DYNAMIC STRIATED METROPOLIS-HASTINGS SAMPLER FOR HIGH-DIMENSIONAL MODELS

DANIEL F. WAGGONER, HONGWEI WU, AND TAO ZHA

Abstract. Having efficient and accurate samplers for simulating the posterior distribution is crucial for Bayesian analysis. We develop a generic posterior simulator called the "dynamic striated Metropolis-Hastings (DSMH)" sampler. Grounded in the Metropolis-Hastings algorithm, it pools the strengths from the equi-energy and sequential Monte Carlo samplers while avoiding the weaknesses of the standard Metropolis-Hastings algorithm and those of importance sampling. In particular, the DSMH sampler possesses the capacity to cope with extremely irregular distributions that contain winding ridges and multiple peaks; and it is robust to how the sampling procedure progresses across stages. The high-dimensional application studied in this paper provides a natural platform for testing any generic sampler.

Date: December 25, 2015.

Key words and phrases. Dynamic striation adjustments, simultaneous equations, monetary policy, inflation coefficient, winding ridges, multiple peaks, independent striated draws, irregular posterior distribution, importance weights, tempered likelihood, effective sample size.

JEL classification: C32, C63, E17.

We are grateful to three anonymous referees and the editor Helmut Herwartz for thoughtful comments, which have led to significant improvement of the paper. This research is supported in part by the National Science Foundation Grant SES 1127665 and by the National Natural Science Foundation of China Research Grants 71473168 and 71473169. Simulations are programed in C++, tailored to parallel and grid computations. The views expressed herein are those of the authors and do not necessarily reflect the views of the Federal Reserve Bank of Atlanta or the Federal Reserve System or the National Bureau of Economic Research.

I. Introduction

We develop a new posterior simulation method that allows researchers to estimate high-dimensional economic and statistical models that have irregular likelihoods with multiple peaks and complicated winding ridges. We undertake this research mainly because, in recent years, the Bayesian estimation and evaluation of multivariate dynamic models have played a central role in assessing how well the model fits to the data and in selecting the best-fit model for forecasting and for policy analysis (Geweke, 1999; Christiano, Eichenbaum, and Evans, 1999, 2005; An and Schorfheide, 2007; Smets and Wouters, 2007).

Standard Markov Chain Monte Carlo (MCMC) methods, such as the Metropolis-Hastings algorithm, work well for estimating models with likelihoods or posterior distributions that have smooth Gaussian shapes. For high-dimensional economic and statistical models, however, the likelihood or the posterior distribution can be non-Gaussian with highly irregular shapes and multiple peaks. These problems can severely compromise the accuracy of previous MCMC samplers for Bayesian inference.

To tackle such problems we develop a new posterior simulation method, called the dynamic striated Metropolis-Hastings (DSMH) sampler. It draws the strengths of two recently developed samplers: the equi-energy (EE) algorithm (Kou, Zhou, and Wong, 2006) and the sequential Monte Carlo (SMC) algorithm (Chopin, 2004; Durham and Geweke, 2012; Herbst and Schorfheide, 2014). The basic idea behind these two techniques is to start with a tractable initial distribution one can sample from and then to transform this initial distribution gradually to the desired posterior distribution through a sequence of stages. At each stage the sample from the previous stage is used to form a new sample for the current stage. The sample gathered at the final stage is the same as the sample generated from the desired posterior distribution. The key difference between these techniques are how information from the previous stage is transmitted to the current stage; the sampling quality depends crucially on this detail.

By pooling the strengths of these two samplers, our newly developed simulation method—the DSMH sampler—makes a significant contribution to the literature in several dimensions. First, it improves the EE sampler by adapting the procedure of sampling the target distribution at each stage when the sampler progresses from the previous stage to the next. This *dynamic* adjustment marks a major departure from the EE sampler and plays an indispensable role in the DSMH sampler. In theory we show that convergence holds for our dynamically adjusted sampler. In practice this dynamic feature holds the key to the DSMH sampler for avoid getting stuck in a local parameter region with highly correlated draws. Consequently the DSMH sampler is capable of exploring the entire posterior distribution.

Second, like the SMC sampler, the DSMH sampler takes advantage of parallel computing and approximates marginal data densities or marginal likelihoods as a by-product. The SMC sampler relies on importance sampling to reweight the sample of random draws (particles) at each stage. The problem inherent in importance sampling is that particles tend to collapse so that only a small fraction of the sample receives most weights. The remedy, called "the mutation step," is to use the Metropolis-Hastings algorithm to resample new particles when the importance sampler begins to collapse. By contrast, because the DSMH sampler is grounded in the Metropolis-Hastings algorithm and utilizes importance weights *only for an initial draw* at each stage, it does not suffer the degeneracy problem inherent in the SMC sampling. The adaption to maintain a sufficient number of draws at each stage enables the DSMH sampler to traverse the entire parameter space efficiently.

Third, we apply the DSMH sampler to structural vector autoregressions (SVAR) models. This application is relevant and important for several reasons. Many multivariate dynamic models such as dynamic stochastic general equilibrium (DSGE) models are closely connected to SVAR models (Ingram and Whiteman, 1994; Del Negro and Schorfheide, 2004). Understanding how the DSMH sampler works for SVAR models provides a first step toward extending application to other multivariate dynamic models. We show that an exact Gibbs sampler exists at every stage of our sampler. This powerful result allows us to obtain accurate posterior draws from the Gibbs sampler at each stage and compare this "true" distribution to the distribution simulated from the DSMH sampler. Moreover, since configuration of the tuning parameters is an important part of the DSMH sampler (as in any Monte Carlo simulation technique), the parameter values that work for our SVAR application serve as an informative benchmark for other applications in which an exact Gibbs sampler is no longer available.

It is known that the posterior distributions for reduced-form VAR models or SVAR models with recursive identification are well behaved. Thus, a successful application of the DSMH sampler to these models does not necessarily mean that the sampler is capable of exploring irregular high-dimensional distribution. To challenge our sampler as well as other relevant samplers, we use three-variable SVAR models with non-recursive identification as in Sims and Zha (2006). Moreover, we choose to work on the *unnormalized* SVAR model so as to make the posterior distribution populated with at least as many as 2^n isolated peaks, where n is the number of equations. This combination of non-recursive identification and un-normalization makes the posterior distribution incredibly complex, full of complicated ridges between peaks.

The importance of using this SVAR cannot be overestimated. As SVAR models are often used as a benchmark for a host of economic models (Christiano, Eichenbaum, and Evans, 2005), our model in this paper, a much smaller and simpler version of Sims and Zha (2006)'s model, represents the complexity that would be encountered by many other economic models. There are several advantages for using such a model as a platform to test generic samplers:

- The three variables in our model are most commonly used in macroeconomics: output gap, inflation, and the interest rate.
- Therefore, it is not an artificial model but rather an empirical model that is used for practical policy analysis and has become a workhorse for modern macroeconomics.
- The model is realistically high-dimensional in the sense that the curse of dimensionality is not too overwhelming to render infeasible the task of testing the quality of a generic sampler.
- Unlike many other economic models, the model has the posterior distribution that can be simulated independently and thus enables researchers to perform accuracy comparison across competing generic samplers.

In summary, the model's posterior distribution serves as a fair but serious platform for testing any generic Monte Carlo sampler. We apply the four generic Monte Carlo samplers to this testing model: the widely used standard random-walk Metropolis sampler, the EE sampler, the SMC sampler, and the DSMH sampler. By generating independent draws from the Gibbs sampler, we are able to evaluate and compare how well each of these samplers works against the underlying distribution. We find that the DSMH sampler outperforms the other three samplers.[1]

The rest of the paper is organized as follows. Section II develops the DSMH sampler with theoretical justifications. Section III addresses a number of practical issues that are relevant to the end user. Section IV discusses two major difficulties that lie at the heart of estimation of multivariate dynamic models. Section V presents two challenging SVAR models that put the generic DSMH sampler to the test. Both models have highly irregular posterior distributions. SectionVI offers concluding remarks.

II. The Dynamic Striated Metropolis-Hastings Sampler

In this section we give a detailed description of the DSMH sampler and discuss a general condition under which convergence holds. Because the DSMH sampler combines the strengths of both the EE and SMC samplers, we contrast it with each of these other samplers throughout the section to facilitate an understanding of our new sampler. The detailed pseudo-code for our newly developed DSMH sampler is provided in Appendix A.

II.1. **The Generic Algorithm.** Let $Y_T = (y_1, \ldots, y_T)$ denote the observable variables, where T is the total number of observations and y_t denotes an $n \times 1$ vector of variables

[1] This finding by no means implies that the DSMH sampler is always superior; rather, it shows the power of the DSMH sampler and presents a challenging criterion for other samplers to meet.

observed at time t. The likelihood function is denoted by $p(Y_T|\theta)$, where $\theta \subset \Theta \subset \mathbb{R}^m$ is a vector of parameters. Combining the likelihood and the prior probability density $\pi(\theta)$, we obtain the posterior kernel $p(Y_T|\theta)\pi(\theta)$.[2] The DSMH sampler proceeds through a series of stages, each associated with a target probability distribution on Θ. The initial stage's target distribution must be tractable, i.e. one must be able to sample independently from the distribution and be able to compute its probability density, not just its kernel. The final stage's distribution must be the posterior probability distribution. In practice, we follow Herbst and Schorfheide (2014) and Bognanni and Herbst (2014) and use the prior density function $\pi(\theta)$ as an initial distribution. The initial distribution is then gradually transformed until, at the final stage, the target is the posterior distribution.[3] At each stage one obtains a sample from the target distribution using the sample obtained from the previous stage. In theory the sample at the final stage is from the posterior distribution; in practice the DSMH sampler is designed to ensure that the sample is representative, even for complicated posterior distributions in high-dimensional spaces.

II.1.1. *Stages.* We transform the posterior distribution by tempering the likelihood. For any real number λ satisfying $0 \leq \lambda \leq 1$, define a tempered posterior kernel as

$$f_\lambda(\theta) = p(Y_T|\theta)^\lambda \, \pi(\theta).$$

To simplify notation, we omit Y_T as an argument in $f_\lambda(\theta)$ with the understanding that $f_\lambda(\theta)$ depends on the data Y_T. The value λ controls the degree of tempering.[4] When $\lambda = 1$, $f_1(\theta)$ is the posterior kernel; when $\lambda = 0$, $f_0(\theta)$ is the prior density.

To define stages we choose λ_i, for $0 \leq i \leq H$, such that $0 = \lambda_0 < \lambda_1 < \cdots < \lambda_{H-1} < \lambda_H = 1$. For $0 \leq i \leq H$, the target distribution for the i^{th} stage is $f_{\lambda_i}(\theta)$, which we use to denote both the probability kernel and the actual distribution itself. Note that the initial target distribution f_{λ_0} is the prior distribution and the final target distribution $f_{\lambda_H}(\theta)$ is the posterior kernel. Recommendations for how to choose λ_i are given in Section III.2.

[2] A probability kernel is non-negative and integrates to a finite positive number that may not be one, while a probability density is non-negative and integrates to exactly one.

[3] Although the basic idea is the same for the DSMH, EE, and SMC samplers, how this idea is implemented differs substantially across these samplers. Even within the SMC sampler, for instance, the implementation differs considerably between the method of Durham and Geweke (2012) and that of Herbst and Schorfheide (2014). Other adaptive samplers such as those proposed by Bauwens, Bos, van Dijk, and van Oest (2004) and Hoogerheide, Opschoor, and van Dijk (2012) have succeeded in sampling lower-dimensional irregular distribution and may, after suitable modifications, weather the challenge of our high-dimensional irregular distribution. While it is simply infeasible to compare all generic samplers in one paper, our high-dimensional model serves as an important benchmark for testing various other generic samplers.

[4] The EE literature refers to $1/\lambda$ as temperature and uses it to measure the degree of tempering. We prefer λ since it is this term that directly appears in all formulae.

For each stage i we obtain a sample from $f_{\lambda_i}(\theta)$, which we denote by $\left\{\theta^{(i,\ell)}\right\}_{\ell=1}^{NG}$, where NG is the total number of simulations. The sample $\left\{\theta^{(0,\ell)}\right\}_{\ell=1}^{NG}$ consists of independent random draws from the prior distribution and the sample $\left\{\theta^{(H,\ell)}\right\}_{\ell=1}^{NG}$ comes from the posterior distribution. In general, the sample $\left\{\theta^{(i,\ell)}\right\}_{\ell=1}^{NG}$ depends on the sample $\left\{\theta^{(i-1,\ell)}\right\}_{\ell=1}^{NG}$. This dependence allows the DSMH sampler to take full advantage of the sample at previous stages and the nature of this dependence marks a major departure from the SMC sampler as discussed in the following two sections.

