the invariant distribution. MCMC methods construct this kernel.

The most popular MCMC technique is known as the Metropolis-Hastings algorithm, which works as follows:

Metropolis-Hastings Algorithm

1. Given θ_i, simulate θ^* from $q(\cdot|\theta_i)$.
2. With probability $\alpha = max\{\frac{p(\theta^*|Y)q(\theta^i|\theta^*)}{q(\theta^*|\theta^i)p(\theta^i|Y)}, 1\}$ set $\theta^{i+1} = \theta^*$, else $\theta^{i+1} = \theta^i$.

q is the density of some distribution that is easy to simulate from. It is known as the proposal or instrument. Under some weak conditions on q and the support of θ, draws generated by the Metropolis-Hastings will converge to draws from the posterior distribution (see Tierney (1994)). Of course, the exact form of Metropolis-Hastings depends on the exact specification of $q(\cdot|\theta_i)$. It is advantageous to have a q which approximates the posterior well, at least locally. The desired q is the one for which the chain converges quickest. Unfortunately, for the class of simulators used with DSGE models, there are few results concerning the convergence speed (geometric or uniform ergodicity), and those results that exist often depend crucially on unknown constants. Therefore, to assess convergence, statisticians often resort to looking at empirical measures of convergence, such as the autocorrelation of the chain, cross behavior of multiple chains, or differences in statistics based on multiple blocks of a single chain. By these measures, there is reason to be cautious about the standard implementations of Metropolis-Hastings.

The random walk Metropolis-Hastings (RWMH) algorithm for DSGE models was first proposed by Schorfheide (2000) and reviewed by An and Schorfheide (2007). The proposal here is given by,

$$q(\theta^*|\theta_i) \sim \theta_i + \epsilon, \quad \epsilon \sim \mathcal{N}(0, -cH^{-1}),$$

where H is Hessian of the log posterior evaluated at the mode and c is a scaling factor. This is due to the fact that under regularity conditions, asymptotically (for the number of observations) the posterior will be normally distributed about the posterior mode with a variance given by $-H^{-1}$. The random walk Metropolis-Hastings is ideally suited for posteriors about which there is little information. Since $q(\theta^*|\theta_i) = q(\theta^*|\theta_i)$, only the posteriors enter the Metropolis-Hastings probability, so q influences moves only through generating the proposals. In this way, it represents a "hill climbing" algorithm or simulated annealing. Indeed, the RWMH has served economists well in estimating early, small-scale DSGE models. However, as the complexity of DSGE models has increased, the RWMH is straining under the task of simulating from large-dimensional parameter vectors. The scaling factor c is crucial. If c is too large, the algorithm will tend to reject too much, while if c is too small, exploration of the posterior will take too long. Figure 1 illustrates this phenomenon. As the dimensionality of the parameter vector increases, the chance that any random perturbation of the current value of the chain moves into a higher-density region necessarily falls. A small scale is needed and the chain moves slowly. Figure 2 shows the sample autocorrelations of draws from four parameters of the Smets and Wouters (2005) model. The autocorrelation of h, the habit persistence parameter, is about 0.8 for a displacement of 100 draws. This indicates extremely high dependence of the chain; movement around the parameter space is quite slow. The RWMH simply takes too long to explore large-dimensional parameter spaces, as pointed out formally in Neal (2003). Moreover, slow convergence associated with small scaling factors also makes chains more likely to get trapped near a mode, as in An and Schorfheide (2007).

It is clear that the simulator have trouble converging as the dimensionality of the parameter vector increases. One solution to this problem is to break the parameter vectors and iterate them over the conditional densities, the so-called "Block Metropolis" algorithm. Breaking the parameter vector up makes it easier to construct good proposals q, so that the simulators other than the random walk become feasible. The question that naturally arises is how the blocks of parameters should be constructed.

There are no theoretical results on the optimal blocking structure in MCMC, except for in a few special cases detailed in Roberts and Sahu (1997). The upshot, though, is that the blocks should be "as independent as possible," according to Robert and Casella (2004). The intuition for this rule is clear: if A and B are independent, then sampling $p(A|B)$ and $p(B|A)$ iteratively will produce draws from $p(A, B)$, since $p(A|B) = p(A)$ and $p(B|A) = p(B)$. On the other hand, if A and B are perfectly correlated, then sampling $p(A|B)$ amounts to solving a deterministic function for a in b. The subsequent draw from $P(B|A)$ will amount to solving for b in a via the inverse of the original, that is, b will be the same value as before and the chain will not move throughout the parameter space.

Unfortunately, given the complexity of DSGE models, particularly in the solution of linear rational expectations system, it is not possible to know the correlation structure of the parameters *a priori*. Indeed, with an irregular distribution the relationship among parameters may vary throughout the parameter space, so that a fixed block structure is not optimal. This paper uses information contained in the curvature of the likelihood to select blocks in estimation. Specifically, we look at the second derivatives collected in the Hessian of the posterior as a measure of the *relatedness* of the parameters. If an element of the hessian $H_{ij}(\theta) = \partial^2 p(\theta|Y)/\partial\theta_i\partial\theta_j$ is "large" – in a sense to be formalized later in the paper – then we conclude that θ_i and θ_j are related and should be blocked together. We repeat this blocking procedure frequently throughout the simulation to account for the fact that the correlation structure may be changing throughout the parameter space, due to the irregular nature of the posterior.

This paper's contribution is construct a block Metropolis-Hastings algorithm that improves upon standard simulation practices in two ways. First, we provide a heuristic for blocking the parameters depending on the local curvature of the posterior as discussed above. Second, we develop proposal distributions based on Newton's method, increasing significantly the likelihood that a proposal is generated from a high-density region of the posterior. The resulting algorithm is intuitive and easy to implement, and it explores the posterior efficiently.

The final contribution of this paper is an expression exact Hessian for the posterior of a DSGE model. The derivation is an extension of Iskrev (2008), who shows how to construct an implicit function between the reduced-form matrices of the state transition equation and the structural matrices of the canonical linear rational expectations system. The exact Hessian is a faster and more accurate estimation than the numerical Hessian economists typically use, and it may be of independent interest in non-MCMC settings.

3 Literature Review

Recently, econometricians have paid more attention to posterior simulation of DSGE models. Curdia and Reis (2009) and Chib and Ramamurthy (2010) have advocated block estimation. Curdia and Reis (2009) groups the parameters by type: economic vs. exogeneous. Chib and Ramamurthy (2010), to which this work is most similar, propose grouping parameters randomly. We provide evidence that the Hessian-based blocking procedure advocated here outperforms both of these procedures in terms of statistical efficiency. Chib and Ramamurthy (2010) advocates generating proposals by estimating the conditional medium for each block using simulated annealing, which is very costly in terms of computational time. A novel and unrelated approach to posterior simulation comes from Kohn et al. (2010). Their procedure is an adaptive Metropolis-Hastings in which the proposal is a mixture of an random walk proposal, an independence proposal, and a t-copula estimated from previous draws of the chain. This is a promising approach to estimation, but it requires one to set many tuning parameters, which may be daunting to the applied macroeconomist. Additionally, since the initial proposal is heavily weighted towards a random walk, the algorithm can suffer from the same problem of getting trapped in local modes like the regular random walk Metropolis-Hastings experiences. The details of Chib and Ramamurthy (2010) and Kohn et al. (2010) are discussed briefly in the Appendix.

Using Metropolis-Hastings to simulate the posterior distribution of a statistical model goes back to Tierney (1994) and the references therein. Chib and Greenberg (2005) provide an excellent overview the Algorithm. Block Metropolis-Hastings is not a new technique in posterior simulation. Besag et al. (1995) use the procedure to successfully simulate from several complex conditionals in a "Metropolis-within-Gibbs" procedure. Roberts and Sahu (1997) examine optimal block structure for simulating from a multivariate normal random variable via Gibbs sampling. Their work contains some of the few available theorems regarding blocking. Recently, a strand of literature in adaptive MCMC has pursued blocking based on principal components of estimates of the variance matrix (see, for example, Andrieu and Thoms (2008)). We do not pursue that approach here because it has little applicability to large-parameter vectors.

The idea of using information contained in the slope and curvature of the likelihood is not new. Metropolis-Hastings algorithms based on Langevin diffusions use the gradient of the likelihood to inform proposals (Roberts and Rosenthal (1998)). MCMC inspired by Hamilton dynamics (see, for example, Neal (2010)) also use the gradient to generate proposals.

The proposals considered in this paper are most closely related to Qi and Minka (2002), Geweke (1999), and Geweke and Tanizaki (2003). Qi and Minka (2002) advocate using Newton's method to generate proposal densities, but not in a block setting, while Geweke and Tanizaki (2003) construct proposals based on Taylor approximations for a univariate simulation. Tjelmeland and Eidsvik (2004) and Liu et al. (2000) stress how local optimization steps in proposals can ameliorate difficulties in estimating multimodal distributions. Using "tailored" proposals generally dates to Chib and Greenberg (1994).

4 Exact Hessian

For large DSGE models, evaluating the Hessian for the log posterior of a DSGE model numerically is costly in terms of computational time.[4] Indeed, this is one reason that the single-block random walk algorithm proposed by Schorfheide (2000) has remained so popular among economists. In that algorithm, the Hessian need only be computed once before the chain can be simulated. In an MH algorithm with multiple blocks, however, it becomes necessary to compute the Hessian for each conditional density (i.e., at every step). Furthermore, when one constructs blocks based on the Hessian, one must calculate it again.

To avoid some of the computation costs of repeatedly evaluating the likelihood while computing the Hessian, we construct the exact Hessian of the log likelihood of the DSGE model by extending the results of Iskrev (2008, 2010), who shows how to compute the derivatives,

$$\frac{\partial vec(T(\theta))}{\partial \theta'} \text{ and } \frac{\partial vec(R(\theta))}{\partial \theta'}$$

by establishing an implicit function between $\{vec(T), vec(R)\}$ and $\{vec(\Gamma_0), vec(\Gamma_1), vec(\Gamma_2), vec(\Gamma_3)\}$. This function is constructed by substituting (2) in (1) and taking expectations.

We extend his result to calculate the Hessians,

$$\frac{\partial}{\partial \theta'} vec\left(\frac{\partial vec(T(\theta))}{\partial \theta'}\right) \text{ and } \frac{\partial}{\partial \theta'} vec\left(\frac{\partial vec(R(\theta))}{\partial \theta'}\right). \tag{4}$$

A detailed derivation is given in the Appendix. The end user must supply the hessians of $\{vec(\Gamma_0), vec(\Gamma_1), vec(\Gamma_2), vec(\Gamma_3)\}$., but these are easy to compute via symbolic differentiation. Most software packages economists use can perform the task. Combining (4) with the Hessians of $Q(\theta), D(\theta)$, and, $Z(\theta)$, which are simple to calculate for DSGE models, one can use standard techniques to calculate the Hessian of the log likelihood based on the state space representation of the DSGE model. To reduce the number of matrix operations, we show how to derive the gradient and hessian of the likelihood based on the Chandrasekhar recursions (Morf (1974)). A detailed derivation is left to the Appendix.

This approach has advantages in speed and accuracy. For example, for a 12 parameter DSGE with 2 observables for 80 periods, numeric evaluation using a two-sided, finite difference method takes about 4 seconds, while the analytic evaluation takes about 0.3 second. In a model in which the analytic derivatives and Hessian are available (i.e., a simple RBC with full depreciation of capital), the error from the exact Hessian is of the order 1E-15, or close to machine zero, while the error from the numeric Hessian is of the order 1E-2.

[4] For an alternative approach to differentiating of the likelihood function, see Bastani and Guerrieri (2008).

It should be noted that the implementation of the exact Hessian involves a number of calculations involving kronecker products of potentially large matrices. For very large systems, these calculations can take some time, particularly because the kronecker product is not implemented efficiently in most programming languages. To overcome this, I have implemented my own interface for the kronecker product. While, I achieve substantial gains in speed through this implementation, for very large systems the calculations can still be cumbersome; in this case it may be preferable to use a numeric hessian either through the exact derivative or through standard numeric techniques.

5 The Algorithm

Here we present the block Metropolis-Hastings algorithm. There are two departures from the standard algorithms: the construction of proposals and the blocking of the parameters. First we will explain the proposals for generic blocks of parameters, then we will discuss how to select the blocks.

Proposal Distributions. Suppose that the parameters are grouped into $b = 1, \ldots, B$ blocks, and let $\theta_{i,b}$ be the bth block at the ith draw in the chain. Let $\theta_{i,-b}$ be the most up-to-date draws of θ for all the other parameters. The task is to generate a proposal for $\theta_{i+1,b}$ conditional on $\theta_{i,-b}$. Let $g(\theta_{i,b}|\theta_{i,-b})$ be the gradient of the log posterior and $H(\theta_{i,b}|\theta_{i,-b})$ be the Hessian of the log posterior.[5] The idea is to center the proposal distribution in a region with high density, but not necessarily close to the posterior. We do that by picking the center using a single-step Newton search. Let $q(\cdot|\theta_{i,b}, \theta_{i,-b}) = t(\mu_b^*, \Sigma_b^*, \nu)$ where

$$\mu_b^* = \theta_{i,b} + \xi - sg(\theta_{i,b} + \xi|\theta_{i,-b})H(\theta_{i,b} + \xi|\theta_{i,b})^{-1} \quad (5)$$

$$\Sigma_b^* = -H(\mu_b^*)^{-1} \quad (6)$$

ξ is the displacement factor, a hyperparameter which controls how much optimization depends on the initial conditional. In a problem with many modes with strong basins of attraction, it might be advantageous to set this higher to ensure escaped local modes. Finally, s, the step size, is a random number independent of $\theta_{i,b}$ which is distributed uniformly on $[0, \bar{s}]$. [6] We generate a proposal from this

[5] The quadratic approximation to the posterior may be poor; we use a model trust region to control for the quality of the approximation. This means is the posterior in not locally quadratic, (i.e., the difference between a actual change and expected change in the posterior as a result of a step increment is large), we add a large λ to the diagonals of the Hessian. Also general, $-H^{-1}$ will not be positive definite, we compute the modified Cholesky in this case.

[6] Qi and Minka (2002) call this a "learning rate." For this kernel to preserve the invariant distribution

distribution; call it $\theta^\star$.

$$\theta^* \sim t(\mu_b^*, \Sigma_b^*, \nu)$$

Note that to preserve the (local) reversibility of the chain, we must find the symmetric parameters associated with a same step $\theta^\star$.

$$\begin{aligned} \mu_b^{\sim} &= \theta^\star + \xi - sg(\theta^\star + \xi|\theta_{i,-b})H(\theta^\star + \xi|\theta_{i,b})^{-1}, && (7) \\ \Sigma_b^{\sim} &= -H(\mu_b^{\sim})^{-1}. && (8) \end{aligned}$$

The updating of block b is made via a Metropolis-Hastings step, with probability,

$$\alpha = \min\left\{ \frac{p(\theta^\star|Y^T, \theta_{i,-b})q(\theta_{i,b}|\mu_b^{\sim}, \Sigma_b^{\sim}, \nu)}{p(\theta_{i,b}|Y^T, \theta_{i,-b})q(\theta^*|\mu_b^*, \Sigma_b^*, \nu)}, 1 \right\}, \qquad (9)$$

set $\theta_{i+1,b} = \theta^*$, otherwise set $\theta_{i+1,b} = \theta_{i,b}$. The Newton search pushes the mean of the proposal distribution to a high density relative to the current state of the chain. That is, this algorithm produces chains much more likely to "jump" to other modes and avoid overly attracting basins. This produces a chain that is less autocorrelated and does a more thorough job of exploring the parameter space.

It is worth mentioning a key difference between this algorithm and the algorithm proposed by Chib and Ramamurthy (2010). That algorithm also relies on a optimization step. There, simulated annealing is used to find μ_b^*. Under certain conditions, simulated annealing is guaranteed to find the global mode, so the algorithm forces $\mu_b^{\sim} = \mu_b^*$. Since this optimum is global, the proposal is independent of the current state of the chain. However, in practice, the current state of the chain, which is the initial condition in the SA algorithm, does affect μ_b^*. Given this correlation, to preserve balance of the chain, μ_b^{fd} should be found, via SA, starting from θ^* to have a valid Metropolis-Hastings algorithm. Computational experience with the RBC model used in this paper indicates that this is not merely a technical point. Using 1000 draws from random blocks around the posterior mode, $\theta_{i,b}$ was "farther" from μ_b^{fd} then μ_b^* so that roughly 90% of the time, $q(\theta_{i,b}|\mu_b^{fd}, \Sigma_b^{fd}, \nu) < q(\theta_{i,b}|\mu_b^*, \Sigma_b^*, \nu)$. The difference was large enough so that accept/reject decisions based on the two proposal parameterizations were different over 50% of the time. With a single-step Newton-based proposals, we don't have any hope of finding the global mode, but solving for the symmetric distribution parameterization ensures the balance of chain.

