A Note on Efficient Fitting of Stochastic Volatility Models

Chen Gong*
Department of Statistics, University of Pittsburgh
chg87@pitt.edu

David S. Stoffer *
Department of Statistics, University of Pittsburgh
stoffer@pitt.edu

KEYWORDS: Stochastic Volatility, Particle Gibbs, Ancestral Sampling, Efficient Markov Chain Monte Carlo.

Abstract

The stochastic volatility model is a popular tool for modeling the volatility of assets. The model is a nonlinear and non-Gaussian state space model and presents some challenges not seen in general. Many approaches have been developed for Bayesian analysis that rely on numerically intensive techniques such as Markov chain Monte Carlo (MCMC). Convergence and mixing problems still plague MCMC algorithms used for the model. We present an approach that ameliorates the slow convergence and mixing problems when fitting stochastic volatility models. The approach accelerates MCMC by exploiting the geometry of one of the targets. We demonstrate the method on various numerical examples.

*This work was supported in part by NSF DMS-1506882

1. The Problem

Most models of volatility that are used in practice are of a multiplicative form, modeling the return of an asset, say y_t, observed at discrete time points, $t = 1, \ldots, n$, as

$$y_t = \sigma_t \epsilon_t \tag{1}$$

where ϵ_t is an iid sequence and the volatility process σ_t is a non-negative stochastic process such that ϵ_t is independent of σ_s for all $s \leq t$. It is often assumed that ϵ_t has zero mean and unit variance.

The basic univariate discrete-time stochastic volatility (SV) model writes the returns and the non-anticipative log volatility process, $x_t = \log \sigma_t^2$, as

$$x_t = \mu + \phi(x_{t-1} - \mu) + \sigma w_t \tag{2}$$

$$y_t = \beta \exp\{\tfrac{1}{2} x_t\} \epsilon_t, \tag{3}$$

where $x_0 \sim \mathrm{N}(\mu, \frac{\sigma^2}{1-\phi^2})$, $w_t \overset{\text{iid}}{\sim} \mathrm{N}(0, 1)$, ϵ_t is a mean zero, unit variance, iid process with at least finite fourth moment, and are all independent processes. The volatility process x_t is not observed directly, but only through the observations, y_t. The detailed econometric properties of the model are discussed in Shephard (1996) and Taylor (1994, 2008).

The model (2)–(3) is a nonlinear state space model and Bayesian analysis of such models can be approached using various methods, many of which are described in Douc et al. (2014, Chap. 12). Early MCMC approaches to the problem may be found in Carlin et al. (1992), Kim et al. (1998), Jacquier et al. (1994), and Taylor (1994). Other more current approaches may be found in Kastner and Frühwirth-Schnatter (2014), with techniques that are implemented in the R package `stochvol` (Kastner and Hosszejni, 2019); see also Hosszejni and Kastner (2019).

Let $\Theta = (\mu, \beta, \phi, \sigma)$ represent the parameters, denote the observations as $y_{1:n} = \{y_1, \ldots, y_n\}$, and the states (log-volatility) by $x_{0:n} = \{x_0, x_1, \ldots, x_n\}$, with x_0 being the initial state. To run a full Gibbs sampler, we alternate between sampling model parameters and latent state sequences from their respective full conditional distributions. Letting $\mathrm{p}(\cdot)$ denote a generic density, we have the following:

Procedure 1 (Generic Gibbs Sampler for State Space Models)

(i) *Draw* $\Theta' \sim \mathrm{p}(\Theta \mid x_{0:n}, y_{1:n})$

(ii) *Draw* $x'_{0:n} \sim \mathrm{p}(x_{0:n} \mid \Theta', y_{1:n})$

Procedure 1-(i) is generally much easier because it conditions on the complete data $\{x_{0:n}, y_{1:n}\}$. Procedure 1-(ii) amounts to sampling from the joint smoothing distribution of the latent state sequence and is generally more difficult. For linear Gaussian models, however, both parts of Procedure 1 are relatively easy to perform (Frühwirth-Schnatter, 1994; Carter and Kohn, 1994; Shumway and Stoffer, 2017, Chap. 6).

Although the focus of this note is to address Procedure 1-(i), we mention our preferred method for handling Procedure 1-(ii). Andrieu et al. (2010) introduced the particle Markov chain Monte Carlo (PMCMC) method, which proposed a conditional particle filter (CPF). The brilliance of the approach is that the CPF is invariant in the sense that the kernel leaves $\mathrm{p}(x_{0:n} \mid \Theta', y_{1:n})$ invariant; that is, all elements of the chain have the target distribution. Thus, one can avoid methods that merely approximate the distribution.

Unfortunately, CPF suffers from the path degeneracy. Lindsten et al. (2014) solved the problem by introducing CPF with ancestral sampling (CPF-AS). The addition of ancestral sampling improved on the problem of path degeneracy while being robust to the number of particles generated. In fact, the method works very well with a small number of particles. Ancestral sampling maintains the invariance of the kernel and, in addition to being efficient, the CPF-AS algorithm is uniformly ergodic under rather general assumptions (Lindsten et al., 2015). The algorithm is presented in the next section so as not to breakup the exposition.

As previously stated, step Procedure 1-(i) is typically the easier step. Usually one puts normal priors on μ or β and ϕ [or a beta prior on $\frac{1}{2}(\phi+1)$], and inverse gamma on σ^2. That is, current methods proceed as if ϕ is a regression parameter and σ is a scale parameter and this treatment is what leads to the inefficiency for this particular model. The problem for SV models is that ϕ behaves like a scale parameter as well as a regression parameter. For example, the autocorrelation function (ACF) of $\{y_t^2\}$ is given by

$$\mathrm{Cor}(y_t^2, y_{t+h}^2) = \frac{\exp(\sigma_x^2\phi^h) - 1}{\kappa_\epsilon \exp(\sigma_x^2) - 1}, \quad h = 1, 2, \ldots, \tag{4}$$

where κ_ϵ is the kurtosis of the noise, ϵ_t and $\sigma_x^2 = \sigma^2/(1-\phi^2)$. For SV models, the ACF values are small and the decay rate as a function of lag is less than exponential and somewhat linear. This means that if you specify values for ϕ but allow us to control σ (and consequently σ_x), we can make the model ACF to look approximately the same no matter which values of ϕ are chosen. This is accomplished by moving ϕ and σ in opposite directions. Another way of looking at the problem is to let (with $\mu = 0$ and $\beta = 1$) $\xi_t = \frac{1}{2\sigma_x} x_t$ and $\zeta_t = \frac{1}{2} w_t$ so we may write (3) as

$$y_t = \mathrm{e}^{\sigma_x \phi \xi_{t-1}}\, \mathrm{e}^{\sigma \cdot \zeta_t}\, \epsilon_t\,, \tag{5}$$

noting that ξ_{t-1} and ζ_t are independent stationary $\frac{1}{2}\,\mathrm{N}(0,1)$s. It is clear from (5) that σ and ϕ are scale parameters of the ξ_t process and σ is a scale parameter of the ζ_t noise process; we see that we can keep the scale of the data approximately the same by moving ϕ and σ in opposite directions.

For example, Figure 1 shows two data sequences (A and B) of length 1000 generated from two different SV models, (2)–(3), with ($\mu = 0$, $\beta = 1$) Model I: $\phi = .99$, $\sigma = .15$, and Model II: $\phi = .95$, $\sigma = .35$. The figure also compares the sample ACF of each generated series squared (A and B) and the theoretical counterparts for Models I and II. While the AR parameter, ϕ, is very different in each model, the simulated series look very much the same. In addition, the sample ACFs suggest that Series B is from Model I when in fact it is from Model II.