To see the effections of tempering, consider a simple one-dimensional example in which $ay_t = \varepsilon_t$, where ε_t is a standard normal random variable and a is the parameter under consideration. The tempered likelihood is proportional to

$$|a|^{\lambda T} e^{-\frac{1}{2}\lambda T \bar{\sigma}^2 a^2},$$

where $\bar{\sigma}^2 = \sum_{t=1}^{T} y_t^2 / T$. Figure 1 plots tempered likelihoods of a for λ varying from 0.005 to 1.0 with $T = 20$ and $\bar{\sigma} = 1$.[5] This simple example embodies the two essential features of tempering. First, the most tempered likelihood is very flat (the thick solid line in the figure). In fact, as λ tends to zero, the tempered likelihood tends to one on the space in which the likelihood is finite and positive, so that the most tempered posterior kernel is close to the prior of a. Second, as λ tends to one, the tempered likelihoods gradually converge to the likelihood itself (the thick dashed line in the figure) and the peaks of the likelihood function grow from the much flatter peaks of the tempered likelihood functions.

II.1.2. *Striations.* Most of the time the DSMH sampler at the i^{th} stage functions as the standard random-walk Metropolis algorithm with the target distribution $f_{\lambda_i}(\theta)$. But occasionally a proposal draw comes from the sample at the previous stage. Random draws are accepted or rejected with an appropriate Metropolis-Hastings acceptance criterion. How to simulate random draws from the previous stage is crucial to the efficiency of the sampler. When simulating from the tempered distribution at the previous stage, we would like those draws to be similar to the current draw in terms of the level (height) of the likelihood. As a result, those draws from the previous stage are likely to be accepted, and because they are simulated *independently*, the sampler moves efficiently among the values of θ that have similar likelihood values. On the other hand, the random-walk Metropolis component of the sampler allows movements among the values of θ that have significantly different likelihood values or levels. Put differently, proposal draws from the tempered distribution at the previous stage move independently *within* the same level set while serially-correlated random-walk Metropolis proposal draws move *between* level sets.

[5] These kernels are unnormalized and thus have multiple peaks. We use unnormalized likelihoods to illustrate how the sampler handles irregular posterior kernels.

We call a "striation" the set of all values of θ that have similar likelihood values. Striations at the i^{th} stage are defined by a sequence of $M+1$ levels, denoted by $L_{i,k}$, satisfying $0 = L_{i,0} < L_{i,1} < \cdots < L_{i,M-1} < L_{i,M} = \infty$. For $1 \leq k \leq M$, the k^{th} striation is the set

$$S_{i,k} = \{\theta \in \Theta \mid L_{i,k-1} \leq p(Y_T|\theta) < L_{i,k}\}. \tag{1}$$

We choose the levels so that the probability that $\theta \in S_{i,k}$ is equal to $1/M$. This probability is with respect to the distribution at the *previous* stage. If

$$I_{i-1} = \int_{\theta \in \Theta} f_{\lambda_{i-1}}(\theta) d\theta, \tag{2}$$

the levels are chosen to satisfy

$$\frac{1}{M} = \int_{\theta \in S_{i,k}} \frac{f_{\lambda_{i-1}}(\theta)}{I_{i-1}} d\theta.$$

It is generally impossible to find analytic expressions for setting the levels. One can, however, use the sample from the previous stage to set the levels by simply choosing $L_{i,k}$ so that an equal number of draws lie in each striation. We find that this simple rule works well in practice for determining the levels.[6] There are tradeoffs in determining the number of levels. On the one hand, one would like to have striations as small as possible to allow the random-walk Metropolis algorithm to move between striations more efficiently. This argues for a larger M. On the other hand, we need the sample in each striation to be representative, which argues for a smaller M. The value of M should be set so that each striation contains a few thousand draws. In all of our examples, we set M to 50, so that the number of draws per striation was $NG/M = 4,000$.

In general, the levels chosen at stage $i-1$ differ from those chosen at stage i. This is what we mean by dynamically adjusting striations; it marks an important departure from the EE sampler developed by Kou, Zhou, and Wong (2006). In that setup, the levels are the same for all stages so that $L_{i,k} = L_k$ for all i and they suggest a geometric progression so that $L_{k+1} = \gamma L_k$ with $\gamma > 1$ being a key tuning parameter. In addition, γL_{M-1} is set to $\sup_\theta\{p(Y_T)|\theta)\pi(\theta)\}$, so that the sequence is completely determined by γ with no room for flexibility. By allowing levels to differ across stages, our approach has two substantive advantages. First, it is unnecessary to maximize the posterior density function (i.e., find $\sup_\theta\{p(Y_t|\theta)\pi(\theta)\}$), which is a hard problem in and of itself for many high-dimensional economic problems. Indeed, optimization proceeds best if one can sample first and then use sampled draws as starting points for the maximization routine. Second, because the striations are fixed by the EE sampler, we find that most of the striations contain no draws in the later stages of the sampler. This phenomenon would be reflected in the sampled

[6] In principle one could also allow an unequal number of draws to lie in each striation as long as there are enough draws in each striation.

posterior distribution that tends to be too concentrated. Such a result negates one of the main advantages of utilizing striations. Indeed, the term "dynamic" in DSHM refers to the important fact that levels are adapted to contain a sufficient probability in each striation when the sampler progresses from stage to stage. Such a *dynamic* adjustment is critical because it ensures that each striation remains fully populated so that all the information in the previous sample is efficiently exploited.

II.1.3. *Metropolis-Hastings.* We now turn to the details of the Metropolis-Hastings proposal distribution. The proposal distribution is a mixture of a Gaussian distribution and the distribution $f_{\lambda_{i-1}}(\theta)$ at the previous stage. If $\theta^* \equiv \theta^{(i,\ell)}$ is the most recent draw from the DSMH sampler at the i^{th} stage, the proposal density of θ given θ^* is

$$g_i(\theta,\theta^*) = (1-p)\phi_{\mathfrak{c}_i\Omega_i}(\theta-\theta^*) + p\chi(\theta,\theta^*)\frac{Mf_{\lambda_{i-1}}(\theta)}{I_{i-1}},$$

where $\phi_{\mathfrak{c}_i\Omega_i}(\cdot)$ is the mean-zero Gaussian probability density with variance $\mathfrak{c}_i\Omega_i$ and $\chi(\theta,\theta^*)$ is the indicator function that returns one if θ and θ^* are in the same striation and zero otherwise.[7] The mixture form of $g_i(\theta,\theta^*)$ dictates that with probability $1-p$, θ is drawn from the Gaussian distribution centered at θ^* and with probability p, θ is drawn from the distribution at the previous stage but from the same striation that contains θ^*. Draws from the previous stage that lie in a particular striation can be easily obtained by selecting, with equal probability, any of the previously obtained draws $\left\{\theta^{(i-1,\ell)}\right\}_{\ell=1}^{NG}$ that *lie in that striation.*

Kou, Zhou, and Wong (2006) propose an acceptance rule slightly different from the standard Metropolis-Hastings acceptance rule and apply it to the proposal density $g(\theta,\theta^*)$. Let θ be a random draw from the proposal distribution. When θ is sampled from the Gaussian distribution, it is accepted with probability

$$\min\left\{1, \frac{f_{\lambda_i}(\theta)}{f_{\lambda_i}(\theta^*)}\right\}; \tag{3}$$

when θ is sampled from the distribution at the previous stage, it is accepted with probability

$$\min\left\{1, \frac{f_{\lambda_i}(\theta)}{f_{\lambda_i}(\theta^*)}\frac{f_{\lambda_{i-1}}(\theta^*)}{f_{\lambda_{i-1}}(\theta)}\right\}. \tag{4}$$

If the draw is accepted, then $\theta^{(i,\ell+1)} = \theta$ and if rejected, then $\theta^{(i,\ell+1)} = \theta^*$. Kou, Zhou, and Wong (2006) show that the distribution sampled this way converges to the target distribution at each stage.

One could instead use the standard Metropolis-Hastings acceptance rule under which a draw θ from the proposal distribution is accepted with probability

$$\min\left\{1, \frac{f_{\lambda_i}(\theta)}{f_{\lambda_i}(\theta^*)}\frac{g_i(\theta^*,\theta)}{g_i(\theta,\theta^*)}\right\}. \tag{5}$$

[7] Details for the choice of Ω_i are given in Section III.1, $\mathfrak{c}_i$ in Appendix B, and p in Section III.3.

If the draw is accepted, $\theta^{(i,\ell+1)} = \theta$; if rejected, $\theta^{(i,\ell+1)} = \theta^*$. Because the integral I_{i-1} can easily be computed as the DSMH sampler proceeds,[8] the proposal density ratio $\frac{g_i(\theta^*,\theta)}{g_i(\theta,\theta^*)}$ is readily available.

The above two acceptance rules are not very different in practice for the following reason. When θ and θ^* are close to each other, which is more likely when the draw comes from the Gaussian, $\phi_{\mathfrak{c}_i\Omega_i}(\theta - \theta^*)$ is much larger than either $f_{\lambda_{i-1}}(\theta)$ or $f_{\lambda_{i-1}}(\theta^*)$. When θ and θ^* are far apart from each other, which is more likely when the draw comes from the previous stage, both $f_{\lambda_{i-1}}(\theta)$ and $f_{\lambda_{i-1}}(\theta^*)$ are much larger than $\phi_{\mathfrak{c}\Omega_i}(\theta - \theta^*)$. Thus, when θ is sampled from the Gaussian distribution, $g_i(\theta^*, \theta)/g_i(\theta, \theta^*) \approx 1$; when θ is sampled from the previous stage's distribution, $g_i(\theta^*, \theta)/g_i(\theta, \theta^*) \approx f_{\lambda_{i-1}}(\theta^*)/f_{\lambda_{i-1}}(\theta)$. We prefer the alternative acceptance rule suggested by Kou, Zhou, and Wong (2006) because it enables the DSMH sampler to remain efficient even in situations where I_{i-1} may not be well estimated. This is one of the key features that make the DSMH sampler attractive and is highlighted by the application in Section V.3.

II.2. **Theoretical Foundation.** In this section we give a proof of the DSMH sampler's convergence under a very general condition.

Condition 1. *The prior $\pi(\theta)$ is proper, so that $\int_{\theta\in\Theta} \pi(\theta)d\theta = 1$, and there exist algorithms for obtaining draws from this density.*

Condition 1 is extremely general and almost all parametric models, economic and statistical, would meet this condition. It is certainly much less restrictive than those in the SMC literature as it does not impose uniform boundedness on the posterior probability density. Condition 1 is all we need for convergence of the DSHM sampler. To prove the convergence, we show that f_λ is a probability kernel and the sampler converges to f_λ.

Proposition 1. *Under Condition 1, $\int_{\theta\in\Theta} f_\lambda(\theta)d\theta < \infty$ for $0 \le \lambda \le 1$ and almost all Y_T.*

Proof. Condition 1 and Tonelli's Theorem imply that $\int_{\theta\in\Theta} p(Y_T|\theta)\pi(\theta)d\theta < \infty$ for almost all Y_T. Let $A = \{\theta \in \Theta | p(Y_T|\theta) < 1\}$ and $B = \{\theta \in \Theta | p(Y_T|\theta) \ge 1\}$. For $0 \le \lambda \le 1$,

$$\begin{aligned}\int_{\theta\in\Theta} p(Y_T|\theta)^\lambda \pi(\theta)d\theta &= \int_{\theta\in A} p(Y_T|\theta)^\lambda \pi(\theta)d\theta + \int_{\theta\in B} p(Y_T|\theta)^\lambda \pi(\theta)d\theta \\ &\le \int_{\theta\in A} \pi(\theta)d\theta + \int_{\theta\in B} p(Y_T|\theta)\pi(\theta)d\theta.\end{aligned}$$

Since integrals of both $\pi(\theta)$ and $p(Y_T|\theta)\pi(\theta)$ are almost surely finite, the integral of $p(Y_T|\theta)^\lambda\pi(\theta)$ is also almost surely finite. So, under Condition 1, $f_\lambda(\theta)$ is a probability kernel. □

[8] In Section III we discuss techniques for estimating this quantity, which is of independent interest.

Armed with Proposition 1, the following proposition shows that Condition 1 is sufficient to ensure convergence of the DSHM sampler. By "convergence" we mean that the random sequence $\{\theta^{(i,\ell)}\}$ is ergodic and the limiting distribution exists. Of course this random sequence depends on the previous random sequences $\{\theta^{(k,\ell)}\}$ for $0 \leq k < i$. To obtain convergence of the i^{th} sequence, the length of the i^{th} sequence, as well as the lengths of all previous sequences, must increase, although the increase needs not be at the same rate.

Proposition 2. *Under Condition 1, the DSMH sampler at the i^{th} stage converges to $f_{\lambda_i}(\theta)$. In particular, at the final stage, the DSMH sampler converges to the posterior distribution.*

Proof. Theorem 2 of Kou, Zhou, and Wong (2006) assumes that the strations do not vary across stages. For the DSMH sampler, not only do the striations vary across stages, they also depend on the previous stage's sample. However, their proof of Theorem 2 will go through as long as the boundaries of the striations for the i^{th} stage, $L_{i,j}(\{\theta^{(i-1,\ell)}\}_{\ell=1}^{NG})$, converge as NG increases, which is true for the DSMH sampler. Thus Theorem 2 of Kou, Zhou, and Wong (2006) applies and gives sufficient conditions for the convergence when the acceptance rule is given by (3) and (4). In particular, the DSMH sampler at the i^{th} stage converges to $f_{\lambda_i}(\theta)$ provided that (1) it does so at the 0^{th} stage; (2) the Metropolis transition probabilities for jumping between adjacent striations is positive; (3) the probability that θ drawn from the target distribution for the i^{th} stage is in the j^{th} striation is positive for all stages and striations. By Condition 1, there exists a sampler for obtaining draws from the prior, so (1) holds. Because our Metropolis jumping kernel is Gaussian and the measure of any striation is positive, the Metropolis transition probability for jumping between any two striations is positive so that (2) holds. By construction, the probability that θ drawn from the target distribution for the i^{th} stage is in the j^{th} striation is approximately $1/M$, so that (3) holds. Thus the DSMH sampler converges at each stage.