Selecting Blocks. The technique for selecting blocks also uses information from the curvature of the posterior. Given the general principle of grouping parameters that are correlated, we say that the parameters θ^j and θ^k are related if the Hessian,

$$H_{jk}(\theta) = \frac{\partial^2 \ln p(\theta|Y)}{\partial\theta^j\partial^k},$$

(the posterior), s must be independent of $\theta_{i,b}$. We can think of a continuum of kernels, indexed by s. For details, see Geyer (2003).

is large. Intuitively, we are saying that if the slope of the log posterior with respect to θ^j, that is, the change in probability density, is affected by movements in θ^k, then they should be grouped together. We also scale the matrix by the square root of the diagonals.

The exact algorithm used in this version of the paper is listed below.

Selecting Blocks Based on Hessians

- Given θ, compute $H(\theta)$ and create $\tilde{H}(\theta)$ so that

$$\tilde{H}(\theta)_{[jk]} = \frac{H(\theta)_{[jk]}}{\sqrt{\frac{\partial^2 \ln p(\theta)}{\partial\theta^j \partial\theta^j} \frac{\partial^2 \ln p(\theta)}{\partial\theta^k \partial\theta^k}}}.$$

- Compute all possible sets blocks of size m, where $m < n_{para}$. (If the parameter size is small enough, compute all possible blocking sizes).
- For each block scheme W with blocks w, computer

$$score(W) = \sum_{w=1}^{blocks(W)} \sum_{j=1}^{n_{para}} \sum_{k=j+1}^{n_{para}} \frac{\tilde{H}(\theta)^2_{[jk]} 1_{\texttt{j and k in block w}}}{\texttt{(\# of elements in block w)}}$$

- Assign probabilities $\alpha_w(\theta)$ to each scheme based on its score. We use a simple guide which assigns probability 50% to the top schema, 25% to the next scheme, and so on. This should be adjusted based on the number of schemes available.

One thing to note about this blocking rule is that we do not consider blocks of size one. The reason for this allowing for blocks of size one could make the cycles through the blocks take too long. However, it is possible that the last parameter chosen can be in a block alone.

The reason that we must adjust the Metropolis-Hastings proposal is because of *state dependent mixing.* It is no longer true that this mixture of the transition kernels induced by the different blocking schemes preserves the log posterior. Preservation means that the distribution of interest (here the log posterior) is a fixed point of the operator defined by the transition kernel which maps probability distributions to probability distributions. To ensure this property, we must account for the transition from blocking scheme to blocking scheme. The Appendix contains details on validity of the block sampler.

An alternative weighting scheme α_w can be constructed by running a burn in phase of the simulator. At each stage in the burn in phase, a blocking scheme is selected by the above algorithm. After the burn in phase, block schemes are selected randomly against the empirical distribution of blocking schemes chosen during the burn-in phase. This is the approach we take with large models, in which calculating the block score is costly.

Collecting the blocking scheme and proposal distributions, we have the Hessian Block Metropolis-Hastings.

Hessian Block Metropolis-Hastings

1. At the $i+1$th step in the chain, for θ_i calculate $\tilde{H}(\theta_i)^{-1}$ and select blocking schema w with probability $\alpha_w(\theta_i)$ where the probability corresponds to likelihood assigned by the blocking scheme.
2. Partition θ_i based on the scheme w selected.
3. For each block $b = 1, \ldots, B$, perform the Newton-based Metropolis-Hastings step explained above. The Metropolis-Hastings probability must be adjusted to

$$\alpha = \min \left\{ \frac{\alpha_w(\theta^*, \theta_{i,-b}) p(\theta^\star|Y^T, \theta_{i,-b}) q(\theta_{i,b}|\mu_b^{fd}, \Sigma_b^{fd}, \nu)}{\alpha_w(\theta_i) p(\theta_{i,b}|Y^T, \theta_{i,-b}) q(\theta^*|\mu_b^*, \Sigma_b^*, \nu)}, 1 \right\} \tag{10}$$

4. Move on to the next block, go back to (3). Once all blocks have been iterated over, go back to (1). Repeat for N simulations.

Theorem. Add theorem.

6 Toy Example: Multivariate Normal

To understand how the Hessian Block Metropolis-Hastings algorithm works, consider using the simulator on the following multivariate normal. In this section we set s to be deterministcally 1, $\xi = 0$, and $\nu = \infty$.

$$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{pmatrix} \sim \mathcal{N}(\mu, \Sigma)$$

$$\text{where } \mu = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \Sigma = \begin{pmatrix} 1.00 & 0.80 & 0.10 & 0.15 \\ 0.80 & 1.0 & 0.05 & 0.00 \\ 0.10 & 0.05 & 1.00 & 0.90 \\ 0.15 & 0.00 & 0.90 & 1.00 \end{pmatrix}$$

The gradient and hessian of the log pdf is given by:

$$\frac{\partial \ln(x|\mu, \Sigma)}{\partial x'} = -(x-\mu)'\Sigma^{-1} \text{ and } \frac{\partial^2 \ln(x|\mu, \Sigma)}{\partial x \partial x'} = -\Sigma^{-1}$$

Since the log density is quadratic, the Hessian is constant throughout the parameter space. For a moment.

For a single block model, the law of motion for the Markov chain is

$$\begin{aligned}
x^{t+1} &= x^t - \frac{\partial \ln(x^t|\mu,\Sigma)}{\partial x'}\left[\frac{\partial^2 \ln(x^t|\mu,\Sigma)}{\partial x \partial x'}\right]^{-1} + \frac{\partial^2 \ln(x^t|\mu,\Sigma)}{\partial x \partial x'}^{-1/2}\mathcal{N}(0, I_4) \\
&= x^t - (x^t-\mu)'\Sigma^{-1}\Sigma + \sqrt{\Sigma}\mathcal{N}(0, I_4) \\
&= \mu + \sqrt{\Sigma}\mathcal{N}(0, I_4) \\
&= \mathcal{N}(\mu, \Sigma).
\end{aligned}$$

When X is normally distributed, the simulator samples from the correct distribution. It is worth contrasting this with the proposal under the Random Walk Metropolis-Hastings, which is $\mathcal{N}(x^t, \Sigma)$. The proposals differ only in the location. The use of the Newton step has recentered the Hessian-based proposal towards the true center of the distribution.

Consider now sampling in two blocks $[[X_1, X_2], [X_3, X_4]]$. The proposal for $x_{1,2}^{t+1}$ is

$$x_{1,2}^{t+1} = x_{1,2}^t - s\frac{\partial \ln(x^t|\mu,\Sigma)}{\partial x'_{1,2}}\left[\frac{\partial^2 \ln(x^t|\mu,\Sigma)}{\partial x_{1,2}\partial x'_{1,2}}\right]^{-1} + \sqrt{\frac{\partial^2 \ln(x^t|\mu,\Sigma)}{\partial x_{1,2}\partial x'_{1,2}}}\mathcal{N}(0, I_2).$$

Applying the partitioned inverse formula, we have:

$$\begin{aligned}
\frac{\partial \ln(x^t|\mu,\Sigma)}{\partial x'_{1,2}} &= (x_{1,2}-\mu_{1,2})'(\Sigma_{1,2} - \Sigma_{1,2|3,4}\Sigma_{3,4}^{-1}\Sigma_{3,4|1,2})^{-1} \\
&- (x_{3,4}-\mu_{3,4})'(\Sigma_{3,4} - \Sigma_{3,4|1,2}\Sigma_{1,2}^{-1}\Sigma_{1,2|3,4})^{-1}\Sigma_{3,4|1,2}\Sigma_{1,2}^{-1} \\
\frac{\partial^2 \ln(x^t|\mu,\Sigma)}{\partial x_{1,2}\partial x'_{1,2}} &= (\Sigma_{1,2} - \Sigma_{1,2|3,4}\Sigma_{3,4}^{-1}\Sigma_{3,4|1,2})^{-1}.
\end{aligned}$$

This means that,

$$\begin{aligned}
x_{1,2}^{t+1} &= \mu_{1,2} + \Sigma_{1,2}^{-1}\Sigma_{3,4|1,2}(x_{3,4}-\mu_{3,4}) \\
&+ (\Sigma_{1,2} - \Sigma_{1,2|3,4}\Sigma_{3,4}^{-1}\Sigma_{3,4|1,2})^{-1/2}\mathcal{N}(0,1).
\end{aligned}$$

The simulater is drawing from the correct conditional distribution.

Blocking. Consider now the blocking strategies, by direct calculation we have:

Blocks	Score
$[X_1, X_2, X_3, X_4]$	0.65
$[X_1, X_2, X_3], [X_4]$	0.48
$[X_1, X_3, X_4], [X_2]$	0.52
$[X_1, X_2, X_4], [X_3]$	0.43
$[X_2, X_3, X_4], [X_1]$	1.05
$[X_1, X_2], [X_3, X_4]$	1.50
$[X_1, X_3], [X_2, X_4]$	0.22
$[X_1, X_4], [X_2, X_3]$	0.52

The block scheme with the highest score is $\{[X_1, X_2], [X_3, X_4]\}$.

6.1 A Large State Space Model

To assess the convergence properties and general behavior of the algorithm, we introduce a generic state space model from Chib and Ramamurthy (2010) as an example.

$$
\begin{aligned}
s_t &= Ts_{t-1} + \epsilon_t, \quad \epsilon \sim \mathcal{N}(0, Q) && (11) \\
y_t &= D + Zs_t + u_t, \quad u_t \sim \mathcal{N}(0, H) && (12)
\end{aligned}
$$

s_t a 5×1 vector of unobserved states and y_t a 10 vector of observables at time t. The system matrices are parametrized as follows. T is a diagonal matrix with

$$diag(T) = [T_1, T_2, T_3, T_4, T_5]'$$

and $Q = I_5$. Z has the following structure imposed for identification purposes:

$$
\begin{aligned}
Z_{ii} &= 1, \quad \text{For } i = 1, \ldots, 5 \\
Z_{ij} &= 0, \quad \text{For } i = 1, \ldots, 5, \quad j > i
\end{aligned}
$$

These parameters are fixed throughout the experiment. The rest of the parameters are free, with the simulation values given below.

$$
Z = \begin{bmatrix}
1.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
Z_{21,} & 1.00 & 0.00 & 0.00 & 0.00 \\
Z_{31,} & Z_{32} & 1.00 & 0.00 & 0.00 \\
Z_{41,} & Z_{42} & Z_{43} & 1.00 & 0.00 \\
Z_{51,} & Z_{52} & Z_{53} & Z_{54} & 1.00 \\
Z_{51,} & Z_{52} & Z_{53} & Z_{54} & Z_{55} \\
Z_{61,} & Z_{62} & Z_{63} & Z_{64} & Z_{65} \\
Z_{71,} & Z_{72} & Z_{73} & Z_{74} & Z_{75} \\
Z_{81,} & Z_{82} & Z_{83} & Z_{84} & Z_{85} \\
Z_{91,} & Z_{92} & Z_{93} & Z_{94} & Z_{95} \\
Z_{101,} & Z_{102} & Z_{103} & Z_{104} & Z_{105}
\end{bmatrix}
$$

H is a diagonal matrix with:

$$diag(H) = [H_{11}, H_{22}, H_{33}, H_{44}, H_{55}, H_{66}, H_{77}, H_{88}, H_{99}, H_{1010}]'$$

.

Let $\theta_1 = diag(T)$, $\theta_2 = D$, $\theta_3 = vec(lowtr(Z_{1:5,1:5}))$, $\theta_4 = vec(Z_{6:10,6:10})$, and $\theta_5 = \log(diag(H))$. The overall parameter vector $\theta = [\theta_1', \theta_2', \theta_3', \theta_4', \theta_5']'$.

We parameterize θ at the same values as Chib and Ramamurthy (2010). The value are listed in Tables 1 and 2. We estimate the model using the Hessian block Metropolis-Hastings algorithm and the Random Walk Metropolis-Hastings algorithm.

The priors for the parameters are given as follows. Let the prior for θ_1 is $\mathcal{N}(\mu_{\theta_1}, V_{\theta_1})$ where, $\mu_{\theta_1} = 0.5\mathbf{1}_{5\times 1}$ and $V_{\theta_1} = 5 \times I_5$. The prior for θ_2 is $\mathcal{N}(\mu_{\theta_2}, V_{\theta_2})$ where $\mu_{\theta_2} = 0.5 \times 1_{10,1}$ and $V_{\theta_2} = 5 \times I_{10}$. The prior for θ_3 is $\mathcal{N}(\mu_{\theta_3}, V_{\theta_3})$ where $\mu_{\theta_3} = 0$ and $V_{\theta_3} = 5 \times I_{10}$. The prior for θ_4 is $\mathcal{N}(\mu_{\theta_4}, V_{\theta_4})$ where $\mu_{\theta_4} = 0$ and $V_{\theta_4} = 5 \times I_{25}$. Finally, The prior for θ_5 is $\mathcal{N}(\mu_{\theta_5}, V_{\theta_5})$ $\mu_{\theta_5} = -1 \times 1_{10,1}$ and $V_{\theta_5} = I_{10}$. We combine these priors with the likelihood implied by state space representation of the system

We simulate 10,000 draws from the Hessian Block Metropolis-Hastings algorithm and 300,000 from the random walk Metropolis-Hastings. For the Hessian-based algorithm, we set $\bar{s} = 0.7$ and $\xi = 0$. For the random walk Metropolis-Hastings we scale the covariance matrix to ensure an acceptance rate of 35%. On the other hand, the average acceptance rate of the Hessian-based simulator is about 52%. On the whole, it appears that both simulators have converged to the same ergodic distribution. However, the draws from the RWMH are extremely correlated, whereas the draws from the HBMH are much less correlated.

Rather than show plots of the autocorrelation functions of the sixty chains we present of numeric summary of the total inefficiency of the chain sometimes known as Geweke's K. Geweke's K is a measure of the inefficiency of the Markov chain relative to iid draws. It is the theoretical variance of the chain relative to chain which is independently distributed.

$$\kappa_i \approx 1 + 2\sum_{l=1}^{\infty} corr(\theta_{i,t}, \theta_{i,t-l})$$

We approximate it by

$$\hat{\kappa}_i = 1 + 2\sum_{l=1}^{L} \left(1 - \frac{l}{L}\right) c\hat{o}rr(\theta_{i,t}, \theta_{i,t-l})$$

for some suitably chosen L. Here, we chooes $L = 5000$. Table B shows the inefficiency factor for each the the 60 parameters. It is obvious that the Hessian-based algorithm produces is much more efficient. Of course, to end-users, statistical efficiency may not be the most important consideration, given that cheapness of the RWMH in terms of computer cost. In Table B we compute the effective sample size implied by the max and the mean inefficiency factors. The RWMH produces a much smaller effictive sample despite the chain being 30 times longer. More importantly, the number of seconds per draw is longer for the random walk Metropolis-Hastings! This is despite the fact that HBMH took about 16 hours compared to the 3 of the RWMH. There is evidence, then, that the Hessian-based algorithm can be more efficient than the RWMH. Finally, the algorithm of Chib and Ramamurthy (2010) will produces draws very little correlation also, but it takes about 5 times longer.

7 A Simple RBC Model

We use the Hessian-based MH to estimate a simple RBC from Curdia and Reis (2009). It is a standard RBC model with non-separable preferences over consumption and leisure. The model to driven by a technology shock and a government spending shock, which enters the utility function directly, separable from consumption and leisure.

The consumer solves

$$\max_{\{C_t,N_t\}_{t=0}^{\infty}} \quad E_0\left[\sum_{t=0}^{\infty}\beta^t\left\{\frac{[C_t(1-N_t)^{\theta}]^{1-\gamma}-1}{1-\gamma}+V(G_t)\right\}\right] \tag{13}$$

$$\text{subject to} \tag{14}$$

$$C_t+K_t-(1-\delta)K_{t-1}+G_t \quad = \quad W_tN_t+R_tK_{t-1}.$$

The production technology is Cobb-Douglas. Firms solve the following problem,

$$\max(A_tN_t)^{1-\alpha}K_{t-1}^{\alpha}-W_tN_t-R_tK_{t-1}. \tag{15}$$

The exogenous processes for "technology" (A_t) and (G_t) "government" spending are constructed so that,

$$\hat{a}_t \quad = \quad \rho_A\hat{a}_{t-1}+\rho_{AG}g_{t-1}+\sigma_A\varepsilon_{A,t}, \tag{16}$$

$$\hat{g}_t \quad = \quad \rho_G\hat{g}_{t-1}+\rho_{GA}a_{t-1}+\sigma_G\varepsilon_{G,t}. \tag{17}$$

The VAR parameterization can be thought of as a way of estimated, say, automatic stabilizers (as in Smets & Wouters) The rest of the model solution and log linearized are given in the Appendix.