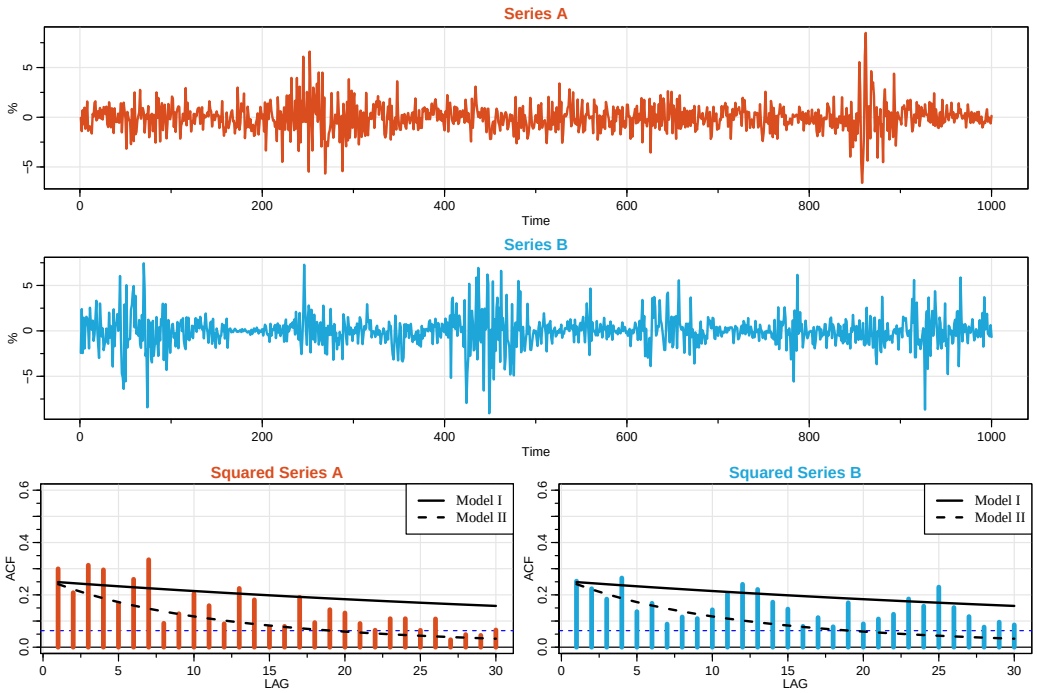

Figure 1: TOP: *Two data sequences (A and B) of length 1000 generated from different two-parameter SV models, I: $\phi = .99$, $\sigma = .15$ and II: $\phi = .95$, $\sigma = .35$.* BOTTOM: *The ACF of each generated series squared (A and B) and the theoretical ACFs of SV models I and II as lines. Series A corresponds to Model I and Series B corresponds to Model II.*

While CPF-AS can improve the mixing of the sampler for Procedure 1-(ii), it does not address mixing for Procedure 1-(i). The problem persists to this day as one can see in one of the vignettes of the R package `stochvol` (Kastner, 2019). The inefficiency of the sampler is due to the fact that (in the two-parameter model) Procedure 1-(i) is typically carried out in two steps by drawing from the univariate posteriors $\mathrm{p}(\phi \mid \sigma, x_{0:n}, y_{1:n})$ and $\mathrm{p}(\sigma \mid \phi, x_{0:n}, y_{1:n})$. There have been suggestions of joint sampling of ϕ and σ^2 (as opposed to σ), e.g., in Kim et al. (1998) or Kastner and Frühwirth-Schnatter (2014). However, the use of independent priors and using the conditionals of one parameter given the others does not seem to improve the efficiency because the methods do not exploit the geometry of the target.

As an example, we performed particle Gibbs with ancestral sampling (PGAS) on Series B ($n = 1000$) shown in Figure 1, which was generated from Model II,

$$x_t = .95x_{t-1} + .35w_t \quad \text{and} \quad y_t = \exp\{\tfrac{1}{2}x_t\}\epsilon_t\,.$$

For Procedure 1-(i) we used normal and inverse gamma priors for ϕ and σ^2, respectively. That is, with prior $\sigma^2 \sim \mathrm{IG}(a_0/2, b_0/2)$, where IG denotes the inverse (reciprocal) gamma distribution,

$$\sigma^2 \mid \phi, x_{0:n}, y_{1:n} \sim \mathrm{IG}\Big(\tfrac{1}{2}(a_0 + n + 1),\ \tfrac{1}{2}\big\{b_0 + \sum_{t=1}^{n}[x_t - \phi x_{t-1}]^2\big\}\Big)\,. \tag{6}$$

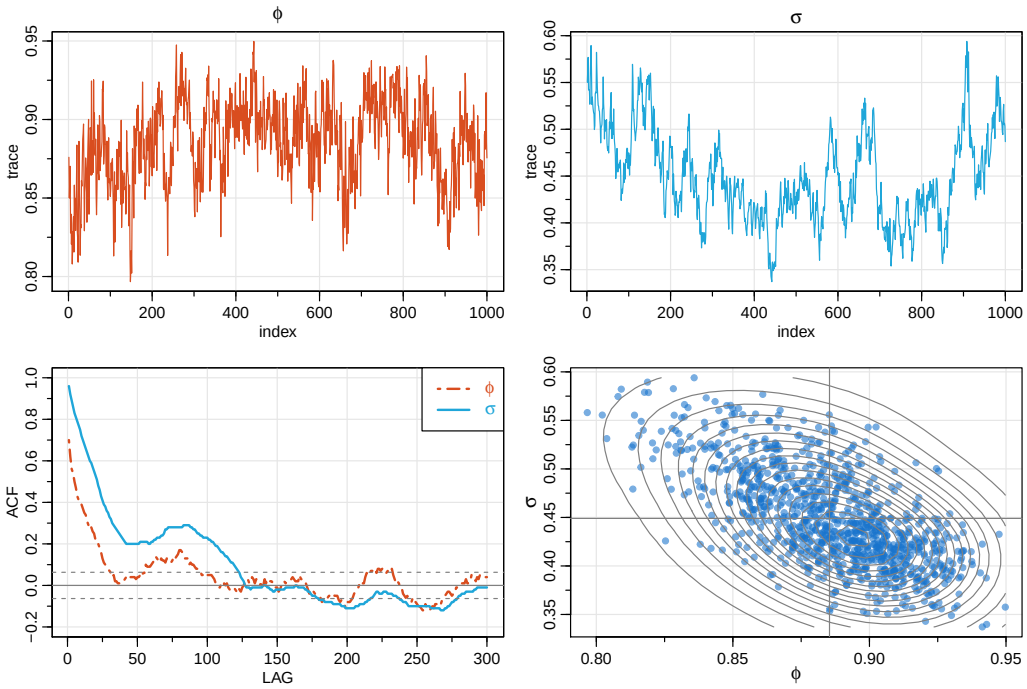

Figure 2: *Particle Gibbs with individual parameter sampling on Series B shown in Figure 1. The actual parameters are* $\phi = .95$ *and* $\sigma = .35$. TOP: *Separate draws of* ϕ *and* σ *in order of the draws.* BOTTOM: *The sample ACF of the traces (left), and a scatterplot (right) of the pairs of values in each draw with the posterior means of* .88 *and* .46 *highlighted.*

With prior $\phi \sim \mathrm{N}(\mu_\phi, \sigma_\phi^2)$, we have $(\phi \mid \sigma, x_{0:n}, y_{1:n}) \sim \mathrm{N}(Bb, B)$, where

$$B^{-1} = \frac{1}{\sigma_\phi^2} + \frac{1}{\sigma^2}\sum_{t=1}^{n} x_{t-1}^2\,, \qquad b = \frac{\mu_\phi}{\sigma_\phi^2} + \frac{1}{\sigma^2}\sum_{t=1}^{n} x_t x_{t-1}\,. \tag{7}$$

For Procedure 1-(ii), we used CPF-AS (Procedure 2) with $N = 20$ particles; details are given in the next section. This example is similar to the experiment discussed in Lindsten and Schön (2013, Sec. 7.1), however we use simulated data so that we know the model is correct (and does not add to convergence problem considerations).