If the acceptance rule is given by (5), then the usual theorems governing the convergence of the Metropolis-Hasting algorithm apply. In particular, if the Metropolis-Hastings transition kernel, given by $g_i(\theta, \theta^*)$, is aperiodic and irreducible with respect to $f_{\lambda_i}(\theta)$, then an induction argument similar to the one used in the proof of Theorem 2 of Kou, Zhou, and Wong (2006) implies that the DSHM draws at the i^{th} stage converge to the distribution $f_{\lambda_i}(\theta)$. See Tierney (1994), Theorem 1, for a discussion of these concepts and this result. Because the support of our proposal distribution is all of $\mathbb{R}^m$, the transition kernel is aperiodic. Because a subset of $\mathbb{R}^m$ is of positive probability with respect to $f_{\lambda_i}(\theta)$ if and only if it is of positive probability with respect to $f_{\lambda_{i-1}}(\theta)$, the transition kernel is irreducible with respect to f_{λ_i}. Thus the sampler converges. □

Propositions 1 and 2 establish the theoretical foundation of the DSMH sampler.

III. Practical Issues

In this section we discuss the tuning parameters that the end user can set and other tuning decisions that are made automatically by our implementation. Table 1 provides an overview of the tuning parameters available to the end user, the recommended settings, and the speed-reliability tradeoffs. The following sections give detailed discussions, all of which are relevant to Table 1.

III.1. **Importance Weights.**

To make the DSMH sampler operational, we must generate a starting value for each group and define $\mathfrak{c}_i\Omega_i$, the variance of the Gaussian distribution for the random-walk Metropolis proposal density. The scale $\mathfrak{c}_i$ is determined by the standard Metropolis tuning procedure. This procedure is described in Appendix B. The importance-weighted draws from the previous stage are utilized to generate the starting values and determine Ω_i.

The unnormalized importance weight of $\theta^{(i-1,\ell)}$ is $\tilde{w}_\ell^{(i)} = f_{\lambda_i}(\theta^{(i-1,\ell)})/f_{\lambda_{i-1}}(\theta^{(i-1,\ell)})$ and the normalized importance weight is $w_\ell^{(i)} = \tilde{w}_\ell^{(i)}/(\sum_{k=1}^{NG}\tilde{w}_k^{(i)})$. Resampling from $\{\theta^{(i-1,\ell)}\}_{\ell=1}^{NG}$ using the normalized importance weights as probabilities delivers a sample from the distribution $f_{\lambda_i}(\theta)$. Resampling produces a representative sample from the distribution $f_{\lambda_i}(\theta)$ if the weights are balanced; otherwise, the sample is unlikely to be reliable.

We take Ω_i to be the importance-weighted estimate of the variance of $f_{\lambda_i}(\theta)$, which is given by

$$\Omega_i = \sum_{\ell=1}^{NG} w_\ell^{(i)}(\theta^{(i-1,\ell)})(\theta^{(i-1,\ell)})' - \mu_i\mu_i', \text{ where } \mu_i = \sum_{\ell=1}^{NG} w_\ell^{(i)}\theta^{(i-1,\ell)}.$$

The starting value for each group is an independent draw from an importance-weighted sample of $\left\{\theta^{(i-1,\ell)}\right\}_{\ell=1}^{NG}$.

The effective sample size (ESS) based on importance weights is useful in determining when the weights have become unbalanced. It is defined as

$$\mathrm{ESS}^{(i,\,\mathrm{IW})} = \frac{\left(\sum_{\ell=1}^{NG}\tilde{w}_\ell^{(i)}\right)^2}{\sum_{\ell=1}^{NG}\left(\tilde{w}_\ell^{(i)}\right)^2} = \frac{1}{\sum_{\ell=1}^{NG}\left(w_\ell^{(i)}\right)^2},$$

where the superscript "IW" stands for importance weights. This value is between 1 and NG, with a larger number indicating that weights are better balanced. Importance weighting forms the basis for SMC samplers and effective sample size is its main diagnostic tool. The idea is that one starts with an independent sample from $f_{\lambda_0}(\theta)$. At each stage the sample is reweighted using the importance weights and the effective sample size is computed. The effective sample size decreases with each successive stage and would quickly collapse to 1 for most high-dimensional dynamic models. When the effective sample size relative to the total number of draws drops below a certain threshold, one resamples from the distribution using the importance weights. For instance, Herbst and Schorfheide (2014) recommend resampling when the effective sample size relative to the total number of draws drops below 50%. After resampling, however, many draws appear multiple times. For this reason, random-walk Metropolis draws are made for each resampled draw. This last step is often described as mutation and the SMC algorithm as the reweighting-resampling-mutating procedure. From this description one can see the central role importance sampling plays in the SMC algorithm.

By contrast, the DSMH sampler is grounded in the Metropolis-Hastings algorithm. Importance sampling is only used to obtain a small number of initial draws at each stage and for computing Ω_i. For this reason, DSMH is less adversely effected as the the importance weights become unbalanced. At each stage and within each striation, the previous stage's draws are independently sampled without reweighting and then accepted or rejected according to the aforementioned acceptance rule.

III.2. **Choosing λ_i.** We find that the DSMH is fairly robust to the choice of λ_i. Kou, Zhou, and Wong (2006) recommend a geometric progression for the λ_i,

$$\lambda_i = \lambda_1^{\frac{H-i}{H-1}}.$$

Given H, the only parameter to set is λ_1, the smallest non-zero λ_i. To answer the question of how small λ_1 should be chosen, consider a model of the form $g_t(y_t, \theta) = \varepsilon_t$, a common form for dynamic multivariate models, where ε_t is an $n \times 1$ vector of exogenous shocks and y_t is an $n \times 1$ vector of observables for $1 \leq t \leq T$. If the probability density of ε_t is $p_\varepsilon(\varepsilon_t, \theta)$, the likelihood function can be expressed as

$$\prod_{t=1}^{T} \left| \det \left[\frac{\partial g_t}{\partial y_t} \right] \right| p_\varepsilon(g_t(y_t), \theta).$$

Whatever complexity the possibly nonlinear function $g_t(y_t, \theta)$ might bring to our problem, the product of determinants adds another dimension of complexity: a high-order polynomial of degree nT. As illustrated in Figure 1, the most tempered likelihood function should be a diffuse mound-shaped distribution; to achieve this objective, λ_1 needs be small enough to wash away the effects of the high-order polynomial. For this reason, λ_1 should be at most $1/(nT)$ and we recommend that it be set to $1/(10nT)$.

One could choose different functional forms for λ_i. For instance, Herbst and Schorfheide (2014) recommend a power function of the form

$$\lambda_i = \left(\frac{i-1}{H-1} \right)^{\gamma},$$

with γ equal to 2 for their SMC sampler. Herbst and Schorfheide (2014) call the choice of λ_i "the tempering schedule." Both as a robustness check and to facilitate comparison with their sampler, we also report results from this tempering schedule.

Although not articulated in Kou, Zhou, and Wong (2006), the distance between λ_{i-1} and λ_i is connected to the effective sample size at each stage. To see this important point, note that

$$\mathrm{ESS}^{(i,\,\mathrm{IW})} = \frac{\left(\sum_{\ell=1}^{i,NG} p\left(Y_T \mid \theta^{(i-1,\ell)}\right)^{\lambda_i - \lambda_{i-1}}\right)^2}{\sum_{\ell=1}^{NG} p\left(Y_T \mid \theta^{(i-1,\ell)}\right)^{2(\lambda_i - \lambda_{i-1})}}.$$

One can see that that as λ_i monotonically decreases to λ_{i-1}, $\mathrm{ESS}^{(\mathrm{IW})}$ monotonically increases to NG. Thus the effective sample size relative to the number of draws is an easily computed and interpreted indicator of how close the distribution $f_{\lambda_{i-1}}(\theta)$ is to the distribution $f_{\lambda_i}(\theta)$. While the DSMH sampler is not as sensitive to a sharp decrease of the effective sample size as the SMC sampler, the two distributions $f_{\lambda_{i-1}}(\theta)$ and $f_{\lambda_i}(\theta)$ should not be too far apart. For this reason, we always recommend to compute the effective sample size relative to the total number of draws. If it gets too small (less than 10% for example), one might need to increase the number of stages or change the value of λ_1 or perform both. In our application, we have not found the need for such an adjustment.

III.3. **Thinning and Probability of Striated Proposal.** The standard Metropolis-Hastings algorithm produces serially correlated draws, especially for high-dimensional problems. For the DSMH sampler, while acceptance of a proposal draw from the sample obtained at the previous stage breaks this dependence, one could still have a long sequence of serially correlated draws, particularly if the probability of making a striated proposal is small. For this reason, we recommend that one posterior draw be saved for every $\mathcal{T}$ posterior draws made at each stage. We call $\mathcal{T}$ "the thinning factor." To save NG draws, therefore, one must simulate approximately $\mathcal{T}NG$ posterior draws. Thus, the thinning factor directly effects the run time of the DSMH algorithm. Double the thinning factor will approximately double the run time of the algorithm. The thinning factor $\mathcal{T}$ is chosen by the user, but we recommend that $\mathcal{T}$ be set to 50. This value works well for the examples considered in this paper, which are representative of many dynamic macroeconomic models. One can always compute the serial correlation of adjacent saved draws to gauge whether this value is too high or too low.

The probability of making a striated proposal draws, p, is related to $\mathcal{T}$ in two ways. The random-walk Metropolis algorithm needs time to explore the distribution locally before a striated proposal draw is accepted and the sampler moves to another region of the parameter space. Since we save only a portion of these draws, it must be that $p < 1/\mathcal{T}$. And since striated proposal draws come from the sample obtained at the previous stage, we need to avoid over-sampling from the previous tempered distribution. If we set $p = 0.1/\mathcal{T}$, the number of striated proposals is equal to 10% of the total number of draws simulated at the previous stage. Heuristically this value is reasonable enough to safeguard against over-sampling.

III.4. **Parallelism.** In this section we discuss how parallelism is used for the DSMH sampler and how it is related to the other literature.

III.4.1. *Central Processing Units.* Central processing units (CPUs) have been traditionally used for scientific computing. Most computers have multiple cores, each of which is essentially an independent processing unit.[9] Inexpensive desktops often have 4 or 8 cores, and high-end workstations could easily have 24 or 32 cores. High-performance cluster machines often have cores numbering in the hundreds.

To effectively use the multiple-core CPU technology, it is best to divide the algorithm into computational blocks so that each block requires no interaction with other blocks. The DSMH sampler is designed to utilize parallelism efficiently through such blocks, which we call "groups." Within each stage, we have G independent groups and simulate N draws for each group. While each group uses the same set of draws from the previous stage, the computation within each group is completely independent of the other groups at the same stage. If each group ran on its own core, then all G groups would finish the stage in approximately the same amount of time as a single group would take to finish on a single core unit. We expect a G-fold improvement by using G cores. Because there is overhead computing time between stages, such as consolidating the draws from each group and sorting them into striations, the improvement is not exactly G-fold but close to it. Some of the between-stage overhead computation could be parallelized, but we have chosen not to do so because the efficiency gain is relatively small.

Because we recommend adjusting N so that the total number of draws remains the same, the difference between a single group and multiple groups is simply the number of starting values used. Starting values for each group are chosen from the importance-weighted sample at the previous stage. For this reason, we recommend that G be set to the number of cores available. To avoid oversampling at the previous stage, N should be at least 100 so that no more than 1% of the draws from the previous stage are used as starting values. If one runs the DSMH algorithm on an inexpensive desktop computer, where there may be only 4 or 8 cores, we recommend that G be set to a multiple of the number of cores available and at least 20 to mitigate the effects of an unfortunate draw of the starting value that may be in a low-probability region of the parameter space. At each stage a total of NG draws are stored

[9] While it is useful to think of each core as a completely independent processing unit, there are usually shared resources. For instance, multiple cores on the same chip must allocate the memory each core needs from a common pool. Hyperthreading is a technology that allows two cores to share computing resources in addition to the common pool of memory. In theory, if an algorithm is able to fully utilize multiple cores, doubling the number of cores available would halve the run time of the program. In practice, the shared resources reduce the efficiency gain slightly.

for use in the next stage. Because it is the total number of draws that matters, G is tailored to the computing environment while N is adjusted to target NG at the desired level.