7.1 Priors & Data

[Table 6 about here.]

The priors are given in Table 6. All parameters are assumed *a priori* to be independent. The priors on α, β, and δ are tightly centered around values to match the steady state real interest rate and the Great Ratios. The priors on γ and θ are assumed to be more diffuse, reflecting greater uncertainty about reasonable values for these parameters. These parameters are crucial for the economic implications of the model. γ controls the how agents savings/investment decision is affected by shocks, while θ controls the response of hours. The priors for the exogenous processes are elicited by estimated VAR on Solow's residuals and the appropriate counterpart for G_t in the data using presample observations.

The model is estimated on quarterly observations of $\{\hat{y}_t, \hat{n}_t\}_{t=1983Q3}^{2007Q4}$. $\hat{y}_t$ detrended real US GDP, while $\hat{n}_t$ is detrended hours. Both series are detrended (individually) via the HP filter with a smoothing parameter of $\lambda = 1600$. The series are plotted (in percentages) in Figure 3.

We estimate the model using 4 different posterior simulators: the random walk Metropolis-Hastings, and three varieties of the Hessian-Based MH discussed above. In the first version we select blocks based on the Hessian, in the second we select blocks randomly (a la Chib and Ramamurthy) and in the third we fix the blocks based on parameter type (a la Curdia and Reis).

7.2 Results

The posterior means and credible sets are listed in Table 8. The first thing that should be apparent is that simulators do not agree on the central features of the distribution. In general, the credible sets for the Hessian MH are much wider than other sets. To investigate why this occurred, we plot the draws of the Hessian MH vs. the draws from the RWMH. It is clear that the Hessian MH algorithm has found a second mode. The Random Block MH likewise finds the second mode, but doesn't remix back to the first after an initial period. Table 7 lists the modes. It seems as though each of these regions explain the data in the same way. Posterior predictive checks in Figure 8 appear identical between the Hessian Metropolis-Hastings and the RW Metropolis. Likewise, the predictive densities implied for future values of output and hours are very similar. Still, though, there are important distinctions between the two modes. Mode 1 features a slightly more persistent exogenous process and a slightly higher variance for the shocks. This is a small but real example of how difficult it is to distinguish exogenous from endogenous persistence. As DSGE models increase in size this kind of "weak" identification will increase, requiring better simulators.

It appears that the Hessian Metropolis Hastings did the best job at finding both modes (although the Random Block did as well, it just didn't mix as well). It would be interesting to see how the chains compare in terms of autocorrelation. Table 9 list Geweke's K, a measure of the variance of the chain relative to theoretically iid draws. See the Appendix for the precise formula and interpretation. The Hessian-based Metropolis-Hastings performs the worst according to Geweke's K. This is because mode-hopping chains are necessarily more persistence than chains which stay on a single mode. For a posterior with a unimodal posterior (or at least, what seems like a unimodal posterior) the Hessian-based MH will perform well on Geweke's K, as we will demonstrate in the next section. Furthermore, the RWMH chain, which demonstrably does not converge, passes the split sample test and multiple chains collectively had a Gelman's R-stat of less 1.001. This lends caution to relying to heavily on convergence diagnostics as stopping rules.

The blocking method also seems to work. About 85% of the time, the block selecting mechanism choses γ and ρ_{AG} in the same block. The correlation between these parameters is about -0.70. Meanwhile, the parameters θ and ρ_{AG} are almost never blocked together, reflecting a posterior correlation of 0.02.

8 Smets and Wouters

The Smets and Wouters (2007) model is a medium-scale DSGE model which serves a benchmark for the current generation. It features 7 observables,and 41 parameters. The complete linearized model is given in the Appendix. For the Smets and Wouters model we simulate the posterior via three techniques: the Hessian block-based Newton method, a random block-based Newton methods, and simple random walk Metropolis-Hastings.

We adjust the Hessian blocking technique slightly, since calculating the full Hessian every step is computationally exhaustive, even with the exact expression of the Hessian. For the first $n_{burn} = 500$ we run the Hessian blocking method, recording the blocks at each simulation step. After this step, we construct an empirical distribution of the top $k = 10$ most selected blocks. Thereafter, at each step we randomly select the block based on this distribution, not the current state of the chain.

The priors are listed in the Tables 10 and **??**. The posteriors look very similar for all the parameters, as in Chib and Ramamurthy (2010).

To examine the speed of convergence, we rely on a empirical measure of the autocorrelation of the chain. We the inefficiency factors for each parameter in Table 12. The Hessian block and random block MH chains are significantly less correlated than the chain constructed from RWMH. On average, however, the parameters from the random block chain are slightly less autocorrelated than the Hessian blocks. However, the maximum K value is much less for the Hessian block MH (48.29) than it is for the random block MH (81.04). This might be explained by the fact that the Hessian block scheme won't ever choose a bad blocking scheme, while the random blocking scheme makes all blocks equally likely.

It's worth looking some of the groupings chosen by the Hessian blocking system.

Table **??** lists some of the more popular blocks. Parameters governing price-stickiness and mark-ups tend to blocked together,

$$[\xi_p, \rho_p, \mu_p],$$

where ξ_p is the Calvo parameter associated with prices, ρ_p is persistence parameter associated with price mark-up and μ_p is the parameter governing on the MA(1) component. However, the corresponding block associated with wage parameters,

$$[\xi_w, \rho_w, \mu_w],$$

is hardly ever chosen, because the correlation of ρ_w and μ_w is close to 0, while the correlation ρ_p and μ_p is nearly 0.8. Meanwhile, the block of $[\bar{\pi}, \bar{l}]$ is chosen 100% of the time, reflecting that the fact that the posterior correlation of these two parameters is roughly -0.60. The habit parameter and persistence to preference shocked are blocked together often, reflecting there positive correlation. The Taylor Rule coefficients ρ_π and ρ_y are always blocked together, but the coefficient on output growth $\rho_{\Delta y}$ is only included in this block about 5% of the time, reflecting the weak correlation of these parameters.

9 News Shock Model

to be added

10 Conclusion

This paper presented a new MCMC algorithm for linear DSGE models. The simulator is based on constructed tailored proposals based on a single-step application of Newton's method. We constructed blocks by examining the curvature of the posterior in order to group parameters that are related to one another. We have shown that the algorithm can be a considerable improvement over the random walk Metropolis-Hastings for linear DSGE models.

Paths for future work include refining the computational of the Hessian and gradient to take into account multithreading capabilities of many modern computing systems, better exploiting the geometry of the posterior to eliminate tuning parameters, and applying the method to new models.

References

An, Sungbae and Frank Schorfheide, "Bayesian Analysis of DSGE Models," *Econometric Reviews*, 2007, *26* (2-4), 113–172.

Andrieu, Christophe and Johannes Thoms, "A Tutorial on Adaptive MCMC," *Statistical Computing*, 2008, *18*, 343–373.

Bastani, Houtan and Luca Guerrieri, "On the Application of Automatic Differentiation to the Likelihood Function for Dynamic General Equilibrium Models," International Finance Discussion Papers 920, Board of Governors of the Federal Reserve System 2008.

Besag, Julian, Peter Green, David Higdon, and Kerrie Mengersen, "Bayesian Computation and Stochastic Systems," *Statistical Science*, Feb 1995, *10* (1), 3–41.

Chib, Siddhartha and Edward Greenberg, "Bayes Inference for Regression Models with ARMA(p,q) Errors," *Journal of Econometrics*, 1994, *64*, 183–206.

_ **and Ivan Jeliazkov**, "Marginal Likelihood From the Metropolis-Hastings Algorithm," *Journal of the American Statistical Association*, 2001, *96*, 270–281.

_ **and Srikanth Ramamurthy**, "Tailored Randomized Block MCMC Methods with Application to DSGE Models," *Journal of Econometrics*, 2010, *155* (1), 19–38.

Curdia, Vasco and Ricardo Reis, "Correlated Disturbances and U.S. Business Cycles," *Working Paper*, 2009.

Fernández-Villaverde, Jesús, "The Econometrics of DSGE Models," Working Paper 14677, NBER 2009.

Geweke, John, "Using Simulation Methods for Bayesian Econometric Models: Inference, Development, and Communication," *Econometric Reviews*, 1999, *18* (1), 1–126.

_ **and Hisashi Tanizaki**, "Note on the Sampling Distribution for the Metropolis-Hastings Algorithm," *Communications in Statisics: Theory and Methods*, 2003, *32* (4), 775–789.

Geyer, Charles J, "The Metropolis-Hastings-Green Algorithm," *mimeo*, 2003.

Green, Peter J., "Reversible Jump Markov Chain Monte Carlo and Bayesian Model Determination," *Biometrika*, 1995, *82*, 711–732.

Iskrev, Nikolay, "Evaluating the Information Matrix in Linearized DSGE Models," *Economics Letters*, 2008, *99*, 607–610.

— , "Local Identification of DSGE Models," *Journal of Monetary Economics*, 2010, *2*, 189–202.

Kohn, Robert, Paolo Giordani, and Ingvar Strid, "Adaptive Hybrid Metropolis-Hastings Samplers for DSGE Models," *Working Paper*, 2010.

Liu, Jun S., Faming Liang, and Wing Hung Wong, "The Multiple Try Method and Local Optimization in Metropolis Sampling," *Journal of the American Statistical Association*, 2000, *95*, 121–134.

Magnus, Jan R. and Heinz Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics*, 2nd ed., Wiley, New York., 1999.

Morf, M., "Fast Algorithms for Multivariate Systems." PhD dissertation, Stanford University 1974.

Neal, Radford, "Slice Sampling (with discussion)," *The Annals of Statistics*, 2003, *31*, 705–767.

Qi, Yuan and Thomas P. Minka, "Hessian-based Markov Chain Monte-Carlo Algorithms," *Unpublished Manuscript*, 2002.

Robert, Christian P. and George Casella, *Monte Carlo Statistical Methods*, Springer, 2004.

Roberts, G. O. and S.K. Sahu, "Updating Schemes, Correlation Structure, Blocking and Parameterization for the Gibbs Sampler," *Journal of the Royal Statistical Society. Series B (Methodological)*, 1997, *59* (2), 291–317.

Roberts, Gareth O. and Jeffrey S. Rosenthal, "Harris Recurrence of Metropolis-within-Gibbs and Trans-dimensional Markov Chains," *The Annals of Applied Probability*, 2006, *16*, 2123–2139.

Roberts, G.O. and Jeffrey S. Rosenthal, "Markov-Chain Monte Carlo: Some Practical Implications of Theoretical Results," *The Canadian Journal of Statistics*, 1998, *25* (1), 5–20.

Schorfheide, Frank, "Loss Function-based Evaluation of DSGE Models," *Journal of Applied Econometrics*, 2000, *15*, 645–670.

Smets, Frank and Raf Wouters, "Comparing Shocks and Frictions in US and Euro Area Business Cycles: A Bayesian DSGE Approach," *Journal of Applied Econometrics*, 2005, *20*, 161 – 183.

Tierney, Luke, "Markov Chains for Exploring Posterior Distributions," *The Annals of Statistics*, 1994, *22* (4), 1701–1728.

Tjelmeland, Hakon and Jo Eidsvik, "On the Use of Local Optimizations with Metropolis-Hastings Updates," *Journal of the Royal Statistical Society (Series B)*, 2004, *66*, 411–427.

A Appendix

A.1 Derivation of $\frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(T)}{\partial\theta'}\right)$

The derivation relies on differential forms. For background see Magnus and Neudecker (1999).

Some Important Matrices and Matrix Facts.

$$\begin{aligned}
K_{mn}vec(A) &= vec(A'), \text{where } A \text{ is an } m\times n \text{ matrix.} && (18)\\
D_n vech(A) &= vec(A), \text{where } A \text{ is a symmetric } n\times n \text{ matrix} && (19)\\
D_n^{+}vec(A) &= vech(A), \text{where } A \text{ is a symmetric } n\times n \text{ matrix} && (20)\\
vec(A\otimes B) &= (I_n\otimes K_{mn}\otimes I_p)(vecA\otimes vecB), && (21)\\
K_{pm}(A\otimes B) &= (B\otimes A)K_{qn}\text{where } A \text{ is an } m\times n \text{ and } B \text{ a } p\times q \text{ matrix} && (22)
\end{aligned}$$

Derivation of $\frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(T)}{\partial\theta'}\right)$. Consider the function derived by Iskrev (2010).

$$(T'\otimes I_{n_s})\frac{\partial vec(\Gamma_0)}{\partial\theta'}+(I_{n_s}\otimes\Gamma_0)\frac{\partial vec(T)}{\partial\theta'}-(T'^2\otimes I_{n_s})\frac{\partial vec(\Gamma_1)}{\partial\theta'}-(T'\otimes\Gamma_1)\frac{\partial vec(T)}{\partial\theta'}-(I_{n_s}\otimes\Gamma_1 T)\frac{\partial vec(T)}{\partial\theta'}-\frac{\partial vec(\Gamma_2)}{\partial\theta'}=0 \tag{23}$$

The differential is this function is:

$$\begin{aligned}
&(dT'\otimes I_{n_s})\frac{\partial vec(\Gamma_0)}{\partial\theta'}+(T'\otimes I_{n_s})\frac{d\partial vec(\Gamma_0)}{\partial\theta'}+(I_{n_s}\otimes d\Gamma_0)\frac{\partial vec(T)}{\partial\theta'}+(I_{n_s}\otimes\Gamma_0)\frac{d\partial vec(T)}{\partial\theta'}\\
&-(d(T'^2)\otimes I_{n_s})\frac{\partial vec(\Gamma_1)}{\partial\theta'}-(T'^2\otimes I_{n_s})\frac{\partial vec(d\Gamma_1)}{\partial\theta'}-(dT'\otimes\Gamma_1)\frac{\partial vec(T)}{\partial\theta'}-(T'\otimes d\Gamma_1)\frac{\partial vec(T)}{\partial\theta'}-(T'\otimes\Gamma_1)\frac{d\partial vec(T)}{\partial\theta'}\\
&-(I_{n_s}\otimes d\Gamma_1 T)\frac{\partial vec(T)}{\partial\theta'}-(I_{n_s}\otimes\Gamma_1 dT)\frac{\partial vec(T)}{\partial\theta'}-(I_{n_s}\otimes\Gamma_1 T)\frac{d\partial vec(T)}{\partial\theta'}-\frac{d\partial vec(\Gamma_2)}{\partial\theta'}=0
\end{aligned} \tag{24}$$

So the partial derivative is given by:

$$\begin{aligned}
&\left(\left(\frac{\partial vec(\Gamma_0)}{\partial\theta'}\right)'\otimes I_{n_s^2}\right)(I_{n_s}\otimes K_{n_sn_s}\otimes I_{n_s})\left(K_{n_sn_s}\frac{\partial vec(T)}{\partial\theta'}\right)\otimes vec(I_{n_s}))+(I_{n_p}\otimes(T'\otimes I_{n_s}))\frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(\Gamma_0)}{\partial\theta'}\right)\\
&+\left(\left(\frac{\partial vec(T)}{\partial\theta'}\right)'\otimes I_{n_s^2}\right)(I_{n_s}\otimes K_{n_sn_s}\otimes I_{n_s})\left(vec(I_{n_s})\otimes\frac{\partial vec(\Gamma_0)}{\partial\theta'}\right)+(I_{n_p}\otimes(I_{n_s}\otimes\Gamma_0))\frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(T)}{\partial\theta'}\right)-\\
&\left(\left(\frac{\partial vec(\Gamma_1)}{\partial\theta'}\right)'\otimes I_{n_s^2}\right)(I_{n_s}\otimes K_{n_sn_s}\otimes I_{n_s})\left(\left[(T\otimes I_{n_s})K_{n_sn_s}\frac{\partial vec(T)}{\partial\theta'}+(I_{n_s}\otimes T')K_{n_sn_s}\frac{\partial vec(T)}{\partial\theta'}\right]\otimes vec(I_{n_s})\right)\\
&\qquad-(I_{n_p}\otimes(T'^2\otimes I_{n_s})\frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(\Gamma_1)}{\partial\theta'}\right)-\\
&\left(\left(\frac{\partial vec(T)}{\partial\theta'}\right)'\otimes I_{n_s^2}\right)(I_{n_s}\otimes K_{n_sn_s}\otimes I_{n_s})\left(\left[K_{n_sn_s}\frac{\partial vec(T)}{\partial\theta'}\otimes vec(\Gamma_1)\right]+\left[K_{n_sn_s}vec(T)\otimes\frac{\partial vec(\Gamma_1)}{\partial\theta'}\right]\right)\\
&\qquad-\left(I_{n_p}\otimes(T'\otimes\Gamma_1)\right)\frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(T)}{\partial\theta'}\right)-\frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(\Gamma_2)}{\partial\theta'}\right)=0
\end{aligned} \tag{25}$$

With this equation and the derivatives and hessians of $\{\Gamma_0, \Gamma_1, \Gamma_2, \Gamma_3\}$, one can solve for $\frac{\partial}{\partial\theta'} vec\left(\frac{\partial vec(T)}{\partial\theta'}\right)$.

