



Contents lists available at ScienceDirect

Journal of Econometrics

journal homepage: www.elsevier.com/locate/jeconom

Tailored randomized block MCMC methods with application to DSGE models

Siddhartha Chib^{a,*}, Srikanth Ramamurthy^b

^a Olin Business School, Washington University in St. Louis, Campus Box 1133, 1 Bookings Drive, St. Louis, MO 63130, United States

^b Selling School of Business, Loyola University Maryland, 4501 N. Charles St., Baltimore, MD 21210, United States

ARTICLE INFO

Article history:

Received 25 January 2009

Received in revised form

25 January 2009

Accepted 20 August 2009

Available online 27 August 2009

Keywords:

Dynamic stochastic general equilibrium models

Markov chain Monte Carlo

Metropolis–Hastings algorithm

Marginal likelihood

Randomized blocks

Tailored proposal densities

Multi modal densities

Simulated annealing

ABSTRACT

In this paper we develop new Markov chain Monte Carlo schemes for the estimation of Bayesian models. One key feature of our method, which we call the tailored randomized block Metropolis–Hastings (TaRB-MH) method, is the random clustering of the parameters at every iteration into an arbitrary number of blocks. Then each block is sequentially updated through an M–H step. Another feature is that the proposal density for each block is tailored to the location and curvature of the target density based on the output of simulated annealing, following Chib and Greenberg (1994, 1995) and Chib and Ergashev (in press). We also provide an extended version of our method for sampling multi-modal distributions in which at a pre-specified mode jumping iteration, a single-block proposal is generated from one of the modal regions using a mixture proposal density, and this proposal is then accepted according to an M–H probability of move. At the non-mode jumping iterations, the draws are obtained by applying the TaRB-MH algorithm. We also discuss how the approaches of Chib (1995) and Chib and Jeliazkov (2001) can be adapted to these sampling schemes for estimating the model marginal likelihood. The methods are illustrated in several problems. In the DSGE model of Smets and Wouters (2007), for example, which involves a 36-dimensional posterior distribution, we show that the autocorrelations of the sampled draws from the TaRB-MH algorithm decay to zero within 30–40 lags for most parameters. In contrast, the sampled draws from the random-walk M–H method, the algorithm that has been used to date in the context of DSGE models, exhibit significant autocorrelations even at lags 2500 and beyond. Additionally, the RW-MH does not explore the same high density regions of the posterior distribution as the TaRB-MH algorithm. Another example concerns the model of An and Schorfheide (2007) where the posterior distribution is multi-modal. While the RW-MH algorithm is unable to jump from the low modal region to the high modal region, and vice-versa, we show that the extended TaRB-MH method explores the posterior distribution globally in an efficient manner.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

Dynamic stochastic general equilibrium (DSGE) models are arguably the dominant framework for modeling and analyzing macroeconomic phenomena. Over the past few years there has been substantial interest in the fitting of these models from a Bayesian perspective implemented by Markov chain Monte Carlo (MCMC) methods (for example, Fernandez-Villaverde and Rubio-Ramirez (2004), Lubik and Schorfheide (2004), Smets and Wouters (2003, 2007), An and Schorfheide (2007), Canova (2007), and De-Jong and Dave (2007)). One reason for the interest in the Bayesian approach is that it provides an avenue (namely through the prior distribution) for incorporating substantive information about the parameters. This proves important because the parameters in

these models are often ill-determined (or unreasonable) when fit by maximum likelihood on account of the stylized nature of the models and the relatively small data sets that can be utilized for the fitting. The attractiveness of Bayesian methods has also grown due to the availability of powerful MCMC simulation methods that provide the technology for sampling the posterior distribution of the parameters (Chib and Greenberg, 1995; Chib, 2001).

One feature of the current generation of Bayesian DSGE fitting methods is that they sample the posterior distribution by what is formally known as a single block random-walk Metropolis–Hastings (M–H) algorithm (RW-MH). In the RW-MH algorithm the parameters are sampled in a single block by drawing a proposal from a random walk process. This proposal value is then accepted as the next draw according to the corresponding M–H probability of move (which in this case is essentially the ratio of the posterior density at the proposed value and the posterior density at the current value); if the proposed value is rejected, the current value is retained as the new value of the Markov chain.

The main appeal of this sampling method is its speed and relative ease of implementation. It has been noticed, however, that

* Corresponding author.

E-mail addresses: chib@wustl.edu (S. Chib), sramamurthy@loyola.edu (S. Ramamurthy).

in the realm of DSGE models, where the posterior distribution can be irregular, the tuning of the RW-MH algorithm is not straightforward. Substantial pre-run tuning effort is often necessary, which can add considerably to the overall time of implementation. For one, it turns out that the effect of the starting value of the Markov chain can wear off slowly, thus necessitating large simulation sample sizes of the order of many hundreds of thousands. In some situations, for instance in a multi-modal problem, the effect of the starting value may not wear off in realistic sampling time. This problem was demonstrated by [An and Schorfheide \(2007\)](#) who noted that the algorithm was unable to escape the region surrounding one of the modes. We revisit this example later in the paper. Another problem is that the variance of the increment in the random walk proposal can be difficult to set, especially in higher-dimensional problems, and the sampler performance can be severely comprised by a poor choice of it. With too small a variance the search process can be extremely slow, whereas with a large variance there can be many rejections and the same value can be repeated many times in the chain. As a result, the sampled sequence tends to exhibit high serial correlations and slow convergence to the posterior distribution.

Our goal in this paper is to suggest new MCMC schemes for the estimation of Bayesian models with specific focus on DSGE models. Essentially, our idea is to combine the efficiency of tailored proposals ([Chib and Greenberg, 1994](#)) with a flexible blocking strategy that virtually eliminates pre-run tuning. In our approach, which we refer to as the Tailored Randomized Block M–H or TaRB-MH algorithm, the parameters of the model are clustered at every iteration into a random number of blocks. Then each block is sequentially updated through an M–H step in which the proposal density is tailored to mimic closely the target density of that block. Since the posterior surface can be irregular in general, we construct our tailored proposal densities from the output of simulated annealing, following [Chib and Ergashev \(in press\)](#). In effect, the TaRB-MH algorithm has the features of a general black box approach that requires little user involvement. We also provide an extension of this algorithm for sampling multi-modal distributions. In this version, which we refer to as the TaRBMJ-MH algorithm, at a pre-specified move jumping iteration (say every 100th), a single-block proposal is generated from one of the modal regions using a mixture proposal density, and this proposal is then accepted according to an M–H probability of move. At the non-mode jumping iterations, the draws are obtained by applying the TaRB-MH algorithm. It should be noted that while these methods are developed with the DSGE applications in mind, the methods are general and can be applied to any Bayesian problem.

We complete our methodological developments by presenting an approach for calculating the model marginal likelihood, which is needed for the comparison of alternative Bayesian models. The method we present is based on the framework of [Chib \(1995\)](#), and its M–H version in [Chib and Jeliazkov \(2001\)](#), with suitable modifications to accommodate the randomized block approach proposed in this paper.

We illustrate the performance of our methods with the aid of several examples, including a 60 dimensional state space model, and two DSGE models. The first DSGE model is the 36 parameter model of [Smets and Wouters \(2007\)](#). The RW-MH algorithm in this model is difficult to tune because of the dimensionality of the parameter space and the complexity of the posterior surface. Even a well tuned RW-MH chain has serial correlations at lags 2500 and beyond. In comparison, the autocorrelations in the sampled output from the TaRB-MH algorithm decay to zero within 30–40 lags for most parameters and the sampler explores high density regions of the posterior distribution that are missed by the RW-MH algorithm.

Our second DSGE example is the model of [An and Schorfheide \(2007\)](#) where the posterior distribution is multi-modal. In this

case, the RW-MH algorithm fails to jump across modal regions even when the sampler is run for a million iterations following a burn-in of 100,000 iterations. This occurs despite a difference of approximately 8 on the log scale between the unnormalized posterior ordinates at the two modes. The TaRBMJ-MH method, on the other hand, jumps to the high mode almost instantaneously and explores the posterior distribution globally in an efficient manner.

The rest of the paper is organized as follows. To motivate the discussion of our methods, as well as for completeness of the paper, we begin with a brief discussion of the general setup of linearized DSGE models in the following section. In Section 3 we present the details of our approach for sampling the posterior distribution and discuss a suitable way of calculating the marginal likelihood. Section 4 deals with the application of our methods to the two DSGE models mentioned above. Concluding remarks are provided in Section 5.

2. The setup

In this paper our focal application is the class of macroeconomic models commonly referred to as DSGE models. These are structural microfounded models set in a general equilibrium framework. Thus, the outcome of the economic variables is determined by the collective actions of the agents in the economy being modeled. Further, in the face of uncertainty, the agents take into account the future expected values of the variables when making their decisions. In the ensuing discussion, as well as in the applications in Section 4, this decision making process is assumed to be based on rational expectations (expectations based on the full information set available up to the current period). For both theoretical and inferential purposes, one is usually interested in analyzing the behavior of the economy around its deterministic steady state. Since these models are typically highly nonlinear, one often works with first order approximations. This reduces the model to a system of stochastic expectational difference equations, popularly known as a linear rational expectations model. For the rest of this section, as well as in the DSGE examples in Section 4, this is our structural model of interest.¹

As discussed in [An and Schorfheide \(2007\)](#), [Sargent \(1989\)](#), [Smets and Wouters \(2003\)](#) and others, the process of transforming a structural model to its reduced form involves two steps. The first step is to solve the model. In the solution step, the expectational terms are integrated out to obtain a stochastic law of motion for the endogenous variables of the model. Here, one generally restricts attention to the subset of parameter space, \mathcal{S}_D , that generates a unique stable solution to the model (the determinate solution). Several solution methods have been proposed in the literature to solve linear rational expectations models. A concise comparative discussion of the various methods is provided by [Anderson \(2008\)](#).

From an empirical standpoint, the convenience of a linear framework is that the output of the solution can be cast as a vector Markov process for the endogenous variables \mathbf{s}_t of the model

$$\mathbf{s}_t = \mathbf{D}(\boldsymbol{\theta})\mathbf{s}_{t-1} + \mathbf{F}(\boldsymbol{\theta})\boldsymbol{\varepsilon}_t \quad (1)$$

where the matrices $\mathbf{D}(\boldsymbol{\theta})$ and $\mathbf{F}(\boldsymbol{\theta})$ are awkward implicit functions of the model parameters $\boldsymbol{\theta}$, obtained from the solution, and $\boldsymbol{\varepsilon}_t$ is the collection of exogenous shocks that is assumed to be a zero mean Gaussian random vector with variance-covariance matrix given by $\boldsymbol{\Omega}(\boldsymbol{\theta})$. It is assumed that the parameters in $\boldsymbol{\Omega}$ lie in the region \mathcal{S}_Ω that is consistent with the positive definiteness constraint and

¹ These stylized assumptions are inconsequential from the point of view of this paper as the MCMC methods we propose can be applied to simulate multivariate target distributions in general.

that any parameter bounds imposed by the underlying theoretical model restrict the parameters to the region \mathcal{S}_L .

For purposes of conducting inference on the parameters, the preceding vector Markov process can be viewed as the state transition equation in a linear Gaussian state space model (SSM). Accordingly, the SSM formulation is completed by defining a measurement equation that relates the states \mathbf{s}_t to a vector of observable data \mathbf{y}_t .

$$\mathbf{y}_t = \mathbf{a}(\boldsymbol{\theta}) + \mathbf{B}(\boldsymbol{\theta})\mathbf{s}_t. \quad (2)$$

Depending on the model and representation of the data, the vector \mathbf{a} and the matrix \mathbf{B} may consist of known constants or involve additional parameters that have to be estimated.

Given a sample of data $\mathbf{y}_{n \times T} = \{\mathbf{y}_t\}$, $t = 1, \dots, T$, the joint density of the data (the likelihood function) given the parameters can be calculated as

$$f(\mathbf{y}|\boldsymbol{\theta}) = \prod_{t=1}^T \frac{1}{(2\pi)^{n/2} |\Sigma_{t|t-1}|^{1/2}} \times \exp \left\{ -\frac{1}{2} \eta'_{t|t-1} \Sigma_{t|t-1}^{-1} \eta_{t|t-1} \right\} I_{\mathcal{S}_L}(\boldsymbol{\theta}) I_{\mathcal{S}_\Omega}(\boldsymbol{\theta}) I_{\mathcal{S}_D}(\boldsymbol{\theta}) \quad (3)$$

where $\eta_{t|t-1} = \mathbf{y}_t - \mathbf{a} - \mathbf{B}\hat{\mathbf{s}}_{t|t-1}$ is the one-step ahead prediction error, $\hat{\mathbf{s}}_{t|t-1}$ is the one-step ahead estimate of the state and $\Sigma_{t|t-1}$ is the conditional variance of the prediction error, each given information up to time $(t-1)$. These quantities are obtained from the usual Kalman filter recursions (see for instance, [Harvey \(1990\)](#)).

It is important to recognize that in most practical models of interest the solution procedure can only be performed numerically. Thus, likelihood evaluation at any point in the parameter space requires tedious computations. Another point is that the parameter space in general is given by the region $\mathcal{S}_L \cap \mathcal{S}_\Omega \cap \mathcal{S}_D$ where the region \mathcal{S}_D that is induced by the determinacy constraint is the most difficult to characterize because it is determined by the output of the solution step. Therefore, in order to verify if a particular value of the parameters is in the admissible region, it is efficient to see if it lies in \mathcal{S}_L , then in \mathcal{S}_Ω , and finally in \mathcal{S}_D .

In the Bayesian context, the empirical state space model is completed with a prior distribution $\pi(\boldsymbol{\theta})$ on the parameters. The posterior distribution of the parameters $\pi(\boldsymbol{\theta}|\mathbf{y})$ is calculated (up to the normalizing constant) as the product of the likelihood function and the prior distribution

$$\pi(\boldsymbol{\theta}|\mathbf{y}) \propto f(\mathbf{y}|\boldsymbol{\theta}) \times \pi(\boldsymbol{\theta}). \quad (4)$$

From the Bayesian point of view, a summary of the posterior (for instance, the mean or the median) serves as an estimate of the parameters. However, as the posterior distribution is typically intractable, such summaries are not obtained easily. Instead, one relies on MCMC methods to draw sample variates from the posterior distribution. These sampled draws are then used to find the posterior mean, posterior quantiles and other summaries of the distribution. In the following section we discuss how this general approach can be effectively operationalized.

3. Posterior sampling and marginal likelihood estimation

3.1. Tailored randomized block M–H algorithm

In MCMC methods, one samples the posterior density by simulating a Markov chain whose invariant density is the specified target density. The sampled draws, beyond a suitable burn-in phase, are then taken as a (correlated) sample from the posterior density. While the general idea is clear, designing a sampler that mixes well (has low serial correlations in the sampled draws),

and converges quickly to the invariant distribution, is challenging, especially in complex models.

The sampling schemes that we focus on belong to the class of tailored multiple block MH algorithms as discussed in [Chib and Greenberg \(1994, 1995\)](#). In a multiple-block MH algorithm, the parameters are grouped into several distinct blocks, and then each block of parameters is updated in sequence by a MH step, conditioned on the most current value of the parameters in the remaining blocks. It is noteworthy that a multiple-block M–H algorithm has not been implemented before in the setting of DSGE models. It is our aim to show, however, that versions of the multiple-block M–H algorithm are capable of effectively sampling the complex posterior distributions that arise from DSGE models.

A key question in the multiple block sampling method is the number and composition of the blocks. One principle is to form groups so that parameters in different blocks are not strongly correlated whereas those within a block are (grouping by correlation). Unlike usual statistical models, however, where location and scale parameters (for instance) may be approximately orthogonal and hence placed in separate groups, in many models, including DSGE models, such a priori knowledge is rarely available. In fact, because of the DSGE solution process, which constructs a highly nonlinear, analytically intractable map from the parameters of the structural model to those of the state-space model, such grouping by correlation is not feasible.

Another complication is that the DSGE model posteriors tend to be irregular so that the “grouping by correlation” criterion is not necessarily ideal. To see why, suppose that a bivariate posterior distribution has positively sloped contours for some portion of the support but has vertical contours for another portion of the support. In that case, it is meaningful in the positively sloped regions to sample the parameters in a group, but to sample the parameters one at a time to escape into the other portion of the support. In other words, in irregular problems, a permanent grouping of parameters in one block is not necessarily optimal.

Our novel idea to blocking, which works well and has never been tried before in the MCMC literature, is to randomize the formation of blocks in every iteration. This means that both the number of blocks and its components are randomized. A related but different randomization approach is utilized to great effect in [Pitt and Shephard \(1997\)](#) and [Elerian et al. \(2001\)](#) where, in the context of state space and diffusion models, respectively, the block sizes for contiguous time ordered variables are randomized. In that blocking strategy, therefore, the time ordered variables are never scrambled, just the length of the sequence of variables that form a block changes. In contrast, in our scheme, the blocks are formed from a scrambled set of parameters. One clear virtue of our randomized blocking scheme is that it avoids the pitfalls from a poor choice of a priori blocks. Another is that it allows for the groupings to change, which is important in irregular problems (as in the preceding illustration). We note that (if necessary) such a scheme could also be organized to allow for some parameters in $\boldsymbol{\theta}_l$ to form a fixed block, and the remaining parameters to form random blocks. In this sense, this randomized blocking strategy is flexible enough to accommodate different model specific blocking needs.