Figure 2 shows the results of the experiment. The top row shows the draws of ϕ and σ in order, the bottom left shows the sample ACF of the traces, and on the right there is a scatterplot of the pairs of values in each draw. One sees the poor mixing in this example. While the sampling procedure with CPF-AS ameliorates the slow convergence problem to some degree, it is not a remedy because of relationship between the parameters seen in the scatterplot and previously discussed leads to *chattering* (a type of meandering) through the sample spaces of the parameters.

To improve the efficiency of the algorithm, we propose a new method for SV models by employing a bivariate prior and sampling the parameters jointly from $\mathrm{p}(\phi, \sigma \mid x_{0:n}, y_{1:n})$. A random walk Metropolis algorithm is used to implement the new method. The new method reduces the sampling inefficiencies significantly because it exploits the geometry of the target distribution.

2. Conditional Particle Filtering with Ancestral Sampling

For the sake of completeness, we present the CPF-AS algorithm used to perform step Procedure 1-(ii). The goal is to repeatedly draw an entire state sequence from the posterior $\mathrm{p}(x_{0:n} \mid \Theta, y_{1:n})$. To ease the notation, we will drop the conditioning arguments in this section. Many of the details (along with references) may be found in Lindsten et al. (2014) and Douc et al. (2014, Part III).

For notation, we will denote the proposal density by $\mathrm{q}(\cdot)$, the target density by $\mathrm{p}(\cdot)$, and the importance function (unnormalized weight) by $\omega = \mathrm{p}\,/\,\mathrm{q}$. Every density shown is conditional on parameters Θ and data $y_{1:t}$ up to time t. At the end of the procedure, we will have a sample of size N from the target of interest, $\mathrm{p}(x_{0:n} \mid \Theta, y_{1:n})$. To keep the exposition simple, a resampling step that was described in Gordon et al. (1993) and subsequently improved by others, and an auxiliary adjustment step as described in Pitt and Shephard (1999) are applied appropriately in the procedures, but we do not explicitly show these steps.

Procedure 2 (Conditional Particle Filter with Ancestral Sampling – [CPF-AS])
Input: *A sequence of conditioned particles $x'_{0:n}$ as a reference trajectory.*

(i) Initialize, $t = 0$:

- *(a) Draw $x_0^j \sim \mathrm{q}(\cdot)$ for $j = 1, \ldots, N-1$ (sample only $N-1$ particles)*
- *(b) Set the Nth particle, $x_0^N = x'_0$.*
- *(c) Compute weights $\omega_0^j \propto \omega_0(x_0^j)$ for $j = 1, \ldots, N$*

(ii) for $t = 1, \ldots, n$:

- *(a) Draw $I_t^j \sim$ Discrete$(\{\omega_{t-1}^i\}_{i=1}^N)$ for $j = 1, \ldots, N-1$*
- *(b) Draw $x_t^j \sim \mathrm{q}(x_t \,|\, x_{0:t-1}^{I_t^j})$ for $j = 1, \ldots, N-1$*
- *(c) Set $x_t^N = x'_t$*
- *(d) Draw $I_t^N \sim$ Discrete$(\{\omega_{t-1}^i\}_{i=1}^N)$ (ancestor sample)*
- *(e) Set $x_{0:t}^j = (x_{0:t-1}^{I_t^j}, x_t^j)$ and $\omega_t^j \propto \omega_t(x_{t-1}^{I_t^j}, x_t^j)$, for $j = 1, \ldots, N$*

While other methods exist, we note again that the resulting Markov kernel leaves its target distribution, $\mathrm{p}(x_{0:n} \mid \Theta, y_{1:n})$, invariant, regardless of the number of particles (Lindsten et al., 2014) and under general conditions is uniformly ergodic (Lindsten et al., 2015). Hence, Procedure 2 enables fast mixing of the particle Gibbs kernel even when using a few particles.

3. Proposed Method

In the SV model, (2)–(3), β and μ are not both needed. In choosing which parameter to keep, Kim et al. (1998) argued that allowing μ to vary and fixing $\beta = \exp(\mu/2)$ has a better interpretation from an economic point-of-view. Henceforth, we follow their restriction on β and allow μ to vary.

In Section 1, we discussed the problems of applying MCMC methods to SV models. Although various techniques such as CPF-AS exist that help solve some of the slow convergence problems when performing Procedure 1-(ii), we still observe poor mixing in Procedure 1-(i) caused by the inverse dependence between ϕ and σ. In this section, we suggest a method to improve the convergence by exploiting the geometry of the target.

To accomplish this goal, we put a bivariate normal prior with a negative correlation coefficient on the pair $\Theta = (\phi, \sigma)$,

$$\begin{pmatrix} \phi \\ \sigma \end{pmatrix} \sim \mathrm{N}_2\left(\begin{bmatrix} \mu_\phi \\ \mu_q \end{bmatrix}, \begin{bmatrix} \sigma_\phi^2 & \rho\sigma_\phi\sigma_q \\ \rho\sigma_\phi\sigma_q & \sigma_q^2 \end{bmatrix} \right), \tag{8}$$

where $\rho < 0$. Allowing possible negative values for σ is an old trick used in optimization to avoid constraints on the parameter space and is akin to the use of the Cholesky decomposition when estimating a covariance matrix to ensure the non-negative definiteness of the matrix. In this case, the draws corresponding to σ^2 will always be non-negative and marginally, has a scaled chi-squared prior distribution. In addition, as is seen in Figure 2, a bivariate normal prior is sensible. The proposition of sampling from (8) is in line with the consideration that exploiting the geometry of the target can improve the efficiency of an MCMC algorithm (Robert et al., 2018).

Note that we have changed the notation slightly by excluding μ from $\Theta = (\phi, \sigma)$ because it may be sampled separately if necessary. To accomplish Procedure 1-(i), note that,

$$\mathrm{p}(\Theta, \mu \mid x_{0:n}, y_{1:n}) \propto \pi(\Theta, \mu)\ \mathrm{p}(x_0 \mid \Theta, \mu) \prod_{t=1}^{n} \mathrm{p}(x_t \mid x_{t-1}, \Theta, \mu)\ \mathrm{p}(y_t \mid x_t, \Theta, \mu)\,, \tag{9}$$

where $\pi(\Theta, \mu)$ is the prior on the parameters. For the generic state space model, the parameters are often taken to be conditionally independent with distributions from standard parametric families (at least as long as the prior distribution is conjugate relative to the model specification). In this case, however, we must work with non-conjugate models, and one option is to replace Procedure 1-(i) with a Metropolis-Hastings step, which is feasible because the complete data density $\mathrm{p}(\Theta, \mu, x_{0:n}, y_{1:n})$ can be evaluated pointwise.