Derivation of $\frac{\partial}{\partial\theta'} vec\left(\frac{\partial vec(R)}{\partial\theta'}\right)$. Consider the function derived by Iskrev:

$$\frac{\partial vec(R)}{\partial\theta'} = -\left(\Gamma_3' \otimes I_{n_s}\right)\left(W'^{-1} \otimes W^{-1}\right)\frac{\partial vec(W)}{\partial\theta'} + \left(I_{n_e} \otimes W^{-1}\right)\frac{\partial vec(\Gamma_3)}{\partial\theta'} \tag{26}$$

where $W = \Gamma_0 - \Gamma_1 T$. Consider first the differential of W,

$$dW = d\Gamma_0 - d\Gamma_1 T - \Gamma_1 dT. \tag{27}$$

So the derivative is given by,

$$\frac{\partial vec(W)}{\partial\theta'} = \frac{\partial vec(\Gamma_0)}{\partial\theta'} - \left(T' \otimes I_{n_s}\right)\frac{\partial vec(\Gamma_1)}{\partial\theta'} - \left(I_{n_s} \otimes \Gamma_1\right)\frac{\partial vec(T)}{\partial\theta'}. \tag{28}$$

And the Hessian of W is,

$$\begin{aligned}
\frac{\partial vec()}{\partial\theta'} vec\left(\frac{\partial vec(W)}{\partial\theta'}\right) \quad = \quad & \frac{\partial vec()}{\partial\theta'} vec\left(\frac{\partial vec(\Gamma_0)}{\partial\theta'}\right) \\
& -\left(\frac{\partial vec(\Gamma_1)}{\partial\theta'} \otimes I_{n_s^2}\right)\left(I_{n_s} \otimes K_{n_s n_s} \otimes I_{n_s}\right)\left(K_{n_s n_s}\frac{\partial vec(T)}{\partial\theta'} \otimes vec(I_{n_s})\right) \\
& -\left(I_{n_p} \otimes T' \otimes I_{n_s}\right)\frac{\partial vec()}{\partial\theta'} vec\left(\frac{\partial vec(\Gamma_1)}{\partial\theta'}\right) \\
& -\left(\frac{\partial vec(T)}{\partial\theta'} \otimes I_{n_e n_s}\right)\left(I_{n_e} \otimes K_{n_s n_y} \otimes I_{n_s}\right)\left(vec(I_{n_e}) \otimes \frac{\partial vec(\Gamma_1)}{\partial\theta'}\right) \\
& -\left(I_{n_p} \otimes I_{n_s} \otimes \Gamma_1\right)\frac{\partial vec()}{\partial\theta'} vec\left(\frac{\partial vec(T)}{\partial\theta'}\right).
\end{aligned} \tag{29}$$

Finally,

$$\frac{\partial vec(W^{-1})}{\partial\theta'} = -\left(W^{-1'} \otimes W^{-1}\right)\frac{\partial vec(W)}{\partial\theta'}. \tag{30}$$

and

$$\frac{\partial\left(W^{-1'} \otimes W^{-1}\right)}{\partial\theta} = \left(I_{n_s} \otimes K_{n_s n_s} \otimes I_{n_s}\right)\left(K_{n_s n_s}\frac{\partial vec(W^{-1})}{\partial\theta'} \otimes vec(W^{-1}) + vec(W^{-1'}) \otimes \frac{\partial vec(W^{-1})}{\partial\theta'}\right) \tag{31}$$

We deduce that,

$$\begin{aligned}
\frac{\partial vec(R)}{\partial\theta'} \quad = \quad & -\left(\frac{\partial vec(W)}{\partial\theta'}' \otimes \Gamma_3' \otimes I_{n_s}\right)\frac{\partial\left(W^{-1'} \otimes W^{-1}\right)}{\partial\theta} \\
- \quad & \left(\frac{\partial vec(W)}{\partial\theta'}' \otimes W^{-1'} \otimes W^{-1}\right)\left(I_{n_s} \otimes K_{n_s n_e} \otimes I_{n_s}\right)\left(K_{n_s n_e}\frac{\partial vec(\Gamma_3)}{\partial\theta'} \otimes vec(I_{n_s})\right) \\
- \quad & \left(I_{n_p} \otimes \left(\Gamma_3' \otimes I_{n_s}\right)\left(W'^{-1} \otimes W^{-1}\right)\right)\frac{\partial vec()}{\partial\theta'} vec\left(\frac{\partial vec(W)}{\partial\theta'}\right) \\
+ \quad & \left(\frac{\partial vec(\Gamma_3)}{\partial\theta'} \otimes I_{n_e n_s}\right)\left(I_{n_e} \otimes K_{n_s n_e} \otimes I_{n_s}\right)\left(vec(I_{n_e}) \otimes \frac{\partial vec(W^{-1})}{\partial\theta'}\right) \\
+ \quad & \left(I_{n_p n_e} \otimes W^{-1}\right)\frac{\partial vec()}{\partial\theta'} vec\left(\frac{\partial vec(\Gamma_3)}{\partial\theta'}\right)
\end{aligned} \tag{32}$$

A.2 Derivation of Exact Gradient and Hessian of the Log Posterior

[7]

A.2.1 Analytic Gradient

Note: Rewrite using Chandrasekar Equations.

Initialization. Let $a_{t|t}$ and $P_{t|t}$ be the "updated" mean and variance of the state vector. We assume that the system stationary, so that it is at the unconditional mean and variance at $t = 0$.

$$\begin{aligned} a_{0|0} &= 0 \qquad (33) \\ P_{0|0} &= TP_{0|0}T' + RQR' \qquad (34) \end{aligned}$$

Clearly $\partial a_{0|0}/\partial\theta' = 0$. For $P_{0|0}$, the differential is

$$dP_{0|0} = dTP_{0|0}T' + TdP_{0|0}T' + TP_{0|0}dT' + dRQR' + RdQR' + RQdR'. \tag{35}$$

So the derivative is

$$\begin{aligned} \frac{\partial vec(P_{0|0})}{\partial\theta'} &= (TP_{0|0} \otimes I_{n_s})\frac{\partial vec(T)}{\partial\theta'} + (T\otimes T)\frac{\partial vec(P_{0|0})}{\partial\theta'} + (I_{n_s} \otimes TP_{0|0})\frac{\partial vec(T')}{\partial\theta'} \\ &\quad + (RQ\otimes I_{n_s})\frac{\partial vec(R)}{\partial\theta'} + (R\otimes R)\frac{\partial vec(Q)}{\partial\theta'} + (I_{n_s} \otimes RQ)\frac{\partial vec(R')}{\partial\theta'} \qquad (36) \end{aligned}$$

Using the matrix facts from above, we can write this derivative as,

$$\begin{aligned} \frac{\partial vec(P_{0|0})}{\partial\theta'} &= \left(I_{n_s^2} - (T\otimes T)\right)^{-1} (\left(I_{n_s^2} + K_{n_s n_s}\right)\left[(TP_{0|0}\otimes I_{n_s})\frac{\partial vec(T)}{\partial\theta'} + (RQ\otimes I_{n_s})\frac{\partial vec(R)}{\partial\theta'}\right] \\ &\quad + (R\otimes R)\frac{\partial vec(Q)}{\partial\theta'}). \qquad (37) \end{aligned}$$

Forecasting. The Kalman filter forecasting equations are

$$\begin{aligned} a_{t+1|t} &= Ta_{t|t} \qquad (38) \\ P_{t+1|t} &= TP_{t|t}T' + RQR' \qquad (39) \end{aligned}$$

The differential of the state forecasting equation is

$$da_{t+1|t} = dTa_{t|t} + Tda_{t|t} \tag{40}$$

Hence the derivative with respect to θ is,

$$\frac{\partial a_{t+1|t}}{\partial\theta'} = (a'_{t|t}\otimes I_{n_s})\frac{\partial vec(T)}{\partial\theta'} + T\frac{\partial a_{t|t}}{\partial\theta'}. \tag{41}$$

The differential of the variance is

$$dP_{t+1|t} = dTP_{t|t}T' + TdP_{t|t}T' + TP_{t|t}dT' + dRQR' + RdQR' + RQdR'. \tag{42}$$

[7] We omit the dependence of the system matrices on θ for simplicity.

Hence the derivative with respect to θ is,

$$\begin{aligned}\frac{\partial vec(P_{t+1|t})}{\partial\theta'} &= \left(I_{n_s^2} + K_{n_s n_s}\right)(TP_{t|t} \otimes I_{n_s})\frac{\partial vec(T)}{\partial\theta'} + (T \otimes T)\frac{\partial vec(P_{t|t})}{\partial\theta'} \\ &+ \left(I_{n_s^2} + K_{n_s n_s}\right)(RQ \otimes I_{n_s})\frac{\partial vec(R)}{\partial\theta'} + (R \otimes R)\frac{\partial vec(Q)}{\partial\theta'} \qquad (43)\end{aligned}$$

The forecast error and variance are defined as:

$$\eta_{t+1} = y_{t+1} - (D + Za_{t+1|t}) \qquad (44)$$

$$F_{t+1} = ZP_{t+1|t}Z' + H \qquad (45)$$

The derivatives for these objects are

$$\frac{\partial\eta}{\partial\theta'} = -\frac{\partial vec(D)}{\partial\theta'} - (a'_{t|t} \otimes I_{n_y})\frac{\partial vec(Z)}{\partial\theta'} - Z\frac{\partial a_{t+1|t}}{\partial\theta'} \qquad (46)$$

$$\frac{\partial F_{t+1}}{\partial\theta'} = \left(I_{n_y^2} + K_{n_y n_y}\right)(ZP_{t+1|t} \otimes I_{n_y})\frac{\partial vec(Z)}{\partial\theta'} + (Z \otimes Z)\frac{\partial vec(P_{t+1|t})}{\partial\theta'} + \frac{\partial vec(H)}{\partial\theta'} \qquad (47)$$

Log Likelihood. The log likelihood is given by

$$\mathcal{L}(Y|\theta) = -\frac{1}{2}\sum_{t=1}^{T}\left(n_y \ln(2\pi) + \ln(det(F_t)) + \eta' F_t^{-1}\eta\right) \qquad (48)$$

It is useful to first construct the derivative of F_t^{-1}. The differential form for matrix inversion is given by $d(X^{-1}) = -X^{-1}dXX^{-1}$. So the derivative of F_t^{-1} is

$$\frac{\partial vec\left(F_t^{-1}\right)}{\partial\theta'} = -\left(F_t^{-1} \otimes F_t^{-1}\right)\frac{\partial vec\left(F_t\right)}{\partial\theta'}. \qquad (49)$$

Using $d\ln(det(X)) = tr(X^{-1}dX)$ and the differential is

$$d\mathcal{L}(Y|\theta) = -\frac{1}{2}\sum_{t=1}^{T}\left(tr(F_t^{-1}dF_t) + \eta' d\left(F_t^{-1}\right)\eta + 2\eta' F_t^{-1}d\eta\right). \qquad (50)$$

So the gradient is given by,

$$\frac{\partial\mathcal{L}(Y|\theta)}{\partial\theta'} = -\frac{1}{2}\sum_{t=1}^{T}\left(vec(F_t^{-1})'\frac{\partial vec(F_t)}{\partial\theta'} + \left(\eta' \otimes \eta'\right)\frac{\partial vec\left(F_t^{-1}\right)}{\partial\theta'} + 2\eta' F_t^{-1}\frac{\partial\eta}{\partial\theta'}\right) \qquad (51)$$

Updating.

$$a_{t+1|t+1} = a_{t+1|t} + P_{t+1|t}Z'F_{t+1}^{-1}\eta_t \qquad (52)$$

$$P_{t+1|t+1} = P_{t+1|t} - P_{t+1|t}Z'F_{t+1}^{-1}ZP_{t+1|t} \qquad (53)$$

It is useful to consider $P_{t+1|t}Z'$ as group. The differential of this matrix is given by

$$d(P_{t+1|t}Z') = dP_{t+1|t}Z' + P_{t+1|t}d(Z') \qquad (54)$$

So the derivative is given by

$$\frac{\partial vec(P_{t+1|t}Z')}{\partial\theta'} = (Z \otimes I_{n_s})\frac{\partial vec(P_{t+1|t})}{\partial\theta'} + \left(I_{n_y} \otimes P_{t+1|t}\right)K_{n_y n_s}\frac{\partial vec(Z)}{\partial\theta'} \qquad (55)$$

The differential for $a_{t+1|t+1}$ is

$$da_{t+1|t+1} = da_{t+1|t} + d(P_{t+1|t}Z')F_{t+1}^{-1}\eta_t + P_{t+1|t}Z'd(F_{t+1}^{-1})\eta_t + P_{t+1|t}Z'F_{t+1}^{-1}d\eta_t \quad (56)$$

Hence the derivative is:

$$\frac{\partial a_{t+1|t+1}}{\partial\theta'} = \frac{\partial a_{t+1|t}}{\partial\theta'} + (\eta'_{t+1}F_{t+1}^{-1} \otimes I_{n_s})\frac{\partial vec(P_{t+1|t}Z')}{\partial\theta'} + (\eta'_{t+1} \otimes P_{t+1|t}Z')\frac{\partial vec(F_{t+1}^{-1})}{\partial\theta'} + P_{t+1|t}Z'F_{t+1}^{-1}\frac{\partial\eta_{t+1}}{\partial\theta'} \quad (57)$$

The differential for $P_{t+1|t+1}$ is

$$dP_{t+1|t+1} = dP_{t+1|t} - d(P_{t+1|t}Z')F_{t+1}^{-1}ZP_{t+1|t} - P_{t+1|t}Z'dF_{t+1}^{-1}ZP_{t+1|t} - P_{t+1|t}Z'F_{t+1}^{-1}d(ZP_{t+1|t}). \quad (58)$$

Hence the derivative is

$$\begin{aligned}\frac{\partial vec(P_{t+1|t+1})}{\partial\theta'} = \frac{\partial vec(P_{t+1|t})}{\partial\theta'} &- \left(I_{n_s^2} + K_{n_sn_s}\right)\left(P_{t+1|t}Z'F_{t+1|t}^{-1} \otimes I_{n_s}\right)\frac{\partial vec(P_{t+1|t}Z')}{\partial\theta'} \\ &- \left(P_{t+1|t}Z' \otimes P_{t+1|t}Z'\right)\frac{\partial vec(F_t^{-1})}{\partial\theta'} \quad (59)\end{aligned}$$

A.2.2 Hessian

A few important derivatives. For an $m \times n$ matrix function X,

$$\begin{aligned}\frac{\partial vec(X \otimes X)}{\partial\theta'} &= (I_n \otimes K_{nm} \otimes I_m)\left[\left(\frac{\partial vec(X)}{\partial\theta'} \otimes vec(X)\right) + \left(vec(X) \otimes \frac{\partial vec(X)}{\partial\theta'}\right)\right] \quad (60) \\ &= (I_n \otimes K_{nm} \otimes I_m)\left(I_{(mn)^2} + K_{(mn)(mn)}\right)\left(\frac{\partial vec(X)}{\partial\theta'} \otimes vec(X)\right) \quad (61)\end{aligned}$$

Forecasting. Begin by taking the differential of **57**.

$$d\frac{\partial a_{t+1|t}}{\partial\theta'} = (da'_{t|t} \otimes I_{n_s})\frac{\partial vec(T)}{\partial\theta'} + (a'_{t|t} \otimes I_{n_s})d\frac{\partial vec(T)}{\partial\theta'} + dT\frac{\partial a_{t|t}}{\partial\theta'} + Td\frac{\partial a_{t|t}}{\partial\theta'} \quad (62)$$

Recognizing that $(I_{n_s} \otimes K_{n_s1} \otimes I_{n_s}) = I_{n_p^3}$, we can write the Hessian of $a_{t+1|t}$ as,

$$\begin{aligned}\frac{\partial}{\partial\theta'}vec\left(\frac{\partial a_{t+1|t}}{\partial\theta'}\right) = &\left(\frac{\partial vec(T)'}{\partial\theta'} \otimes I_{n_s}\right)\left(\frac{\partial a_{t|t}}{\partial\theta'} \otimes vec(I_{n_s})\right) + \left(I_{n_p} \otimes a'_{t|t} \otimes I_{n_s}\right)\frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(T)}{\partial\theta'}\right) \\ &+ \left(\frac{\partial a_{t|t}}{\partial\theta'}' \otimes I_{n_s}\right)\frac{\partial vec(T)}{\partial\theta'} + \left(I_{n_p} \otimes T\right)\frac{\partial}{\partial\theta'}vec\left(\frac{\partial a_{t|t}}{\partial\theta'}\right). \quad (63)\end{aligned}$$