The second component of our approach is that the proposal density in the M–H step is constructed to adapt to both the location and curvature of the posterior density of that block. While this notion of local tailoring dates back to [Chib and Greenberg \(1994, 1995\)](#), we further enhance this approach by basing the tailored densities on the output of the simulated annealing (SA) optimization method (following [Chib and Ergashev \(in press\)](#)). This enhancement is particularly helpful in dealing with irregular distributions, and also for localizing the proposal density far enough away from the current region (depending on where the

global mode for that block of parameters is located) to permit large moves.

Our multiple block tailored M–H algorithm can be summarized as follows.

Algorithm: TaRB-MH algorithm

- Step 1** Initialize $\theta^{(0)} \in \mathcal{S}_L \cap \mathcal{S}_\Omega \cap \mathcal{S}_D$ and fix n_0 (the burn-in) and M (the MCMC sample size)
- Step 2** In each iteration $j, j = 1, \dots, n_0 + M$, randomly generate blocks $(\theta_{j,1}, \theta_{j,2}, \dots, \theta_{j,p_j})$
- Step 3** Within each iteration, sample each block $\theta_{j,l}, l = 1, \dots, p_j$, by an M–H step with a tailored proposal density as described below.
- Step 4** Repeat steps 2–3 $n_0 + M$ times, discard the draws from the first n_0 iterations and save the subsequent M draws $\theta^{(n_0+1)}, \dots, \theta^{(n_0+M)}$.

Generating random blocks in Step 2 above is, of course, straightforward and does not require comment. We therefore focus on an explanation of Step 3. Suppose that at the end of the $(j-1)$ st iteration of the MCMC iteration, there are p_j randomly constructed blocks $(\theta_{j,1}, \dots, \theta_{j,p_j})$. Let $\theta_{j,-l}$ denote the most current value of all the blocks except the l th. Then to construct the tailored proposal density for $\theta_{j,l}$ we find

$$\hat{\theta}_{j,l} = \arg \max_{\theta_{j,l}} \log[f(\mathbf{y}|\theta_{j,l}, \theta_{j,-l}) \times \pi(\theta)]$$

from a suitably formulated version of SA.

In SA one proposes a random modification to the current guess of the maximum which is then accepted or rejected probabilistically. Moves that lower the function value can sometimes be accepted. The probability of accepting such downhill moves declines over iterations according to a “cooling schedule”, thus allowing the method to converge. Specifically, starting from an initial temperature t_0 and an initial guess for the maximum $\theta_{j,l}^{(0)}$, such that $(\theta_{j,1}, \dots, \theta_{j,l}^{(0)}, \dots, \theta_{j,p_j}) \in \mathcal{S}_L \cap \mathcal{S}_\Omega \cap \mathcal{S}_D$, repeated proposals are generated for a randomly chosen element θ_m of $\theta_{j,l}$ using a random walk process

$$\theta'_m = \theta_m + s \mathcal{N}(0, 1)$$

where $s > 0$ is a suitable scale factor. As in the M–H step we restrict the proposals to the constrained set. Perturbations resulting in a higher function value are always accepted, whereas those resulting in a lower function evaluation are accepted with probability

$$p = \exp\{\Delta[\log(f(\mathbf{y}|\theta_{j,l}, \theta_{j,-l}) \times \pi(\theta))]/t\} < 1$$

where $\Delta[\log(f(\mathbf{y}|\theta_{j,l}, \theta_{j,-l}) \times \pi(\theta))]$ is the change in the log of the objective function, computed as the log of the objective function at the perturbed value of the parameters minus the log of the objective function at the existing value of the parameters. As one can verify, this probability is a decreasing function of the temperature, with convergence achieved as $t \rightarrow 0$.

Our implementation of this algorithm is as follows. We first divide the search process into various stages, denoted by $k, k = 1, 2, \dots, K$, with the length of each stage l_k given by $b + l_{k-1}$, where $b \in \mathbb{N}$ is the stage expansion factor. We then specify the initial temperature t_0 which is held constant in each stage but reduced across stages according to the linear cooling schedule $t_k = at_{k-1}$, where $0 < a < 1$ is the cooling constant. Within each stage, the algorithm searches for the maximum, proposing values from a random walk process as mentioned above.

Once $\hat{\theta}_{j,l}$ has been so found, we calculate the curvature of the target posterior distribution of that block as the negative inverse

of the Hessian²:

$$\mathbf{V}_{j,l} = \left(- \frac{\partial^2 \log[f(\mathbf{y}|\theta_{j,l}, \theta_{j,-l}) \times \pi(\theta)]}{\partial \theta_{j,l} \partial \theta'_{j,l}} \right)^{-1} \Bigg|_{\theta_{j,l} = \hat{\theta}_{j,l}}.$$

Our proposal density $q_l(\theta_{j,l}|\theta_{j,-l}, \mathbf{y})$ of $\theta_{j,l}$ is then given by

$$q_l(\theta_{j,l}|\theta_{j,-l}, \mathbf{y}) = t(\theta_{j,l}|\hat{\theta}_{j,l}, \mathbf{V}_{j,l}, \nu), \quad (5)$$

a multivariate student- t density with $\nu > 2$ degrees of freedom. Note that this proposal density depends on the current values of the remaining parameters (blocks) and changes from one iteration to the next as the constituents of the blocks change.

We now draw a proposal value $\theta_{j,l}^\dagger$ from the latter proposal density. If the proposal value violates any of the constraints it is rejected immediately. Otherwise, the proposed value is taken as the new value of the block with the M–H probability of move given by

$$\alpha_l(\theta_{j,l}, \theta_{j,l}^\dagger|\theta_{j,-l}, \mathbf{y}) = \min \left\{ \frac{f(\mathbf{y}|\theta_{j,l}^\dagger, \theta_{j,-l})\pi(\theta_{j,l}^\dagger) t(\theta_{j,l}|\hat{\theta}_{j,l}, \mathbf{V}_{j,l}, \nu)}{f(\mathbf{y}|\theta_{j,l}, \theta_{j,-l})\pi(\theta_{j,l}) t(\theta_{j,l}^\dagger|\hat{\theta}_{j,l}, \mathbf{V}_{j,l}, \nu)}, 1 \right\}. \quad (6)$$

If the proposed value is rejected, the current value of the block is retained as the new value of that block. Step 3 of the algorithm is completed by repeating this process for each block.

As an instructive example, consider the following SSM to which no particular economic interpretation is attached:

$$\mathbf{y}_t = \mathbf{a} + \mathbf{B}\mathbf{s}_t + \mathbf{u}_t$$

$$\mathbf{s}_t = \mathbf{G}\mathbf{s}_{t-1} + \boldsymbol{\varepsilon}_t$$

where \mathbf{y}_t is a 10×1 vector of observables at time t , \mathbf{s}_t is a 5×1 vector of time- t unobserved (latent) states, \mathbf{a} , \mathbf{B} and \mathbf{G} are matrices of appropriate dimensions, $\mathbf{u}_t \sim \mathcal{N}_{10}(\mathbf{0}, \boldsymbol{\Sigma})$ and $\boldsymbol{\varepsilon}_t \sim \mathcal{N}_5(\mathbf{0}, \boldsymbol{\Omega})$. We impose the following identification restrictions: \mathbf{G} is diagonal, $\mathbf{B}_{i,i} = 1$ for $i = 1, \dots, 5$, $\mathbf{B}_{i,j} = 0$ for $i, j = 1, \dots, 5, j > i$, $\boldsymbol{\Sigma} = \text{diag}\{\sigma_i^2\}_{i=1}^{10}$ and $\boldsymbol{\Omega} = \mathbf{I}_5$. This leads to 60 unknown parameters that we collect in the vector $\boldsymbol{\theta}$. To ensure that the variances of the measurement errors are positive, we reparameterize $\boldsymbol{\Sigma}$ as

$$\boldsymbol{\Sigma}^* = \text{diag}\{\sigma_i^{2*}\}_{i=1}^{10}; \quad \sigma_i^{2*} \in \mathcal{R}$$

where $\sigma_i^{2*} = \exp(\sigma_i^{2*})$. Finally, we restrict the parameter space $\boldsymbol{\theta}$ to satisfy stationarity of the vector Markov process \mathbf{s}_t : $\boldsymbol{\theta}_s = \{\boldsymbol{\theta} : \text{abs}(\text{eig}(\mathbf{G})) < 1\}$, which in this case is trivially satisfied if $\mathbf{G}_{i,i} < 1, i = 1, \dots, 5$.

We generate 200 observations from the data generating process $\boldsymbol{\theta}_{DCP}$, summarized in Table 1. The likelihood function $f(\mathbf{y}|\boldsymbol{\theta})$ can be calculated through the Kalman filter as outlined in Section 2. The model is completed by specifying a prior distribution $\pi(\boldsymbol{\theta})$ of the parameters. For notational convenience, we denote the parameters in the principal diagonal of \mathbf{G} as $\boldsymbol{\theta}_1 = \text{vecr}(\{\mathbf{G}_{i,i}\}), i = 1, \dots, 5$, those in \mathbf{a} as $\boldsymbol{\theta}_2$, the free parameters in the lower triangular portion of \mathbf{B} as $\boldsymbol{\theta}_3 = \text{vecr}(\{\mathbf{B}_{i,j}\}), i, j = 2, \dots, 5, j < i$, the remaining parameters in \mathbf{B} as $\boldsymbol{\theta}_4 = \text{vecr}(\{\mathbf{B}_{i,j}\}), i = 6, \dots, 10, j = 1, \dots, 5$ and the diagonal elements of $\boldsymbol{\Sigma}^*$ as $\boldsymbol{\theta}_5 = \{\sigma_i^{2*}\}_{i=1}^{10}$, where vecr

² The covariance matrix computed as the inverse of the negative Hessian is not always guaranteed to be positive definite, especially in the context of DSGE models where the target can be highly irregular. When the negative Hessian is not positive definite, we compute the modified cholesky R of the negative Hessian. Subsequently, we specify $(RR')^{-1}$ as the variance of the proposal density. While there are several versions of the modified cholesky, our implementation is based on algorithm 3.4 in Nocedal and Wright (2000).

Table 1
1-DGP and prior moments of parameters in SSM example.

Param.	DGP					Prior				
$\mathbf{G}_{1,1}, \dots, \mathbf{G}_{5,5}$	0.80	0.20	0.75	0.60	0.10	0.50 (5.00)	0.50 (5.00)	0.50 (5.00)	0.50 (5.00)	0.50 (5.00)
a_1, \dots, a_5	0.20	1.40	1.80	0.10	0.90	0.50 (5.00)	0.50 (5.00)	0.50 (5.00)	0.50 (5.00)	0.50 (5.00)
a_6, \dots, a_{10}	1.00	2.00	0.10	2.20	1.50	0.50 (5.00)	0.50 (5.00)	0.50 (5.00)	0.50 (5.00)	0.50 (5.00)
\mathbf{B}	1.00	–	–	–	–	–	–	–	–	–
	0.50	1.00	–	–	–	0.00 (5.00)	–	–	–	–
	0.60	0.00	1.00	–	–	0.00 (5.00)	0.00 (5.00)	–	–	–
	0.00	0.20	–0.10	1.00	–	0.00 (5.00)	0.00 (5.00)	0.00 (5.00)	–	–
	–0.20	0.00	–0.70	0.00	1.00	0.00 (5.00)	0.00 (5.00)	0.00 (5.00)	0.00 (5.00)	–
	0.00	0.00	–0.40	–0.50	0.00	0.00 (5.00)	0.50 (5.00)	0.00 (5.00)	0.00 (5.00)	0.00 (5.00)
	0.30	0.20	0.00	0.00	–0.30	0.00 (5.00)	0.00 (5.00)	0.50 (5.00)	0.00 (5.00)	0.00 (5.00)
	–0.50	0.00	0.00	0.60	0.00	0.00 (5.00)	0.00 (5.00)	0.00 (5.00)	0.50 (5.00)	0.00 (5.00)
	0.00	–0.50	0.30	–0.10	0.00	0.00 (5.00)	0.00 (5.00)	0.00 (5.00)	0.00 (5.00)	0.50 (5.00)
	0.00	0.00	0.20	0.00	–0.40	0.00 (5.00)	0.00 (5.00)	0.00 (5.00)	0.00 (5.00)	0.50 (5.00)
	$\sigma_1^2, \dots, \sigma_5^2$	1.00	0.30	1.00	0.20	0.61 (0.63)	0.61 (0.63)	0.61 (0.63)	0.61 (0.63)	0.61 (0.63)
	$\sigma_6^2, \dots, \sigma_{10}^2$	0.50	1.00	1.00	0.75	0.61 (0.63)	0.61 (0.63)	0.61 (0.63)	0.61 (0.63)	0.61 (0.63)

Note: Prior variance in parentheses.

stands for row vectorization. Our specified joint prior distribution of the parameters takes the form

$$\pi(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}_1 | \mathbf{g}_0, \mathbf{V}_g) \mathcal{N}(\boldsymbol{\theta}_2 | \mathbf{a}_0, \mathbf{V}_a) \mathcal{N}(\boldsymbol{\theta}_3 | \boldsymbol{\theta}_{30}, \mathbf{V}_{\boldsymbol{\theta}_3}) \mathcal{N}(\boldsymbol{\theta}_4 | \boldsymbol{\theta}_{40}, \mathbf{V}_{\boldsymbol{\theta}_4}) \\ \times \mathcal{N}(\boldsymbol{\theta}_5 | \boldsymbol{\sigma}_0^*, \mathbf{V}_{\boldsymbol{\sigma}^*}) \mathbf{I}_{\boldsymbol{\theta}_5}.$$

For the hyperparameters, we let

$$\begin{aligned} g_{0k} &= 0.5; \quad k = 1, \dots, 5 \\ \mathbf{V}_g &= 5 \times \mathbf{I}_5 \\ a_{0k} &= 0.5; \quad k = 1, \dots, 10 \\ \mathbf{V}_a &= 5 \times \mathbf{I}_{10} \\ \theta_{30k} &= 0; \quad k = 1, \dots, 10 \\ \mathbf{V}_{\boldsymbol{\theta}_3} &= 5 \times \mathbf{I}_{10} \\ \theta_{40k} &= 0; \quad k = 1, \dots, 25 \\ \mathbf{V}_{\boldsymbol{\theta}_4} &= 5 \times \mathbf{I}_{25} \\ \sigma_{0k}^* &= -1; \quad k = 1, \dots, 10 \\ \mathbf{V}_{\boldsymbol{\sigma}^*} &= \mathbf{I}_{10}. \end{aligned}$$

These choices reflect the belief that a priori the parameters are independent, as well as that relatively little is known about them. Further, the $\mathcal{N}(-1, 1)$ prior for σ_i^* implies that σ_i is *lognormally* distributed with an approximate mean of 0.61 and variance of 0.63. The posterior distribution (up to the normalizing constant) can now be written as

$$\pi(\boldsymbol{\theta} | \mathbf{y}) \propto f(\mathbf{y} | \boldsymbol{\theta}) \times \pi(\boldsymbol{\theta}).$$

This is the target function that we would like to explore.

Following step 1 of the TaRB-MH algorithm, we initialize the sampler at $\boldsymbol{\theta} = (\mathbf{g}_0, \mathbf{a}_0, \boldsymbol{\theta}_{30}, \boldsymbol{\theta}_{40}, \boldsymbol{\sigma}_0^*)$. Then steps 2 and 3 are iterated 11,000 times as follows. In every iteration the parameters are first randomly grouped into various blocks. In our scheme this is accomplished by randomly permuting the index of the parameters at the beginning of an iteration. Then block 1 is initialized at the

first element of this shuffled index. A new block is started at every subsequent element with probability 0.15 (or in other words, with probability 0.85 that element is included in the current block). Once the blocks have been so formed in a given iteration, we proceed to step 3, setting the parameters of the SA algorithm at $t_0 = 5$, $a = 0.4$, $K = 8$, $b = 8$ and $s = 0.02$. For each block, we now find the mode and curvature of the current target density to construct a student- t proposal density with 15 degrees of freedom. We then apply the M–H step, revise the value of that block according to the probability given in (6), and repeat this process for the next block. The first 1000 draws are discarded as burn-ins and the last 10,000 are used to summarize the posterior distribution.

Fig. 1 shows the kernel smoothed histogram of the marginal posterior density for 8 select parameters along with the corresponding prior distribution. Notice that the marginal posterior densities for all the parameters except those in \mathbf{a} are roughly centered around the data generating process. Also, the parameters in \mathbf{a} rightly capture the true mean of the data. With 200 observations, the role of the prior distribution is diminished because the likelihood function carries substantial information about the parameters. The figure also includes plots of the autocorrelation function. As seen from these plots, the autocorrelation among the sampled draws drop to zero after a few lags.

To provide a comparative assessment of these results, we sample the posterior distribution in two further ways. In the first, we employ a single block RW-MH sampler where the variance of the increment of this random-walk chain was based on a modified cholesky factorization of (non-invertible) Hessian at the posterior mode. The variance was multiplied by a factor of 10^{-6} to achieve an acceptance rate of 30%. We then ran this chain for 1 million iterations following a burn-in of 250,000 iterations. In the second, we employ a fixed block version of the TaRB-MH algorithm, which we refer to as the TaFB-MH algorithm. In this scheme the parameters in \mathbf{G} , \mathbf{a} , \mathbf{B} , and $\boldsymbol{\Sigma}$ were grouped in separate blocks.