Under these considerations, for the SV model in (2)–(3), we have

$$\begin{aligned}
\mathrm{p}(\Theta \mid \mu, x_{0:n}, y_{1:n}) &\propto \pi(\Theta)\, \mathrm{p}(x_0 \mid \Theta, \mu) \prod_{t=1}^{n} \mathrm{p}(x_t \mid x_{t-1}, \Theta, \mu) \\
&\propto \exp\left\{ -\frac{1}{2(1-\rho^2)} \left[\frac{(\phi-\mu_\phi)^2}{\sigma_\phi^2} + \frac{(\sigma-\mu_\sigma)^2}{\sigma_q^2} - \frac{2\rho(\phi-\mu_\phi)(\sigma-\mu_\sigma)}{\sigma_\phi\sigma_q} \right] \right\} \\
&\quad \cdot \frac{\sqrt{1-\phi^2}}{\sigma} \exp\left\{ -\frac{(x_0-\mu)^2}{2\sigma^2/(1-\phi^2)} \right\} \prod_{t=1}^{n} \frac{1}{\sigma} \exp\left\{ -\frac{[(x_t-\mu)-\phi(x_{t-1}-\mu)]^2}{2\sigma^2} \right\} \\
&\propto \exp\left\{ -\frac{(\phi-\mu_\phi)^2\sigma_q^2 + (\sigma-\mu_\sigma)^2\sigma_\phi^2 - 2\rho\sigma_\phi\sigma_q(\phi-\mu_\phi)(\sigma-\mu_\sigma)}{2(1-\rho^2)\sigma_\phi^2\sigma_q^2} \right\} \\
&\quad \cdot \frac{\sqrt{1-\phi^2}}{\sigma^n} \exp\left\{ -\frac{(1-\phi^2)(x_0-\mu)^2 + \sum_{t=1}^{n}[(x_t-\mu)-\phi(x_{t-1}-\mu)]^2}{2\sigma^2} \right\}.
\end{aligned} \tag{10}$$

As previously suggested, we use a random walk Metropolis step to sample $\Theta = (\phi, \sigma)$ simultaneously from the target posterior distribution $\mathrm{p}(\Theta \mid \mu, x_{0:n}, y_{1:n})$ given in (10). This approach involves choosing a tuning parameter to control the acceptance probability. However, sometimes a good proposal distribution is difficult to choose because both the size and the spatial orientation of the proposal distribution should be considered. We have found that occasionally, the use of an adaptive method can help with the problem and we suggest using a technique that was presented in Andrieu and Thoms (2008, Alg. 4).

Procedure 3 (Sampling Θ)

INPUT: *An initial value,* Θ_0*, and an initial bivariate normal proposal distribution* $\mathrm{N}_2(\mu_0, \lambda_0\Sigma_0)$*. On iteration* $j+1$*, for* $j = 0, 1, 2, \ldots,$

(i) Draw $\vartheta \sim \mathrm{N}_2(\Theta_j, \lambda_j\Sigma_j)$ *and set* $\Theta_{j+1} = \vartheta$ *with probability* $\alpha_{j+1} = \frac{g(\vartheta)}{g(\Theta_j)} \wedge 1$*, where* $g(\Theta)$ *is given on the* RHS *of* (10)*. Otherwise, set* $\Theta_{j+1} = \Theta_j$*.*

(ii) [Optional Adaptive Method] Update

$$\log(\lambda_{j+1}) = \log(\lambda_j) + \gamma_{j+1}[\alpha_{j+1} - \alpha_\star], \tag{11}$$

$$\mu_{j+1} = \mu_j + \gamma_{j+1}(\Theta_{j+1} - \mu_j), \tag{12}$$

$$\Sigma_{j+1} = \Sigma_j + \gamma_{j+1}[(\Theta_{j+1} - \mu_j)(\Theta_{j+1} - \mu_j)' - \Sigma_j], \tag{13}$$

where γ_j *is a scalar nonincreasing sequence of positive step lengths such that* $\sum_{j=1}^{\infty} \gamma_j = \infty$ *and* $\sum_{j=1}^{\infty} \gamma_j^{1+\delta} < \infty$ *for some* $\delta > 0$*;* $\alpha_\star$ *is the expected acceptance rate for the algorithm.*

If Step (ii) is skipped, keep the tuning parameter and the covariance matrix fixed at λ_0 *and* Σ_0*, respectively.*

The optional part makes the algorithm non-Markovian, however, it can adapt continuously to the target distribution. Both the size and the spatial orientation of the proposal distribution will be adjusted by the adaptation procedure. Also, Procedure 3 is straightforward to implement and to use in practice. There are no extra computational costs because only a simple recursion formula for the covariances involved. The algorithm starts by using the accumulating information from the beginning of the sampling and it ensures that the search becomes more efficient at an early stage of the sampling. Haario et al. (2001) establish that the adaptive MCMC algorithms do indeed have the correct ergodicity properties. The hyperparameters in (8) should be chosen with care to ensure the an optimal acceptance rate of about 28% (see Gelman et al., 1996) and we suggest that ρ be moderately negative, $\rho \in (-.6, -.3)$. In addition, if adaptation is not used, the tuning parameter λ_0 should be chosen to maintain a proper acceptance rate; see Douc et al. (2014, Example 5.37) for more details.

If the parameter μ is included in the model, using a diffuse prior (e.g., see Kim et al., 1998), we have

$$\mu \mid \Theta, x_{0:n}, y_{1:n} \sim \mathrm{N}(\nu_\mu, \sigma_\mu^2) \tag{14}$$

Algorithm 1: Joint Particle Gibbs for Stochastic Volatility Models

INPUT: Set the initial value of $\Theta^{[0]} = (\phi, \sigma)^{[0]}, \mu^{[0]}$, and $x_{0:n}^{[0]}$ arbitrarily.

At iteration $j = 1, 2, \ldots,$

(i) Draw $x_{0:n}^{[j]}$ by CPF-AS, Procedure 2, conditioned on $x_{0:n}^{[j-1]}$ and $\Theta^{[j-1]}, \mu^{[j-1]}$.

(ii) With $x_{0:n}^{[j]}$, generate $\Theta^{[j]} = (\phi, \sigma)^{[j]}$ via Procedure 3 and draw $\mu^{[j]}$ from the posterior given in (14) under the current draws $x_{0:n}^{[j]}$ and $\Theta^{[j]}$.

where

$$\nu_\mu = \sigma_\mu^2 \left\{ \frac{1-\phi^2}{\sigma^2} x_0 + \frac{1-\phi}{\sigma^2} \sum_{t=1}^{n} (x_t - \phi x_{t-1}) \right\}$$

and

$$\sigma_\mu^2 = \frac{\sigma^2}{n(1-\phi)^2 + (1-\phi^2)}.$$

Recall that we are fixing $\beta = \exp(\mu/2)$. Finally, our algorithm for the analysis of a SV model is given in Algorithm 1.

4. Examples

4.1. *Two Parameter Model*

In this section, we fit a two-parameter model ($\mu = 0$) to the daily returns of the S&P 500 from January 2005 to October of 2011 shown at the top of Figure 3. The data include the financial crisis of 2008. We compare two methods using CPF-AS (Procedure 2) to sample the state process in both. The first method samples the parameters individually by drawing from the univariate conditionals $\mathrm{p}(\phi \mid \sigma, x_{0:n}, y_{1:n})$ and $\mathrm{p}(\sigma \mid \phi, x_{0:n}, y_{1:n})$ while our method exploits the geometry of the target and samples the parameters jointly as described in Algorithm 1 holding μ at zero.

In each case, we used $N = 20$ particles for the CPF-AS (Procedure 2) and 5000 iterations after a burnin of 100. The posterior mean and a pointwise 95% credible interval of the draws of the state (log-volatility) process is shown in Figure 3. The middle plot shows the results for the non-structured method and the bottom plot shows the results for the structured method. The results are similar, but the trace of the estimated process via the structured method is smoother and less variable than the non-structured method shown in the middle.