For the state forecast variance, begin by taking the differential of **43**.

$$\begin{aligned}d\frac{\partial vec(P_{t+1|t})}{\partial\theta'} = &\left(I_{n_s^2} + K_{n_sn_s}\right)(d(TP_{t|t}) \otimes I_{n_s})\frac{\partial vec(T)}{\partial\theta'} + \left(I_{n_s^2} + K_{n_sn_s}\right)(d(TP_{t|t}) \otimes I_{n_s})d\frac{\partial vec(T)}{\partial\theta'} \\ &+ (dT \otimes T)\frac{\partial vec(P_{t|t})}{\partial\theta'} + (T \otimes dT)\frac{\partial vec(P_{t|t})}{\partial\theta'} + T \otimes T)d\frac{\partial vec(P_{t|t})}{\partial\theta'} \\ &+ \left(I_{n_s^2} + K_{n_sn_s}\right)(d(RQ) \otimes I_{n_s})\frac{\partial vec(R)}{\partial\theta'} + (R \otimes R)\frac{\partial vec(Q)}{\partial\theta'} \quad (64)\end{aligned}$$

Here we have used,

$$\begin{aligned}\frac{\partial vec(TP_{t|t})}{\partial\theta'} &= \left(P_{t|t} \otimes I_{n_s}\right)\frac{\partial vec(T)}{\partial\theta'} + (I_{n_s} \otimes T)\frac{\partial vec(P_{t|t})}{\partial\theta'} \quad (65) \\ \frac{\partial vec(RQ)}{\partial\theta'} &= (Q \otimes I_{n_s})\frac{\partial vec(R)}{\partial\theta'} + (I_{n_e} \otimes R)\frac{\partial vec(Q)}{\partial\theta'} \quad (66)\end{aligned}$$

Using**65**, **66**, and **60**, the Hessian of $P_{t+1|t}$ can be expressed as

$$\begin{aligned}
\frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(P_{t+1|t})}{\partial\theta'}\right) &= \left(\frac{\partial vec(T)}{\partial\theta'}' \otimes (I_{n_s^2}+K_{n_s n_s})\right)(I_{n_s}\otimes K_{n_s n_s}\otimes I_{n_s})\left(\frac{\partial vec(TP_{t|t})}{\partial\theta'}\otimes vec(I_{n_s})\right)\\
+\left(\frac{\partial vec(P_{t|t})}{\partial\theta'}'\otimes I_{n_s^2}\right)&(I_{n_s}\otimes K_{n_s n_s}\otimes I_{n_s})\left(I_{n_s^2}+K_{n_s^2 n_s^2}\right)\left(\frac{\partial vec(T)}{\partial\theta'}\otimes vec(T)\right)\\
&+\left(I_{n_p}\otimes T\otimes T\right)\frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(P_{t|t})}{\partial\theta'}\right)\\
+\left(\frac{\partial vec(R)}{\partial\theta'}'\otimes\left(I_{n_s^2}+K_{n_s n_s}\right)\right)&(I_{n_e}\otimes K_{n_s n_s}\otimes I_{n_s})\left(\frac{\partial vec(RQ)}{\partial\theta'}\otimes vec(I_{n_s})\right)\\
&+\left(I_{n_p}\otimes\left(I_{n_s^4}+K_{n_s^2 n_s^2}\right)(RQ\otimes I_{n_s})\right)\frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(R)}{\partial\theta'}\right)\\
+\left(\frac{\partial vec(Q)}{\partial\theta'}'\otimes I_{n_s^2}\right)&(I_{n_e}\otimes K_{n_e n_s}\otimes I_{n_s})\left(I_{(n_s n_e)^2}+K_{(n_s n_e)(n_s n_e)}\right)\left(\frac{\partial vec(R)}{\partial\theta'}\otimes vec(R)\right)\\
&+\left(I_{n_p}\otimes R\otimes R\right)\frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(Q)}{\partial\theta'}\right)\\
&+\left(I_{n_p}\otimes(I_{n_s^2}+K_{n_s^2 n_s^2})\left(TP_{t|t}\otimes I_{n_s}\right)\right)\frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(T)}{\partial\theta'}\right). \qquad (67)
\end{aligned}$$

To find the Hessian of $P_{0|0}$ set $P_{t+1|t}=P_{t|t}$ and solve in the previous equation.

Log Likelihood. Begin by taking the Hessian of η_{t+1},

$$\begin{aligned}
\frac{\partial}{\partial\theta'}vec\left(\frac{\partial\eta}{\partial\theta'}\right) &= -\frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(D)}{\partial\theta'}\right)-\left(\frac{\partial vec(Z)}{\partial\theta'}'\otimes I_{n_y}\right)\left(\frac{\partial a_{t+1|t}}{\partial\theta'}\otimes vec(I_{n_y})\right)\\
-\left(I_{n_p}\otimes a'_{t|t}\otimes I_{n_y}\right)\frac{\partial}{\partial\theta'}vec&\left(\frac{\partial vec(Z)}{\partial\theta'}\right)-\left(I_{n_p}\otimes Z\right)\frac{\partial}{\partial\theta'}vec\left(\frac{\partial a_{t+1|t}}{\partial\theta'}\right)-\left(\frac{\partial a_{t+1|t}}{\partial\theta'}'\otimes I_{n_y}\right)\frac{\partial vec(Z)}{\partial\theta'}.
\end{aligned}\qquad (68)$$

We have used the fact that $\partial^2 y_{t+1}/\partial\theta\partial\theta'=0$. The Hessian of $F_{t+1|t}$ is given by:

$$\begin{aligned}
\frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(F_t)}{\partial\theta'}\right) &= \left(\frac{\partial vec(Z)}{\partial\theta'}'\otimes\left(I_{n_y^2}+K_{n_y n_y}\right)\right)(I_{n_s}\otimes K_{n_y n_y}\otimes I_{n_y})\left(K_{n_s n_y}\frac{\partial vec(P_{t|t}Z')}{\partial\theta'}\otimes vec(I_{n_y})\right)+\\
&\left(I_{n_p}\otimes\left(I_{n_y^2}+K_{n_y n_y}\right)(ZP_{t+1|t}\otimes I_{n_s})\right)\frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(Z)}{\partial\theta'}\right)\\
+\left(\frac{\partial vec(P_{t+1|t})}{\partial\theta'}'\otimes I_{n_y^2}\right)&(I_{n_s}\otimes K_{n_s n_y}\otimes I_{n_y})\left(I_{(n_y n_s)^2}+K_{(n_y n_s)(n_y n_s)}\right)\left(\frac{\partial vec(vec(Z))}{\partial\theta'}\otimes vec(Z)\right)\\
&+\left(I_{n_p}\otimes Z\otimes Z\right)\frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(P_{t+1|t})}{\partial\theta'}\right). \qquad (69)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial^2\mathcal{L}(\theta|y_{t+1})}{\partial\theta\partial\theta'} &= \frac{\partial^2\mathcal{L}(\theta|y_t)}{\partial\theta\partial\theta'}-\frac{1}{2}\left(I_{n_p}\otimes vec(F^{-1}_{t+1|t})'\right)\frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(F^{-1}_{t+1|t})}{\partial\theta'}\right)-\frac{1}{2}\frac{\partial vec(F_{t+1|t})}{\partial\theta'}'\frac{\partial vec(F^{-1}_{t+1|t})}{\partial\theta'}\\
&- \frac{1}{2}\left(I_{n_p}\otimes\eta'_{t+1}\otimes\eta_{t+1}\right)\frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(F^{-1}_{t+1|t})}{\partial\theta'}\right)\\
&- \frac{1}{2}\frac{\partial vec(F^{-1}_{t+1|t})}{\partial\theta'}'\left(I_{n_y^2}+K_{n_y n_y}\right)\left(\frac{\partial vec(\eta_{t+1})}{\partial\theta'}\otimes\eta'_{t+1}\right)\\
&- \left(\frac{\partial vec(\eta_{t+1})}{\partial\theta'}'\otimes\eta_{t+1}\right)\frac{\partial vec(F^{-1}_{t+1|t})}{\partial\theta'}-\frac{\partial vec(\eta_{t+1})}{\partial\theta'}'F^{-1}_{t+1|t}\frac{\partial vec(\eta_{t+1})}{\partial\theta'}\\
&- \left(I_{n_p}\otimes\eta'_{t+1}F^{-1}_{t+1|t}\right)\frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(\eta_{t+1})}{\partial\theta'}\right)
\end{aligned}$$

Updating.

$$\begin{aligned}
\frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(P_{t+1|t}Z')}{\partial\theta'}\right) &= \left(\frac{\partial vec(P_{t+1|t})}{\partial\theta'}^{\prime}\otimes I_{n_y n_s}\right)\left(I_{n_s}\otimes K_{n_s n_y}\otimes I_{n_s}\right)\left(\frac{\partial vec(Z)}{\partial\theta'}\otimes vec(I_{n_s})\right) \qquad (70)\\
&+ \left(I_{n_p}\otimes Z\otimes I_{n_s}\right)\frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(P_{t+1|t})}{\partial\theta'}\right)\\
&+ \left(\frac{\partial vec(Z)}{\partial\theta'}^{\prime}K_{n_s n_y}\otimes I_{n_y n_s})\right)\left(I_{n_y}\otimes K_{n_s n_y}\otimes I_{n_s}\right)\left(vec(I_{n_y})\otimes\frac{\partial vec(P_{t+1|t})}{\partial\theta'}\right)\\
&+ \left(I_{n_p}\otimes\left(I_{n_y}\otimes P_{t+1|t}\right)K_{n_y n_s}\right)\frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(Z)}{\partial\theta'}\right)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(a_{t+1|t+1})}{\partial\theta'}\right) &= \frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(a_{t+1|t})}{\partial\theta'}\right) \qquad (7\\
&+ \left(\frac{\partial vec(P_{t+1|t}Z')}{\partial\theta'}^{\prime}\otimes I_{n_s}\right)\Bigg(\left(F_{t+1|t}^{-1}\frac{\partial vec(\eta_{t+1})}{\partial\theta'}+\left(I_{n_y}\otimes\eta_{t+1}'\right)\frac{\partial vec(F_{t+1|t}^{-1})}{\partial\theta'}\right)\otimes vec(I_{n_s})\Bigg)\\
&+ \left(I_{n_p}\otimes\eta_{t+1}'F_{t+1|t}^{-1}\otimes I_{n_s}\right)\frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(P_{t+1|t}Z')}{\partial\theta'}\right)\\
&+ \left(\frac{\partial vec(F_{t+1|t}^{-1})}{\partial\theta'}\otimes I_{n_s}\right)\left(\frac{\partial vec(\eta_{t+1})}{\partial\theta'}\otimes vec(P_{t+1|t}Z')+\eta_{t+1}\otimes\frac{\partial vec(P_{t+1|t}Z')}{\partial\theta'}\right)\\
&+ \left(I_{n_p}\otimes\eta_{t+1}'\otimes P_{t+1|t}Z'\right)\frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(F_{t+1}^{-1})}{\partial\theta'}\right)+\left(\frac{\partial vec(\eta_{t+1})}{\partial\theta'}^{\prime}F_{t+1}^{-1}\otimes I_{n_s}\right)\frac{\partial vec(P_{t+1|t}Z}{\partial\theta'}\\
&+ \left(\frac{\partial vec(\eta_{t+1})}{\partial\theta'}^{\prime}\otimes P_{t+1|t}Z'\right)\frac{\partial vec(F_{t+1|t}^{-1})}{\partial\theta'}+\left(I_{n_p}\otimes P_{t+1|t}Z'F_{t+1|t}^{-1}\right)\frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(\eta_{t+1})}{\partial\theta'}\right)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(P_{t+1|t+1})}{\partial\theta'}\right) &= \frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(P_{t+1|t})}{\partial\theta'}\right) \qquad (72)\\
&- \left(\frac{\partial vec(P_{t+1|t}Z')}{\partial\theta'}^{\prime}\otimes\left(I_{n_s^2}+K_{n_s n_s}\right)\right)\left(I_{n_y}\otimes K_{n_s n_s}\otimes I_{n_s}\right)\\
&\times \left(\left[\left(F_{t+1|t}^{-1}\otimes I_{n_s}\right)\frac{\partial vec(P_{t+1|t}Z')}{\partial\theta'}+\left(I_{n_y}\otimes P_{t+1|t}Z'\right)\frac{\partial vec(F_{t+1|t}^{-1})}{\partial\theta'}\right]\otimes vec(I_{n_s})\right)\\
&- \left(I_{n_p}\otimes\left(I_{n_s^2}+K_{n_s n_s}\right)\left(P_{t+1|t}Z'F_{t+1|t}^{-1}\otimes I_{n_s}\right)\right)\frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(P_{t+1|t}Z')}{\partial\theta'}\right)\\
&- \left(\frac{\partial vec(F_{t+1|t}^{-1})}{\partial\theta'}^{\prime}\otimes I_{n_s^2}\right)\left(I_{n_y}\otimes K_{n_y n_s}\otimes I_{n_s}\right)\left(I_{(n_y n_s)^2}+K_{(n_y n_s)(n_y n_s)}\right)\\
&\times \left(\frac{\partial vec(P_{t+1|t}Z')}{\partial\theta'}\otimes vec(P_{t+1|t}Z')\right)\\
&- \left(I_{n_p}\otimes P_{t+1|t}Z'\otimes P_{t+1|t}Z'\right)\frac{\partial}{\partial\theta'}vec\left(\frac{\partial vec(F_{t+1|t}^{-1})}{\partial\theta'}\right)
\end{aligned}$$

A.3 Invariance

Let $f(x)$ be the distribution of interest, where x is a $k \times 1$ vector from the state space X. Let J be the total number of block schema. For each $i = 1 \dots J$, define the $K_i(x, \cdot)$ to be the transition kernel of Block Metropolis Hastings algorithm corresponding to the *ith* blocking schema.

As shown in Chib and Jeliazkov (2001), $K_i(x, \cdot)$ satisfies local reversibility; i.e., detailed balance. See Roberts and Rosenthal (2006) for conditions under which the Block Metropolis-Hastings is Harris-recurrent.

Start with kernel $K_i(x, \cdot)$ which is which preserves the stationary distribution π. Define the *state-dependent mixture* as,

$$\tilde{K}(x, \cdot) = \sum_{i=1}^{n} \alpha_i(x) K_i(x, \cdot).$$

State-dependent mixtures were introduced by Green (1995) and are used extensively in reversible jump MCMC for simulating over distributions of models. Unfortunately, once the probabilities α_i depend on x it is not trivial to show that combined kernel $\tilde{K}(x, \cdot)$ preserves f. Reversible jump algorithms are typically used with kernels that are reversible (hence the name). However, Geyer (2003) shows that if each $\alpha_i(x)K_i(x, \cdot)$ is sub-Markovian,

$$\alpha_i(x) K_i(x, A) \leq 1,$$

then the Markov chain,

$$K^*(x, A) = I(x, A)[1 - K(x, X)] + K(x, A)$$

is reversible (with respect to the dominating measure for the model) for

$$I(x, A) = \begin{cases} 1, & x \in A, \\ 0, & \text{otherwise.} \end{cases}$$

So K^* is Markovian and preserves f, and the Block Metropolis Hastings algorithm is valid.