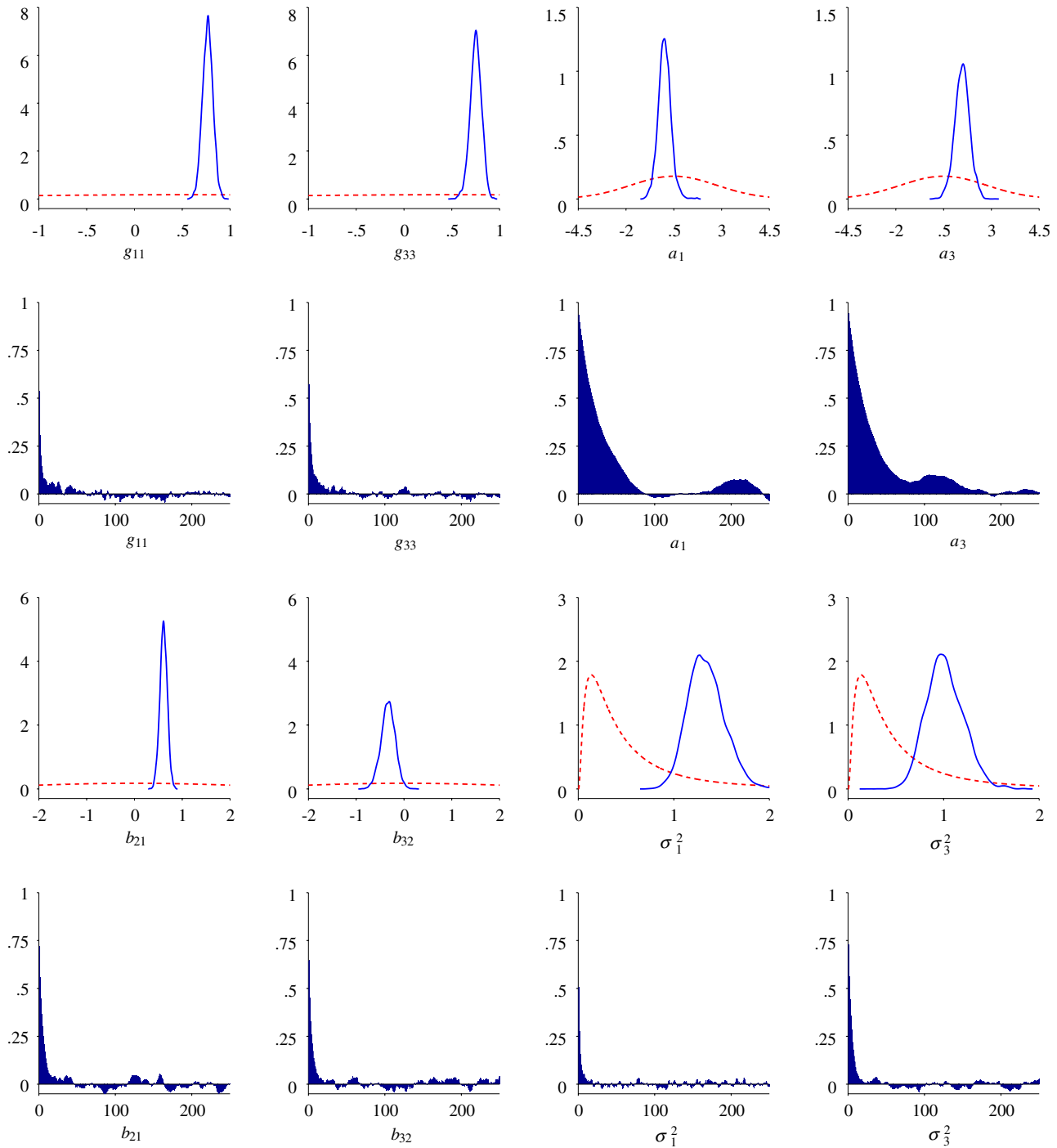


Fig. 1. Sampling results for the SSM example using the TaRB-MH algorithm: Marginal prior-posterior plots and autocorrelation functions for select parameters. Dotted lines indicate the prior distribution, whereas the bold lines indicate the posterior.

In addition the free parameters of \mathbf{B} were grouped into smaller blocks. In this sampling scheme, the SA tuning constants were the same as those in TaRB-MH algorithm. This scheme was run for 11,000 iterations of which the first 1000 draws were discarded.

We compare the efficiency of the sampling schemes primarily in terms of the serial correlations of the sampled outputs. We do not concern ourselves with convergence diagnostics (for example, Zellner and Min (1995)) since each of the MCMC schemes has the posterior distribution as the invariant distribution. Thus, following common practice, we focus on the estimated autocorrelations of the sampled output and the so-called inefficiency factors. The inefficiency factors approximate the ratio of the numerical

variance of the estimate from the MCMC chain relative to that from hypothetical iid draws. For a given sequence of draws the inefficiency factor is computed as

$$1 + 2 \sum_{l=1}^L \rho_k(l) \quad (7)$$

where $\rho_k(l)$ is the autocorrelation at lag l for the k th sequence, and L is the value at which the autocorrelation function tapers off (the higher order autocorrelations are also downweighted by a windowing procedure, but we ignore this aspect for simplicity). A well mixing sampler results in autocorrelations that decay to zero

Table 2aPosterior summary of parameters in **G**, **a** and **Σ**.

Param	TaRB-MH			TaFB-MH			RW-MH		
	Mean	St. dev.	Ineff	Mean	St. dev.	Ineff	Mean	St. dev.	Ineff
$G_{1,1}$	0.77	0.05	4.89	0.77	0.06	3.31	0.76	0.05	632.45
$G_{2,2}$	0.18	0.11	2.44	0.18	0.11	6.07	0.17	0.11	712.82
$G_{3,3}$	0.75	0.06	5.16	0.76	0.06	7.67	0.75	0.06	694.40
$G_{4,4}$	0.55	0.07	3.83	0.56	0.07	12.07	0.55	0.07	836.35
$G_{5,5}$	−0.09	0.12	2.52	−0.09	0.12	3.40	−0.08	0.12	757.94
a_1	0.02	0.34	29.56	0.01	0.38	6.38	0.00	0.24	4560.55
a_2	1.25	0.22	28.78	1.24	0.25	6.07	1.23	0.18	3144.02
a_3	1.47	0.39	38.06	1.46	0.45	7.79	1.48	0.33	1176.63
a_4	0.19	0.17	12.66	0.16	0.20	4.18	0.17	0.17	812.66
a_5	1.05	0.20	32.72	1.05	0.23	6.72	1.04	0.18	680.83
a_6	1.06	0.18	23.94	1.08	0.20	6.23	1.06	0.16	664.40
a_7	1.98	0.14	21.18	1.98	0.16	5.71	1.97	0.12	2515.08
a_8	0.10	0.22	27.53	0.09	0.25	5.99	0.11	0.19	2603.55
a_9	2.06	0.13	42.61	2.06	0.15	5.67	2.07	0.12	994.16
a_{10}	1.57	0.09	39.70	1.56	0.10	5.85	1.57	0.08	650.72
σ_1^2	1.33	0.14	2.81	1.33	0.14	6.81	1.34	0.13	4482.04
σ_2^2	0.30	0.42	6.99	0.29	0.43	25.69	0.27	0.49	1236.32
σ_3^2	1.00	0.19	4.97	1.00	0.20	29.19	0.92	0.09	5009.95
σ_4^2	0.16	0.45	4.05	0.16	0.43	13.65	0.16	0.46	817.80
σ_5^2	0.63	0.30	5.31	0.65	0.29	27.24	0.65	0.27	3483.13
σ_6^2	0.51	0.13	2.34	0.50	0.13	6.61	0.51	0.14	2058.79
σ_7^2	0.92	0.13	2.84	0.92	0.13	10.34	0.92	0.12	2347.75
σ_8^2	1.03	0.14	5.12	1.02	0.13	16.05	1.06	0.14	4995.28
σ_9^2	0.86	0.12	2.56	0.86	0.12	12.14	0.86	0.11	3681.40
σ_{10}^2	0.49	0.14	3.91	0.48	0.15	16.83	0.49	0.13	2649.94

Note: Inefficiency factors (ineff) are based on 5000 lags. The acceptance rates in the three schemes are as follows: 47% in the TaRB-MH, 20%, 45% and 60%, respectively, for the three blocks in the TaFB-MH and 35% in the RW-MH algorithms.

within a few lags (and therefore lead to low inefficiency factors), whereas a poorly mixing sampler exhibits persistent correlations even at large lags.

Not surprisingly, the results from the RW chain are not satisfactory with autocorrelations as high as 0.9 even at lags around 5000. In comparison, the TaFB-MH algorithm performs well with autocorrelations that decay to zero by 250 lags for most parameters. The best mixing is achieved, however, by the TaRB-MH algorithm.

The inefficiency factors from all three sampling schemes, together with a summary of the posterior distribution are given in Tables 2a and 2b. As can be seen in these tables, the TaRB-MH algorithm produces virtually i.i.d. draws for the parameters in **B**, **G** and **Σ**. For the remaining 8 parameters, the highest inefficiency factor in the case of the TaRB-MH algorithm is still less than 50. In comparison, the inefficiency factors range between 3 and 168 in the TaFB-MH algorithm. In particular, the sampling of the parameters in **B** are markedly more efficient with the TaRB-MH algorithm compared to the TaFB-MH. While these differences can be reduced slightly by further partitioning **B** into sub-blocks, the virtues of the TaRB-MH algorithm are quite clear, especially when no such clear blocking strategy is available as in the context of DSGE models. As we show below, the TaRB-MH algorithm performs spiritedly in both DSGE examples. Finally, note that in the RW-MH case, the inefficiency factors are all above 600, with most of them well above 1000, but more importantly, the exploration is highly dependent on the starting value of the chain. On the other hand, the effect of the starting value has no material effect on the TaRB-MH sampler and its effect wears off within 100 to 200 iterations.

3.1.1. Computational burden

All three MCMC schemes were coded in Gauss 8.0 and executed on a Windows Vista 64-bit machine with a 2.66 GHz Intel Core2 architecture CPU. The TaRB-MH algorithm took roughly 30 h to generate 11,000 draws and the TaFB-MH algorithm about 24 h of CPU time. In comparison, the RW-MH algorithm needed around 3 h to complete 1.25 million iterations but the time to find the

posterior mode before the run could start was close to 24 h. The latter was the time that it took for one of several threads of the optimizer to locate a suitable mode that could be utilized for specifying the variance of the RW increment. Considering the significant efficiency gains, which are obtained without the same tuning costs, investment in the TaRB-MH algorithm is well justified.

3.2. Extension to multi-modal problems

In this section we propose a simple extension to our algorithm, still in the same class of tailored MH methods, that can be used to efficiently sample multi-modal distributions. Before that, consider once again the RW-MH algorithm. As mentioned earlier, one strategy for the RW-MH algorithm is to thoroughly scan the parameter space for the various modes and then start the RW chain at the highest one. The problems with this approach are twofold. One is that the mass in the various modal regions is not known a priori. The other, more pronounced issue, concerns truly multi-modal distributions. In this regard, a suitable version of the RW-MH algorithm that is often used is an intuitive modification of the regular RW-MH algorithm. The idea is to propose occasional large jumps that break the attraction to the current mode. This method, however, has limited applicability and often works well only in small dimensional problems.

To describe our modification of the TaRB-MH algorithm, consider for simplicity the case where there are only two well defined modes. We assume that the modal values have been found by initial optimization. Let the location of the two modes be μ_1 and μ_2 . Also, let V_1 and V_2 denote the inverse of the negative Hessian matrices at the two modes. In addition to the proposal densities that are calculated within the TaRB-MH algorithm, define the following mixture proposal

$$q(\theta|y) = p t(\theta|\mu_1, V_1, v_1) + (1 - p) t(\theta|\mu_2, V_2, v_2),$$

where p is the probability of drawing a value from a t density with v_1 degrees of freedom that is centered at μ_1 and variance given

Table 2b
Posterior summary of parameters in **B**.

Param	TaRB-MH			TaFB-MH			RW-MH		
	Mean	St. dev.	Ineff	Mean	St. dev.	Ineff	Mean	St. dev.	Ineff
$B_{2,1}$	0.61	0.08	6.61	0.61	0.08	28.40	0.62	0.08	1661.23
$B_{3,1}$	0.55	0.14	10.71	0.55	0.14	102.08	0.56	0.13	2226.26
$B_{3,2}$	-0.33	0.15	4.25	-0.33	0.14	23.70	-0.34	0.14	873.90
$B_{4,1}$	0.07	0.10	11.17	0.06	0.11	57.06	0.08	0.11	2316.39
$B_{4,2}$	0.34	0.11	6.20	0.35	0.11	31.54	0.33	0.11	1084.67
$B_{4,3}$	-0.01	0.11	16.85	-0.01	0.10	51.07	-0.04	0.11	3999.89
$B_{5,1}$	-0.11	0.09	8.44	-0.11	0.09	83.88	-0.12	0.09	2272.21
$B_{5,2}$	0.14	0.13	4.39	0.14	0.13	24.28	0.14	0.13	1115.01
$B_{5,3}$	-0.51	0.08	4.12	-0.51	0.08	30.40	-0.51	0.08	1056.84
$B_{5,4}$	0.10	0.09	5.31	0.10	0.09	24.05	0.07	0.09	1507.12
$B_{6,1}$	0.02	0.08	11.25	0.02	0.09	168.78	0.02	0.08	3585.38
$B_{6,2}$	0.09	0.10	6.11	0.08	0.10	76.07	0.09	0.10	1112.23
$B_{6,3}$	-0.42	0.08	10.29	-0.41	0.07	59.29	-0.39	0.08	2769.35
$B_{6,4}$	-0.54	0.06	4.02	-0.54	0.06	45.81	-0.54	0.06	1020.02
$B_{6,5}$	0.12	0.09	4.26	0.12	0.09	29.85	0.12	0.09	1505.24
$B_{7,1}$	0.35	0.06	5.61	0.35	0.07	48.23	0.37	0.06	1073.97
$B_{7,2}$	0.38	0.11	4.55	0.37	0.11	43.72	0.36	0.10	914.44
$B_{7,3}$	-0.03	0.06	3.87	-0.03	0.06	29.72	-0.03	0.06	1728.41
$B_{7,4}$	0.00	0.07	4.13	0.00	0.08	50.99	0.03	0.06	4794.97
$B_{7,5}$	-0.39	0.11	3.62	-0.39	0.11	48.57	-0.39	0.11	1138.31
$B_{8,1}$	-0.58	0.09	7.99	-0.59	0.09	100.25	-0.57	0.09	1599.24
$B_{8,2}$	0.11	0.12	5.50	0.12	0.12	69.59	0.11	0.12	1928.35
$B_{8,3}$	0.01	0.09	10.63	0.00	0.09	85.08	-0.02	0.09	4632.40
$B_{8,4}$	0.55	0.08	4.20	0.55	0.08	47.34	0.56	0.08	1118.44
$B_{8,5}$	-0.16	0.11	2.38	-0.17	0.11	38.39	-0.16	0.11	2986.94
$B_{9,1}$	-0.10	0.07	8.47	-0.10	0.07	83.03	-0.12	0.07	3921.19
$B_{9,2}$	-0.59	0.11	6.31	-0.58	0.11	53.51	-0.57	0.11	1250.48
$B_{9,3}$	0.27	0.06	3.62	0.27	0.06	79.24	0.28	0.05	917.19
$B_{9,4}$	-0.11	0.07	4.36	-0.11	0.07	66.67	-0.11	0.07	909.41
$B_{9,5}$	0.00	0.10	4.52	0.01	0.08	44.00	0.02	0.09	2835.37
$B_{10,1}$	0.00	0.05	9.33	0.00	0.05	106.10	0.01	0.05	4926.01
$B_{10,2}$	-0.05	0.08	5.18	-0.06	0.08	47.93	-0.06	0.08	2366.47
$B_{10,3}$	0.17	0.05	3.64	0.17	0.05	61.71	0.17	0.05	1003.02
$B_{10,4}$	-0.02	0.05	4.35	-0.03	0.06	53.08	0.01	0.05	4923.15
$B_{10,5}$	-0.43	0.09	5.44	-0.45	0.09	34.40	-0.44	0.09	2611.21

by \mathbf{V}_1 . We now incorporate this mixture proposal in our regular TaRB-MH algorithm in the following way. Once every few (say a 100) iterations we update all the parameters in one block using this mixture proposal. That is, we first draw a θ^\dagger from the t density centered at the first mode with probability p (from the t density centered at the second mode with probability $1 - p$). This draw is subsequently accepted as the next value of the chain with the probability of move $\alpha_{MJ}(\theta, \theta^\dagger | \mathbf{y})$ given by

$$\alpha_{MJ}(\theta, \theta^\dagger | \mathbf{y}) = \min \left\{ \frac{f(\mathbf{y} | \theta^\dagger) \pi(\theta^\dagger)}{f(\mathbf{y} | \theta) \pi(\theta)} \frac{p t(\theta | \mu_1, \mathbf{V}_1, v_1) + (1-p) t(\theta | \mu_2, \mathbf{V}_2, v_2)}{p t(\theta^\dagger | \mu_1, \mathbf{V}_1, v_1) + (1-p) t(\theta^\dagger | \mu_2, \mathbf{V}_2, v_2)}, 1 \right\}. \quad (8)$$

The next (say) 100 draws of the chain are obtained from the TaRB-MH algorithm. At that point one returns to the mode-jumping step and the whole process is repeated.