Figure 4 displays the results of the parameter estimation using the non-structured method of sampling ϕ and σ. The top of the figure shows the traces of the sampled values after burnin. The corresponding posterior means are about .88 for ϕ and .62 for σ. The bottom of the figure shows the sample ACFs of the traces and a scatterplot of the sampled values. In addition, the ACF plot displays the inefficiency measure as defined in Geyer (1992). The measure was obtained using Geyer's R package, `mcmc` (Geyer and Johnson, 2017). In particular, to evaluate the mixing of

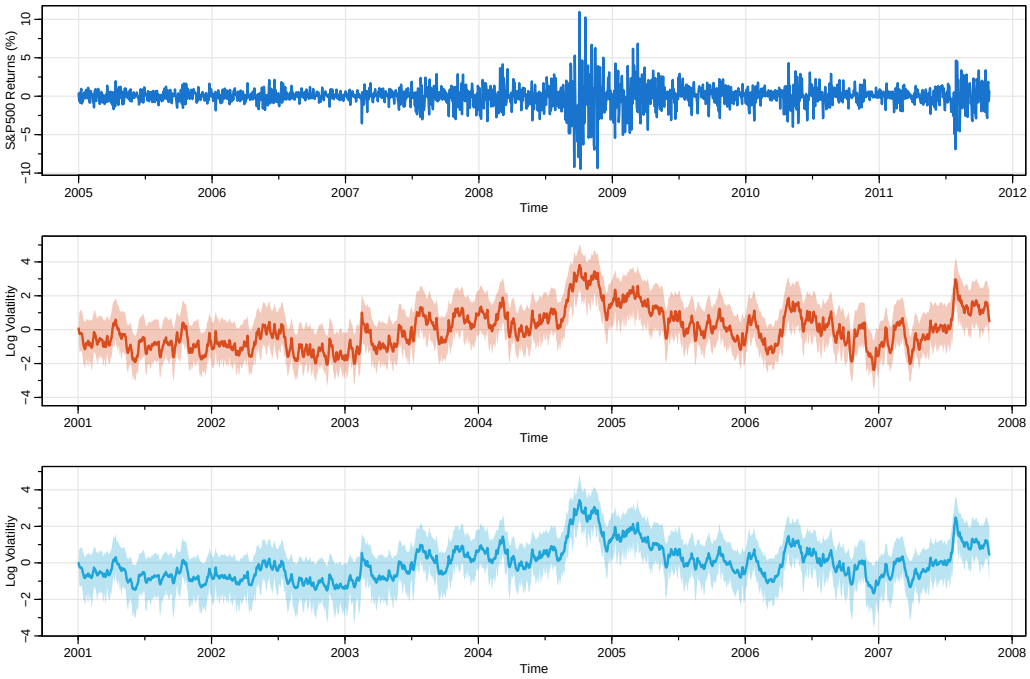

Figure 3: TOP: *Daily returns of the S&P 500 from January 2005 to October of 2011.* MIDDLE: *State process estimated posterior mean and pointwise 95% credible intervals based on one-at-a-time parameter sampling.* BOTTOM: *State process estimated posterior mean and pointwise 95% credible intervals based on the proposed method of structured joint parameter sampling, Algorithm 1. The number of particles used in the particle filter (Procedure 2) for both methods was* $N = 20$.

sampler, we estimate *inefficiency* defined as

$$\text{IF} := 1 + 2\sum_{i=1}^{\infty} \varrho(i), \tag{15}$$

where $\varrho(i)$ is the autocorrelation function of the trace at lag i. The estimated inefficiencies are displayed with the sample ACFs of traces. We note again the slow convergence problem seen in the simulation example. Finally, the bottom right scatterplot shows that ϕ and σ live on ellipses with negative correlation.

Figure 5 displays the results of the parameter estimation using our structured method, Algorithm 1, jointly sampling ϕ and σ. The top of the figure shows the traces of the sampled values after burnin. The corresponding posterior means are .80 for ϕ and .36 for σ. The bottom of the figure shows the sample ACFs of the traces and a scatterplot of the sampled values, which shows an improvement of the established method. In fact, the inefficiencies in the non-targeted sampling method are more the 50% bigger for ϕ and more than 80% bigger for σ.

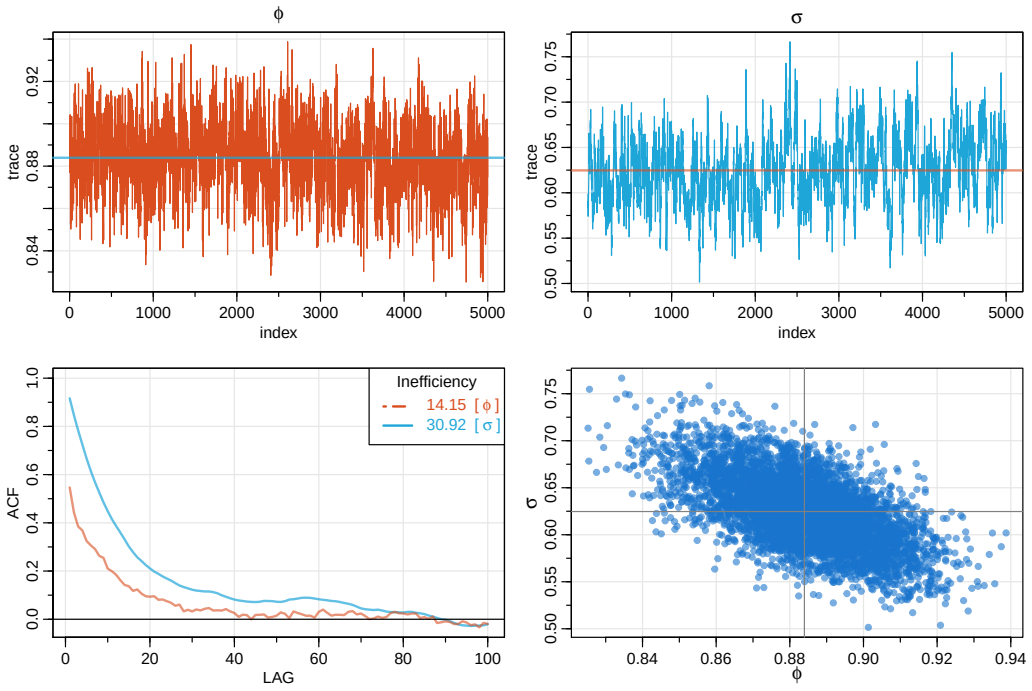

Figure 4: (Individual Sampling) *S&P 500 results of the parameter estimation in a two-parameter model when ϕ and σ are sampled separately. The top shows the traces of the 5000 sampled values after a burnin of 100. The number of particles used in the particle filter (Procedure 2) was $N = 20$. The corresponding posterior means are .88 for ϕ and .62 for σ. The bottom left shows the sample ACFs of the traces and the estimated inefficiency measure as defined in (15). The bottom left shows a scatterplot of the sampled parameters, and exhibits strong negative correlation of $-.61$.*

4.2. *Three Parameter Model*

Next, we fit a three parameter SV model to the S&P 500 series using Algorithm 1; the adaptive part of the Metropolis step, Procedure 3-(ii), was not needed here. To keep the complexity low, we used only $N = 10$ particles for sampling the states (Procedure 2), and then generated 2000 samples after a burnin of 100. The acceptance rate was nearly optimal at 26.1%. The entire estimation process took less than 4 minutes on a workstation running Windows 10 Pro with 32GB of DDR3 RAM, an Intel i7-4770 CPU @ 3.40 GHz, and using Microsoft R, version 3.5.2.

The results of the parameter estimation are shown in Figure 6; the results for the state estimation are similar to the lower plot of Figure 3 and are not shown to save space. The figure shows the trace of the draws (top row), the sample ACF of the draws (middle row) along with the estimated inefficiency, (15), and a histogram of the results (bottom row). The posterior means are displayed in the figure and were .85 for ϕ, .43 for σ and $-.01$ for μ. We note that the results are satisfactory even using this quick analysis. In fact, the efficiency for estimating μ in this case is better than independent sampling (IF = .76). It is also apparent that the previous analysis based on the two-parameter model ($\mu = 0$) was reasonable.