A.4 RBC Model

The optimality conditions for the model imply that for all t

$$\begin{aligned} C_t &: \beta^t \left[C_t(1-N_t)^\theta\right]^{-\gamma}(1-N_t)^\theta + \lambda_t = 0 && (73) \\ N_t &: -\theta\beta^t \left[C_t(1-N_t)^\theta\right]^{-\gamma} C_t(1-N_t)^{\theta-1} + \lambda_t W_t = 0 && (74) \\ K_t &: \lambda_t - \beta\lambda_{t+1}((1-\delta)+R_{t+1}) = 0 && (75) \\ \lambda_t &: W_tN_t + R_tK_{t-1} - C_t - K_t + (1-\delta)K_{t-1} - G_t = 0 && (76) \end{aligned}$$

So that the intertemporal Euler condition is,

$$\left[C_t(1-N_t)^\theta\right]^{-\gamma}(1-N_t)^\theta = \beta E_t\left[\left[C_{t+1}(1-N_{t+1})^\theta\right]^{-\gamma}(1-N_{t+1})^\theta(1-\delta+R_{t+1})\right]. \quad (77)$$

The intratemporal Euler condition is,

$$\theta\frac{C_t}{1-N_t} = W_t. \quad (78)$$

The firm operates in a competitive market and maximizes 1-period profits,

$$\max(A_tN_t)^{1-\alpha}K_{t-1}^\alpha - W_tN_t - R_tK_{t-1} \quad (79)$$

Which of courses implies

$$\begin{aligned} W_t &= (1-\alpha)Y_t/N_t && (80) \\ R_t &= \alpha Y_t/K_{t-1} && (81) \end{aligned}$$

A.4.1 Steady State

$$\begin{aligned} R_{ss} &= 1/\beta - (1-\delta) && (82) \\ K_{ss} &= \alpha Y_{ss}/R_{ss} && (83) \\ I_{ss} &= (1-(1-\delta))K_{ss} && (84) \\ C_{ss} &= Y_{ss} - (1-(1-\delta))K_{ss} - G_{ss} && (85) \\ N_{ss} &= \left(\frac{(1-\alpha)Y_{ss}}{\theta C_{ss}}\right) / \left(1 + \frac{(1-\alpha)Y_{ss}}{\theta C_{ss}})\right) && (86) \\ & && (87) \end{aligned}$$

A.4.2 Log Linearization

$$\begin{aligned}
\hat{\lambda}_t &= E_t[\hat{\lambda}_{t+1} + \frac{R_{SS}}{R_{SS}+1-\delta}\hat{r}_{t+1}] && (88)\\
\hat{\lambda}_t &= -\gamma\hat{c}_t - (1-\gamma)\theta\frac{N_{ss}}{1-N_{ss}}\hat{n}_t && (89)\\
\hat{w}_t &= \hat{c}_t + \frac{N_{ss}}{1-N_{ss}}\hat{n}_t && (90)\\
\hat{w}_t &= \hat{y}_t - \hat{n}_t && (91)\\
\hat{r}_t &= \hat{y}_t - \hat{k}_{t-1} && (92)\\
\hat{i}_t &= \frac{K_{ss}}{I_{ss}}(\hat{k}_t - (1-\delta)\hat{k}_{t-1}) && (93)\\
\hat{y}_t &= (1-\alpha)\hat{a}_t + (1-\alpha)\hat{n}_t + \alpha\hat{k}_{t-1} && (94)\\
\hat{y}_t &= \frac{C_{ss}}{Y_{ss}}\hat{c}_t + \frac{I_{ss}}{Y_{ss}}\hat{i}_t + \frac{G_{ss}}{Y_{ss}}\hat{g}_t && (95)\\
\hat{a}_t &= \rho_A\hat{a}_{t-1} + \rho_{AG}g_{t-1} + \sigma_A\varepsilon_{A,t} && (96)\\
\hat{g}_t &= \rho_G\hat{g}_{t-1} + \rho_{GA}a_{t-1} + \sigma_G\varepsilon_{G,t} && (97)
\end{aligned}$$

We group the parameters into $\theta = [\alpha, \beta, \delta, \theta, \gamma, \rho_A, \rho_G, \sigma_A, \sigma_G, \rho_{AG}, \rho_{GA}]$. We calibrate $G_{ss} = 0.2Y_{ss}$.

A.5 Smets and Wouters Model

The Loglinearized Model:

$$\hat{y}_t = c_y\hat{c}_t + i_y\hat{i}_t + r^{kss}k_y\hat{z}_t + \varepsilon_t^g \tag{98}$$

$$\begin{aligned}\hat{c}_t = {} & \frac{h/\gamma}{1+h/\gamma}\hat{c}_{t-1} + \frac{1}{1+h/\gamma}E_t\hat{c}_{t+1} + \frac{wl^{ss}(\sigma_c-1)}{c^{ss}\sigma_c(1+h/\gamma)}(\hat{l}_t - E_t\hat{l}_{t+1}) \\ & -\frac{1-h/\gamma}{(1+h/\gamma)\sigma_c}(\hat{r}_t - E_t\hat{\pi}_{t+1}) - \frac{1-h/\gamma}{(1+h/\gamma)\sigma_c}\varepsilon_t^b\end{aligned} \tag{99}$$

$$\hat{i}_t = \frac{1}{1+\beta\gamma^{(1-\sigma_c)}}\hat{i}_{t-1} + \frac{\beta\gamma^{(1-\sigma_c)}}{1+\beta\gamma^{(1-\sigma_c)}}E_t\hat{i}_{t+1} + \frac{1}{\phi\gamma^2(1+\beta\gamma^{(1-\sigma_c)})}\hat{q}_t + \varepsilon_t^i \tag{100}$$

$$\hat{q}_t = \beta(1-\delta)\gamma^{-\sigma_c}E_t\hat{q}_{t+1} - \hat{r}_t + E_t\hat{\pi}_{t+1} + (1-\beta(1-\delta)\gamma^{-\sigma_c})E_t\hat{r}_{t+1}^k - \varepsilon_t^b \tag{101}$$

$$\hat{y}_t = \phi_p(\alpha\hat{k}_t^s + (1-\alpha)\hat{l}_t + \varepsilon_t^a) \tag{102}$$

$$\hat{k}_t^s = \hat{k}_{t-1} + \hat{z}_t \tag{103}$$

$$\hat{z}_t = \frac{1-\psi}{\psi}\hat{r}_t^k \tag{104}$$

$$\hat{k}_t = \frac{(1-\delta)}{\gamma}\hat{k}_{t-1} + (1-(1-\delta)/\gamma)\hat{i}_t + (1-(1-\delta)/\gamma)\varphi\gamma^2(1+\beta\gamma^{(1-\sigma_c)})\varepsilon_t^i \tag{105}$$

$$\hat{\mu}_t^p = \alpha(\hat{k}_t^s - \hat{l}_t) - \hat{w}_t + \varepsilon_t^a \tag{106}$$

$$\begin{aligned}\hat{\pi}_t = {} & \frac{\beta\gamma^{(1-\sigma_c)}}{1+\iota_p\beta\gamma^{(1-\sigma_c)}}E_t\hat{\pi}_{t+1} + \frac{i_p}{1+\beta\gamma^{(1-\sigma_c)}}\hat{\pi}_{t-1} \\ & -\frac{(1-\beta\gamma^{(1-\sigma_c)}\xi_p)(1-\xi_p)}{(1+\iota_p\beta\gamma^{(1-\sigma_c)})(1+(\phi_p-1)\varepsilon_p)\xi_p}\hat{\mu}_t^p + \varepsilon_t^p\end{aligned} \tag{107}$$

$$\hat{r}_t^k = \hat{l}_t + \hat{w}_t - \hat{k}_t \tag{108}$$

$$\hat{\mu}_t^w = \hat{w}_t - \sigma_l\hat{l}_t - \frac{1}{1-h/\gamma}(\hat{c}_t - h/\gamma\hat{c}_{t-1}) \tag{109}$$

$$\hat{w}_t = \frac{\beta\gamma^{(1-\sigma_c)}}{1+\beta\gamma^{(1-\sigma_c)}}(E_t\hat{w}_{t+1} + E_t\hat{\pi}_{t+1}) + \frac{1}{1+\beta\gamma^{(1-\sigma_c)}}(\hat{w}_{t-1} - \iota_w\hat{\pi}_{t-1}) \tag{110}$$

$$-\frac{1+\beta\gamma^{(1-\sigma_c)}\iota_w}{1+\beta\gamma^{(1-\sigma_c)}}\hat{\pi}_t + \frac{(1-\beta\gamma^{(1-\sigma_c)}\xi_w)(1-\xi_w)}{(1+\beta\gamma^{(1-\sigma_c)})(1+(\varphi_w-1)\epsilon_w)\xi_w}\hat{\mu}_t^w + \varepsilon_t^w \tag{111}$$

$$\hat{r}_t = \rho\hat{r}_{t-1} + (1-\rho)(r_\pi\hat{\pi}_t + r_y(\hat{y}_t - \hat{y}_t^*)) + r_{\Delta y}((\hat{y}_t - \hat{y}_t^*) - (\hat{y}_{t-1} - \hat{y}_{t-1}^*)) + \varepsilon_t^r \tag{112}$$

$$\varepsilon_t^a = \rho_a\varepsilon_{t-1}^a + \eta_t^a \tag{113}$$

$$\varepsilon_t^b = \rho_b\varepsilon_{t-1}^b + \eta_t^b \tag{114}$$

$$\varepsilon_t^g = \rho_g\varepsilon_{t-1}^a + \rho_{ga}\eta_t^a + \eta_t^g \tag{115}$$

$$\varepsilon_t^i = \rho_i\varepsilon_{t-1}^i + \eta_t^i \tag{116}$$

$$\varepsilon_t^r = \rho_r\varepsilon_{t-1}^r + \eta_t^r \tag{117}$$

$$\varepsilon_t^p = \rho_r\varepsilon_{t-1}^p + \eta_t^p - \mu_p\eta_{t-1}^p \tag{118}$$

$$\varepsilon_t^w = \rho_w\varepsilon_{t-1}^w + \eta_t^w - \mu_w\eta_{t-1}^w \tag{119}$$

$$\tag{120}$$

$$\begin{aligned}
\hat{y}_t^* &= c_y\hat{c}_t^* + i_y\hat{i}_t^* + r^{kss}k_y\hat{z}_t^* + \varepsilon_t^g && (121)\\
\hat{c}_t^* &= \frac{\lambda/\gamma}{1+\lambda/\gamma}\hat{c}_{t-1}^* + \frac{1}{1+\lambda/\gamma}E_t\hat{c}_{t+1}^* + \frac{wl^{ss}(\sigma_c-1)}{c^{ss}\sigma_c(1+\lambda/\gamma)}(\hat{l}_t^* - E_t\hat{l}_{t+1}^*) && (122)\\
&\quad -\frac{1-\lambda/\gamma}{(1+\lambda/\gamma)\sigma_c}r_t^* - \frac{1-\lambda/\gamma}{(1+\lambda/\gamma)\sigma_c}\varepsilon_t^b\\
\hat{i}_t^* &= \frac{1}{1+\beta\gamma^{(1-\sigma_c)}}\hat{i}_{t-1}^* + \frac{\beta\gamma^{(1-\sigma_c)}}{1+\beta\gamma^{(1-\sigma_c)}}E_t\hat{i}_{t+1}^* + \frac{1}{\phi\gamma^2(1+\beta\gamma^{(1-\sigma_c)})}\hat{q}_t^* + \varepsilon_t^i && (123)\\
\hat{q}_t^* &= \beta(1-\delta)\gamma^{-\sigma_c}E_t\hat{q}_{t+1}^* - r_t^* + (1-\beta(1-\delta)\gamma^{-\sigma_c})E_tr_{t+1}^{k*} - \varepsilon_t^b && (124)\\
\hat{y}_t^* &= \phi_p(\alpha k_t^{s*} + (1-\alpha)\hat{l}_t^* + \varepsilon_t^a) && (125)\\
\hat{k}_t^{s*} &= k_{t-1}^* + z_t^* && (126)\\
\hat{z}_t^* &= \frac{1-\psi}{\psi}\hat{r}_t^{k*} && (127)\\
\hat{k}_t &= \frac{(1-\delta)}{\gamma}\hat{k}_{t-1}^* + (1-(1-\delta)/\gamma)\hat{i}_t + (1-(1-\delta)/\gamma)\varphi\gamma^2(1+\beta\gamma^{(1-\sigma_c)})\varepsilon_t^i && (128)\\
\hat{\mu}_t^{p*} &= \alpha(\hat{k}_t^{s*} - \hat{l}_t^*) - \hat{w}_t^* + \varepsilon_t^a && (129)\\
\hat{\mu}_t^{p*} &= 1 && (130)\\
\hat{r}_t^{k*} &= \hat{l}_t^* + \hat{w}_t^* - \hat{k}_t^* && (131)\\
\hat{\mu}_t^{w*} &= -\sigma_l\hat{l}_t^* - \frac{1}{1-\lambda/\gamma}(\hat{c}_t^* + \lambda/\gamma\hat{c}_{t-1}^*) && (132)\\
\hat{w}_t^* &= \mu_t^{w*} && (133)
\end{aligned}$$

With,

$$\begin{aligned}
\gamma &= \bar{\gamma}/100 + 1 && (134)\\
\pi^* &= \bar{\pi}/100 + 1 && (135)\\
\bar{r} &= 100(\beta^{-1}\gamma^{\sigma_c}\pi^* - 1) && (136)\\
r_{ss}^k &= \gamma^{\sigma_c}/\beta - (1-\delta) && (137)\\
w_{ss} &= \left(\frac{\alpha^\alpha(1-\alpha)^{(1-\alpha)}}{(\phi r_{ss}^k)^\alpha}\right)^{\frac{1}{1-\alpha}} && (138)\\
i_k &= (1-(1-\delta)/\gamma)\gamma && (139)\\
l_k &= \frac{1-\alpha}{\alpha}\frac{r_{ss}^k}{w_{ss}} && (140)\\
k_y &= \phi l_k^{(\alpha-1)} && (141)\\
i_y &= (\gamma - 1 + \delta)k_y && (142)\\
c_y &= 1 - g_y - i_y && (143)\\
z_y &= r_{ss}^k k_y && (144)\\
& && (145)
\end{aligned}$$

A.6 News Shock Model

B Graphs and Tables

	Random Walk Metropolis		Hessian Based Metropolis		
Parameter	True	Mean	90 % CI	Mean	90 % CI
T11	0.80	0.810	[0.664, 0.871]	0.806	[0.661, 0.875]
T22	0.20	0.146	[-0.177, 0.261]	0.136	[-0.173, 0.260]
T33	0.75	0.703	[0.501, 0.775]	0.704	[0.502, 0.775]
T44	0.60	0.675	[0.466, 0.751]	0.674	[0.466, 0.751]
T55	0.10	0.035	[-0.379, 0.186]	0.029	[-0.352, 0.179]
D1	0.20	0.408	[-0.491, 0.815]	0.401	[-0.492, 0.816]
D2	1.40	1.396	[0.906, 1.609]	1.407	[0.907, 1.609]
D3	1.80	2.176	[1.096, 2.586]	2.150	[1.077, 2.577]
D4	0.10	0.187	[-0.416, 0.421]	0.182	[-0.408, 0.417]
D5	0.90	0.634	[0.038, 0.882]	0.630	[0.042, 0.860]
D6	1.00	0.805	[0.364, 0.981]	0.814	[0.367, 0.983]
D7	2.00	2.043	[1.663, 2.182]	2.045	[1.673, 2.183]
D8	0.10	0.157	[-0.500, 0.401]	0.152	[-0.506, 0.397]
D9	2.20	2.304	[1.913, 2.444]	2.316	[2.190, 2.399]
D10	1.50	1.456	[1.187, 1.558]	1.469	[1.182, 1.535]
Z21	0.50	0.409	[0.166, 0.484]	0.410	[0.164, 0.487]
Z31	0.60	0.706	[0.316, 0.847]	0.693	[0.303, 0.840]
Z32	0.00	-0.046	[-0.461, 0.109]	-0.039	[-0.436, 0.104]
Z41	0.00	-0.001	[-0.322, 0.115]	0.004	[-0.297, 0.115]
Z42	0.20	0.101	[-0.131, 0.224]	0.090	[-0.131, 0.225]
Z43	-0.10	-0.029	[-0.422, 0.106]	-0.021	[-0.428, 0.104]
Z51	-0.20	-0.261	[-0.570,-0.150]	-0.267	[-0.575,-0.156]
Z52	0.00	0.043	[-0.224, 0.164]	0.043	[-0.227, 0.164]
Z53	-0.70	-0.529	[-0.701,-0.432]	-0.526	[-0.716,-0.444]
Z54	0.00	0.024	[-0.178, 0.136]	0.024	[-0.188, 0.111]