It is interesting to note that the probability of move could also be defined in terms of the component of the proposal density from which θ^\dagger is drawn, as in the Hybrid MCMC algorithm described in Geweke (2005, Section 4.6). The latter scheme is implemented in the rejoinder section of An and Schorfheide (2007) but there is one crucial difference between the two approaches. In our approach, the mode jumping step is embedded within the multiple block TaRB-MH framework whereas the scheme implemented in the rejoinder section of An and Schorfheide (2007) operates on the entire set of parameters in a single block. This difference is important because sampling the parameters in one block is inefficient for making moves within a given mode, as we have argued earlier. As a result, our mode-jumping algorithm is more effective as we show in Section 4.2 in the context of their example.

In summary, the TaBMJ M-H algorithm has the following form.

Algorithm: TaBMJ-MH algorithm

Step 1 Initialize $\theta^{(0)} \in \mathcal{S}_L \cap \mathcal{S}_\Omega \cap \mathcal{S}_D$ and fix n_0 (the burn-in) and M (the MCMC sample size)

Step 2a If not a mode-jumping iteration, then

Step i Randomly generate blocks $(\theta_{j,1}, \theta_{j,2}, \dots, \theta_{j,p_j})$

Step ii For $l = 1, \dots, p_j$, sample $\theta_{j,l}^\dagger$ from the tailored proposal density $t(\theta_{j,l} | \hat{\theta}_{j,l}, \mathbf{V}_{j,l}, v)$ and accept this proposal with the probability of move $\alpha_l(\theta_{j,l}, \theta_{j,l}^\dagger | \theta_{j,-l}, \mathbf{y})$

Step 2b If a mode-jumping iteration, then

with probability p draw θ^\dagger from the proposal density $t(\theta | \mu_1, \mathbf{V}_1, v_1)$ and with probability $(1-p)$ draw θ^\dagger from the proposal density $t(\theta | \mu_2, \mathbf{V}_2, v_2)$ and accept proposal with the probability of move $\alpha_{MJ}(\theta, \theta^\dagger | \mathbf{y})$

Step 3 Repeat Steps 2a–2b $n_0 + M$ times, discard the draws from the first n_0 iterations and save the subsequent M draws $\theta^{(n_0+1)}, \dots, \theta^{(n_0+M)}$.

The generalization of this method to more than two modes is straightforward. If there are M distinct modes, denoted by μ_k , $k = 1, 2, \dots, M$, then one could let

$$q(\theta | \mathbf{y}) = \sum_{k=1}^M p_k t(\theta | \mu_k, \mathbf{V}_k, v_k); \quad \sum_k p_k = 1$$

and embed this proposal into the TaRB-MH algorithm as Step 2b in the preceding algorithm.

To illustrate our approach, consider a target function given by the bivariate mixture normal distribution

$$\pi(\theta | \mathbf{y}) = c \mathcal{N}(\mu_1, \Sigma_1) + (1 - c) \mathcal{N}(\mu_2, \Sigma_2) \quad (9)$$

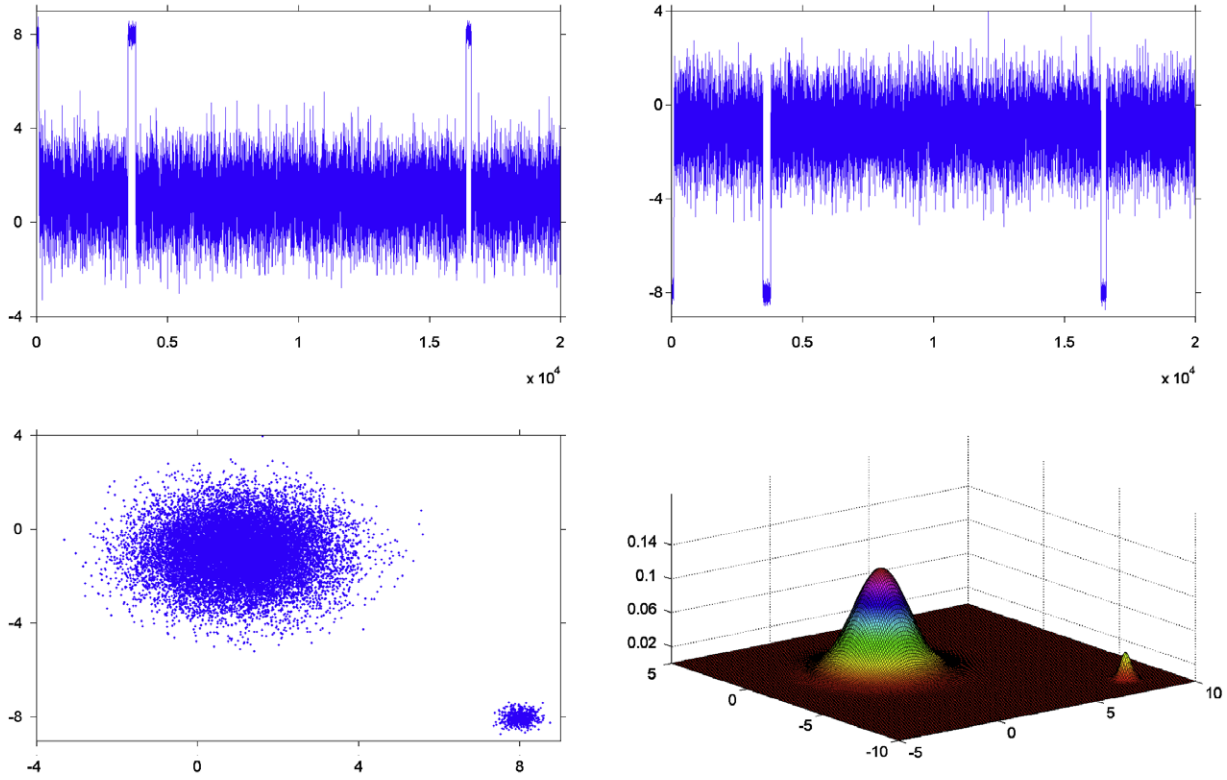


Fig. 2. Sampling from a bivariate contaminated normal using the TaRBMJ-MH algorithm: The top panel plots the actual draws (without any burn-ins) against time for the two variables. The bottom left panel shows a bivariate scatter plot of the draws. The actual target function is the 3-dimensional surface plot in the bottom right panel.

where $\mu_1 = [1, -1]'$, $\mu_2 = 6\mu_1$, $\Sigma_1 = \text{diag}(1.3, 1.3)$, $\Sigma_2 = \text{diag}(0.05, 0.05)$ and $c = 0.99$. The motivation for the specific choice of c comes from the posterior distribution in the [An and Schorfheide \(2007\)](#) model. The significance of c is that the difference in the ordinates at the two modes is roughly 1.34 in the log scale. Although this is a simple problem in two dimensions, the challenge arises because of the tiny second mode in the tails of an otherwise unimodal distribution. For this problem, we initialize the TaRBMJ-MH algorithm at the low mode. The mode jumping step is proposed every 100 iterations with $p = 0.5$. The simulation results from 20,000 draws are shown in [Fig. 2](#). The proposal densities used for this experiment are both student t -distributions, with 5 degrees of freedom for the regular TaRB-MH algorithm and 30 degrees of freedom for the mode jumping step. For the sake of illustration we plot all 20,000 draws without any burn-ins to show when the initial jump from the low to the high mode occurred. The top row plots the time series of the draws for the two random variables. Note that the sampler rightly revisits the low mode roughly 1% of the time. The bottom row includes a bivariate scatter plot of the draws (left) along with a 3 dimensional surface plot of the actual density.

For a second example, consider a four component mixture of twelve dimensional normals. To make the problem interesting, the mean and the covariance matrix of each of the four normal components are selected so that the modal regions are widely separated from one another. Specifically, let

$$\mu = [1.41, 0.81, 0.49, 0.80, 1.07, 0.30]'$$

The four modes are then specified as follows: $\mu_1 = [\mu', 1.5 \times \mu']'$, $\mu_2 = [-10 \times \mu', 10 \times \mu']'$, $\mu_3 = [10 \times \mu', 5 \times \mu']'$ and $\mu_4 = [4 \times \mu', -15 \times \mu']'$. These choices lead to the following modal values

$$\mu_1: (1.41, 0.81, 0.49, 0.80, 1.07, 0.30, 2.11, 1.21, 0.73, 1.20, 1.61, 0.45)$$

$$\begin{aligned} \mu_2: & (-14.07, -8.09, -4.86, -7.97, -10.74, -2.98, 14.07, \\ & 8.09, 4.86, 7.97, 10.74, 2.98) \\ \mu_3: & (14.07, 8.09, 4.86, 7.97, 10.74, 2.98, 7.04, 4.05, 2.43, \\ & 3.99, 5.37, 1.49) \\ \mu_4: & (5.63, 3.24, 1.95, 3.19, 4.30, 1.19, -21.11, \\ & -12.14, -7.30, -11.96, -16.11, -4.48). \end{aligned}$$

To describe the covariance matrices in the four components, let

$$\Sigma^1 = \begin{bmatrix} 0.0885 & -0.0023 & 0.0077 & 0.0041 & -0.0229 & -0.0025 \\ -0.0023 & 0.0055 & 0.0015 & 0.0028 & 0.0013 & 0.0001 \\ 0.0077 & 0.0015 & 0.0031 & 0.0018 & -0.0011 & -0.0002 \\ 0.0041 & 0.0028 & 0.0018 & 0.0029 & 0.0004 & -0.0012 \\ -0.0229 & 0.0013 & -0.0011 & 0.0004 & 0.0169 & 0.0004 \\ -0.0025 & 0.0001 & -0.0002 & -0.0012 & 0.0004 & 0.0024 \end{bmatrix}$$

$$\Sigma^2 = \begin{bmatrix} 0.1365 & -0.0009 & 0.0063 & 0.0075 & -0.0119 & -0.0101 \\ -0.0009 & 0.0082 & 0.0021 & 0.0048 & 0.0045 & -0.0049 \\ 0.0063 & 0.0021 & 0.0020 & 0.0018 & 0.0011 & -0.0018 \\ 0.0075 & 0.0048 & 0.0018 & 0.0137 & 0.0042 & -0.0078 \\ -0.0119 & 0.0045 & 0.0011 & 0.0042 & 0.0153 & -0.0051 \\ -0.0101 & -0.0049 & -0.0018 & -0.0078 & -0.0051 & 0.0096 \end{bmatrix}.$$

These matrices are chosen to mimic the covariance structure of the parameters at the two modes in the [An and Schorfheide \(2007\)](#) model that we discuss in the following section. Subsequently, the four covariances matrices are defined as

$$\begin{aligned} \mathbf{V}_1 &= \begin{bmatrix} \Sigma^1 & 0 \\ 0 & \Sigma^1 \end{bmatrix} & \mathbf{V}_2 &= \begin{bmatrix} \Sigma^1 & 0 \\ 0 & \Sigma^2 \end{bmatrix} \\ \mathbf{V}_3 &= \begin{bmatrix} \Sigma^2 & 0 \\ 0 & \Sigma^1 \end{bmatrix} & \mathbf{V}_4 &= \begin{bmatrix} \Sigma^2 & 0 \\ 0 & \Sigma^2 \end{bmatrix}. \end{aligned}$$

Finally, 5%, 75%, 5% and 15% probabilities are assigned to the four components, respectively. Our target distribution then takes the form

$$\begin{aligned} \pi(\theta|\mathbf{y}) &= 0.05 \mathcal{N}(\mu_1, \mathbf{V}_1) + 0.75 \mathcal{N}(\mu_2, \mathbf{V}_2) \\ &\quad + 0.05 \mathcal{N}(\mu_3, \mathbf{V}_3) + 0.15 \mathcal{N}(\mu_4, \mathbf{V}_4). \end{aligned}$$

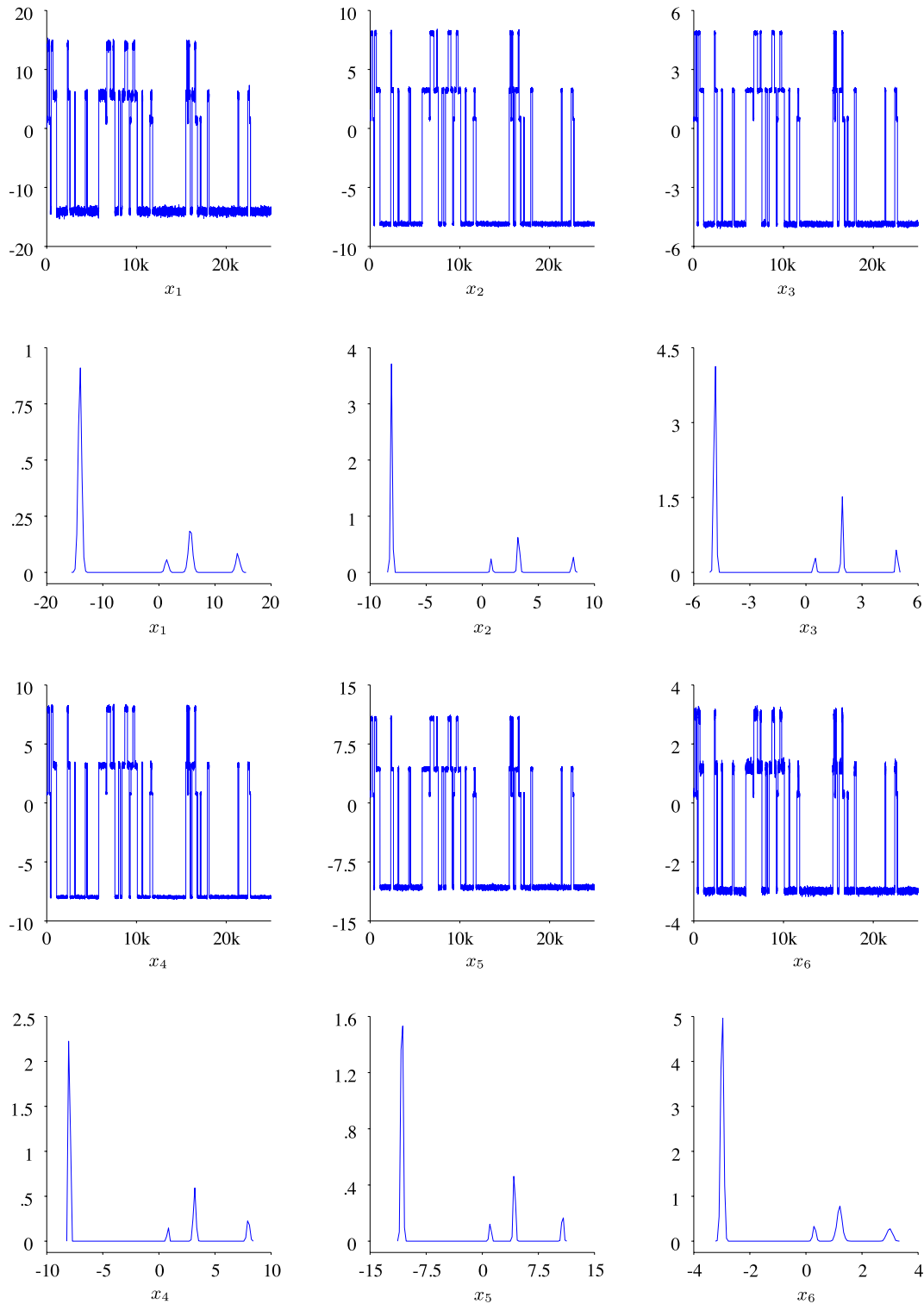


Fig. 3. Sampling from a four component mixture of 12-variate normals using the TaBMJ-MH algorithm: Rows 1 and 3 show the sampled draws against time (without any burn-ins). The simulated marginal density (kernel smoothed histogram) plots for the selected variables are shown in rows 2 and 4.

Notice that these covariance matrices imply that the first six variables are independent of the remaining six in each of the four components. This allows for a natural blocking strategy that involves grouping the first six variables in one block and the remaining six in a second block. Indeed, as one might anticipate, such blocking does lead to the most efficient sampling within each of the modal regions. However, because such information is usually not known a priori, it is illustrative to work with fully randomized blocks in the TaRB-MH steps within the TaBMJ-MH algorithm to

demonstrate its potential in the context of a known, closed form target function.

The results from 25,000 iterations of the TaBMJ-MH algorithm are summarized in Fig. 3. Rows 1 and 3 plot the time series of the actual draws for the first six variables (the results are identical for all twelve variables). Notice the fluid movement of the sampler across the four modes. Also worth mentioning is the fact that the sampler rightly spends roughly the same proportion of time in each of the modal regions as the probability content in those regions.

Finally, the kernel smoothed histogram of the draws are shown in rows 2 and 4. These plots indicate an accurate replication of the target density.

3.3. Marginal likelihood computation

One of the advantages of the Bayesian framework is that it is possible to compare competing models in a formal way through marginal likelihoods and Bayes factors. In this section we show how the framework of Chib (1995), and its M–H version in Chib and Jeliazkov (2001), can be modified to accommodate the randomized block approach proposed in this paper.

In the Chib framework, one starts by noting that the marginal likelihood (which is integral of the sampling density with respect to the prior density) can be expressed in the form

$$m(\mathbf{y}) = \frac{f(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta}|\mathbf{y})}, \quad (10)$$

by a rearrangement of Bayes theorem. Importantly, this is an identity in $\boldsymbol{\theta}$. Thus, the marginal likelihood can be obtained by evaluating this identity at a value $\boldsymbol{\theta}^*$ (say), which is usually taken to be a high density point in the support of the posterior. Clearly, the numerator components in the latter expression are readily computed. As for the component in the denominator, we follow the approach developed in Chib (1995) and decompose the posterior ordinate into marginal and conditional components each of which is then estimated in turn.