III.4.2. *Graphical Processing Units.* The use of graphical processing units (GPUs) is increasing as their price comes down. Moreover, tools for programming GPU applications have become more sophisticated so that it is much easier to compile the existing C, C++, or FORTRAN code on the GPUs. For instance, see Aldrich, Fernández-Villaverde, Gallant, and Rubio-Ramírez (2011) and Durham and Geweke (2012) for examples of how GPUs can be exploited for economic models. The DSMH algorithm is suitable for running in the GPU environment, though there are issues one needs take into account. GPUs are most efficient when the same sequence of instructions is executed over a set of data points. This is certainly the case for the DSMH sampler but care needs be taken about branch points (if-then-else statements). In this architecture, while one core executes an if-then statement, the other cores that execute the if-else statements wait. Conversely, when the other cores execute the if-else statements, the core that executes the if-then statement sits idle. This process functions as if each branch of the if-then-else statement were executed by every core. It is for this reason that an algorithm containing many branch points is less suitable for running on the GPUs. The DSMH sampler for each group contains only two branch points: one branch point is to make either a Metropolis-Hastings proposal draw or a striated proposal draw and in each case the other branch point is to accept or reject the proposal. Furthermore, the steps in each of the branches are algebraic operations or standard function evaluations on the previously computed values except that the likelihood and prior density must be computed if a Metropolis-Hastings proposal draw is made. Since expensive computations for most dynamic models are the evaluation of the likelihood, the code in the pair of branch points is executed in approximately the same amount of time as a single evaluation of the likelihood. If the evaluation of the likelihood is amenable to the GPU environment as shown by Aldrich, Fernández-Villaverde, Gallant, and Rubio-Ramírez (2011) and Durham and Geweke (2012), so is the DSHM sampler.

III.5. **Marginal Likelihood.** The marginal likelihood, often called the marginal data density (MDD) in the macroeconomics literature, is

$$p(Y_T) = \int_{\Theta} p(Y_T \mid \theta)\pi(\theta)d\theta. \tag{6}$$

Computing the MDD is necessary for calculating the Bayes factor or the posterior odds ratio when preforming model comparison. Estimation of the MDD is a by-product of the SMC sampler and can be obtained with no extra computational costs.[10] Such a by-product is

[10] For the EE sampler of Kou, Zhou, and Wong (2006), there is no discussion of how to estimate the MDD.

also true of the DSMH sampler. Since $f_0(\theta) = \pi(\theta)$ is a proper probability density under Condition 1, $I_0 = \int_{\theta\in\Theta} f_0(\theta)d\theta = 1$. For $1 \le i \le H$,

$$I_i = I_{i-1}\int_{\theta\in\Theta}\frac{f_{\lambda_i}(\theta)}{f_{\lambda_{i-1}}(\theta)}\frac{f_{\lambda_{i-1}}(\theta)}{I_{i-1}}d\theta.$$

Thus if $\hat{I}_{i-1}$ is an estimate of I_{i-1}, then I_i can be estimated from the sample $\{\theta^{i-1,\ell}\}_{\ell=1}^{NG}$ by

$$\hat{I}_i = \frac{\hat{I}_{i-1}}{NG}\sum_{\ell=1}^{NG}\frac{f_{\lambda_i}(\theta^{i-1,\ell})}{f_{\lambda_{i-1}}(\theta^{i-1,\ell})} = \frac{\hat{I}_{i-1}}{NG}\sum_{\ell=1}^{NG}\tilde{w}_\ell^{(i)}. \tag{7}$$

From (2) and (6) one can see that $I_H = p(Y_T)$. Hence the MDD can be approximated by $\hat{I}_H$. The estimates $\hat{I}_i$ are extremely fast to compute, but are inaccurate if the weights become unbalanced. As discussed at the end of Section II.1.3, the simulated sample can still be representative even if the importance weights are unbalanced. In Section V.4 we show that the DSMH sampler is robust to relatively unbalanced weights as well as different tempering schedules.

IV. Specific Difficulties for High-Dimensional Models

Consider economic or statistical models of the general form

$$Y_T = \mathcal{M}(\mathcal{E}_T;\theta,\psi), \tag{8}$$

where $\mathcal{E}_T = (\varepsilon_1,\cdots,\varepsilon_T)$ are unobserved exogenous shocks and ψ is a vector of nuisance parameters (e.g., unobserved regimes in Markov-switching models). This general form includes VAR and DSGE models as a special case. For illustrative purposes, consider $\mathcal{M}(\cdot)$ in the following parametric form:

$$A(\theta,\psi)Y_T = c(\theta,\psi) + \mathcal{E}_T, \tag{9}$$

where $A(\cdot)$ is an $nT \times nT$ matrix and $c(\cdot)$ is an nT-dimensional vector. Note that any linear state-space model can be expressed in the form of (9), where $A(\cdot)$ and $c(\cdot)$ are often complicated non-linear functions of θ and ψ. We assume, in our application, that y_t and ε_t have the same dimension of n, and $\mathcal{E}_T$ has the standard Gaussian distribution.[11] The likelihood function for model (9) becomes

$$p(Y_T|\theta,\psi) = |\det A(\theta,\psi)|\exp\left(-\frac{1}{2}\left(A(\theta,\psi)Y_T - c(\theta,\psi)\right)'\left(A(\theta,\psi)Y_T - c(\theta,\psi)\right)\right). \tag{10}$$

The posterior kernel is $p(Y_T|\theta,\psi)\pi(\theta,\psi)$, where $\pi(\theta,\psi)$ is the prior probability density. From expression (10) one sees two major difficulties for estimating this kind of models:

(1) The determinant function is a multivariate polynomial of degree nT.

(2) $A(\cdot)$ and $c(\cdot)$ may be complicated nonlinear functions of the parameters.

[11] All these special assumptions can be relaxed, for the DSMH sampler applies to model (8) as long as the prior density meets Condition 1.

To deal with the first difficulty, we consider SVAR models in which there are no nuisance parameters, both $A(\theta)$ and $c(\theta)$ are linear in θ, and the exponent in the exponential term of (10) is quadratic. If the determinant term were not present in (10), the likelihood function would be a Gaussian probability density. The determinant term, however, induces multiple peaks and complicated ridges into the likelihood function as well as the posterior kernel. Because the determinant term is prevalent in multivariate dynamic models (including DSGE models), SVAR models provide a natural benchmark for testing the DSMH simulator.

For SVAR models, Waggoner and Zha (2003a) develop an efficient Gibbs sampler. Since it is the exact Gibbs sampler, one can apply the method developed by Chib (1995) to accurately compute the MDD (see also Fuentes-Albero and Melosi (2013)). For recursive SVAR models, the Gibbs sampler produces independent draws. If the identification is non-recursive, the draws are serially correlated but convergence is so rapid that it is feasible to draw a large number of starting values independently and then apply the Gibbs sampler to obtain independent draws. Thus we can use the posterior draws generated by the Gibbs sampler as the "truth" to gauge the accuracy of the DSMH sampler and help develop diagnostic tools for more general models.

One class of more general models we study in this paper are Markov-switching SVARs (MSSVARs) proposed by Sims and Zha (2006). This class involves nuisance parameters, namely the hidden Markov states. In addition to the first difficulty discussed above, we now encounter the second difficulty: $A(\cdot)$ and $c(\cdot)$ are much more complicated functions of the underlying parameters. Because of this additional difficulty, posterior simulations become even more challenging. Sims, Waggoner, and Zha (2008) use the Metropolis-within-Gibbs algorithm to make posterior draws, but in general any specific Gibbs design depends on a particular model specification and is prone to analytical and programming errors when the model specification changes. The DSMH sampler is generic. In Section V.5 we use the turning parameters and the diagnostic tool gained from our experiments with the benchmark SVAR model to show how the DSMH sampler works for this complicated example.

V. Application

In this section we present two simultaneous-equation high-dimensional models for the purpose of testing the DSMH sampler: an SVAR model and a Markov-switching SVAR model. While both models pose significant challenges for any generic sampler, our main focus is on the three-variable simultaneous-equation monthly model without Markov-switching parameters.

V.1. **Benchmark SVAR.** Structural vector autoregressions have the following representation:

$$y_t' A_0 = C' + \sum_{h=1}^{l} y_{t-h}' A_h + \varepsilon_t', \text{ for } 1 \le t \le T, \tag{11}$$

where

- l is the lag length;
- ε_t is an n-dimensional column vector of unobserved random i.i.d. standard Gaussian shocks at time t;
- A_0 is an invertible $n \times n$ matrix and A_h is an $n \times n$ matrix for $1 \le h \le l$;
- C is an $n \times 1$ vector of constant terms.

The initial conditions $y_0, \cdots, y_{1-l}$ are taken as given. In our notation, the parameter vector θ is the collection of all the parameters in model (11). The prior distribution takes the form suggested by Sims and Zha (1998), which expands on the original Minnesota prior of Litterman (1986). The likelihood function is proportional to

$$|A_0|^T \prod_{k=1}^{n} \exp\left(-\frac{T}{2}\theta'\Sigma\theta\right), \tag{12}$$

where Σ is a symmetric and positive definite matrix that depends on the data. For any exclusion restrictions placed on some of the parameters, Waggoner and Zha (2003a) show that the likelihood form (12) remains the same but θ is composed of only the parameters that are not excluded. With Sims and Zha (1998)'s prior, the posterior kernel takes the form (12) as well. Moreover, that paper derives a Gibbs sampler for any posterior kernel of form (12) and shows that the Gibbs sampler is efficient and the sampled distribution converges very rapidly.

If we raise the expression (12) to any positive power, the function form remains the same as (12). Thus, the Gibbs sampler of Waggoner and Zha (2003a) applies to $f_\lambda(\theta)$ for $0 \le \lambda \le 1$. We formalize this result in the following proposition.

Proposition 3. *For model* (11) *with exclusion restrictions and the prior of Sims and Zha (1998), there exists a Gibbs sampler for* $f_\lambda(\theta)$ *for* $0 \le \lambda \le 1$.

This result is powerful because it enables researchers to gauge how well a particular sampler performs at each stage indexed by λ.

V.2. **Monthly Empirical Model.** To show how the DSMH sampler handles the first difficulty highlighted in Section IV, we apply the DSMH sampler to a three-variable monthly SVAR model with thirteen lags.[12] The three variables are those commonly used by monetary

[12] We follow Sims and Zha (2006) and use thirteen lags to remove possible residual seasonality, even though the data we use are seasonally adjusted. Our results, however, do not hinge on whether we use twelve lags

DSGE models: log output gap (x_t), GDP-deflator inflation (π_t), and the federal funds rate (R_t). The U.S. data are monthly from 1988:1 to 2014:6, covering the post-Volcker period of U.S. history. Output gap is measured by the difference between actual real GDP and potential real GDP published by the Congressional Budget Office. Both actual and potential GDP series as well as GDP deflator are interpolated to monthly frequency using the methodology suggested by Leeper, Sims, and Zha (1996) and Bernanke, Gertler, and Watson (1997). Federal funds rates are monthly average effective rates and annualized.

A majority of applications in the SVAR literature concern restrictions imposed on A_0 and we follow this approach in our application. The identification for the three-variable SVAR follows Sims and Zha (2006) and is summarized as

$$A_0 = \begin{bmatrix} a_{0,11} & a_{0,12} & 0 \\ a_{0,21} & a_{0,22} & 0 \\ a_{0,31} & 0 & a_{0,33} \end{bmatrix}, \tag{13}$$

where columns represent equations. The identification (13) is non-recursive but the model is *globally identified*, see Rubio-Ramírez, Waggoner, and Zha (2010). The identifying restrictions are consistent with new-Keynesian models but with fewer restrictions than the stylized model of Rudebusch and Svensson (1999) to maintain the fit of the model. The first equation (the first column) characterizes the aggregate demand behavior in which output gap responds to both inflation and the interest rate.[13] The second equation (the second column) is consistent with the Phillips-curve relationship in which inflation reacts to output gap. The last equation (the third column) characterizes the monetary policy behavior that responds to output gap and inflation with only one-month delay (this assumption is reasonable for monthly data because the monetary authority has no information about GDP and its price deflator within the month). The hyperparmaters for the prior, in the notation of Sims and Zha (1998), is $\lambda_1 = 0.7$, $\lambda_2 = 0.5$, $\lambda_3 = 0.1$, $\lambda_4 = 1.2$, $\mu_5 = 1.0$, and $\mu_6 = 1.0$. Our empirical results are not sensitive to this prior setting.

There are substantive reasons we choose model (11) to test the DSMH sampler for high-dimensional problems. First, SVARs have served as a benchmark for other multivariate dynamic models such as DSGE models. Second, model (11) with a long lag length or a large number of variables presents a challenging high-dimensional problem. Even for our "small-scale" three-variable SVAR model, the number of parameters is well over a hundred,

or thirteen lags. We simply use the conventional setup in the empirical VAR literature to show how various generic samplers can handle a practical example that is used for empirical policy analysis and has common dynamic features shared by a host of high-dimensional economic models.

[13] One could make a further restriction such that the coefficient of inflation and that of the interest rate have the same magnitude but with opposite signs. This restriction means that output gap responds to the current (realized) real interest rate.