Table 1: Posterior Results for the Generic State Space System, Part 1

	Random Walk Metropolis		Hessian Based Metropolis		
Parameter	True	Mean	90 % CI	Mean	90 % CI
Z61	0.00	-0.031	[-0.295, 0.056]	-0.035	[-0.302, 0.053]
Z71	0.30	0.219	[0.044, 0.286]	0.219	[0.047, 0.279]
Z81	-0.50	-0.437	[-0.697,-0.340]	-0.425	[-0.698,-0.347]
Z91	0.00	0.100	[-0.086, 0.166]	0.097	[-0.010, 0.159]
Z101	0.00	0.006	[-0.174, 0.063]	0.004	[-0.187, 0.059]
Z62	0.00	-0.052	[-0.357, 0.049]	-0.045	[-0.342, 0.050]
Z72	0.20	0.348	[-0.019, 0.471]	0.345	[-0.015, 0.475]
Z72	0.00	-0.001	[-0.371, 0.131]	-0.015	[-0.384, 0.149]
Z92	-0.50	-0.338	[-0.641,-0.235]	-0.333	[-0.500,-0.250]
Z102	0.00	0.046	[-0.240, 0.143]	0.049	[-0.110, 0.126]
Z63	-0.40	-0.373	[-0.618,-0.284]	-0.364	[-0.494,-0.295]
Z73	0.00	0.014	[-0.204, 0.090]	0.013	[-0.101, 0.070]
Z83	0.00	0.019	[-0.299, 0.135]	0.020	[-0.149, 0.112]
Z93	0.30	0.313	[0.125, 0.381]	0.308	[0.125, 0.362]
Z103	0.20	0.161	[-0.048, 0.227]	0.153	[0.052, 0.235]
Z64	-0.50	-0.545	[-0.732,-0.477]	-0.541	[-0.646,-0.489]
Z74	0.00	-0.015	[-0.253, 0.062]	-0.014	[-0.248, 0.064]
Z84	0.60	0.647	[0.391, 0.737]	0.648	[0.401, 0.739]
Z94	-0.10	0.025	[-0.185, 0.095]	0.023	[-0.188, 0.097]
Z104	0.00	-0.068	[-0.262, 0.001]	-0.068	[-0.267,-0.001]
Z65	0.00	0.099	[-0.200, 0.194]	0.108	[-0.204, 0.192]
Z75	-0.30	-0.186	[-0.480,-0.068]	-0.194	[-0.485,-0.067]
Z85	0.00	-0.004	[-0.335, 0.135]	-0.005	[-0.332, 0.136]
Z95	0.00	0.061	[-0.244, 0.167]	0.046	[-0.115, 0.130]
Z105	-0.40	-0.347	[-0.621,-0.233]	-0.349	[-0.501,-0.260]
log(H11)	-0.60	-0.529	[-1.065,-0.283]	-0.568	[-0.966,-0.365]
log(H22)	-1.40	-1.467	[-2.655,-1.093]	-1.492	[-2.647,-1.035]
log(H33)	-0.20	-0.289	[-0.989, 0.023]	-0.291	[-0.985,-0.024]
log(H44)	-1.10	-1.164	[-1.902,-0.823]	-1.176	[-1.902,-0.819]
log(H55)	-0.50	-0.451	[-1.395,-0.023]	-0.461	[-1.395,-0.041]
log(H66)	-0.85	-0.867	[-1.309,-0.683]	-0.869	[-1.312,-0.690]
log(H77)	0.00	-0.050	[-0.374, 0.080]	-0.052	[-0.382, 0.083]
log(H88)	0.00	0.021	[-0.358, 0.168]	-0.015	[-0.357, 0.168]
log(H99)	-0.35	-0.352	[-0.761,-0.203]	-0.354	[-0.761,-0.206]
log(H1010)	-0.50	-0.52	[-0.911,-0.309]	-0.462	[-0.918,-0.301]

Table 2: Postior Results for the Generic State Space System, Part 2

	RWMH	HMH		RWMH	HMH
T11	1770.504	068.070	Z62	469.062	7.764
T22	497.635	26.081	Z72	1180.399	4.009
T33	1080.512	25.018	Z72	658.940	1.898
T44	1245.188	35.984	Z92	852.893	1.204
T55	957.894	4.513	Z102	673.523	1.777
Z21	646.790	0.991	Z63	650.033	9.426
Z31	1054.846	10.445	Z73	489.433	1.791
Z32	500.266	1.245	Z83	634.595	13.276
Z41	652.728	4.936	Z93	516.957	2.947
Z42	380.393	4.937	Z103	428.844	4.177
Z43	570.691	16.374	Z64	808.732	4.178
Z51	943.256	1.683	Z74	273.838	1.527
Z52	813.701	10.193	Z84	597.955	2.551
Z53	1165.414	3.605	Z94	328.814	1.022
Z54	565.163	3.979	Z104	357.225	2.063
D1	4289.261	45.873	Z65	945.645	1.901
D2	3255.506	33.838	Z75	1041.226	3.961
D3	3646.995	52.297	Z85	1102.199	1.130
D4	2713.929	3.166	Z95	836.050	12.589
D5	2942.756	23.538	Z105	1574.284	5.181
D6	2394.646	3.817	H11	2336.803	3.764
D7	2194.708	26.471	H22	4172.562	1.863
D8	3411.031	21.437	H33	3058.559	4.094
D9	2055.815	26.974	H44	2960.845	12.767
D10	1594.918	4.680	H55	4023.465	8.907
Z61	930.084	2.627	H66	1870.005	12.351
Z71	631.925	3.578	H77	1223.108	2.173
Z81	637.414	6.175	H88	1094.616	0.998
Z91	729.320	6.356	H99	1436.564	1.178
Z101	575.784	3.501	H1010	1686.031	1.102

Table 3: Inefficiency factors for the Generic State Space Model

Table 4: Geweke's K

	Based on Mean Geweke‘s K	
	Random Walk Metropolis	Hessian Based Metropolis
Effective Sample Size	216.58	974.11
Seconds per “independent draw	83.11	81.11
	Based on Max Geweke‘s K	
	Random Walk Metropolis	Hessian Based Metropolis
Effective Sample Size	46.297	200.11
Seconds per “independent” draw	388	327

Table 5: Wall Time per Indepedent Draw

C Other Algorithms

C.1 Chib and Ramamurthy (2010)

- At step i, split θ in j blocks randomly.
- For each block $b = 1 \dots j$.
  - Find the mode $\hat{\theta}_b$ conditional on $\theta_{i,-b}$ which is the all the parameters not in b at their most recent values. To do this optimization, use a simulated annealing algorithm proposed by Chib and Greenberg. Compute the Hessian at the mode $H(\hat{\theta}_b)$.
  - Generate a proposal θ_b^* from $q(\cdot|\hat{\theta}_b, \theta_{i,-b}) = t(\hat{\theta}_b, -H(\theta_b)^{-1}, \nu)$.
  - Accept this proposal with probability,

    $$\alpha = \min\left\{1, \frac{f(\theta_b^*, \theta_{i,-b}|Y)q(\theta_{i,b}|\hat{\theta}_b, \theta_{i,-b})}{f(\theta_{i,b}, \theta_{i,-b}|Y)q(\theta_b^*|\hat{\theta}_b, \theta_{i,-b})}\right\}$$

  - Move on to next block
- Move on to next block

C.2 Kohn et al. (2010)

The core of the algorithm is the proposal distribution which is a mixture of three components:

- A random walk component $q_{1,i+1}(\theta|\Theta_i)$ which a mixture of three normals:

  $$q_{1,i+1} = \alpha\beta N(\theta_i, \kappa\Sigma_{i+1}) + (1-\alpha)\beta N(\theta_i, \kappa_2\Sigma_{i+1}) + (1-\beta)N(\theta_i, \kappa_3 I)$$

  Σ_{i+1} is a recursively estimated measure of empirical variance of the chain.

- An independence chain component based on the empirical mean and variance of chain, $q_{2,i+1}(\theta) = N(\bar{\theta}_i, \Sigma_{i+1})$.
- T copula with mixutre of normal marginal distributions.

  $$q_{3,i+1} = \tilde{\beta} q_{31,i+1}(\theta|\Theta_i) + (1-\tilde{\beta}) q_{32,i+1}(\theta|\theta_i)$$

  where q_{31} is t copula based on the mixture of normal distributions with parameters $\{\lambda_j\}, j = 1 \ldots, d = n_{para}$, estimating via ML by way of clustering algorithm:

  $$q_{31} = \frac{t_{d,v}(x|\mu, \Sigma)}{\Pi_{i=1}^{d} t_{1,\nu}(x^j|0,1)} \Pi_{i=1}^{d} f_j(\theta^j|\lambda_j)$$

  where x^j and θ^j are related by

  $$T_{1,\nu}(x^j|0,1) = F_j(\theta^j|\lambda_j)$$

  \$q$_{32}$ is a fat-tailed version of q_{31} obtained by inflating the mixture of normal variances by a factor of 9.

The weights on the three components change over time. More weight is gradually placed on the copula until it dominates the proposal.

The full proposal is a hybrid of the proposals:

$$q_t = \alpha_{1t} q_{1,t} + \alpha_{2t} q_{2,t} + (1 - \alpha_{1t} - \alpha_{2t}) q_{3,t}$$

The weights $\{\alpha_{1t}, \alpha_{2t}\}$ evolve according to a deterministic hybrid schedule. The idea is start out with a heavy weight on the random walk component, occasionally taking independence steps, as a estimate of the covariance matrix and mean of the distribution is built up, the independence sampler is used more often with occasional random walk moves to avoid getting stuck at difficult points.

Finally, at prespecifed intervals, the parameters $\{\lambda_j\}$ of the mixture of normals used in the t-copula are re-estimated using the entire history of draws.

Parameter	Distribution	Support	Para(1)	Para(2)
α	Beta	$[0,1]$	0.340	0.020
β	Beta	$[0,1]$	0.990	0.004
δ	Beta	$[0,1]$	0.050	0.005
$\gamma - 1$	Normal	$[0,\infty)$	1.000	0.307
θ	Normal	$[0,\infty)$	2.400	0.707
ρ_A	Beta	$[0,1]$	0.600	0.150
ρ_G	Beta	$[0,1]$	0.500	0.150
σ_A	Inv. Gamma	$[0,\infty)$	0.050	4.000
σ_G	Inv. Gamma	$[0,\infty)$	0.580	4.000
ρ_{AG}	Normal	$(-\infty,\infty)$	0.000	0.010
ρ_{GA}	Normal	$(-\infty,\infty)$	0.000	6.000

Table 6: Prior

Parameter	Mode 1	Mode 2
α	3.93	3.92
β	0.99	0.99
δ	0.05	0.05
$\gamma - 1$	0.53	1.16
θ	2.05	2.33
ρ_A	0.93	0.97
ρ_G	0.85	0.82
σ_A	0.04	0.04
σ_G	0.77	0.65
ρ_{AG}	0.01	0.02
ρ_{GA}	-3.21	- 3.83
Posterior	663.1	663.3

Table 7: Prior

	Prior		Random Walk MH		Hessian Block MH		Random Block MH		Fixed Block MH	
Parameter	Mean	90 % CI	Mean	90 % CI	Mean	90 % CI	Mean	90 % CI	Mean	90 % CI
α	0.340	[0.269, 0.366]	0.396	[0.334, 0.424]	0.395	[0.300, 0.424]	0.393	[0.327, 0.416]	0.388	[0.328, 0.415]
β	0.990	[0.981, 0.995]	0.991	[0.983, 0.995]	0.993	[0.986, 0.999]	0.993	[0.987, 0.996]	0.991	[0.984, 0.996]
δ	0.050	[0.031, 0.057]	0.047	[0.033, 0.052]	0.046	[0.026, 0.053]	0.048	[0.036, 0.053]	0.044	[0.031, 0.049]
$\gamma - 1$	0.998	[0.097, 1.384]	0.616	[0.002, 0.978]	0.743	[0.000, 1.405]	1.183	[0.372, 1.521]	0.539	[0.003, 0.979]
θ	2.408	[0.017, 3.321]	2.100	[0.248, 3.032]	2.095	[0.001, 3.217]	2.588	[0.220, 3.637]	2.026	[0.269, 2.937]
ρ_A	0.599	[0.233, 0.798]	0.927	[0.875, 0.955]	0.945	[0.895, 0.993]	0.969	[0.945, 0.988]	0.916	[0.855, 0.958]
ρ_G	0.501	[0.070, 0.701]	0.803	[0.650, 0.895]	0.803	[0.675, 0.948]	0.818	[0.701, 0.881]	0.802	[0.660, 0.889]
σ_A	0.250	[0.080, 0.387]	0.041	[0.033, 0.045]	0.041	[0.032, 0.044]	0.041	[0.035, 0.044]	0.041	[0.034, 0.044]
σ_G	1.101	[0.346, 1.704]	0.876	[0.361, 1.188]	1.294	[0.338, 1.484]	0.666	[0.331, 0.829]	0.843	[0.373, 1.134]
ρ_{AG}	-0.000	[-0.033, 0.013]	0.023	[0.016, 0.026]	0.019	[0.001, 0.025]	0.010	[0.004, 0.014]	0.022	[0.014, 0.025]
ρ_{GA}	0.047	[-19.743, 7.662]	-3.309	[-5.141,-2.257]	-3.549	[-7.497,-1.848]	-3.721	[-5.853,-2.624]	-4.271	[-6.668,-1.530]
σ_{AG}	0.000	[0.000, 0.000]	0.034	[0.034, 0.034]	0.000	[0.000, 0.000]	0.000	[0.000, 0.000]	0.000	[0.000, 0.000]

Table 8: Posterior Estimates

Parameter	Prior	RWMH	Hessian Block MH	Random Block MH	Fixed Block MH	Adaptive MH
α	1.019	42.346	7.205	7.344	69.313	43.957
β	0.871	38.476	31.770	16.453	42.539	39.810
γ	1.056	33.117	157.344	80.556	55.239	45.284
θ	0.894	29.068	20.311	19.968	12.851	34.355
ρ_A	0.813	32.476	115.093	138.397	56.629	46.826
ρ_G	0.872	77.440	37.951	12.993	45.465	35.918
σ_A	0.890	38.666	8.122	5.971	2.095	44.249
σ_G	1.016	91.009	67.138	32.205	46.578	31.505
ρ_{AG}	0.931	42.504	251.163	164.934	33.580	54.145
ρ_{GA}	1.087	88.185	20.510	34.535	113.815	71.407
σ_{AG}	1.000	1.000	1.000	1.000	1.000	318.060

Table 9: Geweke's K

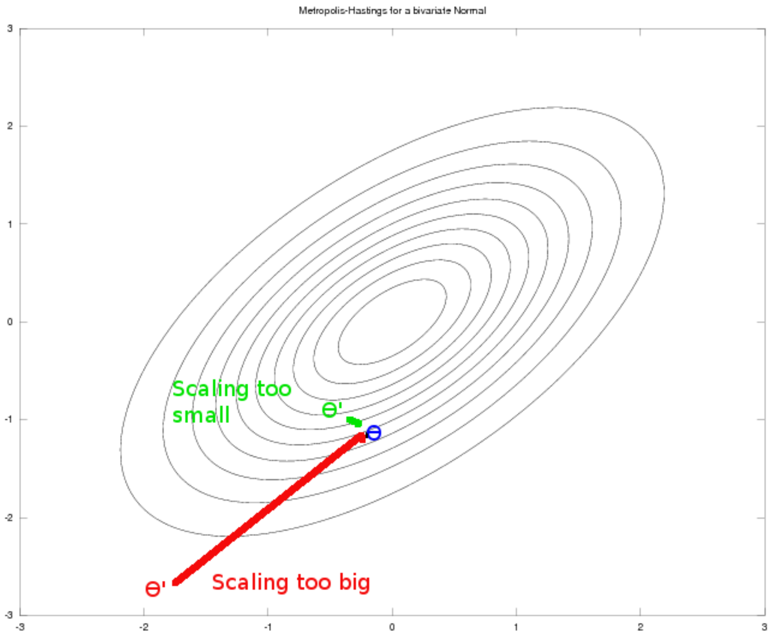

Figure 1: The effect of c on the proposals in the random walk Metropolis-Hastings

Parameter	Distribution	Support	Para(1)	Para(2)
φ	Normal	$(-\infty, \infty)$	4.000	1.500
σ_c	Normal	$(-\infty, \infty)$	1.500	0.370
h	Beta	$[0, 1]$	0.700	0.100
ξ_w	Beta	$[0, 1]$	0.500	0.100
σ_l	Normal	$(-\infty, \infty)$	2.000	0.750
ξ_p	Beta	$[0, 1]$	0.500	0.100
ι_w	Beta	$[0, 1]$	0.500	0.150
ι_p	Beta	$[0, 1]$	0.500	0.150
ψ	Beta	$[0, 1]$	0.500	0.150
λ_p	Normal	$(-\infty, \infty)$	1.250	0.120
r_π	Normal	$(-\infty, \infty)$	1.500	0.250
ρ	Beta	$[0, 1]$	0.750	0.100
r_y	Normal	$(-\infty, \infty)$	0.120	0.050
r_{Δ_y}	Normal	$(-\infty, \infty)$	0.120	0.050
$\bar{\pi}$	Gamma	$[0, \infty)$	0.620	0.100
$\beta^{-1} - 1$	Gamma	$[0, \infty)$	0.250	0.100
$\bar{l}$	Normal	$(-\infty, \infty)$	0.000	2.000
$\bar{\gamma}$	Normal	$(-\infty, \infty)$	0.400	0.100
α	Normal	$(-\infty, \infty)$	0.300	0.050
ρ_a	Beta	$[0, 1]$	0.500	0.200
ρ_b	Beta	$[0, 1]$	0.500	0.200
ρ_g	Beta	$[0, 1]$	0.500	0.200
ρ_i	Beta	$[0, 1]$	0.500	0.200
ρ_r	Beta	$[0, 1]$	0.500	0.200
ρ_p	Beta	$[0, 1]$	0.500	0.200
ρ_w	Beta	$[0, 1]$	0.500	0.200
μ_p	Beta	$[0, 1]$	0.500	0.200
μ_w	Beta	$[0, 1]$	0.500	0.200
ρ_{ga}	Beta	$[0, 1]$	0.500	0.200
σ_a	Inv. Gamma	$[0, \infty)$	0.100	2.000
σ_b	Inv. Gamma	$[0, \infty)$	0.100	2.000
σ_g	Inv. Gamma	$[0, \infty)$	0.100	2.000
σ_i	Inv. Gamma	$[0, \infty)$	0.100	2.000
σ_r	Inv. Gamma	$[0, \infty)$	0.100	2.000
σ_p	Inv. Gamma	$[0, \infty)$	0.100	2.000
σ_w	Inv. Gamma	$[0, \infty)$	0.100	2.000