In order to apply this approach we fix the number of blocks in the posterior ordinate estimation step to the average number of blocks (say B) that are realized in the TaRB-MH run. Then, we construct the blocks $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_B$ with each block consisting of randomly chosen components from $\boldsymbol{\theta}$ and write

$$\pi(\boldsymbol{\theta}^*|\mathbf{y}) = \pi(\boldsymbol{\theta}_1^*|\mathbf{y})\pi(\boldsymbol{\theta}_2^*|\mathbf{y}, \boldsymbol{\theta}_1^*) \dots \pi(\boldsymbol{\theta}_B^*|\mathbf{y}, \boldsymbol{\theta}_1^*, \dots, \boldsymbol{\theta}_{B-1}^*). \quad (11)$$

In this decomposition, the typical ordinate is $\pi(\boldsymbol{\theta}_l^*|\mathbf{y}, \boldsymbol{\theta}_1^*, \dots, \boldsymbol{\theta}_{l-1}^*)$, $l = 1, \dots, B$, which can now be estimated in a manner akin to that in Chib (1995) by utilizing the output from appropriate reduced runs in which the parameters in $\boldsymbol{\theta}_l$ form a fixed block, and the remaining free parameters form random blocks.

Specifically, following Chib and Jeliazkov (2001), the simulation-consistent estimate of the l th ordinate is obtained as

$$\begin{aligned} \hat{\pi}(\boldsymbol{\theta}_l^*|\mathbf{y}, \boldsymbol{\theta}_1^*, \dots, \boldsymbol{\theta}_{l-1}^*) \\ = \frac{n_1^{-1} \sum_{g=1}^{n_1} \alpha(\boldsymbol{\theta}_l^{(g)}|\mathbf{y}, \boldsymbol{\theta}_1^*, \dots, \boldsymbol{\theta}_{l-1}^*, \boldsymbol{\Psi}^{l+1,(g)}) q_l(\boldsymbol{\theta}_l^*|\boldsymbol{\Psi}_{l-1}^*, \boldsymbol{\Psi}^{l+1,(g)}, \mathbf{y})}{n_1^{-1} \sum_{j=1}^{n_1} \alpha(\boldsymbol{\theta}_l^{(j)}|\mathbf{y}, \boldsymbol{\theta}_1^*, \dots, \boldsymbol{\theta}_{l-1}^*, \boldsymbol{\Psi}^{l+1,(j)})} \end{aligned} \quad (12)$$

where $\boldsymbol{\Psi}_{l-1} = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_{l-1})$, $\boldsymbol{\Psi}^{l+1} = (\boldsymbol{\theta}_{l+1}, \dots, \boldsymbol{\theta}_B)$, and $\alpha(\boldsymbol{\theta}_l^{(g)}|\mathbf{y}, \boldsymbol{\theta}_1^*, \dots, \boldsymbol{\theta}_{l-1}^*, \boldsymbol{\Psi}^{l+1,(g)})$, $q_l(\boldsymbol{\theta}_l^*|\boldsymbol{\Psi}_{l-1}^*, \boldsymbol{\Psi}^{l+1,(g)}, \mathbf{y})$ and $\alpha(\boldsymbol{\theta}_l^{(j)}|\mathbf{y}, \boldsymbol{\theta}_1^*, \dots, \boldsymbol{\theta}_{l-1}^*, \boldsymbol{\Psi}^{l+1,(j)})$ are obtained from (5)–(6) with $\boldsymbol{\theta}_{-l} \equiv (\boldsymbol{\Psi}_{l-1}, \boldsymbol{\Psi}^{l+1})$.

In the preceding expression, the average in the numerator is with respect to draws from the conditional distribution $\pi(\boldsymbol{\theta}_l^{(g)}|\mathbf{y}, \boldsymbol{\theta}_1^*, \dots, \boldsymbol{\theta}_{l-1}^*, \boldsymbol{\Psi}^{l+1,(g)})$. Accordingly, the draws $\{\boldsymbol{\theta}_l^{(g)}|\mathbf{y}, \boldsymbol{\theta}_1^*, \dots, \boldsymbol{\theta}_{l-1}^*, \boldsymbol{\Psi}^{l+1,(g)}\}$ in the numerator are obtained from the l th stage reduced MCMC run, in which the parameters in the preceding blocks, collected in $\boldsymbol{\Psi}_{l-1}$, are held fixed at $\boldsymbol{\Psi}_{l-1}^*$. Furthermore, the sampling in each of these reduced runs is implemented by the TaRB-MH algorithm, with the parameters in $\boldsymbol{\theta}_l$ forming a fixed block, and the parameters in $\boldsymbol{\Psi}^{l+1}$ forming random blocks. As in the main run, this randomization serves to increase the efficiency of the sampling in the reduced runs.

On the other hand, the average in the denominator is taken with respect to the conditional product measure $\pi(\boldsymbol{\Psi}^{l+1}|\mathbf{y}, \boldsymbol{\Psi}_l^*) q_l(\boldsymbol{\theta}_l|\mathbf{y},$

Table 3

Log marginal likelihood estimates (with numerical standard errors) for the 4, 5 and 6 dimensional dynamic factor models based on the output from the TaRB-MH algorithm.

4 factor	5 factor	6 factor
–3239.37 (0.0942)	–3235.28 (0.0900)	–3257.80 (0.0970)

$\boldsymbol{\Psi}_{l-1}^*, \boldsymbol{\Psi}^{l+1})$. Conveniently, draws from $\pi(\boldsymbol{\Psi}^{l+1}|\mathbf{y}, \boldsymbol{\Psi}_l^*)$ are available from the calculation of the $(l+1)$ st stage numerator which are supplemented with a drawing of $\boldsymbol{\theta}_l$ from $q_l(\boldsymbol{\theta}_l|\mathbf{y}, \boldsymbol{\Psi}_{l-1}^*, \boldsymbol{\Psi}^{l+1})$. In the final stage, we draw n_1 values of $\boldsymbol{\theta}_B$ from $q_B(\boldsymbol{\theta}_B|\mathbf{y}, \boldsymbol{\Psi}_{B-1}^*)$ and compute the denominator average based on these draws.

Having estimated $\pi(\boldsymbol{\theta}^*|\mathbf{y})$, the log of the marginal likelihood is then available as

$$\begin{aligned} \log \hat{m}(\mathbf{y}) = \log f(\mathbf{y}|\boldsymbol{\theta}^*) + \log \pi(\boldsymbol{\theta}^*) \\ - \sum_{l=1}^B \log \hat{\pi}(\boldsymbol{\theta}_l^*|\mathbf{y}, \boldsymbol{\theta}_1^*, \dots, \boldsymbol{\theta}_{l-1}^*). \end{aligned} \quad (13)$$

We summarize this readily implementable calculation in algorithmic form as follows.

Algorithm: Marginal likelihood

Step 1 Partition $\boldsymbol{\theta}$ into B blocks, each block consisting of randomly chosen elements from $\boldsymbol{\theta}$

Step 2 In the l th stage, $l = 1, \dots, B$, repeat n_1 times:

- Generate $\boldsymbol{\theta}_l^{(g)}, \boldsymbol{\Psi}^{l+1,(g)}$ from $\pi(\boldsymbol{\theta}_l, \boldsymbol{\Psi}^{l+1}|\mathbf{y}, \boldsymbol{\Psi}_{l-1}^*)$ by the TaRB-MH algorithm, randomizing only over the parameters in $\boldsymbol{\Psi}^{l+1}$, and calculate the l th stage numerator summand in (12)
- Supplement the preceding draw with a draw $\boldsymbol{\theta}_{l-1}^{(g)}$ from $q_{l-1}(\boldsymbol{\theta}_{l-1}|\mathbf{y}, \boldsymbol{\Psi}_{l-2}^*, \boldsymbol{\Psi}^{l,(g)})$ and calculate the $(l-1)$ st stage denominator summand in (12)
- Store these values

Step 3 For the final stage, draw n_1 values $\{\boldsymbol{\theta}_B\}$ from $q_B(\boldsymbol{\theta}_B|\mathbf{y}, \boldsymbol{\Psi}_{B-1}^*)$ and compute the l th stage denominator in (12)

Step 4 Calculate the log of the marginal likelihood as in (13).

As an illustrative example, we revisit the dynamic factor model example introduced in Section 3.1. A question that is of considerable interest in the context of pure (atheoretical) factor models is the dimension of the (latent) state vector. Accordingly, we perform a comparative analysis of three different models \mathcal{M}_1 , \mathcal{M}_2 and \mathcal{M}_3 with four, five and six factors, respectively. Recall that the true model, one from which the data is generated, is \mathcal{M}_2 . Thus, to crosscheck whether this model is indeed better supported by the data, we estimate the marginal likelihood for each of the three models using the method described above and compute the Bayes factors. In all three models the prior distribution is of the same form as in Table 1 except for the adjustment to the dimensionality of the parameter space (54 in the case of the 4-factor model and 65 in the case of the 6-factor model). The marginal likelihood estimates, together with the numerical standard errors (see Chib (1995) for details), are summarized in Table 3.

In all three cases, the marginal likelihood estimates are based on a six stage scheme with a simulation sample size of 10,000 draws. As can be seen from the table, the difference in the log-marginal likelihood between the five factor and four factor model is roughly 4, and that between the five factor and six factor model is approximately 22. These results show that the correct \mathcal{M}_2 model is preferred over both the \mathcal{M}_1 and \mathcal{M}_3 models.

4. Applications

In this section we apply the methods discussed above to two DSGE models. In the first example we re-estimate the large scale DSGE model in Smets and Wouters (2007) using the TaRB-MH algorithm and compare the results to those in the published version that are based on the RW-MH algorithm. Subsequently, we illustrate the TaRBMJ-MH algorithm in the context of the bimodal problem in An and Schorfheide (2007).

4.1. Smets and Wouters (2007) model

Our first DSGE application is the model in Smets and Wouters (2007) (SW07). The SW07 model, which is close in spirit to that of Christiano et al. (2005), is fit to US macroeconomic data for the period 1966:1–2004:4. These models, together with those in Smets and Wouters (2003) and Del Negro et al. (2007) among others, constitute an important class of evolving, large scale, DSGE models featuring sticky nominal price and wage contracts, habit formation, variable capital utilization and investment adjustment costs.

The linearized model in SW07 comprises of 14 equations in 14 endogenous variables: output (y_t), consumption (c_t), investment (i_t), value of capital stock (q_t), capital (k_t^s), installed capital (k_t), capital utilization (z_t), rental rate of capital (r_t^k), price mark-up (μ_t^p), inflation (π_t), wage mark-up (μ_t^w), real wage (w_t), labor hours (l_t) and interest rate (r_t) under a sticky-price-wage setting, and 7 exogenous driving processes. This system is stacked with the corresponding version of the flexible-price-wage model that determines the potential output, y_t^* , together with c_t^* , i_t^* , l_t^* , k_t^* , k_t^{s*} , z_t^* , w_t^* , r_t^{k*} , q_t^* , μ_t^{w*} , μ_t^{p*} and r_t^* . A complete description of the model is given in Appendix.

For estimation, the data comprises quarterly time series of the log difference of real GDP, log difference of real consumption, log difference of real investment, log difference of real wage, log hours worked, log difference of GDP deflator and the federal funds rate for the US economy from 1966:I to 2004:IV. The endogenous variables relate to the vector of observables through the relation

$$y_t = \begin{bmatrix} dlGDP_t \\ dlCONS_t \\ dlINV_t \\ dlWAG_t \\ lHOURS_t \\ dlP_t \\ FEDFUNDS_t \end{bmatrix} = \begin{bmatrix} \bar{y} \\ \bar{y} \\ \bar{y} \\ \bar{y} \\ \bar{l} \\ \bar{\pi} \\ \bar{r} \end{bmatrix} + \begin{bmatrix} y_t - y_{t-1} \\ c_t - c_{t-1} \\ i_t - i_{t-1} \\ w_t - w_{t-1} \\ l_t \\ \pi_t \\ r_t \end{bmatrix} \quad (14)$$

where l denotes 100 times log and dl refers to the log difference. Also, \bar{y} is the quarterly trend growth rate of real GDP, consumption, investment and wages, $\bar{\pi}$ is the quarterly steady-state inflation rate and \bar{r} is the steady-state nominal interest rate. Of the 41 underlying parameters in the model, 5 are held fixed. The prior distribution of the remaining parameters (as specified in SW07) is summarized in Table 4. Our interest centers around the 36 dimensional posterior distribution resulting from the sampling density and prior distribution of the parameters.

4.1.1. Sampling results from the TaRB-MH algorithm

As one might anticipate, sampling the posterior distribution of the parameters in this model poses significant challenges on account of the high dimensionality of the parameter space. To appreciate the complexity of the problem, we begin by focusing on the task of finding the posterior mode. Our extensive experimentation with the SA optimizer using various starting values, stage lengths, scale factors, initial temperatures and temperature reduction factors produced an unnormalized posterior modal ordinate in the vicinity of -877.72 on the log scale. In comparison, the modal

Table 4

Prior distribution in the Smets and Wouters (2007) model.

Parameter	Density	Mean	Std. dev.
φ	Normal	4.00	1.50
σ_c	Normal	1.50	0.37
h	Beta	0.70	0.10
ξ_w	Beta	0.50	0.10
σ_l	Normal	2.00	0.75
ξ_p	Beta	0.50	0.10
ι_w	Beta	0.50	0.15
ι_p	Beta	0.50	0.15
ψ	Beta	0.50	0.15
Φ	Normal	1.25	0.12
r_π	Normal	1.50	0.250
ρ	Beta	0.75	0.10
r_y	Normal	0.12	0.05
$r_{\Delta y}$	Normal	0.12	0.05
$\bar{\pi}$	Gamma	0.62	0.10
$100(\beta^{-1} - 1)$	Gamma	0.25	0.10
\bar{l}	Normal	0.00	2.00
\bar{y}	Normal	0.40	0.10
α	Normal	0.30	0.05
σ_a	Inverse Gamma	0.10	2.00
σ_b	Inverse Gamma	0.10	2.00
σ_g	Inverse Gamma	0.10	2.00
σ_l	Inverse Gamma	0.10	2.00
σ_r	Inverse Gamma	0.10	2.00
σ_p	Inverse Gamma	0.10	2.00
σ_w	Inverse Gamma	0.10	2.00
ρ_a	Beta	0.50	0.20
ρ_b	Beta	0.50	0.20
ρ_g	Beta	0.50	0.20
ρ_l	Beta	0.50	0.20
ρ_r	Beta	0.50	0.20
ρ_p	Beta	0.50	0.20
ρ_w	Beta	0.50	0.20
μ_p	Beta	0.50	0.20
μ_w	Beta	0.50	0.20
ρ_{ga}	Beta	0.50	0.20

Remark: The Inverse Gamma distribution here refers to the Inverse Gamma type-I distribution as in the AS07 model.

ordinate in SW07 is -906.29 . This reiterates the point in Sims et al. (forthcoming) that it is difficult to find the dominant mode in large scale macroeconomic models. Consequently, relying on the mode as the starting value of a RW chain, with the variance of the increment obtained from the negative inverse of the Hessian at the mode, is not necessarily a reliable strategy. As we have argued above, the TaRB-MH (because it does not work with the entire parameter vector in one block) does not suffer from these problems. It finds even higher density regions than those found by a well tuned SA optimizer and it explores the parameter space more fully than the RW-MH algorithm.

Following the template in Section 3.1, we initialize the TaRB-MH chain at the prior mean. The values chosen for the SA parameters are also similar. However, because it is more time consuming to evaluate the likelihood function in this problem (on account of the solution step), we reduced the number of stages K to 4 and the stage length increment b to 6. Though the tailoring is coarser, the results are not sensitive to these choices. Finally, we set the degrees of freedom in the student- t proposal density to 10. The results are summarized in Table 5. For comparison, this table also includes published the results from SW07. Our results are based on 10,000 iterations of the TaRB-MH algorithm following a burn-in of 1000 iterations, whereas the reported RW-MH results are based on 250,000 iterations beyond a burn-in of 50,000 iterations. The important differences in the results from the two algorithms are listed below.

- Posterior mean is significantly different for parameters $\bar{\pi}$ and \bar{l} . Note that the corresponding 90% intervals are shifted for these parameters. Interestingly, the posterior ordinate at the mean of the TaRB-MH sample is significantly higher than that

Table 5

Posterior summary of the structural and shock parameters in the SW07 model.