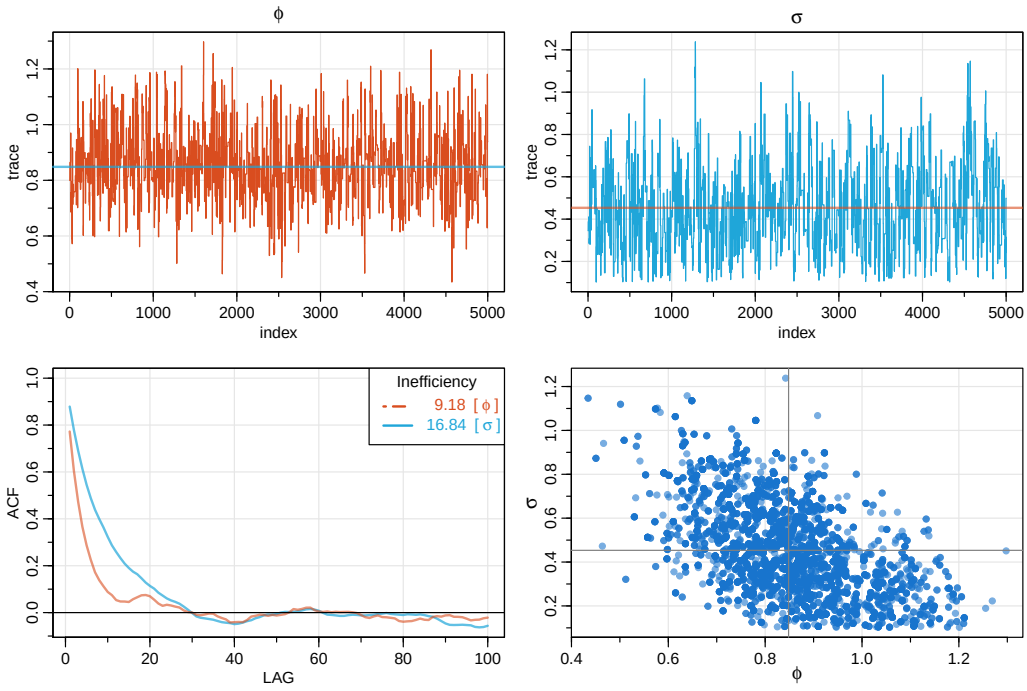

Figure 5: (Joint Sampling) *S&P 500 results of the parameter estimation in a two-parameter model using the proposed method of sampling ϕ and σ jointly, Algorithm 1. The top shows the traces of the 5000 sampled values after a burnin of 100. The number of particles used in the particle filter (Procedure 2) was $N = 20$. The corresponding posterior means are .85 for ϕ and .45 for σ and the correlation between the draws is $-.55$. The bottom shows the sample ACFs of the traces and a scatterplot of the sampled values, which indicate the sample parameters are uncorrelated. In addition, estimated inefficiency measures are improved over the counterparts in Figure 4.*

5. Stochastic Volatility with Multiple Observations

It is often reasonable to assume that similar assets are being driven by the same volatility process. In this case, the model presented in Asai et al. (2006) can be used. The model assumes a univariate volatility process is driving a number of similar assets and is given by,

$$x_t = \phi x_{t-1} + \sigma w_t \tag{16}$$

$$y_{it} = \beta_i \exp\left\{\frac{x_t}{2}\right\} \epsilon_{it}\,, \quad i = 1, \ldots, p, \tag{17}$$

where the y_{it} are the returns of the ith asset, $w_t \overset{\text{iid}}{\sim} \mathrm{N}(0,1)$, and $\boldsymbol{\epsilon}_t = (\epsilon_{1t}, \ldots, \epsilon_{pt})' \overset{\text{iid}}{\sim} \mathrm{N}_p(0, \mathrm{I})$. In this model, μ is removed to avoid overparameterization and each β_i is a scale parameter for the ith asset.

We can easily apply our proposed method to this model. That is, as in the univariate case, we put a bivariate normal prior on the state parameters, ϕ and σ. Then, because β_i, for $i = 1, \ldots, p$, is a scale parameter, a reasonable choice is to use independent inverse gamma priors for β_i^2 as in

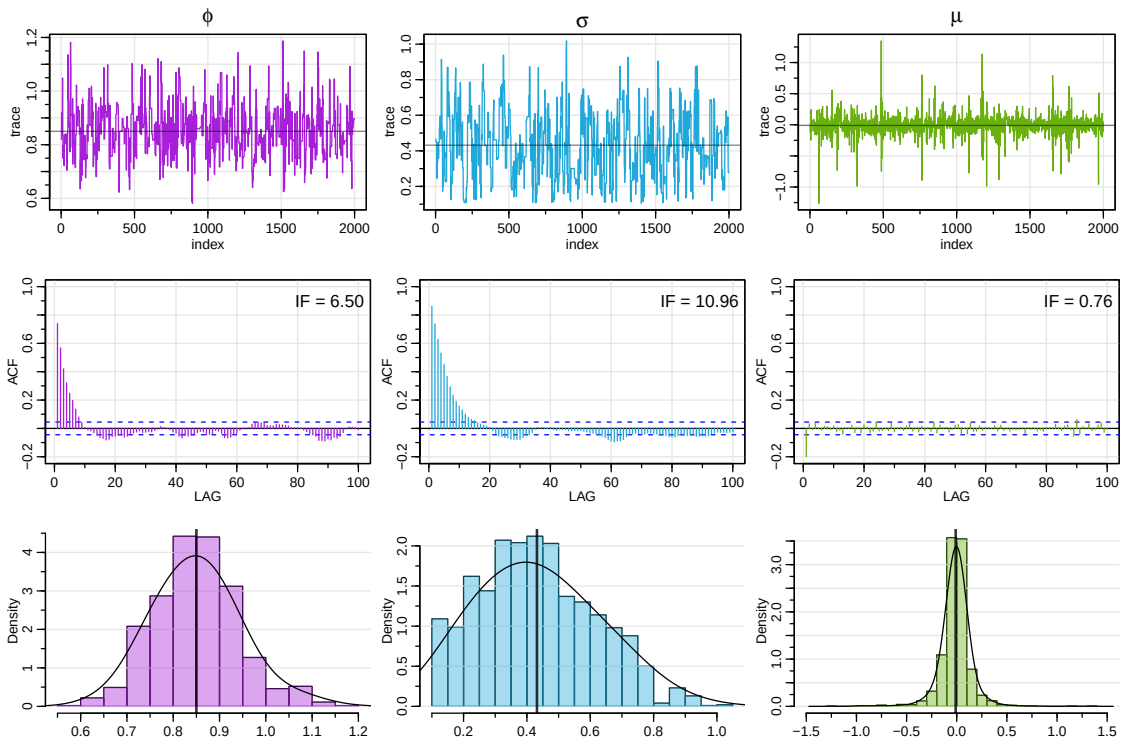

Figure 6: *S&P 500, results of the parameter estimation in a three-parameter model using the proposed method of sampling ϕ and σ jointly, Algorithm 1.* TOP: *Traces of the 2000 sampled values of the parameters after burnin.* MIDDLE: *Sample ACF of the draws along with the estimated inefficiency, (15).* BOTTOM: *Histogram of the results. The posterior means are displayed in the figure and were .85 for ϕ, .43 for σ and $-.01$ for μ. The correlation between the joint draws of ϕ and σ is $-.54$.*

(6). That is, if $\beta_i^2 \sim \text{IG}(a_i/2, b_i/2)$, then the posterior is

$$\beta_i^2 \mid \Theta, x_{0:n}, y_{1:n} \sim \text{IG}\Big(\tfrac{1}{2}(a_i + n + 1),\ \tfrac{1}{2}\Big\{b_i + \sum_{t=1}^{n} \frac{y_{it}^2}{\exp(x_t)}\Big\}\Big)\,. \tag{18}$$

Hence, for this model, we can simply add a third step to Algorithm 1, which is to sample β_i^2 from (18) for $i = 1, \ldots, p$. We summarize these steps in Algorithm 2.