Table 10: Priors for the Smets & Wouters model

Block	Frequency	Description	Posterior Correlations
$[r_\pi, r_y]$	100%	Taylor Rule coefficients for Inflation and Output	$corr(r_\pi, r_y) = 0.72$
$[\sigma_c, h, \rho_b]$	90%	Risk aversion, habit formation, and persistence of preference shock	$corr(\sigma_c, h) = -0.40$
$[\xi_p, \rho_p, \mu_p]$	95%	Calvo parameter for prices, persistence of price mark up, MA coefficient of price mark-up	$corr(\rho_p, \mu_p) = 0.72$
			$corr(\xi_p, \mu_p) = 0.08, corr(\xi_p, \rho_p) = -0.49$
$[\bar{\pi}, \bar{l}]$	100%	Steady state wages and steady state hours	$corr(\bar{\pi}, \bar{l}) = -0.60$
$[\xi_w, \mu_w]$	35%	Calvo parameter for wages and MA coefficient on wage markup	$corr(\xi_w, \mu_w) = 0.08$
$[\iota_w, \iota_p]$	30%	Wage and Price Indexation	$corr(\iota_w, \iota_p) = -0.23$

Table 11: Some Blocking Statistics from Smets and Wouters Model **??**

Parameter	Hessian Block MH	Random Block MH	Random Walk MH
φ	19.17	6.35	126.42
σ_c	18.26	10.94	310.38
h	16.73	29.06	203.93
ξ_w	11.18	7.32	208.02
σ_l	10.57	27.44	156.26
ξ_p	5.39	81.04	205.45
ι_w	9.32	2.69	111.23
ι_p	48.29	1.31	115.46
ψ	9.13	34.41	85.17
λ_p	7.14	12.38	143.63
r_π	2.43	4.50	72.143
ρ	8.55	20.29	97.20
r_y	6.57	4.72	135.85
r_{Δ_y}	6.50	3.85	97.19
$\bar{\pi}$	3.55	4.90	135.86
$\beta^{-1} - 1$	1.38	4.68	138.79
$\bar{l}$	1.52	6.36	116.83
$\bar{\gamma}$	35.52	12.47	289.02
α	8.81	20.26	103.96
ρ_a	8.64	3.52	216.38
ρ_b	45.58	8.22	462.00
ρ_g	20.91	7.84	273.64
ρ_i	11.16	4.36	73.85
ρ_r	9.39	15.26	151.77
ρ_p	21.67	5.34	269.69
ρ_w	17.58	5.48	472.46
μ_p	40.30	23.38	215.70
μ_w	2.15	19.61	413.06
ρ_{ga}	3.20	1.55	413.06
σ_a	13.23	8.32	136.56
σ_b	38.06	6.00	313.87
σ_g	8.77	4.08	207.99
σ_i	12.86	5.97	178.92
σ_r	3.75	2.64	276.11
σ_p	21.10	2.88	83.15
σ_w	14.46	3.87	199.59
Mean	14.51	11.78	193.74

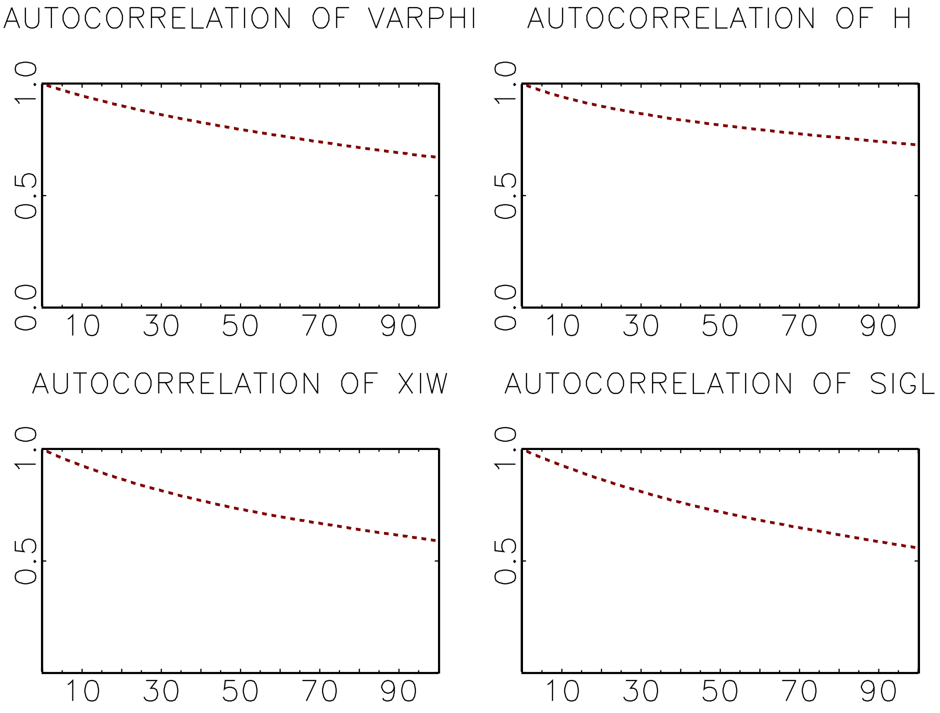

Figure 2: Autocorrelations of draws of four parameters from Smets and Wouters Model

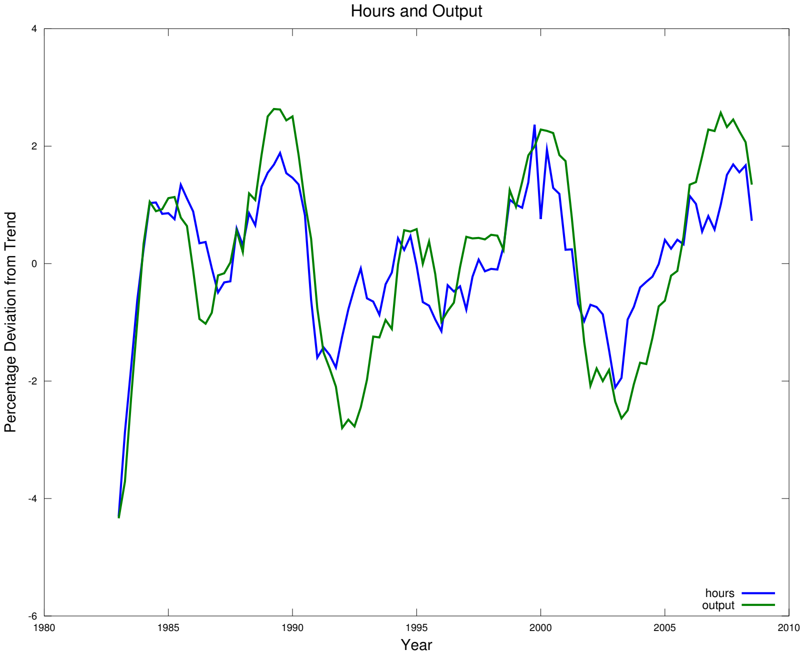

Figure 3: Data for the RBC Model

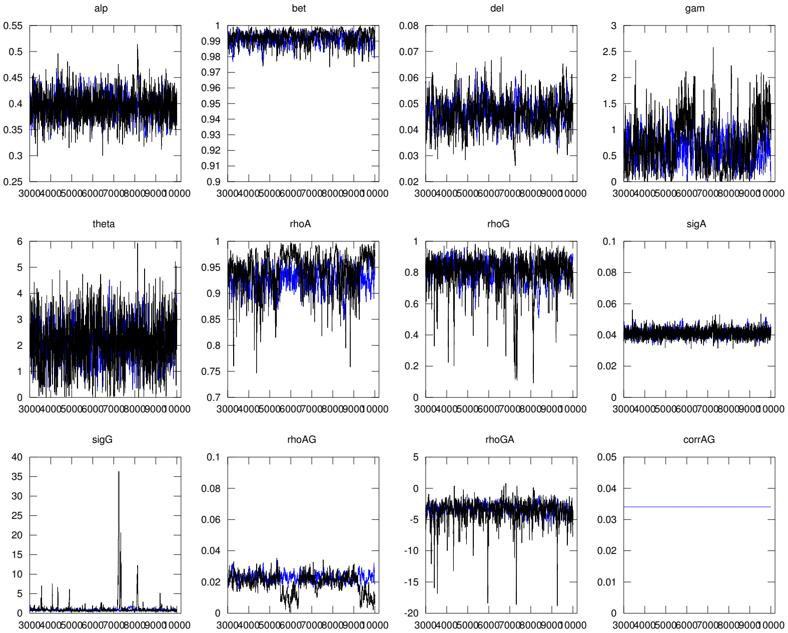

Figure 4: Draws from RWMH (Blue) vs. Hessian-based MH (Black)

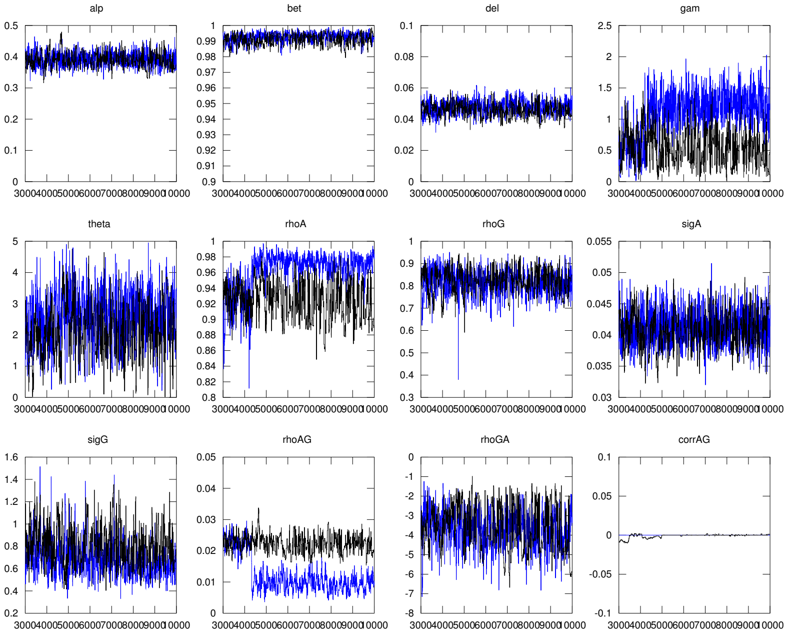

Figure 5: Draws from Random Block MH (Blue) vs. Adaptive Hybrid MH (Black)

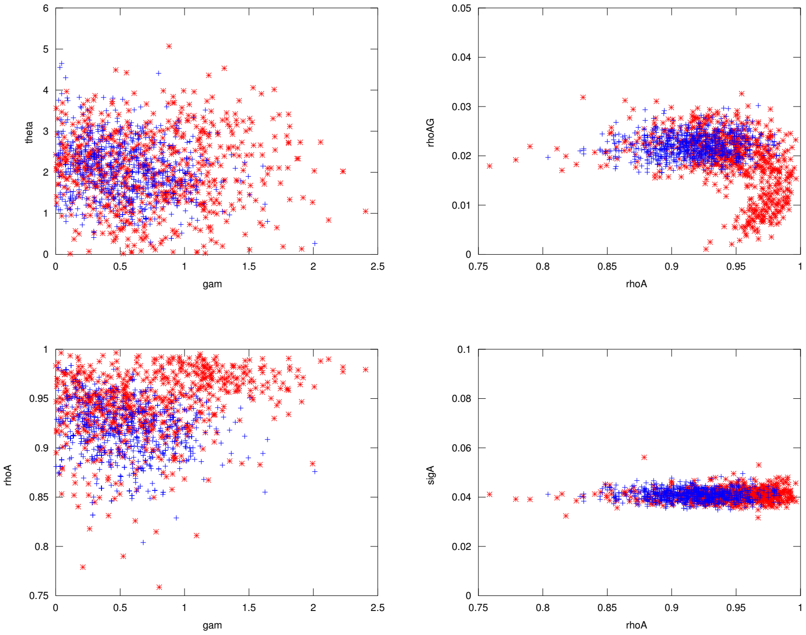

Figure 6: Draws from RWMH (Blue) vs. Hessian-based MH (Red)

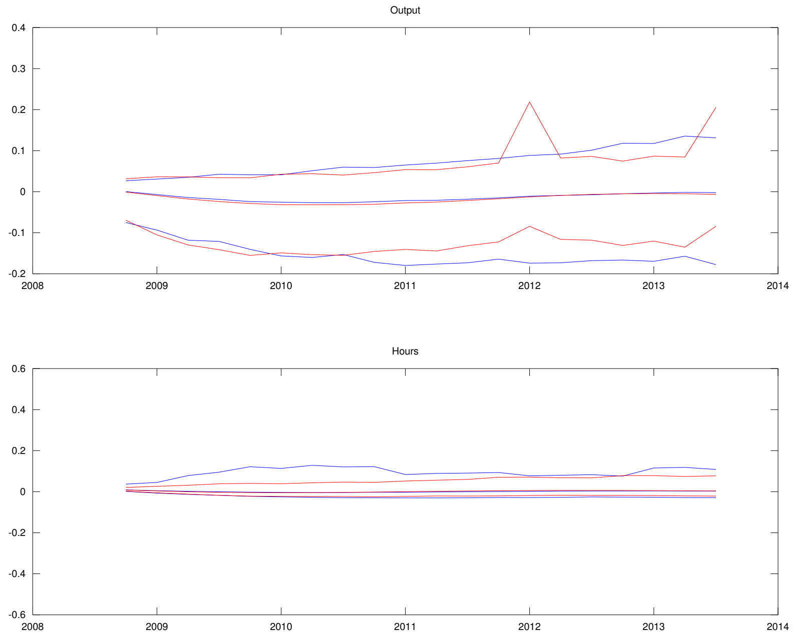

Figure 7: Predictive Density Mean and Credible Set for Hessian-based MH (Blue) and RWMH (Red)

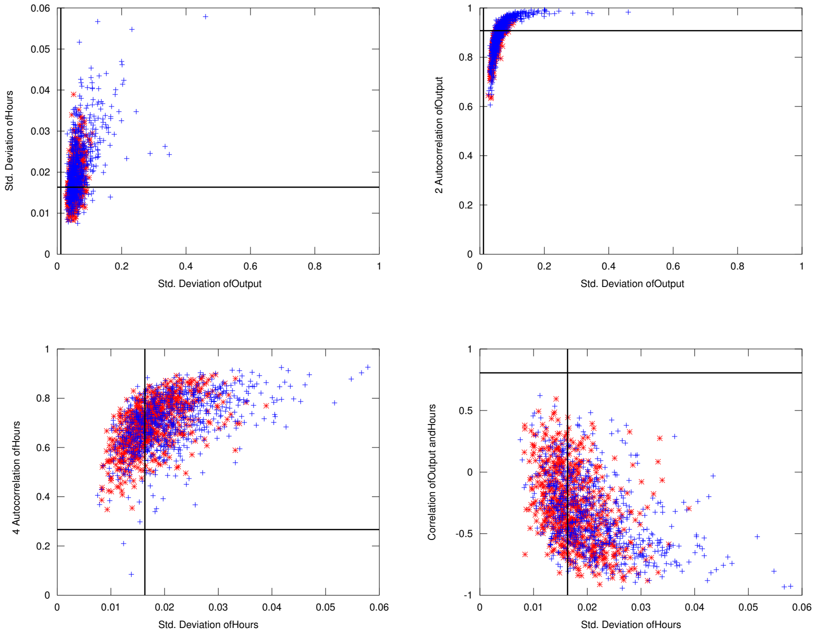

Figure 8: Posterior Predictive Checks for Hessian-based MH (Red) and RWMH (Red)