Parameter	SW07 posterior		TaRB-MH posterior		
	Mean	90% interval	Mean	90% interval	Inefficiency factors
φ	5.74	[3.97, 7.42]	5.77	[3.69, 8.06]	9.73
σ_c	1.38	[1.16, 1.59]	1.36	[1.03, 1.71]	12.14
h	0.71	[0.64, 0.78]	0.75	[0.66, 0.82]	15.64
ξ_w	0.70	[0.60, 0.81]	0.65	[0.52, 0.79]	54.28
σ_l	1.83	[0.91, 2.78]	1.98	[0.96, 3.21]	28.79
ξ_p	0.66	[0.56, 0.74]	0.62	[0.49, 0.75]	45.22
ι_w	0.58	[0.38, 0.78]	0.59	[0.31, 0.83]	8.65
ι_p	0.24	[0.10, 0.38]	0.23	[0.08, 0.41]	18.13
ψ	0.54	[0.36, 0.72]	0.59	[0.36, 0.81]	5.93
Φ	1.60	[1.48, 1.73]	1.57	[1.42, 1.74]	6.44
r_π	2.04	[1.74, 2.33]	2.00	[1.64, 2.37]	11.21
ρ	0.81	[0.77, 0.85]	0.80	[0.75, 0.85]	11.42
r_y	0.08	[0.05, 0.12]	0.08	[0.03, 0.13]	26.12
$r_{\Delta y}$	0.22	[0.18, 0.27]	0.23	[0.17, 0.29]	5.68
$\bar{\pi}$	0.78	[0.61, 0.96]	0.66	[0.48, 0.85]	3.81
$100(\beta^{-1} - 1)$	0.16	[0.07, 0.26]	0.16	[0.06, 0.29]	6.09
\bar{l}	0.53	[-1.3, 2.32]	0.95	[-0.07, 2.56]	9.39
$\bar{\gamma}$	0.43	[0.40, 0.45]	0.41	[0.37, 0.46]	9.51
α	0.19	[0.16, 0.21]	0.19	[0.15, 0.23]	4.68
σ_a	0.45	[0.41, 0.50]	0.46	[0.41, 0.53]	4.17
σ_b	0.23	[0.19, 0.27]	0.25	[0.18, 0.30]	15.23
σ_g	0.53	[0.48, 0.58]	0.53	[0.47, 0.59]	2.57
σ_l	0.45	[0.37, 0.53]	0.43	[0.34, 0.55]	33.14
σ_r	0.24	[0.22, 0.27]	0.25	[0.22, 0.28]	4.30
σ_p	0.14	[0.11, 0.16]	0.14	[0.10, 0.18]	14.89
σ_w	0.24	[0.20, 0.28]	0.26	[0.21, 0.32]	11.62
ρ_a	0.95	[0.94, 0.97]	0.96	[0.93, 0.98]	6.15
ρ_b	0.22	[0.07, 0.36]	0.21	[0.04, 0.49]	24.42
ρ_g	0.97	[0.96, 0.99]	0.98	[0.96, 0.99]	7.53
ρ_l	0.71	[0.61, 0.80]	0.74	[0.61, 0.86]	37.54
ρ_r	0.15	[0.04, 0.24]	0.15	[0.04, 0.30]	6.64
ρ_p	0.89	[0.80, 0.96]	0.89	[0.75, 0.98]	48.92
ρ_w	0.96	[0.94, 0.99]	0.98	[0.96, 1.00]	21.80
μ_p	0.69	[0.54, 0.85]	0.66	[0.38, 0.84]	38.23
μ_w	0.84	[0.75, 0.93]	0.83	[0.63, 0.94]	43.72
ρ_{ga}	0.52	[0.37, 0.66]	0.50	[0.32, 0.69]	2.61

Remark: The TaRB-MH sampler resulted in acceptance rates between 40 and 45 % for all the parameters.

at the mean of the SW07 RW-MH sample (-871.66 compared to -888.84). A comparative summary of the unnormalized posterior ordinates at the mode and the mean in the published results together with the results that we obtain using the SA algorithm for finding the mode and the TaRB-MH algorithm for finding the posterior mean are reported in Table 6.

- For the remaining parameters, the posterior means are roughly comparable. However, the TaRB-MH algorithm explores more of the posterior, as indicated by the wider 90% intervals for all but one parameter (ρ_w).
- In the case of the TaRB-MH algorithm, the inefficiency factors are in the range of 2.61–54.28, with most values below 30. In contrast, the inefficiency factors from our well tuned implementation of the RW-MH sampler (initialized at the mode found by SA) are above 2500. This points to the high degree of autocorrelation in the draws from the RW-MH sampler, as also noted in SW07. To illustrate the differences in the mixing of the two samplers over the parameters space, we plot the sampled draws from the posterior distribution for select parameters together with their ACFs in Fig. 4. The top panel corresponds to the draws from the TaRB-MH sampler and the bottom panel corresponds to the RW-MH sampler. As these plots show, the RW-MH chain is highly persistent with the ACFs retaining significant mass even at lag 1000. On the other hand, the TaRB-MH ACFs decay quickly (within 30–40 lags for most parameters).

We note in conclusion that convergence diagnostics (for instance, Zellner and Min (1995) amongst others) could also be useful in detecting the limited exploration of the RW-MH sampler.

Table 6

Summary of posterior ordinates at the mode and mean in the Smets and Wouters (2007) model.

	SW07	SA/TaRB-MH
Mode	−906.29	−877.72
Mean	−888.84	−871.66

4.2. An and Schorfheide (2007) model

In this section we turn our attention to multimodal posterior distributions. In the realm of DSGE models, one such case is that of An and Schorfheide (2007) (AS07) where, in a model with 13 parameters the posterior distribution appears to have two distinct modes separated by a deep valley. The authors show that despite a difference of approximately 8 on the log scale between the unnormalized posterior ordinates at the two modes, the RW-MH sampler is unable to escape the attraction of the individual modes even after a million iterations following a burn-in of 100,000 iterations. In contrast, we find that the TaRB-MH algorithm navigates quickly to the high modal region when initialized at the prior mean or at the low mode. Even more efficient movement between the modes occurs with the TaRB-MH method.

Consider then the linearized DSGE model in AS07 which is given by

$$\hat{y}_t = \mathbb{E}_t \hat{y}_{t+1} + \hat{g}_t - \mathbb{E}_t \hat{g}_{t+1} - \frac{1}{\tau} (\hat{r}_t - \mathbb{E}_t \hat{\pi}_{t+1} - \mathbb{E}_t \hat{z}_{t+1}) \quad (15)$$

$$\hat{\pi}_t = \beta \mathbb{E}_t \hat{\pi}_{t+1} + \kappa (\hat{y}_t - \hat{g}_t) \quad (16)$$

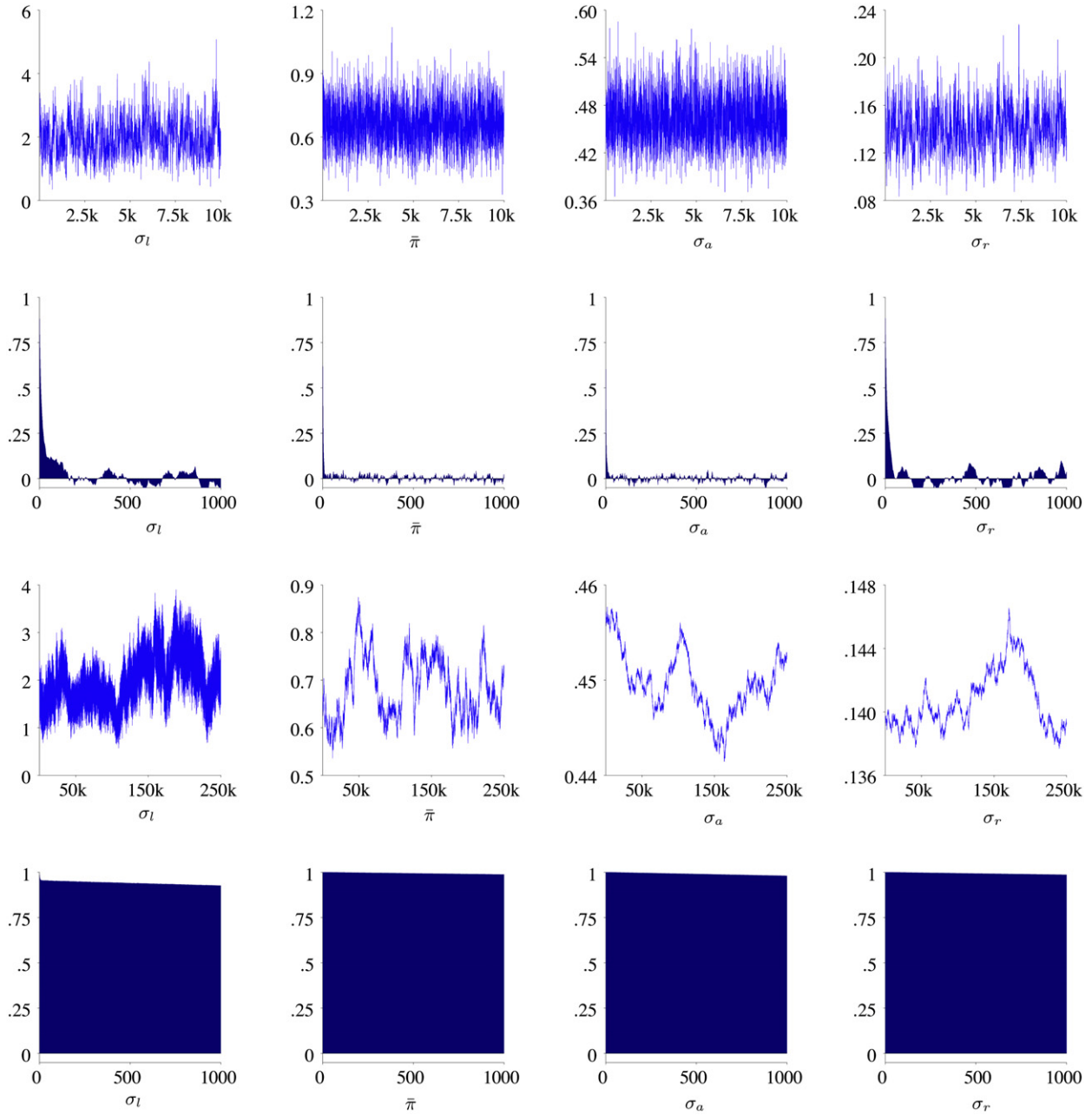


Fig. 4. Sampling results for the SW07 model: Time series plots of draws from the posterior and corresponding autocorrelation functions for select parameters using the TaRB-MH algorithm (top panel) and the RW-MH algorithm (bottom panel).

$$\hat{c}_t = \hat{y}_t - \hat{g}_t \quad (17)$$

$$\hat{r}_t = \rho_r \hat{r}_{t-1} + (1 - \rho_r) \psi_1 \hat{\pi}_t + (1 - \rho_r) \psi_2 (\Delta \hat{y}_t + \hat{z}_t) + \varepsilon_{r,t} \quad (18)$$

where y_t , π_t , r_t and c_t denote output, inflation, nominal interest rate and consumption, respectively, and the hats denote log-deviation of the variables from their steady state or average values. In addition to the interest rate shock $\varepsilon_{r,t}$ in the Taylor rule (Taylor, 1993) above, the system is driven by two other shocks, namely, a consumption shock and a technology shock. The evolution of these exogenous processes is governed by the independent AR(1) processes

$$\hat{g}_t = \rho_g \hat{g}_{t-1} + \varepsilon_{g,t} \quad (19)$$

$$\hat{z}_t = \rho_z \hat{z}_{t-1} + \varepsilon_{z,t} \quad (20)$$

where $\varepsilon_{i,t}$ is normally distributed with mean 0 and standard

deviation σ_i . As before, we collect the shocks in the vector $\varepsilon_t = [\varepsilon_{r,t}, \varepsilon_{g,t}, \varepsilon_{z,t}]'$. Then $\varepsilon_t \sim \mathcal{N}_3(\mathbf{0}, \mathbf{\Omega})$, where $\mathbf{\Omega} = \text{diag}[\sigma_r^2, \sigma_g^2, \sigma_z^2]$. The model is solved, subject to the determinacy constraint, to obtain a stochastic law of motion for the 7-dimensional state vector \mathbf{s}_t defined as

$$\mathbf{s}_t = [\hat{\pi}_t, \hat{y}_t, \hat{r}_t, \hat{g}_t, \hat{z}_t, \mathbb{E}_t \hat{\pi}_{t+1}, \mathbb{E}_t \hat{y}_{t+1}]'$$

The simulated data used in the fitting are assumed to be the series of quarterly observations on per capita GDP growth rates $\hat{y}_t = \gamma^Q + 100(\hat{y}_t - \hat{y}_{t-1} + \hat{z}_t)$, annualized inflation rates $\pi_t = \pi^A + 400\hat{\pi}_t$, and annualized nominal interest rate $r_t = \pi^A + r^A + 4\gamma^Q + 400\hat{r}_t$, where γ^Q , r^A , and π^A are related to the steady states of the relevant variables. Table 7 contains the parameters of the data generating process, along with the prior distribution, both of which are taken from AS07. It is assumed that the sample size is

Table 7

DGP and prior distribution for the model parameters in An and Schorfheide (2007).

Parameter	DGP	Prior Density	Mean	Standard deviation
τ	2.00	Gamma	2.00	0.50
κ	0.15	Gamma	0.20	0.10
ψ_1	1.50	Gamma	1.50	0.25
ψ_2	1.00	Gamma	0.50	0.25
ρ_r	0.60	Beta	0.50	0.20
ρ_g	0.95	Beta	0.80	0.10
ρ_z	0.65	Beta	0.66	0.15
r^A	0.40	Gamma	0.50	0.50
π^A	4.00	Gamma	7.00	2.00
γ^Q	0.50	Normal	0.40	0.20
σ_r	0.20	Inverse Gamma	0.50	0.26
σ_g	0.80	Inverse Gamma	1.25	0.65
σ_z	0.45	Inverse Gamma	0.63	0.33

Remark: The $IG(\tilde{\alpha}, \tilde{\beta})$ distribution on σ is the derived distribution from the assumption that $\sigma^2 \sim IG(\alpha, \beta)$. It is sometimes referred to as the Inverse Gamma type-1 distribution.

80. The SSM of interest is given by

$$\begin{bmatrix} \hat{Y}_t \\ \pi_t \\ r_t \end{bmatrix}_{y_t} = \begin{bmatrix} \gamma^{(Q)} \\ \pi^A \\ \pi^A + r^A + 4\gamma^Q \end{bmatrix}_a + \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & -1 \\ 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_B s_t \quad (21)$$

$$s_t = D(\theta)s_{t-1} + F(\theta)\epsilon_t. \quad (22)$$

4.2.1. Sampling results from the TaRBMJ-MH algorithm

As mentioned above, an interesting aspect of this model is the occurrence of possibly multiple modes. Specifically, by maximization of the posterior distribution with respect to the parameters one can find two distinct optima with the optima differing primarily in terms of the components $(\tau, \psi_2, \rho_r, \rho_z, \sigma_g, \sigma_z)$. The modal values are presented in Table 8. One point to note is that the difference in the (unnormalized) posterior ordinates at the two modes is around 8.3 on the log scale (-192.12 at the low mode vs. -183.78 at the high mode). One can understand the implication of this difference on the relative probability mass in the two regions by the following simple exercise.

Denote the high mode by μ_1 and the low mode by μ_2 . Assume that the underlying posterior distribution can be approximated by a two component mixture of 13-dimensional normals as follows

$$p \mathcal{N}(\theta|\mu_1, \Sigma_1) + (1-p) \mathcal{N}(\theta|\mu_2, \Sigma_2)$$

where the Σ s are the negative inverse Hessians at the respective modes. Under this working assumption, p is the only unknown. From the knowledge of the ordinates at the two modes one can write

$$\frac{p \mathcal{N}(\mu_1|\mu_1, \Sigma_1) + (1-p) \mathcal{N}(\mu_1|\mu_2, \Sigma_2)}{p \mathcal{N}(\mu_2|\mu_1, \Sigma_1) + (1-p) \mathcal{N}(\mu_2|\mu_2, \Sigma_2)} = \exp\{8\}$$

which yields $p = 0.99995$. This suggests that the mass in the second mode is around 0.005%, which is small enough to be ignorable. Arguably, this calculation is only suggestive of the actual probability of the low modal region. Nonetheless, it leads to the conjecture that the mass in the low modal region is not substantial as one might intuitively expect from the disparity in the heights of the two modes.

For sampling the posterior distribution in this model we initialized the TaRBMJ-MH sampler at the low mode.³ For the usual TaRB-MH iterations with fully randomized blocks, the parameters for the SA algorithm were chosen as follows. We set the number of stages K to 10, the increment of stage length b to 15, the initial temperature to 75 and the linear temperature reduction factor a to 0.8. Finally, the degrees of freedom in the t -proposal density was set to 2. We also assigned equal probability ($p = 0.5$) to both modes in the mode jumping proposal which was called in every 100th iteration. The mixture proposal used in this step was a t -density with 5 degrees of freedom. The sampler was run for 10,000 iterations without any burn-in to observe the movement of the sampler throughout the course of the MCMC iterations. While the acceptance rate is not very sensitive to these settings, it is worth noting that the resulting acceptance rate was around 45% for all the parameters.

As expected, the transition from the low modal region to the high modal region was almost instantaneous (in that the first time a value from the high mode component of the mixture distribution was proposed it was accepted). This jump occurred at the 200th iteration. However, the reverse transition through the mode jumping step occurred only once in the 300th iteration. Beyond this, any value generated through the low mode component in the mode jumping step got rejected. However, in the non-mode jumping steps, the usual TaRB-MH sampler continued exploring both modal regions efficiently. This points to the potential of this algorithm in the global exploration of the posterior distribution.