126 to be exact. Third, the posterior involves a term of the form $|\det A_0|^T$. This is a polynomial of degree nT, which for our monthly SVAR model is a polynomial of degree 915! Since our identification is non-recursive, this high degree means that the likelihood function contains many complicated winding ridges. Fourth, because we chose to work with the unnormalized posterior kernel, there are 8 peaks for our model.[14] This combination of not imposing a normalization together with a identification scheme that involves simultaneity makes the likelihood function as well as the posterior kernel unusually complicated. For all these reasons, our SVAR model provides a natural platform for testing generic samplers.

According to Proposition 3, one is able to generate posterior draws very efficiently at each stage i using the Gibbs sampler. If identification is recursive, the Gibbs sampler produces independent draws. If the identification is non-recursive as in our case, the random draws are serially correlated but, as shown in Waggoner and Zha (2003a), convergence is so rapid that it is feasible to draw independently a large number of starting values and apply the Gibbs sampler to obtain independent draws. Furthermore, one can apply the method developed by Chib (1995) to accurately compute the MDD. Thus we use the posterior draws generated by the Gibbs sampler as the "truth" to gauge the quality of the DSMH sampler and offer an invaluable tool for improving the efficiency of this sampler by selecting appropriate values of the tuning parameters.

V.3. **Estimation Results.** The tuning parameters for the DSMH sampler are set as $N = 2000$, $G = 100$, $H = 50$, $M = 50$, and $\mathcal{T} = 50$ (see Table 1 for further reference). In the multiuser environment of our high performance cluster, it is difficult to get meaningful timing results since we have less control over the total load on the machine at any given time. On our desktop workstation, a dual processor machine with a total of 24 hyperthreading cores, it took a little less than 2 hours to complete a run with the above settings when we utilized 20 (out of 24) cores for our computations. The total number of simulations across all stages was 500,000,000. As a comparison, we apply the standard random-walk Metropolis sampler with the same value of N, G, and $\mathcal{T}$. We also used the SMC sampler of Herbst and Schorfheide (2014) (or Bognanni and Herbst (2014)) and the EE sampler of Kou, Zhou, and Wong (2006). The number of posterior draws for the EE sampler and the SMC sampler is configured to make it compatible with the DSMH sampler for the same given amount of computing time. Specifically, for the SMC sampler of Herbst and Schorfheide (2014) the number of stages is 50

[14] Given the likelihood form (12) and the symmetric prior, changing the sign of an equation in model (11) does not change the posterior density value. Since there are 2^n possible ways to change signs, there are 2^n peaks in any unnormalized SVAR model. For detailed discussions, see Waggoner and Zha (2003b) and Hamilton, Waggoner, and Zha (2007). As a scientific-reporting procedure, it is best to store the unnormalized posterior draws. If a researcher chooses a particular normalization rule for specific purposes, the normalization can be applied to the stored unnormalized draws without additional computational costs.

and at each stage 200,000 draws are saved. In the mutation step, 50 random-walk Metropolis draws are made for each of the 200,000 resampled starting values and the last Metropolis draw is saved. Bognanni and Herbst (2014) recommend many more stages, but only one random-walk Metropolis draw for each of the resampled starting values. To keep the total number of simulations the same, we have used 2,500 stages (far more than 50 stages) and saved 200,000 draws per stage. We do not report the results of this simulation exercise as they are similar to those obtained by the SMC sampler of Herbst and Schorfheide (2014). For the EE sampler of Kou, Zhou, and Wong (2006), the settings are the same as in the DSMH sampler. Again, with all the four samplers together, a total number of simulations across all stages was 500,000,000. For each stage, we use 200,000 independent draws through the Gibbs algorithm as the benchmark for comparison. For both the SMC and DSMH algorithms, we use the two existing tempering schedules: geometric and quadratic.

In our application, there are four parameters related to the *simultaneous relationship* between output gap and inflation: $a_{0,11}$, $a_{0,21}$, $a_{0,12}$, and $a_{0,22}$. For a clear illustration, we concentrate on the posterior distribution of these four parameters and two additional lagged parameters $a_{7,11}$ and $a_{7,21}$.[15] Because it is impossible to display the six-dimensional distribution, we display a two-dimensional distribution at a time. Although winding ridges and multiple peaks in a higher-dimensional parameter space are much worse than what a set of two-dimensional graphs can reveal, these graphs nonetheless give us a glimpse of the distributional complexity we deal with.

Figure 2 displays the two-dimensional marginal posterior distribution of $a_{0,11}$ and $a_{0,22}$, formed from the posterior draws. By “marginal” we mean the joint posterior probability of $a_{0,11}$ and $a_{0,22}$ after integrating out all other parameters. The top panel of Figure 2, used as the “truth,” displays the four local peaks connected by two shallow crossing ridges. The bottom panel confirms that the straight random-walk Metropolis sampler fail to sample the distribution with multiple peaks. The second panel shows that the SMC sampler is able to trace out all the peaks, but assigns two much probability to shallow cross. As a result, the height of the four peaks is lower than that displayed in the top panel. The DSMH sampler, by contrast, is able to trace out the four peaks and the cross proportionately (the third panel) so that the height is close to that displayed in the top panel.

The inability of the SMC sampler to meet the height of the distribution is manifested in Figure 3 displaying the joint posterior distribution of $a_{0,12}$ and $a_{0,22}$. The shark-mouth shape of the distribution displayed by both the Gibbs sampler and the DSMH sampler is clearly misrepresented by the ring shape generated by the SMC sampler. The straight random-walk Metropolis sampler fares much worse by missing not only the other local peak

[15] According to the form (11), $a_{7,11}$ is the coefficient of the first variable in the first equation at the seventh lag and $a_{7,21}$ is the coefficient of the second variable in the second equation at the seventh lag.

but also a good portion of winding ridges around the local peak covered by the Metropolis draws. The reason that the straight random-walk Metropolis sampler cannot even cover the region near the peak is due to the higher-dimensional problem not revealed by the multiple two-dimensional distributions.

Figure 4, displaying the joint distribution of $a_{7,11}$ and $a_{7,21}$, presents another example of the multimodal distribution missed by the SMC sampler, which generates a single-mound distribution instead. To be sure, while the straight random-walk Metropolis sampler fails in almost all dimensions, the SMC sampler performs remarkably well in many other dimensions, which we do not report. But the multiple two-dimensional examples reported here indicate that this three-variable empirical model challenges the SMC sampler severely. If one were able to display a six-dimensional probability distribution of $a_{0,11}$, $a_{0,12}$, $a_{0,21}$, $a_{0,22}$, $a_{7,11}$, and $a_{7,21}$, the picture would look much more complicated than each of the three figures. The joint probability distribution of all 126 parameters would be beyond our visualization and imagination. From all three figures one can see that the DSMH sampler is capable of generating the posterior draws representative of the underlying irregular posterior distribution.

There is another important lesson learned from this experiment: the inability to trace out the unnormalized posterior distribution can affect estimation of the normalized parameters. Take the monetary policy equation (the third equation) as an example. Once this equation is normalized, the interest rate (the lefthand variable) responds to lagged interest rates, output gap, and inflation (the righthand variables). As proposed by Sims and Zha (2006), the average long-run coefficient of inflation in this interest rate rule, denoted by $\bar{b}_\pi$, can be estimated as

$$\bar{b}_\pi = \frac{E\left(\sum_{h=1}^{l} a_{h,23}/a_{0,33}\right)}{1 - E\left(\sum_{h=1}^{l} a_{h,33}/a_{0,33}\right)}, \tag{14}$$

where E stands for the mathematical expectation with respect to the posterior distribution. Following Herbst and Schorfheide (2014), we compute the numerical standard error (NSE) for $\bar{b}_\pi$ by running the generic sampler repeatedly using a different random seed. The estimates and the corresponding NSEs of $\bar{b}_\pi$ by different samplers are reported below.

$\bar{b}_\pi$	DSMH	DSMH	SMC	SMC	EE	Truth
	(geometric)	(quadratic)	(geometric)	(quadratic)		(Gibbs)
Estimate	1.195	1.182	0.881	1.093	1.183	1.187
NSE	0.0161	0.0136	0.0172	0.00406	0.0176	0.00203

The NSEs for these generic samplers are all very small. By this measure, the inflation coefficient in the monetary policy equation is well estimated. These small NSEs, however, fail to indicate the bias. The value of $\bar{b}_\pi$ is above one according to the Gibbs sampler, implying that monetary policy is hawkish by reacting to inflation strongly. The estimate from The DSMH sampler is very close to the "truth" from both the geometric and quadratic tempering schedules, especially when one takes account of the NSEs. The EE sampler fares surprising well in this dimension. But the values estimated by the SMC sampler with both tempering schedules are downward biased with tight NSEs. The SMC sampler with the geometric tempering schedule delivers the value that is qualitatively biased, giving an inaccurate conclusion that monetary policy reacts to inflation dovishly.

To be sure, the SMC sampler in the literature does not use the geometric tempering schedule. But the point of the above results is to illustrate that the DSMH algorithm is robust to tempering scheduling. This is a very important property because, when the truth is unknown, we are also uncertain of which tempering schedule is best for a particular problem. The SMC sampler's sensitivity to tempering scheduling is further illustrated in the following section where we discuss another important *normalized* object: the marginal data density.

V.4. **Marginal Likelihood and Effective Sample Size.** As discussed in Section III.5, the MDD or the marginal likelihood is a by-product of the DSMH sampler. For the SMC sampler, the quality of this by-product depends on the quality of importance weights mainly because the sampler is grounded in importance sampling. Grounded in the Metropolis-Hastings algorithm, the DSMH sampler by contrast is much less sensitive to the quality of importance weights. Table 2 records the estimated integral constant according to (7), the NSE, and the ESS at each stage, along with the estimated integral constant based on independent posterior draws simulated through the algorithm of Chib (1995). The two generic samplers are considered: the DSMH and SMC algorithms. The numerical standard error for the estimated integral constant from importance weights is calculated as

$$\text{NSE} \equiv \text{NSE}\left(\log \hat{I}_i\right) = \sqrt{\frac{1}{G}\sum_{j=1}^{G}\left[\log \hat{I}_i^{(j)} - \frac{1}{G}\sum_{k=1}^{G}\log \hat{I}_i^{(j)}\right]^2},$$

where the superscript (j) stands for the j^{th} group of posterior draws. We do not report the NSEs for the independent draws based on Gibbs simulations because the errors are all within second decimal points.

As one can see, the estimates of log integral constants by the SMC sampler are biased by large margin. The NSEs do not detect such a bias because they are *unrealistically* small relative to the actual error of the estimated integral constant from importance weights. If we had not known the accurate estimate of the integral constant in the "Truth" column, we would not have had the sense of the magnitude of the error. Thus, the NSE does not measure how biased the MDD estimate is.

The ESSs do indicate that the importance weights become increasingly unbalanced for the SMC sampler, especially toward the later stages. The ESSs deteriorates drastically for the DSMH sampler too, but the sampler is more robust to this deterioration for the reasons articulated in previous sections. Even when the ESSs are adequate for the SMC in earlier stages, the estimate of the log integral constant has begun to show a bias. The ESSs are not the whole story. Table 3 reports the same objects with the quadratic tempering schedule favored by the SMC literature. For the quadratic tempering schedule, the ESS increases as the stage increases, but starts out at very low levels, often less than 10%. This is one of the reasons the estimate bias from the SMC sampler is still very severe, suggesting that one needs increase the number of stages especially at the beginning. If the number of stages increases, the number of random-walk Metropolis draws would have to decrease so as to have a fair comparison with the DSMH sampler. This is indeed what Bognanni and Herbst (2014) advocate. We find, however, that the results generated by following their recommendation are very similar.

The following table tabulates the estimate of log MDD at the final stage for various samplers.

log MDD	DSMH	DSMH	SMC	SMC	EE	Truth
	geometric	quadratic	geometric	quadratic		Gibbs
Estimate	4422.17	4422.10	4325.3	4348.65	4465.77	4422.00
NSE	0.19	0.03	4.63	0.49	0.31	0.03

As one can see, the EE sampler fares even worse and the bias of the estimate is the worse among the competing algorithms.

V.5. **Markov-switching SVARs.** Because the DSMH sampler is generic, choosing appropriate values of its tuning parameters that work for high-dimensional problems is challenging. The values that work well for the constant SVAR model serve as a very useful benchmark when one applies the DSMH sampler to other high-dimensional problems. In this section we apply it to a Markov-switching SVAR model in the form of

$$y_t' A_0 = C + \sum_{h=1}^{l} y_{t-h}' A_h + \varepsilon_t' \Xi_{s_t}^{-1}, \text{ for } 1 \leq t \leq T, \tag{15}$$

where $\Xi_{s_t}^{-1}$ is a diagonal matrix with the diagonal elements depending on a Markov process represented by s_t with the $\kappa \times \kappa$ transition matrix $Q = [q_{i,j}]$ such that $q_{i,j} = \text{Prob}(s_t = i | s_{t-1} = j)$ for $i, j = 1, \ldots, \kappa$. Markov-switching SVAR models have been effective in addressing relevant issues related to the 2008 financial crisis (Hubrich and Tetlow, Forthcoming). Time-varying volatility such as changing uncertainty has been a prominent feature in the data (Cogley and Sargent, 2005; Justiniano and Primiceri, 2008; Bloom, 2009; Fernández-Villaverde, Guerrón-Quintana, Rubio-Ramírez, and Uribe, 2011). Sims, Waggoner, and Zha

(2008) show that one tractable way to model time-varying volatility is to allow shock variances to follow a Markov process. If the identifying restrictions on A_0 are non-recursive, we no longer have the exact Gibbs sampler to generate independent draws.