For an example, we consider the daily NYSE returns for three banks, Bank of American (BOA), Citigroup (Citi), and J.P. Morgan (JPM) from January 2005 to November 2017. The data are displayed in Figure 7; also shown is the estimated log-volatility, which we describe shortly.

We used Algorithm 2 with $N = 20$ particles to generate 2000 draws after a burnin of 200 iterations. The procedure was non-adaptive and the acceptance rate was 29.8%. The entire procedure took about 12 minutes on the same machine mentioned in the other examples. The parameter estimation summary is displayed in Figure 8 and the display is similar to the previous example. The displays suggest that the algorithm is mixing well. The top shows the traces of the draws for each parameter and indicates the posterior means, .86 for ϕ, .32 for σ, and 1.64, 1.62, and 1.42 for the βs of BOA, Citi, and JPM, respectively. The middle plot shows the sample ACFs of the traces

Algorithm 2: Joint Particle Gibbs for (16) – (17)

INPUT: Set the initial value of $\Theta^{[0]} = (\phi, \sigma)^{[0]}$, $\beta_{1:p}^{[0]}$, and $x_{0:n}^{[0]}$ arbitrarily.

At iteration $j = 1, 2, \ldots,$

(i) Draw $x_{0:n}^{[j]}$ by CPF-AS, Procedure 2, conditioned on $x_{0:n}^{[j-1]}$ and $\Theta^{[j-1]}$.

(ii) With $x_{0:n}^{[j]}$, generate $\Theta^{[j]} = (\phi, \sigma)^{[j]}$ via Procedure 3 and draw $\beta_{1:p}^{2\,[j]}$ from the posteriors given in (18) under the current draws $x_{0:n}^{[j]}$ and $\Theta^{[j]}$.

along with the inefficiencies. The bottom row of Figure 8 displays the posterior distributions of each parameter along with the location of the posterior mean.

The resulting posterior of the log-volatility is shown at the bottom of Figure 7. Shown are the posterior mean and a swatch displaying pointwise 99% credible intervals. We also display a lowess fit as a thin line to emphasize the volatility trend. Notice that the impending financial crisis of 2008 is visible at least one year prior as the volatility starts a trend upwards just prior to 2007. It seems that there is an advantage to using multiple similar sources to estimate volatility.

6. Conclusions

The conditional particle filter with ancestral sampling (Procedure 2) was a breakthrough for analyzing nonlinear state space models by establishing a computationally efficient, invariant and uniformly ergodic method of sampling from the posterior of the hidden state trajectories, Procedure 1-(ii). The method works well for many cases if drawing from the posterior of the parameters, Procedure 1-(i), is not problematic. Unfortunately, this situation does not include the case of stochastic volatility models because in the state equation, the autoregressive parameter, ϕ and the noise variance, σ^2 in (2) have a tendency to work in opposite directions.

Prior attempts to handle SV models had less than optimal efficiency because ϕ was treated as a regression parameter while σ was treated as a scale parameter. Consequently, these parameters were sampled individually as they typically are in these situations. Even block techniques that sampled ϕ and σ^2 jointly have the same problem because the joint distribution is derived from the conditionals and do not exploit the geometry of the target.

We have presented a method that exploits the geometry of the target by sampling the state parameters jointly using a bivariate normal distribution. While it is possible that a sampled pair yields values of $|\phi| > 1$ or $\sigma < 0$, it is not a problem. For example, the state process is assumed to be stationary, so realistically, one only needs $|\phi| \neq 1$ (e.g., see Shumway and Stoffer, 2017, Example 3.4), which will not happen (with probability 1 in all but pathological cases). Also, sampled values of σ^2 will always be non-negative.

Finally, we mention that we did not supply every particular numerical detail (e.g., hyperparameters and tuning parameters) of our examples. Instead, for the sake of reproducibility, we supply

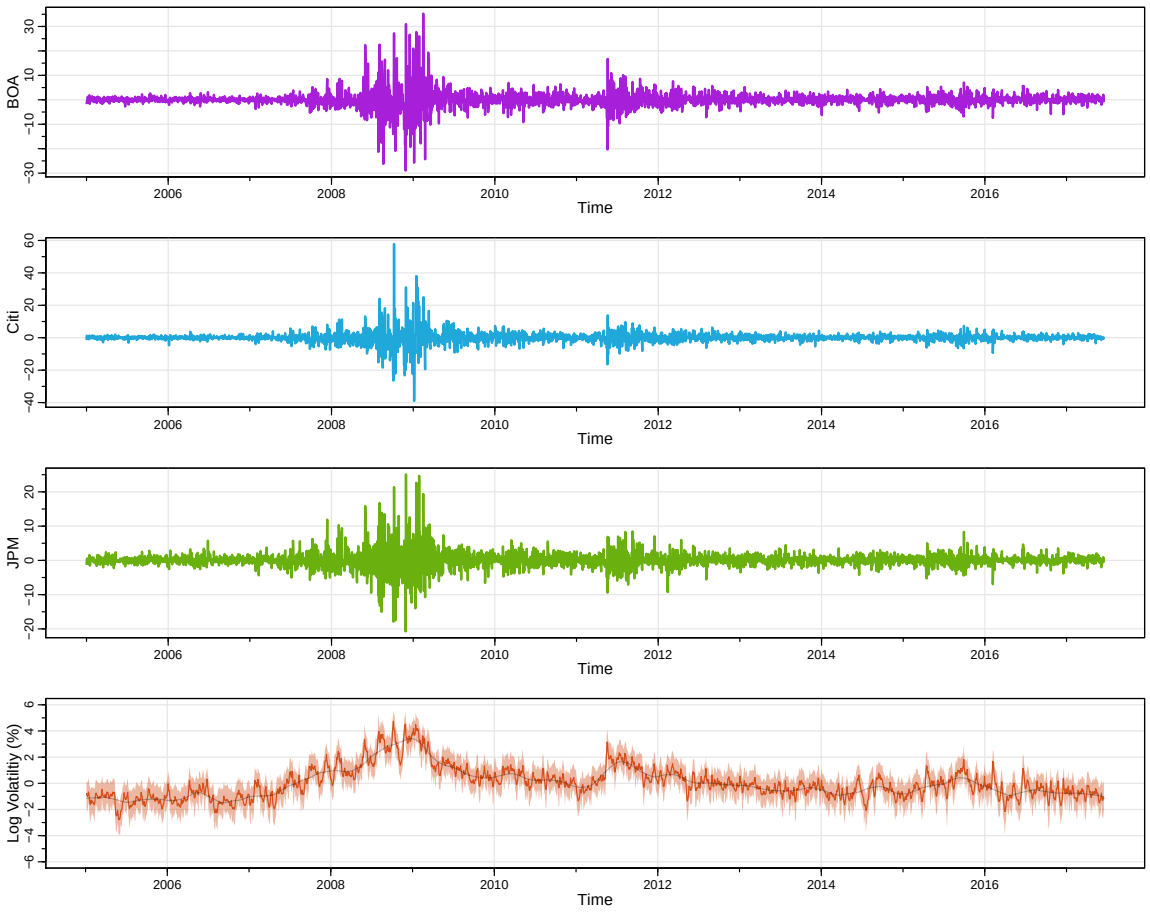

Figure 7: *The daily NYSE returns (as percentages) for three banks, Bank of American (BOA), Citigroup (Citi), and J.P. Morgan (JPM) from January 2005 to November 2017.* BOTTOM: *The resulting posterior of the log-volatility based on Algorithm 2. Shown are the posterior mean and a swatch displaying the limits of* 99% *of the sampled states. We also display a lowess smooth as a thin line to emphasize the volatility trend.*

the data and the R code (R Core Team, 2019) for every example on GitHub; see Gong and Stoffer (2019) for the URL. Additional information may also be found in Gong (2019).

References

Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342.

Andrieu, C. and Thoms, J. (2008). A tutorial on adaptive MCMC. *Statistics and Computing*, 18(4):343–373.