The results from the TaRBMJ-MH sampler are summarized in Table 8. This table also contains the modal information of the parameters. Particularly noteworthy in this table is the movement of the sampler across the two modal regions as indicated by the 90% intervals (column 6 in table). Also worth recognizing is the fine mixing of the sampler as indicated by the inefficiency factors. The highest inefficiency factor is around 50, with the average inefficiency factor across all the parameters being 27.⁴ These factors combined together suggest an efficient, well mixing sampler. The movement of the sampler across the support of the posterior is better illustrated in Fig. 5. The plots show the sampled draws from the posterior distribution for the six parameters that differ across the two modes, together with the respective modal values (indicated by the red lines). As one can see, the sampler visited the low modal region a few times. However, the proportion of draws from this region is not substantial enough to define a distinct mode in the marginal density plots.

A couple of remarks follow.

1. The first is that embedding the mode jumping step within the RW-MH scheme does not result in similar expansive exploration of the posterior distribution. As noted earlier, this is because it is practically infeasible to tune a RW chain to traverse both modal regions. Thus, once in the high modal (low modal) region, the RW sampler fails to move to the low modal (high modal) region by itself. In combination with the fact that the mode jumping step does not lead to frequent transitions to the low modal region means that exploration is limited to mostly the high modal region. In addition, the RW sampler does not explore the high modal region very effectively. Neither problem occurs with the TaRBMJ-MH algorithm. The TaRB-MH step within that algorithm explores the high-modal region more effectively and by itself produces moves intermittently to the low

³ The results from two other chains, one initialized at the dominant mode and the other initialized at the prior mean are virtually identical to those from this chain.

⁴ These inefficiency factors are low given the irregularity of the posterior surface in this problem and in relation to what can be achieved without recourse to our sampling scheme.

Table 8

Posterior sampling results using the TaRBMJ-MH algorithm for the An and Schorfheide (2007) model.

Parameter	Posterior modes		Posterior			
	Mode 1	Mode 2	Mean	Numerical S.E.	90% interval	Inefficiency factors
τ	2.05	1.41	2.12	0.0392	[1.04, 3.74]	29.91
κ	0.16	0.18	0.17	0.0059	[0.03, 0.42]	34.54
ψ_1	1.55	1.57	1.66	0.0149	[1.12, 2.42]	19.88
ψ_2	0.96	0.81	1.00	0.0101	[0.70, 1.35]	36.38
ρ_r	0.59	0.49	0.59	0.0054	[0.41, 0.72]	49.38
ρ_g	0.94	0.97	0.92	0.0033	[0.79, 0.98]	41.57
ρ_z	0.58	0.80	0.54	0.0094	[0.21, 0.83]	31.57
r^A	0.64	0.62	0.68	0.0083	[0.07, 1.43]	5.54
π^A	4.06	4.00	4.16	0.0212	[3.28, 5.52]	14.48
γ^Q	0.50	0.54	0.48	0.0050	[0.11, 0.80]	8.655
σ_r	0.22	0.24	0.23	0.0012	[0.18, 0.32]	11.61
σ_g	0.76	1.07	0.76	0.0120	[0.45, 1.33]	28.57
σ_z	0.54	0.30	0.61	0.0111	[0.30, 1.01]	38.86

Remark: The results reported in this table are based on the low mode as the starting value. The TaRBMJ-MH algorithm was run for 10,000 iterations, with the mode jumping step using the mixture distribution proposed every 100th iteration.

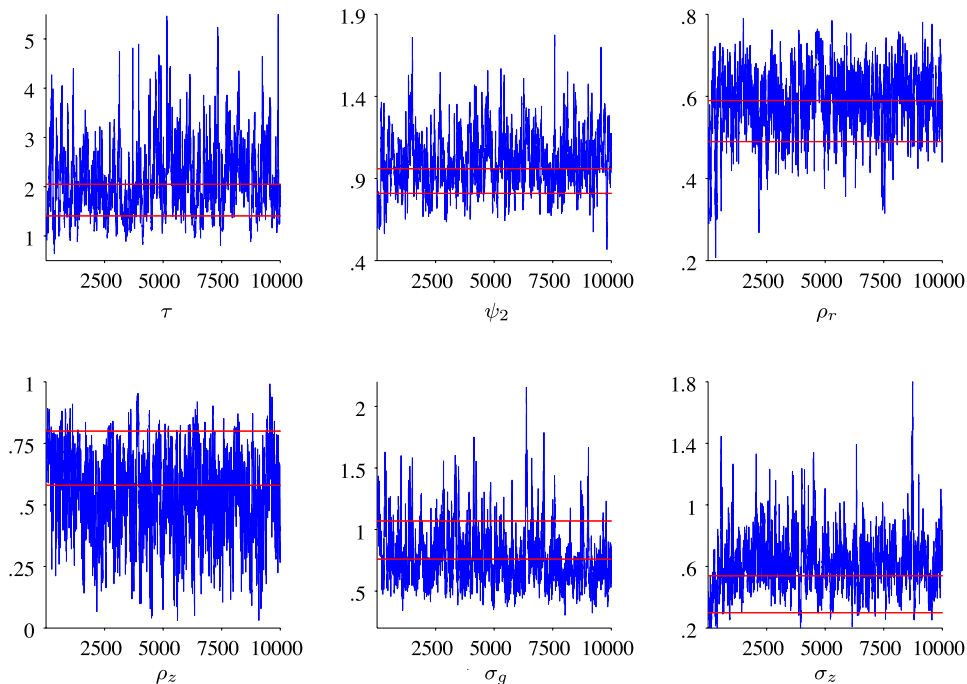


Fig. 5. Sampling results for the AS07 model using the TaRBMJ-MH algorithm: Time series plots of draws from the posterior for select parameters together with the modal values indicated by the red lines.

modal region (quite apart from the moves to that region in the mode-jumping step), thus leading to an exploration of the posterior that is difficult to achieve by any other means.

- The second remark relates to the mode jumping method discussed in the rejoinder section of An and Schorfheide (2007). In our efforts to replicate their results, we first executed that method with the settings that are reported in the rejoinder. We refer to this as Implementation I. In detail, An and Schorfheide (2007) construct the following mixture density

$$q(\theta|\mathbf{y}) = 0.5 t(\theta|\mu_l, c_l \mathbf{V}_l, v_l) + 0.5 t(\theta|\mu_h, c_h \mathbf{V}_h, v_h),$$

where μ_l (μ_h) and \mathbf{V}_l (\mathbf{V}_h) denote the low (high) mode and the variance at the low (high) mode, $c_l = 3$, $c_h = 4$, $v_l = 5$, and $v_h = 2$. Once a proposal θ^\dagger is generated from the j -th component of the mixture, the acceptance probability is calculated as

$$\tilde{\alpha}_j(\theta, \theta^\dagger|\mathbf{y}) = \min \left\{ \frac{f(\mathbf{y}|\theta^\dagger)\pi(\theta^\dagger)}{f(\mathbf{y}|\theta)\pi(\theta)} \frac{t(\theta|\mu_j, \mathbf{V}_j, v_j)}{t(\theta^\dagger|\mu_j, \mathbf{V}_j, v_j)}, 1 \right\}.$$

An and Schorfheide (2007) report that 25% of the draws generated from the high mode component were accepted.

In our implementation of this method we were unable to reproduce that rate. Instead, we found that fewer than 2% of the draws from the high mode component were accepted. We were able to increase the acceptance rate to 6.6% in Implementation II with the parameters $c_l = 2$, $c_h = 2$, $v_l = 2$, and $v_h = 1$. Neither implementation, however, is competitive with the TaRBMJ-MH algorithm as can be seen from comparing the results in Table 8 with those in Table 9. One clear difference is that the 90% intervals are narrower in Table 9 than those in Table 8. This suggests that the rejoinder method does not adequately explore the posterior surface. To illustrate this point, we provide in Fig. 6 a plot of the (unnormalized) posterior ordinate at each sampled draw. The top panel corresponds to the draws from the method in the rejoinder whereas the bottom panel depicts the results from the TaRBMJ-MH algorithm. These plots show the extensive movement of the TaRBMJ-MH sampler, particularly in the low modal region. In sum, these results suggest that the TaRB-MH algorithm supplemented by a mode jumping step is a substantial improvement over the mode jumping approach discussed in the rejoinder of An and Schorfheide (2007).

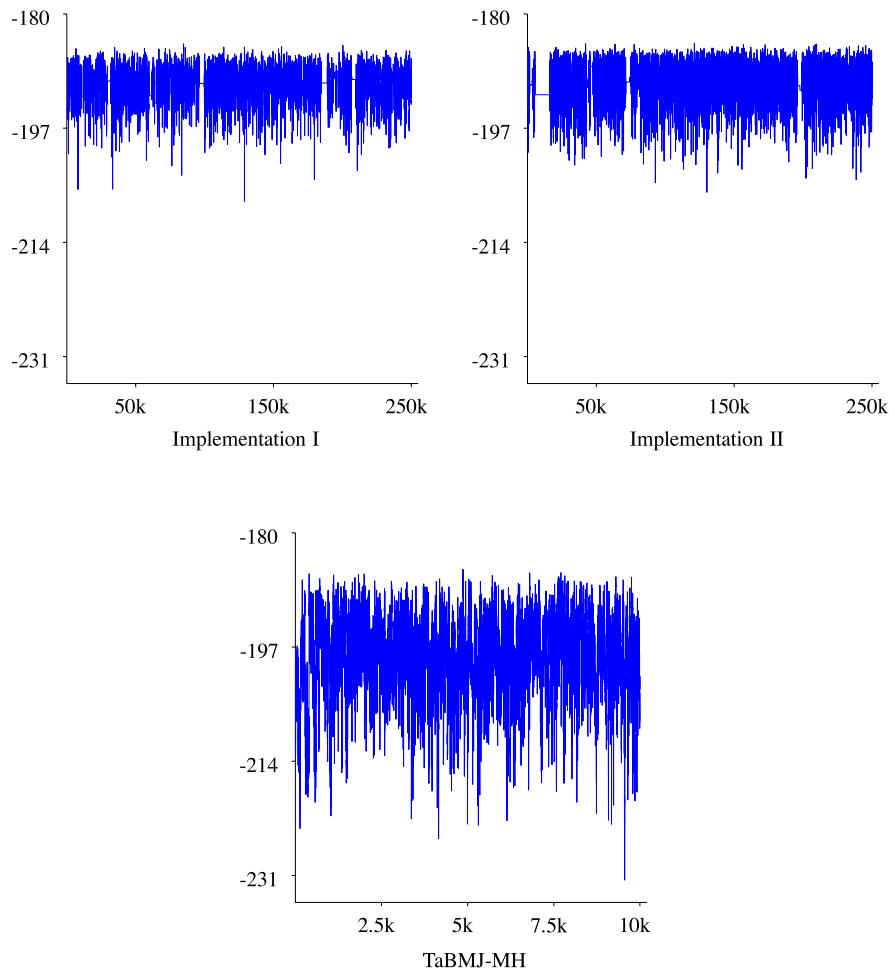


Fig. 6. Trace plot of posterior ordinate over draws from the Hybrid MCMC algorithm (top panel) and the TaRBMJ-MH algorithm (bottom panel) for the AS07 model.

Table 9

Posterior sampling results using the mode jumping algorithm in the rejoinder section of An and Schorfheide (2007).

Parameter	Implementation I		Implementation II	
	Mean	90% interval	Mean	90% interval
τ	2.14	[1.42, 3.14]	2.15	[1.51, 3.02]
κ	0.18	[0.08, 0.30]	0.18	[0.09, 0.30]
ψ_1	1.61	[1.28, 2.04]	1.61	[1.29, 2.04]
ψ_2	0.98	[0.81, 1.17]	0.97	[0.80, 1.17]
ρ_r	0.59	[0.49, 0.67]	0.59	[0.49, 0.67]
ρ_g	0.94	[0.88, 0.98]	0.93	[0.89, 0.97]
ρ_z	0.55	[0.30, 0.75]	0.54	[0.32, 0.75]
r^A	0.64	[0.13, 1.23]	0.63	[0.11, 1.19]
π^A	4.09	[3.33, 4.86]	4.08	[3.40, 4.77]
γ^Q	0.50	[0.26, 0.72]	0.51	[0.27, 0.74]
σ_r	0.23	[0.19, 0.28]	0.23	[0.20, 0.28]
σ_g	0.75	[0.55, 0.99]	0.74	[0.50, 0.99]
σ_z	0.59	[0.41, 0.82]	0.59	[0.41, 0.77]

Remark: Implementation I refers to the settings reported in the rejoinder section of An and Schorfheide (2007). Implementation II refers to the settings mentioned in this paper that result in slightly higher acceptance rates.

We conclude this section with a note on the marginal likelihood computation. In Section 3.2 we showed how the marginal likelihood can be estimated within the Chib (1995) framework when sampling is done with the TaRB-MH method and illustrated it in the context of an arbitrary large scale SSM example. In this section, we apply that method to the An and Schorfheide model. Recall that the marginal likelihood computation requires decomposing the

Table 10

Log marginal likelihood estimates (with numerical standard errors) for the An and Schorfheide (2007) model based on the outputs from the TaRB-MH and RW-MH algorithms.

TaRB-MH			RW-MH	
n_1	2 stage	3 stage	n_1	1 stage
5,000	−206.64 (0.1009)	−206.18 (0.0944)	75,000	−204.37 (0.7448)
10,000	−206.58 (0.0729)	−206.29 (0.0671)	150,000	−206.24 (0.4764)
15,000	−206.55 (0.0586)	−206.34 (0.0544)	250,000	−205.91 (0.4060)

posterior ordinate into marginal-conditional densities. Unlike the fixed block case discussed in Chib and Jeliazkov (2001), there is some freedom here in selecting the number of blocks (stages), as well as the components of these blocks. Therefore, it is worthwhile to consider the marginal likelihood estimate, and the numerical standard errors that result, as the number of stages (and the sample size n_1 in the reduced MCMC runs) are varied. Accordingly, we consider both two and three stage schemes for calculating the posterior ordinate. We consider these many stages because the average number of blocks in the full MCMC run for this model is around 3. We also compare our results to the estimate of the marginal likelihood when sampling is done with the help of the RW-MH algorithm. The estimates are summarized in Table 10.

For the two stage and three stage computations under the TaRB-MH algorithm, we calculate the marginal likelihood from 5000,

10,000 and 15,000 draws in the reduced runs. Under the single block RW-MH scheme, the computations are carried out in 1-stage, with the draws in the numerator of Eq. (13) being those from the posterior sampling and those in the denominator being those from $\mathcal{N}(\theta^*, c\Sigma)$, where Σ is the variance matrix computed at the posterior mode. In this calculation we use the output from our most efficient implementation of the RW-MH algorithm and illustrate the results for samples of sizes 75,000, 150,000 and 250,000.

It is noteworthy that the numerical standard error of the marginal likelihood estimate (which are reported in parentheses in the table) is lower under the more efficient sampling scheme. This is in keeping with the discussion in Chib and Jeliazkov (2001). As one can see from the table, the numerical standard errors under the less efficient RW-MH algorithm are about 7 to 8 times higher than those under the TaRB-MH algorithm for any sample length. These factors combined together suggest that a stable and efficient estimate of the marginal likelihood can be obtained with a sample size of 5000 from the TaRB-MH algorithm, with either 2 or 3 stages.

5. Conclusion

In this paper we provide new MCMC approaches for estimating DSGE models. The organizing principle is the sampling of the posterior density by a specially constructed version of a (randomized) multiple-block M-H algorithm that is designed to deal with the sort of considerations that arise in the context of these models. Specifically, the blocks and its constituents are constructed randomly and the proposal density in the M-H sampling is found from the output of SA. The combined thrust of these innovations proves extremely effective as we show in the context of several examples, including two DSGE models. We also provide a straightforward extension of our general approach that can be used for sampling distributions in which one suspects multiple modes. On both dimensions, this improves significantly on what has been achieved so far. Finally, we show how the model marginal likelihood, which is needed in the comparison of alternative DSGE models, can be computed efficiently from the output of our tailored randomized block sampling scheme.

Finally, it is worthwhile mentioning that 10,000 iterations of the TaRB-MH algorithm take on average 6 to 7 times longer to complete than 250,000 iterations of the RW-MH algorithm. This computing time however does not reflect the time required to tune the RW-MH algorithm, which, as noted above, increases substantially with the dimensionality of the parameter space. A more serious problem with a naive comparison of computing times is that it fails to recognize that the RW-MH simply does not work in many circumstances. It is therefore vital to have the correct answer even if it is costly to obtain in terms of computing time. A final point is that computing time is not a constant and has rapidly declined over time and can be expected to continue to fall in the near future. In sum, given that the methods proposed here are reliable and efficient, they have the potential for broadening the appeal and practicality of Bayesian methods in the fitting of DSGE models.

Acknowledgements

The authors would like to gratefully acknowledge the many useful comments from the editor and three reviewers.

Appendix. Linearized model in Smets and Wouters (2007)

This Appendix provides the complete SW07 model that includes the system of equations under both the sticky-price-wage and flexible price-wage settings (the latter marked by asterisks), together with the steady states of the relevant variables. We also

include the state vector \mathbf{s}_t , the vector of innovations \mathbf{e}_t and the vector of expectational errors $\boldsymbol{\eta}_t$ so that the canonical form of the model (for Sims' 2003 solution method) can be constructed. The writing of this section has benefited considerably from Giorgio Primiceri's Matlab code of the canonical representation.