To put the DSMH sampler to the test, we make the problem unusually demanding by estimating again the *unnormalized* version of model (15) with a two-state Markov process for s_t. The nonlinearity increases the complexity of the problem considerably by expanding the magnitude of the first difficulty and introducing the second difficulty discussed in Section IV. The tuning parameters are set as in the previous section: $N = 2000$, $G = 100$, $H = 50$, $M = 50$, and $\mathcal{T} = 50$. Using 20 (out of 24) cores of our workstation, a run with the above tuning parameters took about 10 hours to complete, five times as long as for the example without Markov-switching. Most of this additional computing time is spent in computing the likelihood function. With Markov-switching, evaluation of the likelihood function requires the Hamilton filter (Hamilton, 1989), which must loop through all the data for each evaluation. With monthly data, this is an expensive computation.

Table 4 reports log values of the integral constants ($\log \hat{I}_i$) estimated by the Mueller method described in Liu, Waggoner, and Zha (2011), those estimated by importance weights, the NSEs, and the EESs at various stages.[16] In comparison to Tables 2 and 3, there is no "Truth" column because the exact Gibbs sampler is unavailable for this simultaneous-equation Markov-switching SVAR model.

There are, however, several reasons for our confidence in the estimates reported in Table 4. First, the Mueller method serves as cross-verification of the quality of the estimated MDD through updated importance weights. The estimates by these two methods are very close. Second, in this Markov-switching case, the Mueller method and the bridge-sampling algorithm (Meng and Wong, 1996) deliver essentially the same MDD estimate. Third, the DSMH sampler with the geometric schedule gives the estimate 4496.34, which is again very close to the other estimates.

To summarize, we use the two three-variable SVAR models studied in this paper to demonstrate two essential qualities of the DSMH sampler. First, the DSMH sampler has a remarkable capacity to trace out entire complicated distributions in the high-dimensional parameter space. Second, by combining the strengths of the SMC and equi-energy samplers, the DSMH sampler is able to achieve computational efficiency for accurate statistical inferences within a feasible computing time frame.

[16] We use Sims, Waggoner, and Zha (2008)'s elliptic probability density as a proposal density for the Mueller method. Other efficient methods, including the bridge-sampling method (Meng and Wong, 1996), give almost identical results.

VI. Conclusion

We have shown that the DSMH sampler developed in this paper is capable of simulating incredibly irregular posterior distributions full of complicated ridges connecting multiple peaks in the high-dimensional parameter space. We have intentionally set the bar high by estimating two *unnormalized* monthly SVAR models with simultaneous equations and over a hundred parameters. To illustrate how hard the problem is, we have displayed part of the complexity inherent in this high-dimensional posterior distribution. The generic DSMH sampler has proven to be a remarkably efficient posterior simulator dealing with such complexity.

As common in any posterior simulator, technical details such as the appropriate range of values for tuning parameters become indispensable. This task is challenging due to the nature of high dimension and unusual irregularity imbedded in the posterior distribution. The values of tuning parameters that work for our benchmark SVAR model provide a benchmark for estimating other dynamic structural models for which the exact Gibbs sampler may not be available. Indeed, we have applied the DSMH sampler to a Markov-switching SVAR model and shown that the sampler remains efficient.

We exclusively focus on SVAR models not only because they serve as a benchmark for other multivariate dynamic models but also because they provide a natural experiment for testing any simulator against the "truth" (the independent draws generated by the Gibbs sampler). If the simulator fails to trace out the labyrinthine shape of the posterior distribution as graphically displayed in the paper, it sends a strong signal about its capability of estimating other multivariate dynamic models when there is no known "truth" about the underlying posterior distribution. It is our hope that the DSMH sampler, thoroughly tested against high-dimensional SVAR models, will prove to be as powerful in other applications as in our application.

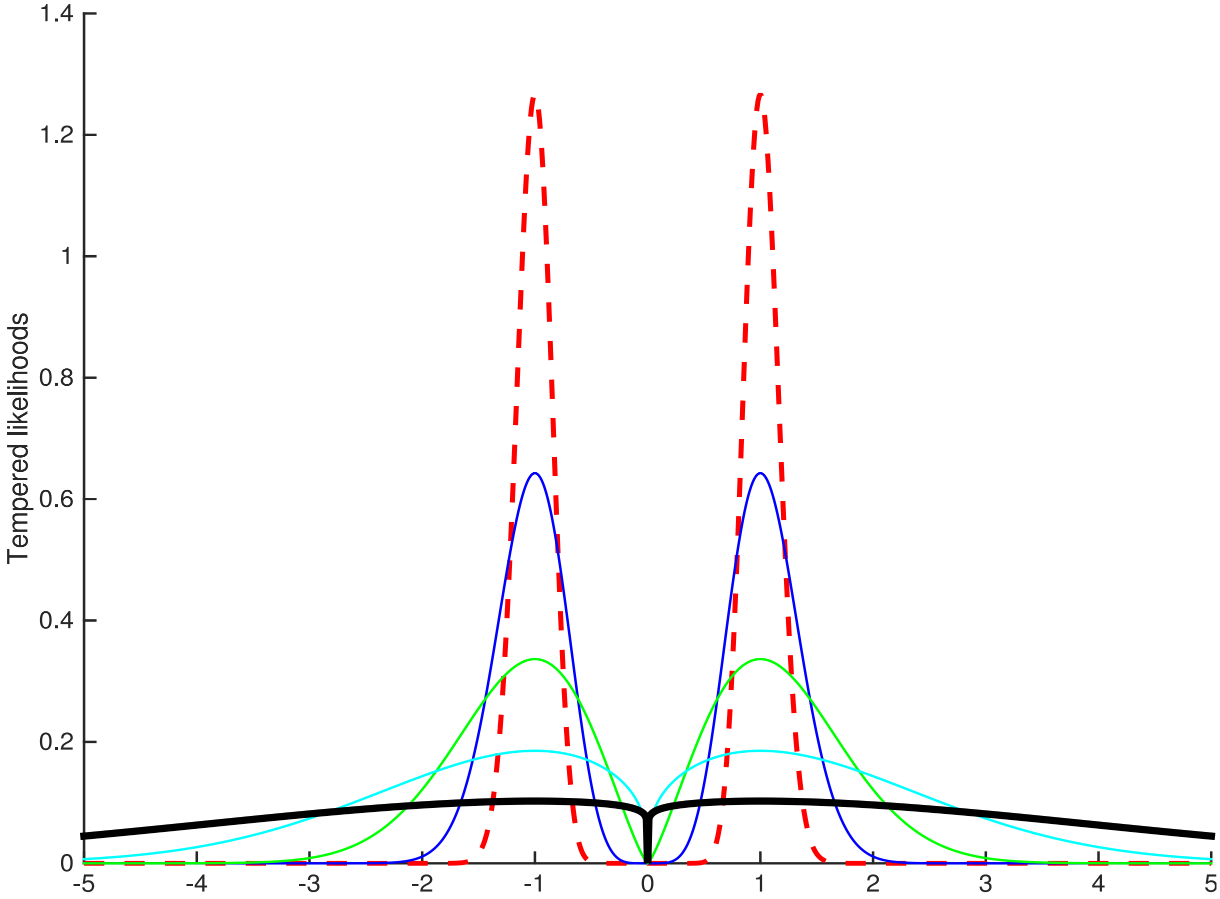
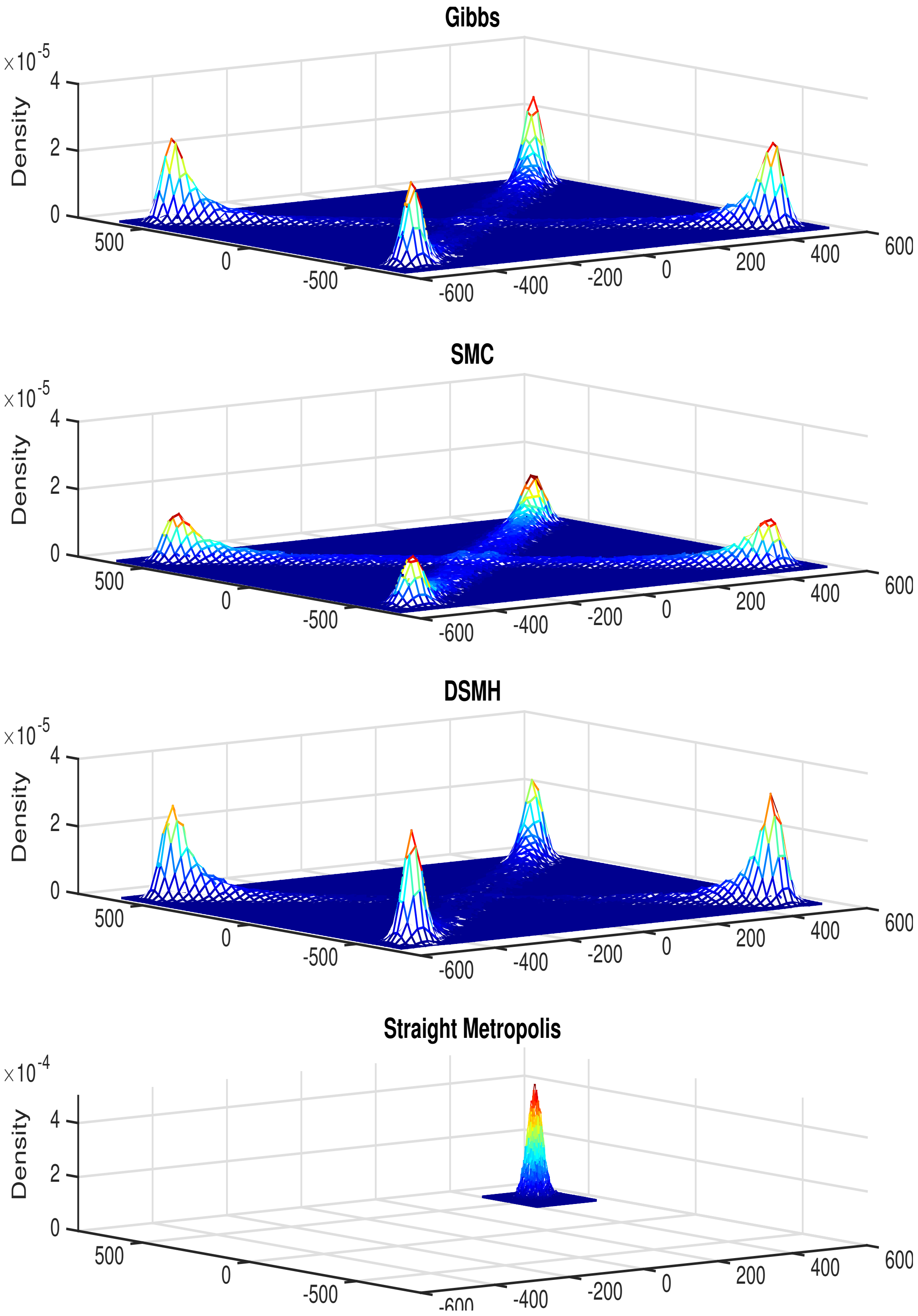


Figure 1. An illustrative example for tempered likelihoods. The thick dashed line is the likelihood (the most peaked) and the thick solid line is the most tempered likelihood (the flattest).


Gibbs

SMC

DSMH

Straight Metropolis

FIGURE 2. The two-dimensional probability density of $a_{0,11}$ (x-axis) and $a_{0,22}$ (y-axis) (after integrating out all other parameters). The probability density is formed empirically from the posterior draws generated by four algorithms: the Gibbs sampler, the SMC sampler, the DSMH sampler, and the straight random-walk Metropolis sampler.