Asai, M., McAleer, M., and Yu, J. (2006). Multivariate stochastic volatility: a review. *Econometric Reviews*, 25(2-3):145–175.

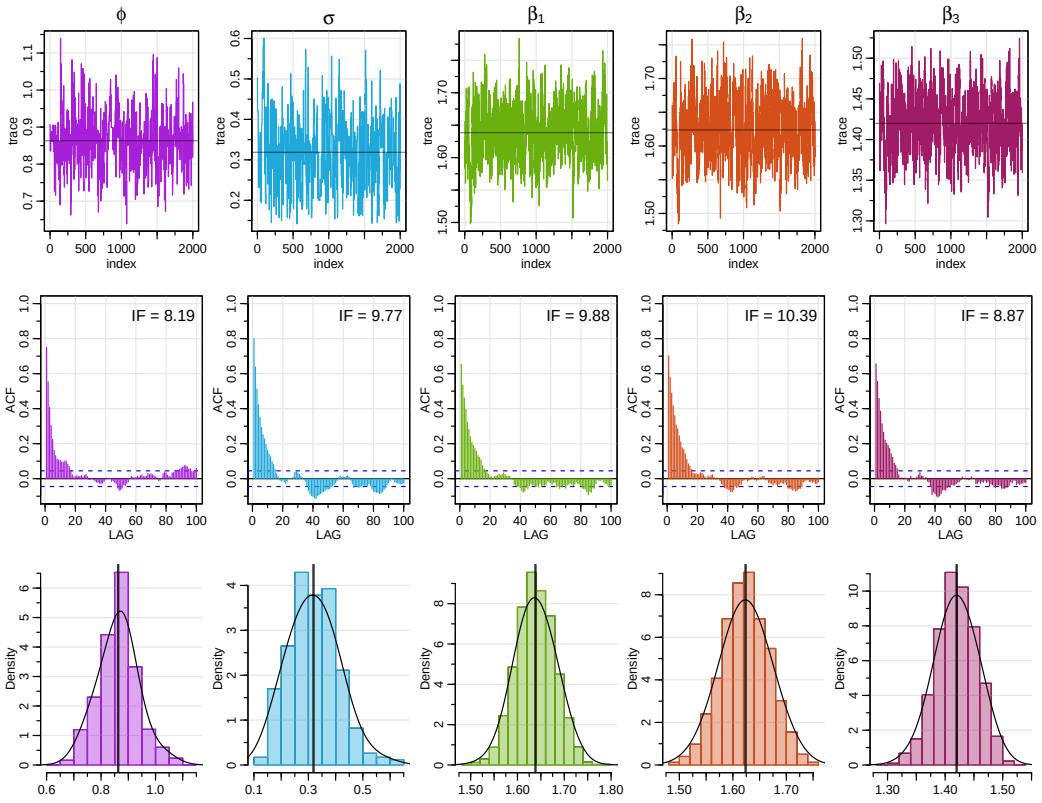

Figure 8: *For the bank returns data shown in Figure 7, the results of fitting an MSV model based on Algorithm 2.* TOP: *The traces of the draws for each parameter and indicates the posterior means,* .86 *for* ϕ*,* .32 *for* σ*, and* 1.64*,* 1.62*, and* 1.42 *for the* β*s of BOA, Citi, and JPM, respectively.* MIDDLE: *The sample ACFs of the traces along with the inefficiencies.* BOTTOM: *The estimated posterior distributions of each parameter along with the location of the posterior mean.*

Carlin, B. P., Polson, N. G., and Stoffer, D. S. (1992). A Monte Carlo approach to nonnormal and nonlinear state-space modeling. *Journal of the American Statistical Association*, 87(418):493–500.

Carter, C. K. and Kohn, R. (1994). On Gibbs sampling for state space models. *Biometrika*, 81(3):541–553.

Chib, S. and Greenberg, E. (1996). Markov chain Monte Carlo simulation methods in econometrics. *Econometric Theory*, 12(3):409–431.

Douc, R., Moulines, E., and Stoffer, D. S. (2014). *Nonlinear Time Series: Theory, Methods and Applications with R Examples*. CRC Press, Boca Raton.

Frühwirth-Schnatter, S. (1994). Data augmentation and dynamic linear models. *Journal of Time Series Analysis*, 15(2):183–202.

Gelman, A., Roberts, G. O., and Gilks, W. R. (1996). Efficient Metropolis jumping rules. In

Bayesian statistics, 5 (Alicante, 1994), Oxford Sci. Publ., pages 599–607, New York. Oxford Univ. Press.

Geyer, C. J. (1992). Practical Markov chain Monte Carlo. *Statist. Sci.*, 7(4):473–483.

Geyer, C. J. and Johnson, L. T. (2017). *mcmc: Markov Chain Monte Carlo*. R package version 0.9-5.

Gong, C. (2019). *Particle Gibbs Methods in Stochastic Volatility Models*. PhD thesis, University of Pittsburgh.

Gong, C. and Stoffer, D. S. (2019). Stochastic Volatility Models – R code. https://github.com/nickpoison/Stochastic-Volatility-Models/. [GitHub Repository].

Gordon, N., Salmond, D., and Smith, A. F. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc. F, Radar Signal Process.*, 140:107–113.

Haario, H., Saksman, E., and Tamminen, J. (2001). An adaptive Metropolis algorithm. *Bernoulli*, 7(2):223–242.

Hosszejni, D. and Kastner, G. (2019). Modeling univariate and multivariate stochastic volatility in r with stochvol and factorstochvol. *arXiv preprint arXiv:1906.12123*.

Jacquier, E., Polson, N. G., and Rossi, P. E. (1994). Bayesian analysis of stochastic volatility models. *Journal of Business & Economic Statistics*, 20(1):69–87.

Kastner, G. (2019). Dealing with stochastic volatility in time series using the r package stochvol. *arXiv preprint arXiv:1906.12134*.

Kastner, G. and Frühwirth-Schnatter, S. (2014). Ancillarity-sufficiency interweaving strategy (asis) for boosting mcmc estimation of stochastic volatility models. *Computational Statistics & Data Analysis*, 76:408–423.

Kastner, G. and Hosszejni, D. (2019). *stochvol: Efficient Bayesian Inference for Stochastic Volatility (SV) Models*. R package version 2.0.4.

Kim, S., Shephard, N., and Chib, S. (1998). Stochastic volatility: Likelihood inference and comparison with ARCH models. *The Review of Economic Studies*, 65(3):361–393.

Lindsten, F., Douc, R., and Moulines, E. (2015). Uniform ergodicity of the particle gibbs sampler. *Scandinavian Journal of Statistics*, 42(3):775–797.

Lindsten, F., Jordan, M. I., and Schön, T. B. (2014). Particle gibbs with ancestor sampling. *The Journal of Machine Learning Research*, 15(1):2145–2184.

Lindsten, F. and Schön, T. B. (2013). Backward simulation methods for Monte Carlo statistical inference. *Foundations and Trends in Machine Learning*, 6(1):1–143.

Pitt, M. K. and Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599.

R Core Team (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Robert, C. P., Elvira, V., Tawn, N., and Wu, C. (2018). Accelerating mcmc algorithms. *Wiley Interdisciplinary Reviews: Computational Statistics*, 10(5):e1435.

Shephard, N. (1996). Statistical aspects of ARCH and stochastic volatility. *Monographs on Statistics and Applied Probability*, 65:1–68.

Shumway, R. H. and Stoffer, D. S. (2017). *Time Series Analysis and Its Applications: With R Examples*. Springer, New York, 4th edition.

Taylor, S. J. (1994). Modeling stochastic volatility: A review and comparative study. *Mathematical Finance*, 4(2):183–204.

Taylor, S. J. (2008). *Modelling Financial Time Series*. World Scientific, 2nd edition.