The following auxiliary parameters (in terms of the parameters in Table 10) are useful for the calculations below: $\beta = 100(\hat{\beta} + 100)^{-1}$, $\gamma = \bar{\gamma}/100$, $\lambda_p = \phi_p - 1$. In addition, the following parameters are fixed as in SW07: $\delta = 0.025$, $\lambda_w = \phi_w - 1 = 0.5$, $g_y = 0.18$, $\varepsilon_w = \varepsilon_p = 10$. Also, for the steady state calculations below, define $l^{ss} = e^{\bar{l}}$. Subsequently, $l^{ss} = \bar{l}$.

The steady state values:

$$\begin{aligned}\tilde{r} &= 100(\beta^{-1}e^{\gamma\sigma_c} - 1) \\ \bar{r} &= \bar{\pi} + \tilde{r} \\ g_y &= 1/(1 - g_y) \\ r^{kss} &= \frac{e^{\gamma\sigma_c}}{\beta} - 1 + \delta \\ \mu^{pss} &= 1/(1 + \lambda_p) \\ w^{ss} &= \left[\frac{\mu^{pss}(1 - \alpha)^{(1-\alpha)}}{(\alpha - \alpha)(r^{kss})^\alpha} \right]^{(1/(1-\alpha))} \\ kl^{ss} &= (w^{ss}/r^{kss})(\alpha/(1 - \alpha)) \\ Fl^{ss} &= (kl^{ss})^\alpha - r^{kss}kl^{ss} - w^{ss} \\ yl^{ss} &= kl^{ss\alpha} - Fl^{ss} \\ k^{ss} &= kl^{ss}l^{ss} \\ i^{ss} &= (1 - (1 - \delta)e^{-\gamma})k^{ss}e^{\gamma} \\ F &= Fl^{ss}l^{ss} \\ y^{ss} &= yl^{ss}l^{ss} \\ c^{ss} &= (y^{ss}/g_y) - i^{ss}.\end{aligned}$$

The linearized model:

$$\begin{aligned}y_t &= \frac{c^{ss}}{y^{ss}}c_t + \frac{i^{ss}}{y^{ss}}i_t + \frac{k^{ss}r^{kss}}{y^{ss}}z_t + u_t^g \\ y_t^* &= \frac{c^{ss}}{y^{ss}}c_t^* + \frac{i^{ss}}{y^{ss}}i_t^* + \frac{k^{ss}r^{kss}}{y^{ss}}z_t^* + u_t^{g*} \\ c_t &= \frac{\lambda e^{-\gamma}}{1 + \lambda e^{-\gamma}}c_{t-1} + \frac{1}{1 + \lambda e^{-\gamma}}E_t c_{t+1} \\ &\quad + \frac{(\sigma_c - 1)w^{ss}l^{ss}}{c^{ss}\sigma_c(1 + \lambda e^{-\gamma})(1 + \lambda_w)}(l_t - E_t l_{t+1}) \\ &\quad - \frac{1 - \lambda e^{-\gamma}}{\sigma_c(1 + \lambda e^{-\gamma})}(r_t - E_t \pi_{t+1}) + u_t^b \\ c_t^* &= \frac{\lambda e^{-\gamma}}{1 + \lambda e^{-\gamma}}c_{t-1}^* + \frac{1}{1 + \lambda e^{-\gamma}}E_t c_{t+1}^* \\ &\quad + \frac{(\sigma_c - 1)w^{ss}l^{ss}}{c^{ss}\sigma_c(1 + \lambda e^{-\gamma})(1 + \lambda_w)}(l_t - E_t l_{t+1}^*) \\ &\quad - \frac{1 - \lambda e^{-\gamma}}{\sigma_c(1 + \lambda e^{-\gamma})}r_t^* + u_t^{b*} \\ i_t &= \frac{1}{1 + \beta e^{(1-\sigma_c)\gamma}}i_{t-1} + \frac{\beta e^{(1-\sigma_c)\gamma}}{1 + \beta e^{(1-\sigma_c)\gamma}}E_t i_{t+1} \\ &\quad + \frac{1}{(1 + \beta e^{(1-\sigma_c)\gamma})\varphi e^{2\gamma}}q_t + u_t^i \\ i_t^* &= \frac{1}{1 + \beta e^{(1-\sigma_c)\gamma}}i_{t-1}^* + \frac{\beta e^{(1-\sigma_c)\gamma}}{1 + \beta e^{(1-\sigma_c)\gamma}}E_t i_{t+1}^* \\ &\quad + \frac{1}{(1 + \beta e^{(1-\sigma_c)\gamma})\varphi e^{2\gamma}}q_t^* + u_t^{i*}\end{aligned}$$

$$q_t = \beta(1-\delta)e^{-\gamma\sigma_c}E_t q_{t+1} + (1-\beta(1-\delta)e^{-\gamma\sigma_c})E_t r_{t+1}^k - r_t \\ + E_t \pi_{t+1} - \frac{\sigma_c(1+\lambda e^{-\gamma})}{1-\lambda e^{-\gamma}}u_t^b$$

$$q_t^* = \beta(1-\delta)e^{-\gamma\sigma_c}E_t q_{t+1}^* + (1-\beta(1-\delta)e^{-\gamma\sigma_c})E_t r_{t+1}^{k*} - r_t^* \\ - \frac{\sigma_c(1+\lambda e^{-\gamma})}{1-\lambda e^{-\gamma}}u_t^b$$

$$y_t = \frac{y^{ss}+F}{y^{ss}}\alpha k_t^s + \frac{y^{ss}+F}{y^{ss}}(1-\alpha)l_t + \frac{y^{ss}+F}{y^{ss}}u_t^a$$

$$y_t^* = \frac{y^{ss}+F}{y^{ss}}\alpha k_t^{s*} + \frac{y^{ss}+F}{y^{ss}}(1-\alpha)l_t^* + \frac{y^{ss}+F}{y^{ss}}u_t^a$$

$$k_t^s = z_t + k_{t-1}$$

$$k_t^{s*} = z_t^* + k_{t-1}^*$$

$$z_t = \frac{1-\psi}{\psi}r_t^k$$

$$z_t^* = \frac{1-\psi}{\psi}r_t^{k*}$$

$$k_t = (1-\delta)e^{-\gamma}k_{t-1} + (1-(1-\delta)e^{-\gamma})i_t \\ + (1-(1-\delta)e^{-\gamma})((1+\beta e^{(1-\sigma_c)\gamma})\varphi e^{2\gamma})u_t^i$$

$$k_t^* = (1-\delta)e^{-\gamma}k_{t-1}^* + (1-(1-\delta)e^{-\gamma})i_t^* \\ + (1-(1-\delta)e^{-\gamma})((1+\beta e^{(1-\sigma_c)\gamma})\varphi e^{2\gamma})u_t^{i*}$$

$$\mu_t^p = \alpha r_t^k + (1-\alpha)w_t - u_t^a$$

$$\mu_t^{p*} = \alpha r_t^{k*} + (1-\alpha)w_t^* - u_t^a$$

$$\pi_t = \frac{\iota_p}{1+\iota_p\beta e^{(1-\sigma_c)\gamma}}\pi_{t-1} + \frac{\beta e^{(1-\sigma_c)\gamma}}{1+\iota_p\beta e^{(1-\sigma_c)\gamma}}E_t \pi_{t+1} \\ + \frac{1-\beta e^{(1-\sigma_c)\gamma}\xi_p(1-\xi_p)}{(1+\iota_p\beta e^{(1-\sigma_c)\gamma})\xi_p((\phi_p-1)\varepsilon_p+1)}\mu_t^p + u_t^p$$

$$\mu_t^{p*} = 1$$

$$r_t^k = w_t - k_t + l_t$$

$$r_t^{k*} = w_t^* - k_t^* + l_t^*$$

$$\mu_t^w = w_t - \sigma_l l_t - \frac{1}{1-\lambda e^{-\gamma}}c_t + \frac{\lambda e^{-\gamma}}{1-\lambda e^{-\gamma}}c_{t-1}$$

$$\mu_t^{w*} = -\sigma_l l_t^* - \frac{1}{1-\lambda e^{-\gamma}}c_t^* + \frac{\lambda e^{-\gamma}}{1-\lambda e^{-\gamma}}c_{t-1}^*$$

$$w_t = \frac{1}{1+\beta e^{(1-\sigma_c)\gamma}}w_{t-1} \\ + \left(1 - \frac{1}{1+\beta e^{(1-\sigma_c)\gamma}}\right)(E_t w_{t+1} + E_t \pi_{t+1}) \\ - \frac{1+\iota_w\beta e^{(1-\sigma_c)\gamma}}{1+\beta e^{(1-\sigma_c)\gamma}}\pi_t + \frac{\iota_w}{1+\beta e^{(1-\sigma_c)\gamma}}\pi_{t-1} \\ - \frac{(1-\xi_w)(1-\xi_w\beta e^{(1-\sigma_c)\gamma})}{(1+\beta e^{(1-\sigma_c)\gamma})\xi_w((\phi_w-1)\varepsilon_w+1)}\mu_t^w + u_t^w$$

$$w_t^* = \mu_t^{w*}$$

$$r_t = \rho r_{t-1} + (1-\rho)(r_{\pi}\pi_t) - [(1-\rho)r_y + r_{\Delta y}]y_t \\ - [(1-\rho)r_y + r_{\Delta y}]y_t^* - r_{\Delta y}y_{t-1} + r_{\Delta y}y_{t-1}^* + u_t^r.$$

The exogenous shock processes:

$$u_t^a = \rho_a u_{t-1}^a + \varepsilon_t^a$$

$$u_t^b = \rho_b u_{t-1}^b + \varepsilon_t^b$$

$$u_t^g = \rho_g u_{t-1}^g + \varepsilon_t^g + \rho_{ga}\varepsilon_t^a$$

$$u_t^i = \rho_i u_{t-1}^i + \varepsilon_t^i$$

$$u_t^r = \rho_r u_{t-1}^r + \varepsilon_t^r$$

$$u_t^p = \rho_p u_{t-1}^p - \mu_p \varepsilon_{t-1}^p + \varepsilon_t^p$$

$$\varepsilon_t^p = \varepsilon_t^p$$

$$u_t^w = \rho_w u_{t-1}^w - \mu_w \varepsilon_{t-1}^w + \varepsilon_t^w \\ \varepsilon_t^w = \varepsilon_t^w.$$

The expectational error terms:

$$\pi_t = E_{t-1}\pi_t + \eta_t^\pi$$

$$c_t = E_{t-1}c_t + \eta_t^c$$

$$l_t = E_{t-1}l_t + \eta_t^l$$

$$q_t = E_{t-1}q_t + \eta_t^q$$

$$r_t^k = E_{t-1}r_t^k + \eta_t^{rk}$$

$$i_t = E_{t-1}i_t + \eta_t^i$$

$$w_t = E_{t-1}w_t + \eta_t^w$$

$$c_t^* = E_{t-1}c_t^* + \eta_t^{c*}$$

$$l_t^* = E_{t-1}l_t^* + \eta_t^{l*}$$

$$q_t^* = E_{t-1}q_t^* + \eta_t^{q*}$$

$$r_t^{k*} = E_{t-1}r_t^{k*} + \eta_t^{rk*}$$

$$i_t^* = E_{t-1}i_t^* + \eta_t^{i*}.$$

Dummy equations for lagged variables:

$$y_{t-1} = y_{t-1}$$

$$c_{t-1} = c_{t-1}$$

$$i_{t-1} = i_{t-1}$$

$$w_{t-1} = w_{t-1}$$

$$y_{t-1}^* = y_{t-1}^*.$$

To cast the model in the canonical form for Sims (2002) solution method, define the 53 dimensional state vector

$$\mathbf{s}_t = [y_t, k_t^s, l_t, r_t^k, w_t, \pi_t, \mu_t^p, c_t, r_t, z_t, q_t, i_t, k_t, \mu_t^w, E_t \pi_{t+1}, \\ E_t c_{t+1}, E_t l_{t+1}, E_t q_{t+1}, E_t r_{t+1}^k, E_t i_{t+1}, E_t w_{t+1}, y_{t-1}, c_{t-1}, i_{t-1}, \\ w_{t-1}, u_t^a, u_t^b, u_t^g, u_t^i, u_t^r, u_t^p, u_t^w, \varepsilon_t^p, \varepsilon_t^w, y_t^*, k_t^{s*}, l_t^*, \\ r_t^{k*}, w_t^*, \mu_t^{p*}, c_t^*, r_t^*, z_t^*, q_t^*, i_t^*, k_t^*, \mu_t^{w*}, E_t c_{t+1}^*, \\ E_t l_{t+1}^*, E_t q_{t+1}^*, E_t r_{t+1}^{k*}, E_t i_{t+1}^*, y_{t-1}^*]$$

the vector of innovations

$$\mathbf{\varepsilon}_t = [\varepsilon_t^a, \varepsilon_t^b, \varepsilon_t^g, \varepsilon_t^i, \varepsilon_t^r, \varepsilon_t^p, \varepsilon_t^w]$$

and the vector of endogenous rational expectations errors

$$\mathbf{\eta}_t = [\pi_t - E_{t-1}\pi_t, c_t - E_{t-1}c_t, l_t - E_{t-1}l_t, q_t - E_{t-1}q_t, \\ r_t^k - E_{t-1}r_t^k, i_t - E_{t-1}i_t, w_t - E_{t-1}w_t, c_t^* - E_{t-1}c_t^*, l_t^* - E_{t-1}l_t^*, \\ q_t^* - E_{t-1}q_t^*, r_t^{k*} - E_{t-1}r_t^{k*}, i_t^* - E_{t-1}i_t^*].$$

Based on this information, the canonical form of the model can be readily formulated.

References

- An, Sungbae, Schorfheide, Frank, 2007. Bayesian analysis of DSGE models. *Econometric Reviews* 26 (2), 113–172.
- Anderson, Gary S., 2008. Solving linear rational expectations models: A horse race. *Computational Economics* 31 (2), 95–113.
- Canova, Fabio, 2007. *Methods for Applied Macroeconomic Research*. Princeton University Press, Princeton, NJ.
- Chib, Siddhartha, 1995. Marginal likelihood from the Gibbs output. *Journal of the American Statistical Association* 90, 1313–1321.
- Chib, Siddhartha, 2001. Markov chain Monte Carlo methods: Computation and inference. *Handbook of Econometrics* 5, 3569–3649.
- Chib, Siddhartha, Ergashev, Bakhodir, 2008. Analysis of multi-factor affine yield curve models. *Journal of the American Statistical Association* (in press).
- Chib, Siddhartha, Greenberg, Edward, 1994. Bayes inference in regression models with ARMA (p, q) errors. *Journal of Econometrics* 64, 183–206.
- Chib, Siddhartha, Greenberg, Edward, 1995. Understanding the Metropolis-Hastings algorithm. *American Statistician* 49, 327–335.

- Chib, Siddhartha, Jeliazkov, Ivan, 2001. Marginal likelihood from the Metropolis–Hastings output. *Journal of the American Statistical Association* 96, 270–281.
- Christiano, Lawrence J., Eichenbaum, Martin, Evans, Charles L., 2005. Nominal rigidities and the dynamic effects of a shock to monetary policy. *Journal of Political Economy* 113 (1), 1–45.
- DeJong, David N., Dave, Chetan, 2007. *Structural Macroeconometrics*. Princeton University Press, Princeton, NJ.
- Del Negro, Marco, Schorfheide, Frank, Smets, Frank, Wouters, Rafael, 2007. On the fit and forecasting performance of new keynesian models. *Journal of Business and Economic Statistics* 25 (2), 123–162.
- Elerian, Ola, Chib, Siddhartha, Shephard, Neil, 2001. Likelihood inference for discretely observed non-linear diffusions. *Econometrica* 69, 959–993.
- Geweke, John F., 2005. *Contemporary Bayesian Econometrics and Statistics*, 2nd ed. MIT Press, Cambridge, MA.
- Harvey, Andrew C., 1990. *The Econometric Analysis of Time Series*. In: *Wiley Series in Probability and Statistics*.
- Fernandez-Villaverde, Jesus, Rubio-Ramirez, Juan F., 2004. Comparing dynamic equilibrium economies to data. *Journal of Econometrics* 123 (1), 153–187.
- Lubik, Thomas A., Schorfheide, Frank, 2004. Testing for indeterminacy: An application to us monetary policy. *American Economic Review* 1 (5), 1123–1175.
- Nocedal, Jorge, Wright, Stephen J., 2000. *Numerical Optimization*. Springer.
- Pitt, Michael, Shephard, Neil, 1997. Likelihood analysis of non-Gaussian measurement time series. *Biometrika* 84, 653–667.
- Sargent, Tom, 1989. Two models of measurements and the investment accelerator. *Journal of Political Economy* 97 (2), 251–287.
- Sims, Christopher A., 2002. Solving linear rational expectations models. *Computational Economics* 20 (1–2), 1–20.
- Sims, Christopher A., Waggoner, Daniel F., Zha, Tao, *Methods for Inference in Large Multiple-Equation Markov-Switching Models*. *Journal of Econometrics*, (forthcoming).
- Smets, Frank, Wouters, Rafael, 2003. An estimated dynamic stochastic general equilibrium model of the euro area. *Journal of European Economic Association* 97 (3), 586–606.
- Smets, Frank, Wouters, Rafael, 2007. Shocks and frictions in US business cycles: A bayesian DSGE approach. *American Economic Review* 97 (3), 586–606.
- Taylor, John B., 1993. Discretion versus policy rules in practice. *Carnegie-Rochester Conference Series on Public Policy* 39, 195–214.
- Zellner, Arnold, Min, Chung-Ki, 1995. Gibbs sampler convergence criteria. *Journal of the American Statistical Association* 90, 921–927.