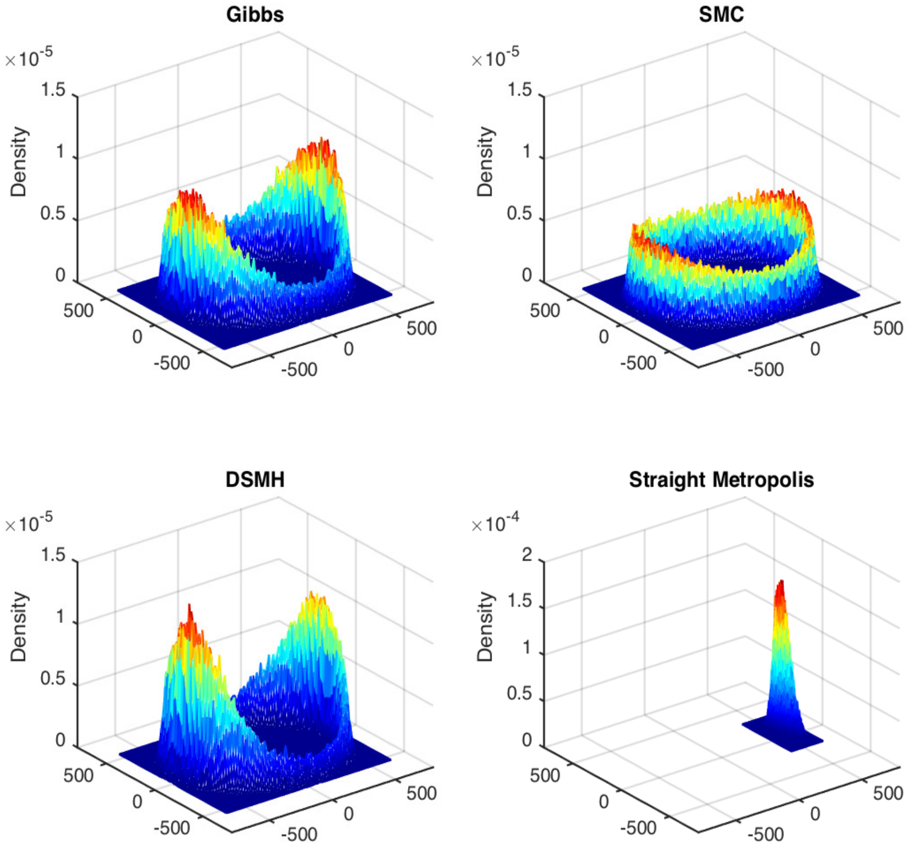


FIGURE 3. The two-dimensional probability density of $a_{0,12}$ (y-axis) and $a_{0,22}$ (x-axis) (after integrating out all other parameters). The probability density is formed empirically from the posterior draws generated by four algorithms: the Gibbs sampler, the SMC sampler, the DSMH sampler, and the straight random-walk Metropolis sampler.

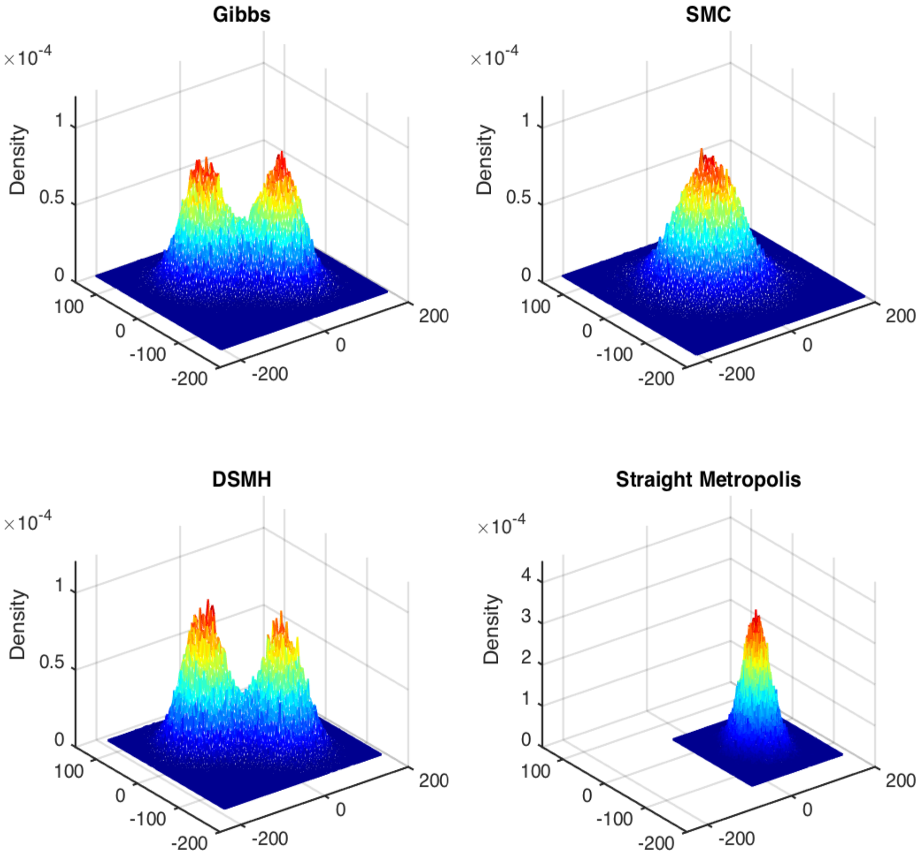


FIGURE 4. The two-dimensional probability density of $a_{7,11}$ (y-axis) and $a_{7,21}$ (x-axis) (after integrating out all other parameters). The probability density is formed empirically from the posterior draws generated by four algorithms: the Gibbs sampler, the SMC sampler, the DSMH sampler, and the straight random-walk Metropolis sampler.

TABLE 1. Recommended values of tuning parameters

Parameter	Recommended Value	Faster Run Time	More Reliable Sample
NG	Problem specific	Smaller	Larger
H	50	Smaller	Larger
$\mathcal{T}$	50	Smaller	Larger
λ_1	$1/(10nT)$	-	-
p	$1/(10\mathcal{T})$	-	-
M	50	-	-
$[\alpha_0, \alpha_1]$	$[0.2, 0.3]$	-	-
GK	10,000	Smaller	Larger

Note: G should be a multiple of the number of computing cores with N and K adjusting to target NG and KG at their desired levels. α_0, α_1, and K are for selecting the Metropolis scale parameter.

Table 2. Estimated log integral constants ($\log I_i$), NSEs, and ESSs for different samplers with log marginal data densities at the final stage under the geometric tempering schedule

Stage	Truth	DSMH	NSE	ESS	SMC	NSE	ESS
1	-0.6397	-0.6443	0.00	0.74	-0.6338	0.00	0.74
2	-0.7264	-0.7324	0.01	0.99	-0.7836	0.16	0.98
3	-0.8184	-0.8249	0.01	0.99	-0.9191	0.16	0.98
4	-0.9128	-0.9204	0.01	0.99	-1.0516	0.17	0.98
5	-1.0086	-1.0159	0.01	0.98	-1.1809	0.18	0.98
6	-1.0998	-1.1054	0.01	0.98	-1.3046	0.19	0.97
7	-1.1831	-1.1899	0.01	0.98	-1.4185	0.20	0.97
8	-1.2522	-1.2588	0.01	0.97	-1.5180	0.21	0.97
9	-1.3009	-1.3081	0.01	0.97	-1.5963	0.22	0.96
10	-1.3161	-1.3258	0.01	0.97	-1.6430	0.23	0.96
11	-1.2949	-1.3001	0.01	0.96	-1.6493	0.24	0.96
12	-1.2165	-1.2236	0.01	0.96	-1.6016	0.25	0.95
13	-1.0636	-1.0706	0.01	0.95	-1.5079	0.28	0.94
14	-0.8207	-0.8258	0.01	0.95	-1.3355	0.31	0.93
15	-0.4591	-0.4616	0.01	0.94	-1.0527	0.33	0.93
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
36	276.60	276.56	0.05	0.65	266.13	1.13	0.44
37	340.01	339.98	0.05	0.63	327.88	1.23	0.39
38	417.17	417.14	0.06	0.61	403.09	1.33	0.34
39	510.95	510.93	0.06	0.58	494.63	1.43	0.30
40	624.91	624.83	0.07	0.55	605.84	1.56	0.25
41	763.07	763.06	0.07	0.52	740.91	1.70	0.20
42	930.77	930.72	0.08	0.49	904.90	1.85	0.16
43	1134.1	1134.0	0.08	0.46	1103.8	2.05	0.11
44	1380.4	1380.3	0.10	0.42	1344.9	2.27	0.08
45	1678.8	1678.8	0.11	0.39	1637.0	2.54	0.05
46	2040.1	2040.1	0.12	0.36	1990.9	2.83	0.03
47	2477.5	2477.6	0.13	0.33	2419.4	3.18	0.01
48	3006.8	3006.9	0.15	0.30	2938.1	3.59	0.01
49	3647.3	3647.4	0.16	0.25	3565.8	4.07	0.00
50	4422.0	4422.2	0.19	0.23	4325.3	4.63	0.00

Note: "Truth" represents integral constants calculated from independent sampling through Gibbs and the columns under "NSE" and "ESS" corresponds to the particular sampler indicated at the top of the previous column.

Table 3. Estimated log integral constants ($\log I_i$), NSEs, and ESSs for different samplers with log marginal data densities at the final stage under the quadratic tempering schedule

Stage	Truth	DSMH	NSE	ESS	SMC	NSE	ESS
1	-1.2738	-1.2812	0.00	0.26	-1.2790	0.00	0.26
2	-0.0851	-0.1217	0.10	0.23	-8.2207	7.72	0.07
3	4.7143	4.6761	0.04	0.33	-2.5369	5.54	0.10
4	13.203	13.183	0.05	0.40	4.1351	5.19	0.10
5	25.458	25.414	0.06	0.46	14.829	4.47	0.11
6	41.424	41.394	0.05	0.51	29.799	3.38	0.17
7	61.162	61.133	0.04	0.56	48.577	2.39	0.24
8	84.661	84.618	0.04	0.61	70.414	2.13	0.27
9	111.87	111.83	0.04	0.63	96.062	1.84	0.31
10	142.82	142.80	0.04	0.66	125.54	1.58	0.36
11	177.51	177.49	0.04	0.69	158.64	1.45	0.39
12	215.90	215.92	0.03	0.71	195.51	1.32	0.42
13	258.06	258.07	0.04	0.74	236.17	1.20	0.46
14	303.90	303.93	0.03	0.75	280.53	1.12	0.49
15	353.48	353.49	0.04	0.76	328.63	1.05	0.51
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
36	2250.4	2250.4	0.04	0.90	2197.5	0.57	0.76
37	2381.5	2381.5	0.04	0.90	2327.2	0.56	0.77
38	2516.2	2516.2	0.03	0.90	2460.6	0.56	0.77
39	2654.6	2654.7	0.03	0.91	2597.6	0.55	0.78
40	2796.8	2796.8	0.03	0.91	2738.4	0.54	0.78
41	2942.7	2942.7	0.04	0.91	2882.8	0.54	0.78
42	3092.2	3092.3	0.04	0.91	3030.9	0.53	0.78
43	3245.5	3245.5	0.04	0.91	3182.7	0.53	0.79
44	3402.5	3402.5	0.04	0.91	3338.2	0.52	0.79
45	3563.1	3563.2	0.04	0.92	3497.4	0.52	0.79
46	3727.5	3727.6	0.03	0.92	3660.3	0.51	0.79
47	3895.4	3895.6	0.03	0.92	3826.8	0.51	0.80
48	4067.3	4067.4	0.03	0.92	3997.1	0.51	0.80
49	4242.8	4242.9	0.04	0.92	4171.0	0.50	0.80
50	4421.9	4422.1	0.04	0.92	4348.7	0.50	0.80

Note: "Truth" represents integral constants calculated from independent sampling through Gibbs and the columns under "NSE" and "ESS" corresponds to the particular sampler indicated at the top of the previous column.

Table 4. Regime-switching BVAR: estimated log integral constants ($\log I_i$), NSEs, ESSs, and log marginal data densities at the final stage under the quadratic tempering schedule

Stage	DSMH (Mueller)	DSMH (IW)	NSE	ESS
1	-1.1009	-0.9846	0.00	0.25
2	0.8707	0.9862	0.05	0.34
3	6.2041	6.3094	0.04	0.43
4	14.993	15.110	0.04	0.44
5	27.447	27.579	0.04	0.47
6	43.706	43.806	0.04	0.51
7	63.689	63.834	0.04	0.55
8	87.439	87.654	0.04	0.60
9	115.10	115.26	0.04	0.64
10	146.45	146.64	0.03	0.68
11	181.48	181.78	0.03	0.71
12	220.44	220.67	0.03	0.73
13	263.03	263.29	0.03	0.75
14	309.42	309.64	0.03	0.77
15	359.33	359.71	0.03	0.79
⋮	⋮	⋮	⋮	⋮
36	2276.8	2277.7	0.07	0.82
37	2409.6	2411.1	0.08	0.82
38	2546.9	2548.3	0.07	0.83
39	2687.2	2689.4	0.07	0.83
40	2831.6	2834.3	0.08	0.83
41	2981.1	2983.1	0.07	0.84
42	3133.2	3135.7	0.08	0.84
43	3288.7	3292.2	0.07	0.85
44	3450.2	3452.6	0.06	0.86
45	3613.9	3616.8	0.06	0.86
46	3782.1	3784.8	0.06	0.87
47	3951.8	3956.7	0.06	0.87
48	4129.1	4132.4	0.06	0.88
49	4309.2	4312.0	0.06	0.88
50	4493.0	4495.4	0.05	0.89

Note: "DSMH (Mueller)" represents integral constants estimated by applying the Mueller method to DSMH draws, and "DSMH (IW)" stands for integral constants derived from the importance weights as a byproduct according to (7).

Appendix A. Pseudo Code for the DSMH Sampler

Algorithm for the DSMH sampler:

```
(1) For $1 < i \leq H$, set
        $\lambda_i = \exp\left( \frac{H-i}{H-1} \log(\lambda_1) \right).$
(2) Set $p = 1/(10\mathcal{T})$.
(3) Draw $NG$ samples from $f_{\lambda_0}$. Denote the draws by $(\theta^{(0,\ell)})_{\ell=1}^{NG}$ and sort them so that $f_{\lambda_0}(\theta^{(0,\ell)}) \leq f_{\lambda_0}(\theta^{(0,\ell+1)})$. It is assumed that there exist algorithms for obtaining draws from $f_{\lambda_0}$.
(4) For $i$ from 1 to $H$ do
    (a) Compute the variance matrix $\Omega_i$ using the importance-weighted draws from the previous stage.
            $\Omega_i = \sum_{\ell=1}^{NG} w_\ell^i \theta^{(i-1,\ell)} (\theta^{(i-1,\ell)})' - \left( \sum_{\ell=1}^{NG} w_\ell^i \theta^{(i-1,\ell)} \right) \left( \sum_{\ell=1}^{NG} w_\ell^i \theta^{(i-1,\ell)} \right)'$
    (b) Tune $\mathfrak{c}_i$ so that the Gaussian Metropolis jumping kernel with variance matrix $\mathfrak{c}_i \Omega_i$ has an acceptance rate between $\alpha_0$ and $\alpha_1$.
    (c) For $1 \leq k \leq M-1$, let
            $L_{i,k} = f_{\lambda_{i-1}}(\theta^{(i-1,\mathrm{floor}(kNG/M))}).$
    (d) For $j$ from 1 to $G$ do
        (i) Draw $\theta^{(i,j,0)}$ from the previous stage's draws using the importance weights.
        (ii) For $\ell$ from 1 to $N\mathcal{T}$ do
            (A) Draw $u_1$ and $u_2$ from the uniform distribution on $(0,1)$.
            (B) If $u_1 < p$, set $k$ so that $L_{i,k-1} \leq f_{\lambda_{i-1}}(\theta^{(i,j,\ell)}) < L_{i,k}$ and set $\hat{\theta}^{(i,j,\ell+1)} = \theta^{(i-1,\mathrm{floor}((k-1+u_2)NG/H))}$. Set
                    $\theta^{(i,j,\ell+1)} = \begin{cases} \hat{\theta}^{(i,j,\ell+1)} & \text{if } \frac{f_{\lambda_i}(\hat{\theta}^{(i,j,\ell+1)}) f_{\lambda_{i-1}}(\theta^{(i,j,\ell)})}{f_{\lambda_i}(\theta^{(i,j,\ell)}) f_{\lambda_{i-1}}(\hat{\theta}^{(i,j,\ell+1)})} > u_2 \\ \theta^{(i,j,\ell)} & \text{otherwise} \end{cases}.$
            (C) If $u_1 \geq p$, draw $\hat{\theta}^{(i,j,\ell+1)}$ from the Gaussian distribution with mean $\theta^{(i,j,\ell)}$ and variance $\mathfrak{c}_i \Omega_i$. Set
                    $\theta^{(i,j,\ell+1)} = \begin{cases} \hat{\theta}^{(i,j,\ell+1)} & \text{if } \frac{f_{\lambda_i}(\hat{\theta}^{(i,j,\ell+1)})}{f_{\lambda_i}(\theta^{(i,j,\ell)})} > u_2 \\ \theta^{(i,j,\ell)} & \text{otherwise} \end{cases}.$
            (D) Save $\theta^{(i,j,\ell)}$ if $\ell$ is a multiple of $\mathcal{T}$.
    (e) Merge all the saved draws. Denote these draws by $(\theta^{(i,\ell)})_{\ell=1}^{NG}$ and sort them so that $f_{\lambda_i}(\theta^{(i,\ell)}) \leq f_{\lambda_i}(\theta^{(i,\ell+1)})$.
```

Appendix B. Pseudo-code for Tuning the Metropolis Scale

For $1 \leq i \leq H$, the algorithm for tuning $\mathfrak{c}_i$ given Ω_i is:

```
(1) Set
        $\mathfrak{c}_i = \begin{cases} \mathfrak{c}_{i-1} & \text{if } i > 1 \\ 1 & \text{if } i = 1 \end{cases}.$
(2) For $j$ from 1 to $G$ do
    (a) Draw $\theta^{(i,j,0)}$ from the previous stage's draws using the importance weights.
    (b) For $\ell$ from 1 to $K$ do
        (i) Draw $u$ from the uniform distribution on $(0,1)$ and set $c_j = 0$.
        (ii) Draw $\hat{\theta}^{(i,j,\ell+1)}$ from the Gaussian distribution with mean $\theta^{(i,j,\ell)}$ and variance
             $\mathfrak{c}_i \Omega_i$ and set
                 $\theta^{(i,j,\ell+1)} = \begin{cases} \hat{\theta}^{(i,j,\ell+1)} & \text{if } \frac{f_{\lambda_i}(\hat{\theta}^{(i,j,\ell+1)})}{f_{\lambda_i}(\theta^{(i,j,\ell)})} > u \\ \theta^{(i,j,\ell)} & \text{otherwise} \end{cases}.$
             If $\hat{\theta}^{(i,j,\ell+1)}$ was accepted, increment $c_j$.
(3) Set
        $\alpha = \frac{\sum_{j=1}^{G} c_j}{KG}.$
    If $\alpha \in (\alpha_0, \alpha_1)$ then done, otherwise set
        $\mathfrak{c}_i = \begin{cases} \frac{1}{5}\mathfrak{c}_i & \text{if } \alpha \leq (\frac{\alpha_0+\alpha_1}{2})^5 \\ \frac{\log(\frac{\alpha_0+\alpha_1}{2})}{\log(\alpha)}\mathfrak{c}_i & \text{if } (\frac{\alpha_0+\alpha_1}{2})^5 < \alpha < (\frac{\alpha_0+\alpha_1}{2})^{\frac{1}{5}} \\ 5\mathfrak{c}_i & \text{if } (\frac{\alpha_0+\alpha_1}{2})^{\frac{1}{5}} \leq \alpha \end{cases}$
    and return to step (2).
```

In our application, we choose $[\alpha_0,\ \alpha_1] = [0.2,\ 0.3]$ and $K = 500$. Since G varies between 20 and 50, KG is between 10,000 and 25,000. For many stages, we find that the tuning process exits after one pass so that $\mathfrak{c}_i = \mathfrak{c}_{i-1}$.

References

Aldrich, E. M., J. Fernández-Villaverde, A. R. Gallant, and J. F. Rubio-Ramírez (2011): "Tapping the Supercomputer Under Your Desk: Solving Dynamic Equilibrium Models with Graphics Processors," *Journal of Economic Dynamics and Control*, 35(3), 386–393.

An, S., and F. Schorfheide (2007): "Bayesian Analysis of DSGE Models," *Econometric Reviews*, 26(2–4), 113–172.

Bauwens, L., C. S. Bos, H. K. van Dijk, and R. D. van Oest (2004): "Adaptive Radial-Based Direction Sampling: Some Flexible and Robust Monte Carlo Integration Methods," *Journal of Econometrics*, 123(2), 201–225.

Bernanke, B. S., M. Gertler, and M. W. Watson (1997): "Systematic Monetary Policy and the Effects of Oil Price Shocks," *Brookings Papers on Economic Activity*, 1, 91–142.

Bloom, N. (2009): "The Impact of Uncertainty Shocks," *Econometrica*, 77(3), 623–685.

Bognanni, M., and E. Herbst (2014): "Estimating (Markov-Switching) VAR Models without Gibbs Sampling: A Sequential Monte Carlo Approach," Unpublished Manuscript.

Chib, S. (1995): "Marginal Likelihood from the Gibbs Output," *Journal of the American Statistical Association*, 90, 1313–1321.

Chopin, N. (2004): "Central Limit Theorem for Sequential Monte Carlo Methods and its Application to Bayesian Inference," *Annals of Statistics*, 32, 2385–2411.

Christiano, L. J., M. S. Eichenbaum, and C. L. Evans (1999): "Monetary Policy Shocks: What Have We Learned and To What End?," in *Handbook of Macroeconomics*, ed. by J. B. Taylor, and M. Woodford, vol. 1A, pp. 65–148. North-Holland, Amsterdam, Holland.

——— (2005): "Nominal Rigidities and the Dynamic Effects of a Shock to Monetary Policy," *Journal of Political Economy*, 113, 1–45.

Cogley, T., and T. J. Sargent (2005): "Drifts and Volatilities: Monetary Policies and Outcomes in the Post WWII U.S.," *Review of Economic Dynamics*, 8, 262–302.

Del Negro, M., and F. Schorfheide (2004): "Priors from General Equilibrium Models for VARs," *International Economic Review*, 45, 643–673.

Durham, G., and J. Geweke (2012): "Adaptive Sequential Posterior Simulators for Massively Parallel Computing Environments," Unpublished Manuscript.

Fernández-Villaverde, J., P. Guerrón-Quintana, J. F. Rubio-Ramírez, and M. Uribe (2011): "Risk Matters: The Real Effects of Volatility Shocks," *American Economic Review*, 101(6), 2530–2561.

Fuentes-Albero, C., and L. Melosi (2013): “Methods for Computing Marginal Data Densities from the Gibbs Output,” *Journal of Econometrics*, 175(2), 132–141.

Geweke, J. (1999): “Using Simulation Methods for Bayesian Econometric Models: Inference, Development, and Communication,” *Econometric Reviews*, 18(1), 1–73.

Hamilton, J. D. (1989): “A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle,” *Econometrica*, 57(2), 357–384.

Hamilton, J. D., D. F. Waggoner, and T. Zha (2007): “Normalization in Econometrics,” *Econometric Reviews*, 26(2-4), 221–252.

Herbst, E., and F. Schorfheide (2014): “Sequential Monte Carlo Sampling for DSGE Models,” *Journal of Applied Econometrics*.

Hoogerheide, L., A. Opschoor, and H. K. van Dijk (2012): “A Class of Adaptive EM-based Importance Sampling Algorithms for Efficient and Robust Posterior and Predictive Simulation,” *Journal of Econometrics*, 171(2), 101–120.

Hubrich, K., and R. J. Tetlow (Forthcoming): “Financial Stress and Economic Dynamics: the Transmission of Crises,” *Journal of Monetary Economics*.

Ingram, B. F., and C. H. Whiteman (1994): “Supplanting the ”Minnesota” Prior: Forecasting Macroeconomic Time Series Using Real Business Cycle Model Priors,” *Journal of Monetary Economics*, 34(3), 497–510.

Justiniano, A., and G. E. Primiceri (2008): “The Time Varying Volatility of Macroeconomic Fluctuations,” *American Economic Review*, 98(3), 604–641.

Kou, S. C., Q. Zhou, and W. H. Wong (2006): “Equi-energy sampler with applications in statistical inference and statistical mechanics,” *Annals of Statistics*, 34(4), 1581–1619.

Leeper, E. M., C. A. Sims, and T. Zha (1996): “What Does Monetary Policy Do?,” *Brookings Papers on Economic Activity*, 2, 1–78.

Litterman, R. B. (1986): “Forecasting with Bayesian Vector Autoregressions — Five Years of Experience,” *Journal of Business and Economic Statistics*, 4, 25–38.

Liu, Z., D. F. Waggoner, and T. Zha (2011): “Sources of Macroeconomic Fluctuations: A Regime-Switching DSGE Approach,” *Quantitative Economics*, 2, 251–301.

Meng, X.-L., and W. H. Wong (1996): “Simulating Ratios of Normalizing Constants via a Simple Identity: A Theoretical Exploration,” *Statistica Sinica*, 6, 831–860.

Rubio-Ramírez, J. F., D. F. Waggoner, and T. Zha (2010): “Structural Vector Autoregressions: Theory of Identification and Algorithms for Inference,” *Review of Economic Studies*, 77, 665–696.

Rudebusch, G. D., and L. E. O. Svensson (1999): “Policy Roles for Inflation Targeting,” in *Monetary Policy Rules*, ed. by J. B. Taylor, chap. 5, pp. 203–262. University of Chicago Press, Chicago and London.

Sims, C. A., D. F. Waggoner, and T. Zha (2008): "Methods for Inference in Large Multiple-Equation Markov-Switching Models," *Journal of Econometrics*, 146(2), 255–274.

Sims, C. A., and T. Zha (1998): "Bayesian Methods for Dynamic Multivariate Models," *International Economic Review*, 39(4), 949–968.

——— (2006): "Were There Regime Switches in U.S. Monetary Policy?," *American Economic Review*, 96, 54–81.

Smets, F., and R. Wouters (2007): "Shocks and Frictions in US Business Cycles: A Bayesian DSGE Approach," *American Economic Review*, 97, 586–606.

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

Waggoner, D. F., and T. Zha (2003a): "A Gibbs Sampler for Structural Vector Autoregressions," *Journal of Economic Dynamics and Control*, 28(2), 349–366.

——— (2003b): "Likelihood Preserving Normalization in Multiple Equation Models," *Journal of Econometrics*, 114(2), 329–347.

Federal Reserve Bank of Atlanta, Federal Reserve Bank of Atlanta, Federal Reserve Bank of Atlanta, Emory University, and NBER